

RoboCup2015 – Rescue Simulation League Team Description <Apollo (P.R. China)>

< Zhipeng Wei, Zhidong Wu, Huifei Liu, Qichang Li, Ruizhi Pu >

Nanjing University of Posts and Telecommunications , China
Apollo Robotics Innovation Lab , NUPT
wzpnjupt@gmail.com

Abstract. This paper describes the main features of the Apollo-Rescue rescue simulation team that is going to participate in the RoboCup 2015. In the past year, a lot of work has been done. In this paper, a new decision-making module will be introduced and a modified path planning algorithm for complex and dynamic environment will be covered. Moreover we have realized an effective fire control algorithm aimed at the dynamic partition. We have also used new viewer and logger system to debug our software.

1. Introduction

Apollo has actively participated in the RoboCup 2013 and achieved 8-th place last year, which was the best performance for us. The previous year, we had participated in two competitions. After all these competitions in 2014, we made an annual conclusion, found some major problems during these competitions and tried more to improve the performance and overcome the weaknesses of our code. In the first, we improved the decision-making procedure of agents. Different strategies were used to solve task allocation [1], [2] problems for different kind of agents. A multi-agent task allocation strategy based on artificial immune [3] system was used to solve task allocation problem for Ambulance Team, a partitioning-based task allocation strategy was used for Police Force and a convex hull based method were introduced to help Fire Brigades to control the fire spreading. By means of these efforts, in 2014 competitions, we expect to become a high quality rescue team in the near future. More details will be introduced below. Moreover, we have developed a monitor which is used to debug and visualize different aspects of agent behavior. By means of these efforts, we expect to become a high quality rescue team in the near future and achieve better performance in 2015 competitions. More details will be introduced below.

2. Dynamic Partition

In the past, we created static partition based on K-means algorithm, which was done during pre-compute and won't change before simulation finish. But later in the

simulation, urban disaster would get much worse like too many buildings burnt and agents got stuck or die accidental. So the static partition will not meet the requirements of task allocation effectively. For the further promotion of working efficiency, dynamic partition is introduced in the simulation. We recreate partitions in the specific time. In brief, these partitions recreated are valuable remainder areas that don't burn out yet, and the working platoon agents should be reassigned to these partitions to do corresponding jobs. What's more, the partitions will be recreated for the reassignment of resources in the harsh environment, just like the fire is out of control and so on.

3. Agents

We introduce a task cooperation mechanism based on partition. Partition is a very useful way to organize tasks. Agents can focus on a limited task in every partition which allows us to assign resources efficiently. Agents will enter the neighboring active partitions which still have rescued value when they have finished tasks in their own partition or their partitions have been deserted because of the fire out of control. By this way, agents can assign tasks efficiently and we can prevent them from having the same behavior prematurely.

3.1 Fire Brigade

3.1.1 Gray Prediction

Due to the randomness of fire in the relief process, specific spread trend and details unknown, so the entire process of building temperature change can be seen as a gray box, according to the display of results, which is the temperature of the buildings, we are now using the grey prediction [5] to get the next cycle prediction value of temperature. Grey system by sampling the latest value as the sequence to estimate, therefore according to the process of the change of the temperature, dynamic characteristics and output algorithm rules will also adaptive, avoided due to the large number of cumulative variables and possible errors. The team used the grey prediction algorithm for $GM(1,1)$ (Grey Model (GM)), GM represents one of the grey model (1,1) abbreviation, is representative of this model is based on the first-order differential equation with single variable, this single variable and the temperature of the building.

The original building temperature data sequence is as follows:

$$T^{(0)} = \{T^{(0)}(1), T^{(0)}(2), T^{(0)}(3), \dots, T^{(0)}(n)\} \quad (1)$$

According to the gray principle calculation get the final prediction (solution) represented by the following formula:

$$\hat{T}^{(0)}(k+1) = \hat{T}^{(1)}(k+1) - \hat{T}^{(1)}(k), k \in N \quad (2)$$

Then we can get the corresponding condition prediction of the buildings on fire.

Gray prediction is the point of the method of curve fitting extrapolation. Data sequence determines the length parameter of first order differential equations in A and B. Experience has proved that the accuracy of prediction, and not increased with increasing length sequence of join operations, sequence acquisition but depends on the recent time. Unburned building temperature gray prediction, if the prediction results show that the 2 cycle within the building will reach a temperature above the

ignition temperature, namely the 2 cycle combustion, so when the need for pre watering action on the building.

3.1.2 Fire Control Strategy Based On Block

The block holds the characteristics of slowing the spread of fire in the map. Therefore, we used a fire control strategy based on block. We divide blocks mainly through the characteristics of the buildings and main roads which are a road of high priority. We obtain the average width of the map firstly to get a base width. When the width of the road is wider than base width, we make the main road according to the angle between two roads.

In the past, our fire strategy often neglected the barricade factors and caused the situation that a lot of fire points are inaccessible. So in cooperation with the task of Police Force, we can stop the fire in the block peripheral actively. It can avoid wasting of extinguishment resources to achieve the purpose of fire control.

3.2 Ambulance Team

Last year, we carried out a substantial improvement on the AT code, perfect some low-level tools such as life prediction and etc. We made the quantitative analysis in targets selecting and civilians' locating.

3.2.1 Target Selection

In terms of targets' selection, spending too much time moving to target, saving a civilian that could not survive or making agent himself injured due to rescue would have a serious impact on the final score. Combined with several factors such as distance, partition, targets' reachable degree, civilians' healthy degree and surrounding environment for stuck agents, the target selector for AT would select the optimal target. There are two modes for optimal target selection, one is greedy mode which could select local targets in partition, and the other is auction mode based on auction algorithm [5] which could select global targets. An evaluation system has been established for the mode conversion. The evaluation result for the current map could be obtained by the number of available AT, discovered civilians, the fire spreading condition, the map reachable degree, the current simulation phase and so on. The rescue strategy will be determined by this result to reach the best result. In the old code, the static partition based on K-means algorithm had always been the emphasis on rescue strategy. However, this year, dynamic partition has been added on the basis of static partition. With dynamic partition, we can ignore the valueless places to ensure AT will work in the most valuable area all the time. If the temporal summation of all ATs' executing the rescue command is denoted as VT and the temporal summation of all ATs' executing other commands is denoted as OT , the $\eta = VT/OT$ expresses the overall efficiency then. Ceteris paribus, the score of each map was positively correlated with η .

3.2.2 Civilian Search

In RCRSS, the rescue of the civilians is a significant part. The survival of the civilians directly determined the score. So it's necessary to locate the civilians as soon as possible and rescue them. According to the emergency mechanism in the

simulation platform, when the civilian is conscious, there is 10% chance for it to send voice message “Ouch” and “Help” in each cycle. As shown in Figure 1, when the receiver receives the message, it will create a circle centered at the coordinate of itself which the radius is the max voice distance. By computing the intersection for many times, receiver will gradually narrow the trapped possible range down to locate the trapped civilian quickly. In Figure 1, comparing the new search process with the old one, we can clearly find that the quantity of civilian which is found during the search process is increasing significantly, which helps AT to rescue in time.

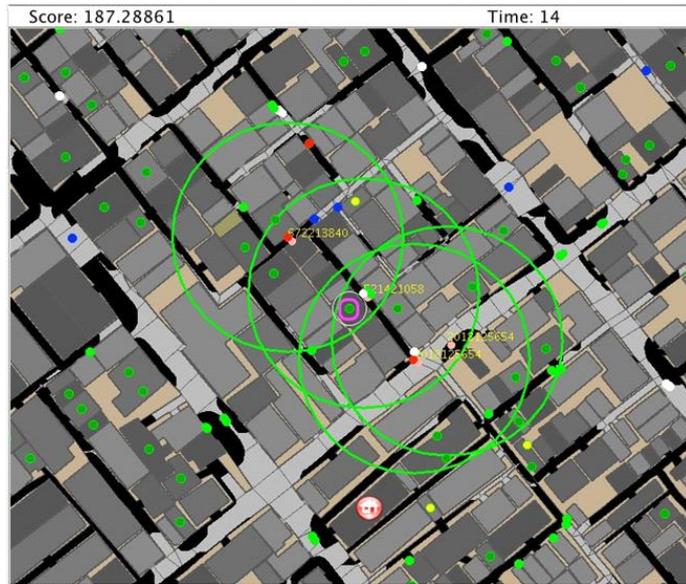


Fig.1. Civilian-Search Module

3.3 Police Force

In RCRSS, police force is responsible for clearing roadblocks. If the clearing effect is not good, the residual roadblocks will greatly hinder other agents' action, and the agents are easily adsorbed into the residual roadblocks, thereby significantly reducing the overall rescue level. It is a meaningful issue that how to effectively clear the roadblocks by using the rectangle and the agents will not be blocked easily after clearing. In the latest code, we proposed a clearing method based on vector weighted sum [6]. Based on vector weighted sum method for node model midline, police force determines the final leading line by the searched path. According to the leading line we can improve the clearing method for roadblocks and improve the efficiency of road clearing.

The path leading to target roadblocks created by the A star algorithm, assuming the path is composed by m nodes, and the path composed by m nodes can be indicated by collection $N = \{n_1, n_2, n_3 \dots n_m\}$; there are $m-1$ adjacent edges among the nodes, and we can use collection $E = \{e_1, e_2, e_3 \dots e_{m-1}\}$ indicate them. The first leading line g_1 can be created by the connection of p_1 and p_2 , among them p_1 is the center point of head

node which comes from the collection N , and p_2 is the center point of adjacent edge e_1 which comes from the collection E , and so on. At last, the connection of the center point of e_{m-1} and the end node p_{m+1} creates the last leading line g_m . So we can get the leading line collection $G = \{g_1, g_2, g_3 \dots g_m\}$ and the center point collection $P = \{p_1, p_2, p_3 \dots p_{m+1}\}$. From collection P , the first vector v_1 is created by point p_1 and p_2 , p_1 is the start point and p_2 is the end point, and so on. At last we can get the vector collection $V = \{v_1, v_2, v_3 \dots v_m\}$. G , V and P can be determined by steps above, and vector weighted sum is based on these collections. Specific method is as the follow program flow chart.

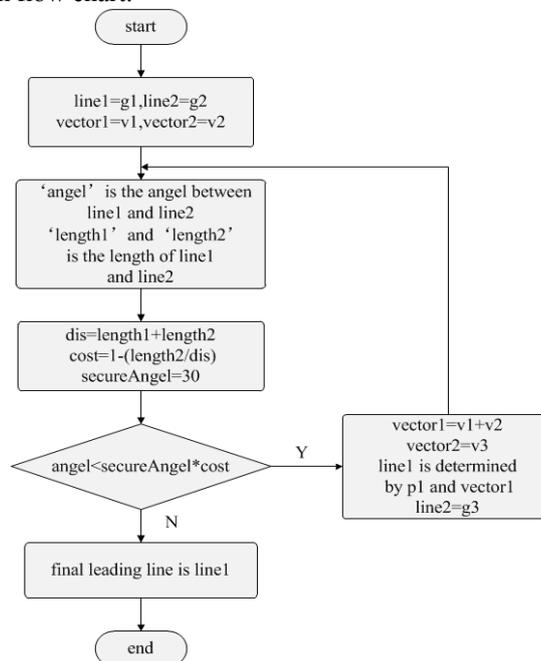


Fig.2. Flow chart of vector weighted sum

Then it will determine a final leading line. This improved method is applied to the system and police force can effectively clear roadblocks along the final leading line. Not only making the path clean, but also improving the clearing efficiency.

We apply two methods to simulation map Kobe1 and select a cleared path (the former does not use improved method but the latter dose). The results are as follows.

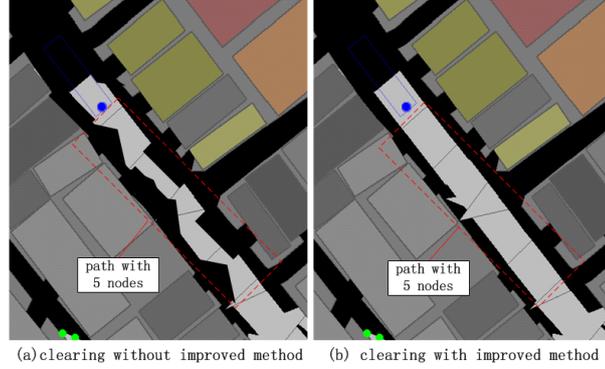


Fig.3. Contrast between clearing with improved method and without

Just as the figures above, the selected path is composed of 5 nodes and we can calculate the pass rate of each node after clearing. The pass rate formula is as follows.

$$passRate = 1 - \frac{bloWidth_Left + bloWidth_Right}{nodeWidth} \quad (3)$$

The parameter *passRate* is the pass rate of a node, *bloWidth_Left* is the average roadblocks width of node left side and *bloWidth_Right* is the right side. The parameter *nodeWidth* is the average width of a node. Each node pass rate is different and we can get the node pass rate after clearing as Fig 4 shows.

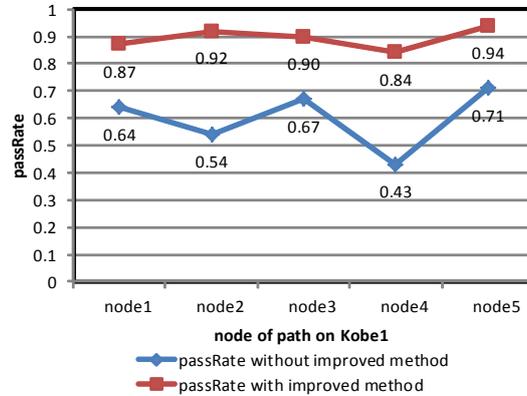


Fig.4. The contrast of node pass rate

The parameter *passRate* is a very important index to the whole rescue efficiency. As we know from figure above, the *passRate* using improved method is higher than the *passRate* not using, which directly proves the improvement of clearing method.

4. Communication

Communication is one of the most important underlying structures in RCRSS, and it plays an important role in agent cooperation, world perception and decision. This

year, we focus on increasing the Channel utilization and improving the message delivery with no communication.

4.1 Optimization for the utilization of Channel

There are lots of Entity IDs in the message. These fields are long but only contain little information, so we can create an index, and send only the index of the Entity ID in the message. In this way, we can decrease the bandwidth which each message occupies significantly. In the past, we bind the message to the channel. However, the sending frequency of different message differs, when there are several channels, we can't make the most of the channel bandwidth, so we introduce the dynamic channel sending into it. According to the estimate of the channel load, we decide the channel which we send messages.

4.2 Communication Strategy Under Severe Environment Based On messenger

Under the only Voice communication environment, we perform message passing through messenger, in the simulation process, we set point in the partition, agent within partitions go to the rendezvous point for communication by saying in the pre agreed time period, in order to promote the message transmission in different partitions, we selected a number of PF as a messenger in the agreed time period to a different partition set point for the exchange of information.

5. Software

5.1 Software Architecture

For every agent, we create different job lists, which added with common jobs and specific jobs, such as, ExtinguishFireJob for Fire Brigades, ClearPathJob for Police Forces and so on, according to their priority. An agent will choose one job to execute in each time step. In addition, we use logger to log job state, time record in order to debug, solving the problems like overtime. At the same time, we can make a quantitative analysis of the data of agent decisions to optimize the life prediction and the civilian value calculation.

5.2 Tools

Viewer and the Logger are used to detect the code's quality and efficiency. We modify the particular map by the map editor to simulate the scene that debugging requires in certain scenarios to raise the efficiency, such as the gas station explosion scenario. Automated scripts are applied to test the large map, which shorten the time and manpower.

6. Acknowledgements

At last, we are very grateful to the server developers for maintaining the rescue simulation server and their kindly technical supports. In addition, we have great thanks to ZJU-Base, SEU-Jolly, CSU_Yunlu for their kindness help and technical communication. We show our gratitude to our teacher Z.W. Liang and other members of Apollo Robotics Innovation Lab who provide us a research environment and inspire our work. We are appreciative of anyone who has support us and inspire our work.

7. References

- [1] Carmen Gervet and Slim Abdennadher. Multi-agent planning for the robocup rescue simulation - applying clustering into task allocation and coordination. In International Conference on Agents and Artificial Intelligence (ICAART), pages 339–342, 2012.
- [2] Benjamin Balaguer, Stephen Balakirsky, Stefano Carpin, and Arnoud Visser. Evaluating maps produced by urban search and rescue robots: lessons learned from robocup. *Autonomous Robots*, 27(4):449–464, 2009.
- [3] Helen Flynn. Machine learning applied to object recognition in robot search and rescue systems. Master’s thesis, University of Oxford, September 2009.
- [4] Masaru Shimizu and Tomoichi Takashi. Training platform for rescue robot operation and pair operations of multi-robots. *Advanced Robotics*, 27(5):385–391, March 2013.
- [5] T. Service and J.Adams, “Coalition formation for task allocation:theory and algorithms,”*Journal of Autonomous Agents and Multi-Agent Systems*, vol. 22, pp. 225–248, 2011.
- [6] S. Liemhetcharat and M. Veloso, “Modeling mutual capabilities in heterogeneous teams for role assignment,” in *Proc. Int. Conf. Intelligent Robots Systems*, 2011, pp. 3638–3644.