RoboCup Rescue 2015 – Rescue Simulation League Team Description S.O.S (Iran)

Hesam AkbariBoshrabadi, Pegah Taheri, MohammadMilad Ameri, Marjan Al-Bouye, Morteza FaghaniLemraski

[1] Robotics Research Center, Department of
Computer Engineering and Information Technology,
Amirkabir University of Technology,
No. 424, Hafez Ave.,
Tehran, Iran
{hitax13, mohamadmilad, marjan.albouye73, morti.faghani, peghtaheri}@gmail.com

Abstract. S.O.S team is well known for its great performance in rescue simulation league. We have achieved lots of trophies in this league. Last year, we achieved 1st place in the robocup's national competition held in Brazil. We have achieved a very stable situation in our base codes by developing it for years by old team members and our most strategy has been shaped during these years in different competitions .Changes pointed below in this paper is designed to improve old strategies and fix newly appeared logical errors and adding new aspects to our strategies.

1. Introduction

The S.O.S. basic agent and its abilities and skills have been described in previous years TDPs (2010-2014), so this paper is to depict the new strategies added to our plan for RoboCup 2015. Following will come the description of these improvements in details [2].
Last year S.O.S members developed enough strategy on full and low communication, But For 2015 Robocup competition the main focus of S.O.S will be on no communication map, in this way we designed new Strategies for Fire brigade and Police force agents, which is described in agents section. In natural disaster all kind of event will be happen. Because of this we will perform so many activities to manipulate our code in the hardest situation.

2. Agents

2.1 Police Force

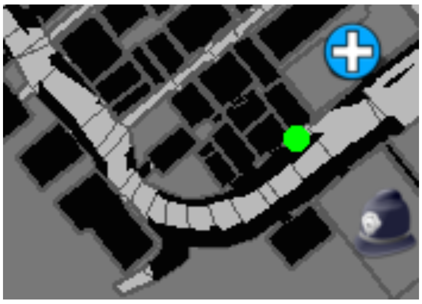
Our last year's TDP didn't discuss lots of new features that were added to the Policeforce code. Hence, in this year's TDP, we are focusing more on describing Police's algorithms and methods.

2.1.1 New clearing system

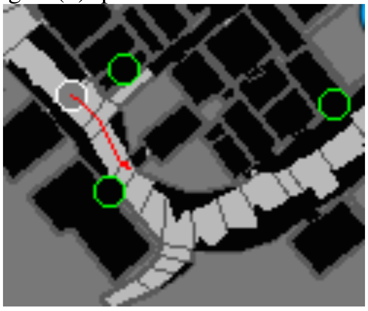
Changing of routine clearance in the simulation server in 2013 competition caused lots of issue including agents getting caught in steep edges of the map. To define this problem it has to be pointed that when an agent is stuck in a concave area with a sharp angle, other agents will surround it and cause a repulsion force to the stuck agent. As a result, lots of cycles will be wasted in the process of freeing that agent.

Streets not having the same direction is what causes this problem. The steep angle between the directions of 2 roads results in the angle in the joints, That will cause the angularity in concave areas in clearance. The simple and basic solution would be clearing the whole road, but we have avoided that, since it will waste lots of cycles.

The S.O.S Police Force has designed and developed a new system and algorithm that will reduce the probability of the agents stucking in edges. When clearing, our algorithm takes into account all of the future paths and then chooses the best path for clearing (based on next steps) while considering all of the options (rather than choosing and clearing the road's major direction), that will cause reduction in stucking in steep edges. Figure (1) and figure (2) we could compare S.O.S new clearing method with ZJU team clearing.



Figure(1): police Force new Clearing



Figure(2): Police Force Clearing of Zju in 2014 robocup competition

2.1.2 Make Fire Brigade Reachable to Fire zone

One of the other changes in Police Force is adding an Interrupt State for connecting a Fire Brigade to its fire. This works in the way that, if a Police Force sees or senses a fire, the agent starts estimating the fire's size. If it guesses that the fire is rather small in size and the Fire Brigade that will be assigned to it is in this agent's clear list, the Police Force will drop its task and first free a path between the Fire Brigade agent and the fire, so the fire will be extinguished sooner and with less waste of energy.

2.1.3 Check Fire Probability State

Another feature added to the Police Force is the Check Fire Probability State. In this state, if the police agent senses some buildings with high temperature, the agent starts estimating the probable place of fire. If the estimated pace of fire is near the agent, the agent clears the road to it, so there would be the possibility to stop fire in its beginning of ignition.

2.1.4 No communication world model

Our police force also updates the world model of Fire Brigade agents, which is very important in the No Communication strategy. In the no communication strategy, if an agent senses fire in his cluster, follows the fire to find its borders. But what is even more important than finding the fire is informing the Fire Brigade agents of the new information of the fire. The police agent finds the Fire Brigade that will be responsible for extinguishing the fire and updates its info for fire. Since in no communication, the exact place of the agents in unknown, the police searches for the most probable places of Fire Brigades, according to their old world model. Then the police agent updates each Fire Brigade's info of the world model, if the fire agent's last update time is more than 15 cycles.

2.1.5 New fire zone clearing

One of the problems in old version of our police was the fact that clearing all of the area around a fire zone for police forces took a lot of time and so the fire would be distributed the time clearing was done. Our solution was to not clear all of the roads around a fire zone, instead, our police finds some points around a fire and only clears the path to them. These points are chosen in a way that it is guaranteed that our Fire Brigades positioning from them can control the fire. This will reduce wasting in time and energy.

2.1.6 Cluster Dividing method

The main feature we are working on, is a strategy we call Cluster Quartering. In this strategy, in precompute time, each Police agent divides its cluster to 4 parts. Then the middle point of each quarter is found and the roads connecting them is selected. After the precompute, when the police is choosing roads for clearing, the most priority is with these roads. After 20 to 30 cycles, while police has done its normal tasks, this strategy has made a connection between different points in each Police Force's cluster. So if a fire starts in this cluster, there would always be a way around it to control it.

2.2 Fire brigade team

Since the FireBrigade's features of our codes are well known to readers, only the newly changed features will be named in this article.

Our most important limit in designing our algorithms is the short think time of server (0.5 – 2 sec).

2.2.1 No communication Strategy

This year, the main focus of fire brigade agents is on no communication map. It was obvious in 2014 robocup competition that most of teams lost lots of points in this strategy. So we've decided to develop special strategies for no communication maps. In the beginning of our strategy, we have developed a method to make a fire brigade escape when they get lost in a fire zone. Then police forces put a special agent with a task to move around fire zone in order to help fire brigades update their world model.

2.2.2 Low and full communication strategy

Optimization of water consumption is significant during natural disasters. Because of this, we have manipulated positioning to conserve energy. Last year, S.O.S developed two methods: 1-visibility positioning and 2-greedy positioning. In last year competition it was seen that all agents got visibility position because of some errors in the strategy. So we decided to improve our strategies to modify this issues.

Because of their wrong information, fire brigades may be trying to extinguish an unsuitable building. Visibility positioning conserves water uses but in some cases, getting an appropriate visible position leads to waste cycles. So we define an optimum number between greedy and visibility position, and different types of maps will have different numbers. The numbers and how deciding between visibility and greedy positioning is as described below:

- No Communication
  - Greedy Positioning: upd [1] < visibilityTime [2] && upd < 10
  - Visibility Positioning: upd > visibilityTime && upd > 10
- Low Communication
  - Greedy Positioning: visibilityTime > 5 && upd < 10
  - Visibility Positioning: upd > 10
- Full Communication
  - Greedy Positioning: (visibilityTime / greedyTime[3])<upd && upd < 10
  - Visibility Positioning: upd > 10

2.2.3 Fire Zone Selector

Last year, S.O.S defined 3 types of fire brigade,

1- Searcher
2- Extinguisher
3- Free

In this year we added additional strategy and modified last year's methods to have a better distribution of fire brigades, so we force searcher fire brigades to stay in their cluster and first priority of extinguisher is their own cluster and the neighbor cluster. Since probability of re ignition due to aftershock exists in map, the extinguisher fire brigade will not leave their cluster when fire zone is located in farther clusters. At last, free fire brigade will not belong to a special cluster and has permission to extinguish fiery buildings in everywhere. From this point of view, distribution of fire brigades will have a significant effect on handling small fire zones, and to achieve that an important score will be considered for small fire zone.

2.2.4 Building selector

The main Concentration of our efforts will be held on selecting the best fiery building for extinguishing. This selection has a great impact on fire propagation in map. To have a better selection we first remove some awful strategies that was used in older versions of our code, like selecting the building inside the map, selecting big buildings, pre extinguishing and unburned islands, in the continues of that, we add some new strategy which is mentioned in below:

1- Location of fiery buildings in a fire zone and its distance to fire brigade's cluster is great way to set suitable score for buildings. This method guides fire brigades to distribute around fire zone to prevent and stop the propagation of fire in map. Equation () shows the formula to set score for building in a fire zone. In this formula, building of fire zone that are near to fire brigade's cluster get positive score and other buildings get negative score.

Ave = Average distance of all fire zone buildings to center of clusters

Distance = distance of a building to center of cluster

$$score = \frac{distance - avg}{S}$$

$$S = \sqrt{\sum_{i=0}^{n} (distance - avg)^2}$$

[1] Upd = Competition Cycle – update cycle of Building

[2] VisibilityTime = Necessary time needed to get visibility position

[3] GreedyTime = Necessary time needed to get greedy position

2- Score of Pre extinguishing will be used only for buildings which have significant area compared to other buildings. Definition of a big building is related to ratio of building's area to average area of all buildings in the map.

3- Setting negative score for big buildings with high temperature: in order to save water and time during competition we put negative score for big buildings in special positions.

4- Early ignited building: low level of water is needed to extinguish early ignited buildings due to low temperature of the building, so we pay attention to this kinds of buildings.

These methods are only some examples of methods to find excellent building other method like random building, critical temperature and etc. will be developed.

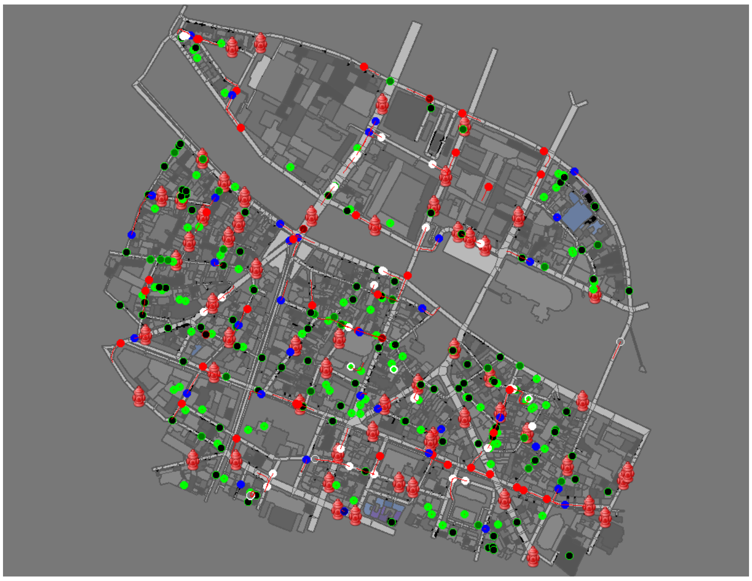
2.3 Ambulance team

Decision making has been changed to centralize deciding, based on each AT's current task the cost of rescuing each civilian by each AT and etc. In our algorithm, each will have two targets to decide between them. One of them is chosen by the Ambulance center. The agent also chooses his target and then decides between these 2. The reason that we find 2 targets is to only rely on center deciding but to also consider problems caused by reachability errors and noise.

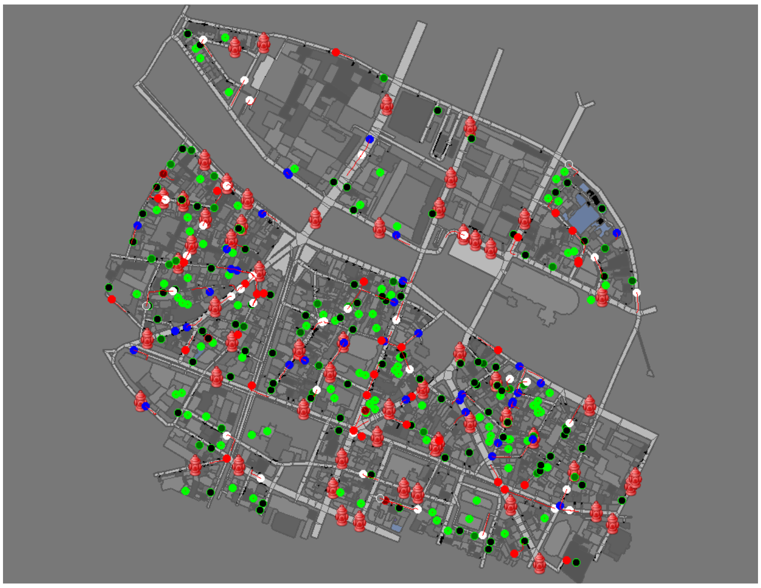
The most of important issue of ambulance agent last year was dead time error, we used a particular filter to determine dead time but the results of past competition showed some mistakes in evaluating dead time. So we decided about utilizing new learning methods like using decision tree to determine the dead time of civilians.

Last year task assigning was individual and every civilian had only one ambulance agent to rescue, but in last cycles of simulation in one map it was seen that most of civilians were not chosen because of low remaining cycles. So we lost the score relating to them. In this year's strategy we want to define new situation that ambulance agents will work simultaneously to rescue civilians with low HP.

Our Ambulance Team has 15 deciding states. Since lots of teams have worked on fire extinguishing, the job of ambulance becomes most important and lifesaving. Even in real world, whilst and after extinguishing the fire, the goal is to save civilians. As seen below, in these maps what saved our team and made us the champions was our ambulance.



MRL Paris3- final 2014
Score : 122.956230
Point : 3



SOS Paris3- final 2014
Score : 127.143252
Point : 8

3. Agent communication

World model and disaster prediction of our strategy are similar to last year but in no communication map we will do some activity to update agent's information by their movement. For example we want to put searcher fire brigade only for turning around fire site to help other agent to select better fiery building. In this point of view we want to develop base strategies that agent could do all of their task same in no and full communication.

4. Software Architecture

Figure (3) and (4) shows the main structure of our team works on every agent from the past up to now. The main state of every agent is shortly describe in figure (3) and in this year we will do all of our effort to develop best strategy that is useful and we hope to utilize from this strategy in real natural disaster.

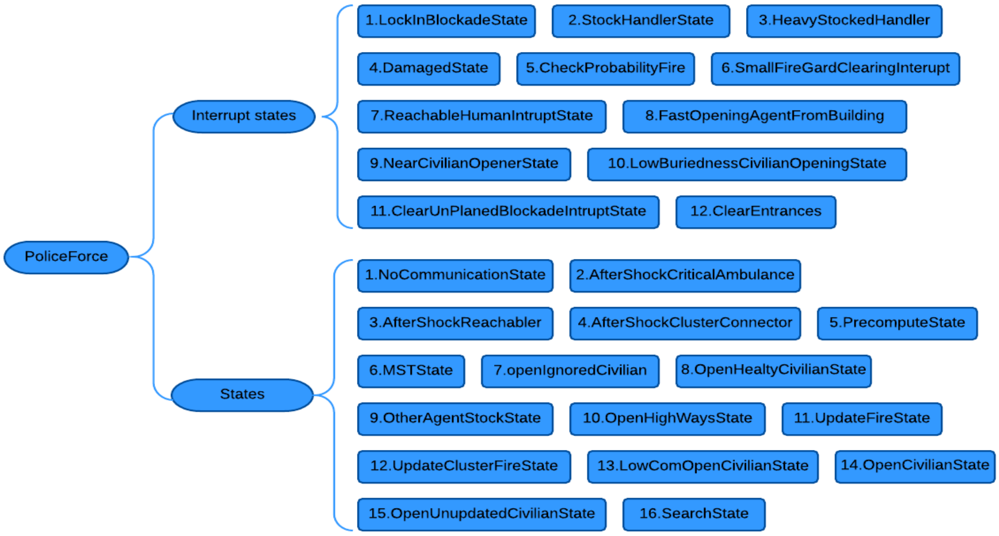

PoliceForce
Interrupt states
1.LockInBlockadeState
2.StockHandlerState
3.HeavyStockedHandler
4.DamagedState
5.CheckProbabilityFire
6.SmallFireGardClearingInterupt
7.ReachableHumanIntruptState
8.FastOpeningAgentFromBuilding
9.NearCivilianOpenerState
10.LowBuriednessCivilianOpeningState
11.ClearUnPlanedBlockadeIntruptState
12.ClearEntrances
States
1.NoCommunicationState
2.AfterShockCriticalAmbulance
3.AfterShockReachabler
4.AfterShockClusterConnector
5.PrecomputeState
6.MSTState
7.openIgnoredCivilian
8.OpenHealtyCivilianState
9.OtherAgentStockState
10.OpenHighWaysState
11.UpdateFireState
12.UpdateClusterFireState
13.LowComOpenCivilianState
14.OpenCivilianState
15.OpenUnupdatedCivilianState
16.SearchState

Fig (3): Sate Architecture of S.O.S Police Force

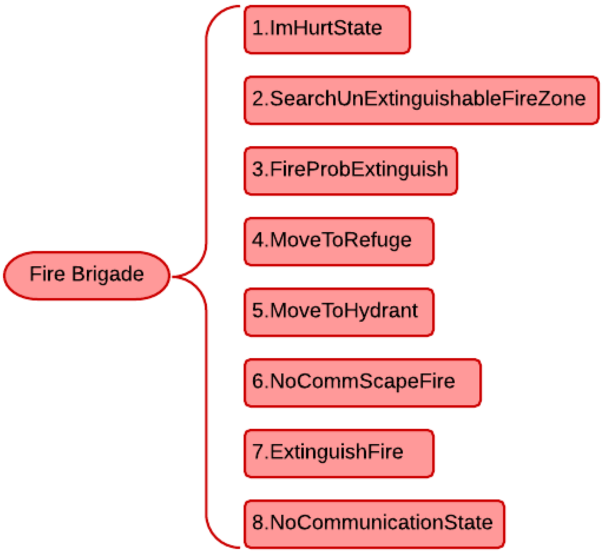


Fig (4): Sate Architecture of S.O.S Fire Brigade

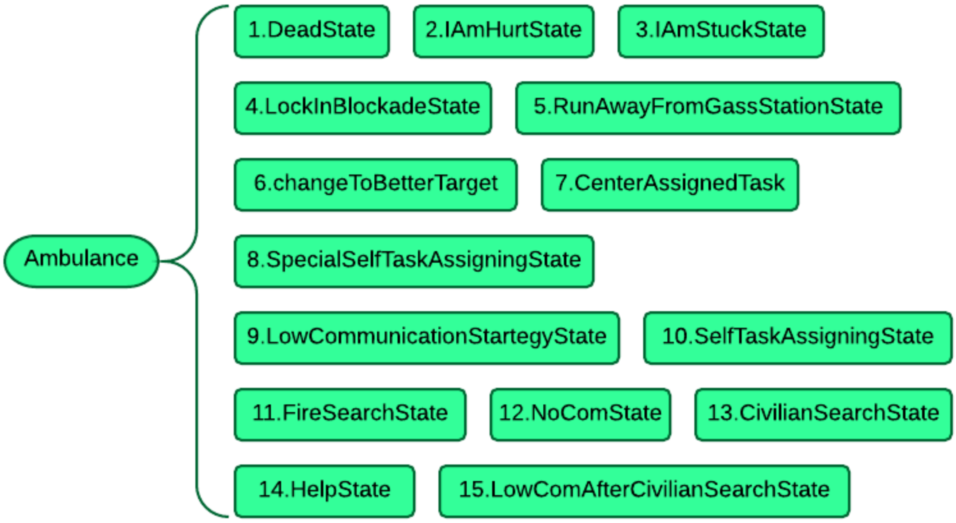


Fig (5): Sate Architecture of S.O.S Ambulance team

5. Software Tools

We utilize eclipse as part of our IDE and Ubuntu as operating system because of its high performance and finally we use SVN for code version control. And also we provide some other tools for debugging our strategies and base [2]. In this years, the most useful tool was Agent World Model Viewer, thus we decide to improve our team capability in this field. Currently we increase our layer to 125 that are responsible for different strategies.

6. Special Thanks and acknowledgements

Special thanks to all of our old team members, who had a big role in achieving today's code.

7. References

[1] A General Computational Recognition Primed Decision Model with Multi-Agent Rescue Simulation Benchmark by Alireza Nowroozi; Mohammad Ebrahim Shiri, Assistant Professor; Angeh Aslanian; Caro Lucas, Full Professor

[2] Modaresi, M et.al: S.O.S. Team Description Paper Proceeding of Robocup 2014.

[3] Golshahi, Y et.al: S.O.S. Team Description Paper Proceeding of Robocup 2013.

[4] Modaresi, M et.al: S.O.S. Team Description Paper Proceeding of Robocup 2012.

[5] Markari, A et.al: S.O.S. Team Description Paper Proceeding of Robocup 2010.

[6] Hashemi, B et.al: S.O.S. Team Description Paper Proceeding of Robocup 2009.

[7] Cormen, T., Leiserson, C., Rivest.: Introduction to Algorithms MIT Press, Cam bridge(2000)

[8] Robotic Rescue Simulation league Rules