RoboCup 2015–Rescue Simulation League Team Description

CSU_Yunlu(China)

Fu Jiang, Xiaoyong Zhang, Tao Cao, Xin Xu, Yong Meng, Xiaoyu He, Mingzhu Shi, Huan Tian, Hongbing Luo, Lu Tang, Jun Peng

School of Information Science and Engineering, Central South University
22# South Shaoshan Road, Tianxin District, Changsha
Hunan 410075, China
Jiangfu0912@csu.edu.cn
http://robocup.csu.edu.cn

Abstract. The main work about CSU_Yunlu Rescue Agent Simulation Team within a year is introduced in this paper. Firstly, an estimation function based on A-Star algorithm is leveraged to determine whether a road is blocked or not. Then the workflow of ambulance team agent will be introduced roughly below. The new clear strategy based on POS team is also proposed.

Keywords: RoboCup Rescue; A-star Algorithm; AT Workflow; Clear Strategy

1 Introduction

The purpose of Rescue Simulation League is to decrease the life and financial losses caused by nature disasters such as earthquakes, floods and etc. In order to achieve this goal, a large urban disaster is simulated. The simulation platform, RoboCup Rescue Simulation System (RCRSS), can simulate the real world's limits and problems as accurately as possible [1]. And the system also exploits the technologies of artificial intelligence and data mining.

Our CSU_Yunlu team has began to participate in RoboCup Rescue Simulation competition since 2006, and we won the third place in RoboCup 2014. Major improvements on our rescue policy and code since 2014 are summarized in the remainder of this section [2].

The main contents of this article are listed as follows:

1) A new mechanism was introduced to determine whether the road is passable or not, which will be used in the A-star algorithm.

2) The workflow of AT agent.

3) A new clear strategy based on POS team was used.

2 Path Planning

A* search (pronounced “A-star search”) is the most widely known best-first search algorithm. It evaluates nodes by considering the cost to reach the node *g(n)*, and the cost to get from the node to the goal *h(n)*:

$$f(n) = g(n) + h(n).$$

f(n) = estimated cost of the cheapest solution through *n*[3]

In the function *h(n)*, we use Euclidean distance to compute and compare the cost. As the main part of this algorithm, *g(n)* is the path cost from the start node to node *n*. When we consider a single road, there are two cases, it can pass through or can’t. If the road is passable, we simply add its length to *g(n)* and turn to the next node. Otherwise, we will add Double.MAX_VALUE. Due to the existence of blockades, it is important but hard to determine whether a road is passable or not. In our new mechanism, we propose a method to solve this problem.

For the most commonly seen Road in the RCRSS, the Road can be consider as a quadrilateral with two passable opposing edges, but the remaining two edges can’t pass through. When an earthquake happened, blockades spread from the boundary to the center, as it shown in figure 1, A and B are blockades, line 1 and line 2 are the unpassable edges of a road. For blockade B, we can find the nearest vertex N away from line 2. Then we make out the segment perpendicular from point N to line 2, and find the pedal P. If the straight-line distance between point N and P is greater than 500mm(that is the radius of an agent), this means agents can bypass blockade B. Then we find out the middle point of point N and P, and mark it as escape point E. Otherwise, there is no escape point E.

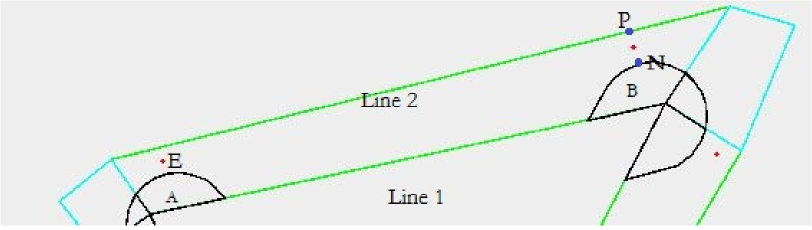

Figure 1. blockades spread from the boundary to the center.

The pedal P can be located in other blockades, as it shown in figure 2. Point P is the pedal that related to blockade A, but it is located in blockade B. In this case, the escape point E is no longer the middle point of segment NP. But we can find the intersection point I of blockade B and segment NP. And if the length of NI is greater than 500, we can find the escape point E, which is the middle point of segment NI. Otherwise, there is no escape point.

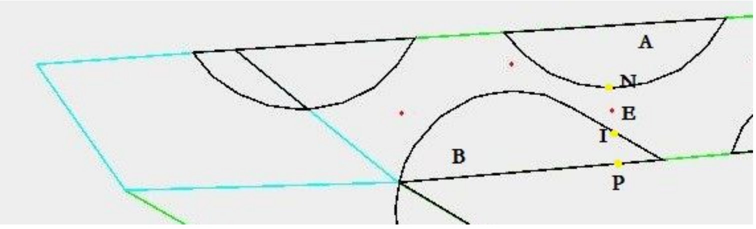

Figure 2. pedal P be located in other blockades.

A blockade can be located in other blockades, too. In figure 3, blockade A is located in blockade B, and we find two escape points E1 and E2. Among them, E1 belongs to blockade A and E2 belongs to blockade B. It is clear that E1 is a useless point, so we discard it. And we make E2 as the escape point of blockade A. So a single escape point can have one or more blockades.

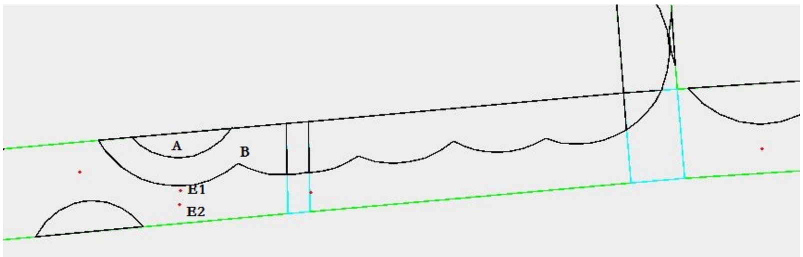

Figure 3. blockade be located in other blockades.

For a single road (suppose the agent has already know all blockades of this road), we can find a list of escape points and each escape point can have one or more blockades. We keep a copy of blockade list of this road, traversal the escape point list and remove its blockades from this copy list. When the traversal is done, if the copy list is not empty, it means that some blockades have no escape point and the road is an unpassable one. If the copy list is empty, there are still efforts to be done.

3 Ambulance Team

The main task for an ambulance team is to rescue the citizens in the building as much as possible and send them to the refuge. Therefore, the task distribution is extremely important and it needs an accurate and efficient decision. In recent years, we optimized some modules such as work partition, task update, task allocation, rescued object management, etc. And recently, we made some following changes based on the original behavior.

1) We set the priority of rescue agents, thus it will be irrelevant to the rescue of citizens.

2) When selecting a rescue target, we must make sure that the previous selected target is always being considered, which can guarantee the priority. We will write a zero information channel massage, when it is used by an ambulance agent to rescue people and at the same time new agents go into the building, and then they will be told whom they are rescuing. What to achieve next is to mark this building as searched according to the specific situation of the other agents at the entrance of the building and to solve the problem that two agents get stuck at the entrance and can’t get out of the building due to the narrow entrance.

3) Take measures to prevent ambulance team from getting stuck by roadblock and doing nothing.

4) After searching a cluster, we only need to add the citizens into other clusters instead of the buildings.

5) The number of buried humans in a cluster will be reset and recalculated in each cycle to guarantee the dynamic reflection of the present refuges’ distribution.

6) When an agent is dying, it should stay at the refuge. And when a gas station

explodes, all the agents around will get badly hurt, so they should stay away from it.

7) When calculating the rescue time, we will add correction time on the basis of rough calculated time. And there should be a relationship between the correction time and the rough calculated time, in general it is a proportionality relationship. For instance, when the subject is beside, the correction time is zero. By this way it can make sure that all the subjects use the same handling method.

4. Police Force

The main task for a police force agent is to clear the blockades. The performance of police force agents will affect the rescue efficiency. Therefore, we add some modules such as work partition, task allocation, priority of the clean targets, etc. At the same time, we optimized some modules such as clear method. Recently, we made the following changes based on the original behavior.

In order to get more rational allocation of tasks，all the police force agents are distributed in their own work area. We use K-means++ algorithm to take the partition. In K-means++ algorithm, a cluster is a group of buildings which are closest to the cluster center. This algorithm cost very short time to divide the map so that we are able to perform more computations during preprocessing.

In our new strategy, we gave a priority to the clean targets. the police force agents will choice to help other agents(fire brigade agents and ambulance agents) who get stuck by blockades as their preferential task, so that PB and AT can work more efficiently. Afterward they will go to clear the critical areas (the entrances of the refuge, crossroads etc.) and then they will clear some important areas such as the entrances of the building. Meanwhile police force agents will broadcast the messages about the building on fire.

In our old strategy, the PF agent clear all blockades around them. We need to clean up the middle region, so that other agents can pass through. For some blockades, there is no need to clear them in the very beginning of the simulation at all. Briefly we usually choice the midpoint of the common edge (it also is the edge of the next area in the path) as our clear direction.

5. Fire Brigade

The rules of fire propagation are complex. We only use the radiated energy of fired building for decision in our previous code, and it leads to the limitation of the fire forecasting and decision-making. So we implement a simplified fire propagation model based on the principle of fire simulator and a particle based on online fire forecasting method.

We cannot get actual temperature of the building every cycle and conditional probability can't be used to empower the value of the particle in this situation. We can use the distribution of particles to predict the true state, then we can give the particle same weight value when it cannot receive the true temperature. When the particles are given same weight, they are in line with uniform distribution. We can estimate the true value with the expectation of the uniform distribution.

Updating of particles' temperature is calculated by equation 1:

$$present(t+1) = present(t) + \Delta t \qquad (1)$$

In equation (1), $present(t)$ is the temperature of the particle in time step t. Δt is the update value of particle and it can be calculated.

The prediction temperature can be calculated as equation (2):

$$T(t)_c = \sum_{i=1}^{N} present(t)_i / N \qquad (2)$$

$T(t)_c$ is the temperature of building in time step t. $present(t)_i$ is temperature of particle i in time step t.

We implement a fire simulator to simulate the propagation of fire. Due to the limitation of agent's perception model, it is difficult for an agent to grasp the overall situation of the fire; or information in their possession is outdated. With the use of this self-implement simulator, we can simulate the propagation of fire. So that fire brigades can update fire information by themselves.

To find the fired building earlier, we defined some critical areas. In the very beginning of the simulation (the top twenty or thirty cycles), fire brigades will search those critical areas. For those critical areas, they are crossroads, which are uniformly distributed on the map. Due to the small number of crossroad, fire brigades can quickly complete the search task. In order to avoid duplication of search, we assigned a static search area, which is divided according to K-means++ algorithm, for each fire brigade.

After the fire brigade understood the overall situation of the fire, dynamic partitioning is prosed. Then we can schedule fire brigades according to those fire zones. There are many scheduling schemes, such as scheduling by distance, or by the amount of needed water. Currently, we are scheduling by distance. However, in some maps, there are more serious wandering phenomenon. Our next plan is optimize the scheduling scheme of fire brigades.

6 Acknowledgment

This project has been carried out with the help of many individuals and support from School of Information Science and Engineering, Central South University.

Thanks POS team and MRL team, we wrote new clear strategy based on their code.

7 References

[1] RoboCup rescue website. http:/roborescue.sourceforge.net/

[2] Jiang F., Peng J., Zhang XY, Cao T.,Guo MH, Jing LL, Gao ZF and Zheng MF RoboCupRescue 2014-CSU_Yunlu Rescue Simulation Team Description. *RoboCup 2014 International Symposium*, Joao Pessoa, Brazil, July, 2014

[3] Stuart J. Russell, Peter Norving. Artificial Intelligence: A Modern Approach (M) .Third Edition. Upper Saddle River: Prentice Hall, 2010.93~94

[4] Peng J., Wu M., Zhang X., Xie Y., Jiang F., Liu Y. A collaborative Multi-Agent model with knowledge-based communication for the RoboCupRescue simulation. *International Symposium on Collaborative Technologies and Systems*, Las Vegas, USA, May, 2006, pp: 341-348.

[5] Peng J., Liu Y., Wu M., Zhang X., Jiang F. State Prediction based Dynamic Cooperation Algorithm in Multi-agent System. *Journal of System Simulation*, 2008, 20 (20), pp: 5511-5515.

[6] Liu R., Sun Z. A DBN Model for Fire Spreading in RoboCupRescue Simulation. Collected papers for RoboCup 2007 conference 2007(21)

[7] Guan D., Xu Y., Tan Y. Particle Filter Based Study of Fire Forecasting in RCRSS. Nanjing: Southeast University, 2009

[8] Hao X., Cheng M., Wong V. A Stackelberg Game for Cooperative

Transmission and Random Access in Cognitive Radio Networks. in Proc. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Toronto, Canada, September, 2011, pp: 411-416.

[9] Scutari G., Palomar D., Pang J., Facchinei F. Flexible Design of Cognitive Radio Wireless Systems. *IEEE Signal Processing Magazine*, 2009, 26 (5), pp: 107-123.