

RoboCupRescue 2015 - Rescue Simulation League

Team Description

Ri-one (Japan)

Yosuke Takeuchi, Yuta Muraki
Akira Kinose, Futoshi Sato, Marina Nakagawa
Naoki Nakayama, Satoshi Suemitsu
Tatsuki Fukumori, Yukino Watanabe

Ritsumeikan University, Japan
is0219sp@ed.ritsumei.ac.jp
<http://rione.org/hp/>

Abstract. This paper describes features of Ri-one, our team, for the RoboCup Rescue Simulation (hereinafter referred to as RCRS) in 2015. Last year, our team was developed by selecting the actions of agents by making use of the center agent giving commands, matching ambulance team agents with civilians, extinguishing fires using the convex hull which determines the degree of influence of a fire to its surrounding buildings, and enhancing the efficiency of police force agents rescuing humans. In contrast, our team this year propose several methods of conveying information through dynamic clustering, suited to each situation. Additionally, we worked on several individual implementations for the following: modifying the manner in which an ambulance team agent approaches a civilian, setting extinguishing positions for fire brigade agents, and producing an even distribution of police force agents around the map. We will explain each of these approach method and the ways in which they are applied to the agents.

1 Introduction

The RoboCup Rescue Simulation is a multi-agent simulation for disaster relief. The RCRS server simulates various circumstances imitating a city after a disaster. The aim is to make use of the virtual agents in order to rescue buried victims from under blockades, and to extinguish fires causing buildings to go up in flames.

Last year our team implemented methods such as using a center agent which broadened the potential number of actions able to be taken by individual agents, matching ambulance team agents with civilians, extinguishing fires on buildings using the convex hull, and improving the efficiency of police force agents rescuing civilians. This year, we worked on a more effective application of the ambulance team agents, specified the direction in which water is applied when a fire, and dispersed police force agents around the map in order to maximize efficiency. Each of the chapters will describe the following contents:

- Chapter 2: Dynamic clustering methods using k-means++ and the Discrete Voronoi diagram
- Chapter 3: Concrete implementations based on Communication methods introduced last year
- Chapter 4: Implementation of each Agent
 - 4.1. Effective methods for ambulance team agents by introducing Ant Colony Optimization.
 - 4.2. Searching optimum areas for fire brigade agents to extinguish fires.
 - 4.3. Enhancing the efficiency of obstacle clearances by distributing police force agents evenly around the map.

2 WorldModel

In this chapter, we intend to propose a solution to the problem which we encountered last year of clustering dynamic information in area partitions, by simultaneously using k-means++ and the Discrete Voronoi diagram.

The main issue with the normal k-means technique was that there was a possibility that approximated answers would endlessly become worse. In order to solve this, k-means++ resets the centre of each cluster before implementing the recursive equation in order to improve computed values. A Voronoi diagram is one which divides the map into sections based on how close regions are in relation to ‘generator’ points which have been initially specified. The method used last year which created partitions according to the positioning of buildings with the k-means method, had too high an order. This meant that the clustering operation was only conducted once at the beginning and that we carried on using this value throughout, consequently not taking enough action with the dynamically moving situation. It was also seen that the approximated values were not very accurate.

From these evaluations, this year we introduced k-means++ to solve these problems of inaccuracy by its method of initializing the center of each cluster before performing the repetition formula. K-means++ also shows improvement in the precision of the values which are executed. By using this together with the Discrete Voronoi diagram which takes a significantly lower order to execute, we propose to cluster information more efficiently in accordance with the dynamically changing conditions.

The strongest advantage to this solution is that it can appoint the coordinates of moving agents and constantly changing fire locations as the initial ‘generator’ points. This allows us to repeatedly run the clustering calculation mid-cycle, which was difficult with the previous method which only used a single algorithm. We expect this new implementation to distribute agents more efficiently and actively, depending on the workload of each agent.

3 Communication

Communication is most important for agents on two points: to make a decision on which task is the most efficient, and to gather as much information as possible for judging what to do. Our team once proposed adding new information to each agent’s world model with specific commands from a center agent. This however seemed to leave unresolved problems as will be discussed further in the following paragraph. In order to solve these we mounted two improved implementations, such that the center agent no longer sends commands to all agents but only to individuals, and that the receiving agents can all act as senders to let the center agent know of their position.

As stated above, our Team Description Paper from last year proposed that “adding new information to each agent’s world model, consisting of several commands for agents to take specific actions” will improve efficiency. The center agent allowed for receiving agents to have more choices with their actions when no apparent action was available from their own world model. However it was not the case that all agents always took commands from the center agent even when available. In fact, given that the centre agent sent orders to all agents, there were cases where more than one agent tried to solve the same problem which lead to inefficiency. There also seemed to be some cases where agents could not operate according to the commands or their world model, in spite of there being actions which needed to have taken place.

Confirming the proposal from last year, we designated all agents to specific areas within the map so not to create overlaps and to increase efficiency with less movement. Within each area, the agent with the operation of highest priority is chosen as the leading agent of that area. These chosen agents then receive commands from the center agent, and make sure that they are carried out. If the agent is not able to finish the tasks on its own, it then conveys

the same information to the necessary number of agents in its area, making sure that there are no duplicates and that the commands are obeyed in the most efficient manner. We also implemented an algorithm which allows agents to send a notification directly to the center agent before aimlessly wondering the map looking for an operation to carry out, so that it can be allocated to an area where it is most needed.

As a result of these new implementations, agents are able to make multiple judgements from numerous information sources, increasing the efficiency. With our new method which allows agents sending a message to the center agent, we have seen success in dispatching agents to effective positions, significantly decreasing the amount of wasted time as when searching an operation by the conventional exhaustive search. However on occasion, factors such as noise and anomalous results of evaluation values of priority, obstructed the command and therefore was not successfully transmitted. Taking these factors into account will be the next step to enhancing the efficiency in the future.

4 Agent Skills

4.1 Ambulance Team

The main purpose of the ambulance team agent is to rescue injured victims, and to carry them to refuge. In RCRS, it is important that the ambulance team keeps victims alive as much as possible. In this section, we intend to propose the following three methods:

- Matching ambulance team agents with victims within its visible scope
- Leaving rescued civilians on the road to rescue other more fatal victims.
- Trusting each agent’s judgement on rescuing the same victims with others.

The first method refers to the rescuing priority between injured agents and injured victims. In previous years, we did not distinguish priorities between agents and civilians in the case of an ambulance team agent rescuing victims. Although both are important to be kept alive as survivors, it is better for an ambulance team agent to be rescued preferably considering that rescued agents will be able to perform more than rescued civilians. Once rescued, agents will be able to go to other victims who are in need of help.

The second method improves rescuing efficiency. When rescued victims are being carried to a refuge it is all well that this victim will stay alive. However at this same time, there is a possibility that other victims who need help may be dying. Based on this, we proposed that if the victim has enough life points to live after being rescued from inside a building or under an obstacle, ambulance team agents will leave this victim on the road with taking it to refuge, in order to help others in more urgent need of help.

When we put the latter proposal into practice, we have to make sure that ambulance team agents pass by the victim left on the road from time to time, in order to obtain health information on how many cycles it has left to live. For this to happen, we searched roads on which ambulance team agents frequently walked, and left the rescued victims in one of those positions by using the method of Ant Colony Optimization (hereinafter referred to as ACO).

Here is the explanation of ACO: the ACO is the algorithm based on habits of ants. An ant after discovering food goes back to its nest spraying pheromone on the paths which it passes, and other ants will then track this pheromone. Pheromone has a property to evaporate when time passes. To begin with, an ant who comes to stand at a junction will choose one of the routes randomly. Now assume that the walking speed of all ants is the same. If pheromone is left on a short route to the destination, other ants will track it and leave their track of pheromone on top of it before it evaporates. On longer routes, the pheromone will evaporate before other ants are able to track it. Since ants tend to choose routes where pheromone is more dense, the short route will be the most appropriate and most likely to remain.

In our team, we applied this ACO method to search not the shortest route but the route which ambulance team agents pass frequently. For instance, we increment the value of pheromone to the location of where an ambulance team agent is currently standing and decrement this value by each cycle. The more time passes, the more the pheromone goes up on roads which agents walk on quite often. This enables us to determine and leave rescued victims on roads where ambulance team agents frequently pass.

The last method of the three, refers the selection of rescuing. That is, we have decided to trust the decision of each ambulance team agent when deciding whether or not to save a victim which is already having help from another agent. Last year, our team introduced a way of analyzing victims who need to be rescued judging from various standpoints, in a manner similar to the Hungarian Algorithm. This was done so as not to cause overlapping selection of hurt civilians among agents. However the degree of damage which the victims are suffering from at any one time, is shown by the equation below:

A degree of damage when victim is buried = Initial degree of damage * The number of cycles passed since initial damage

Considering this equation, it is evident that we should be rescuing victims in few a cycles as possible. Given that it takes less time to rescue a survivor in need of help with plural numbers of agents, we speculated that duplication may not necessarily be a bad thing.

4.2 Fire Brigade

The purpose of the fire brigade is to extinguish fires caused by disaster. It is also important to prevent the spread of fire and to limit the damage by which it causes. Burning buildings raise the temperature of the surrounding buildings and gradually spread out. This means that fire brigade agents must extinguish the outer circumference of burning areas so that they prevent the spread of fire.

Fire is able to spread even while it is being extinguished. From this fact, it is evident that fire brigade agents must extinguish the outer circumference of burning areas and change their extinguishing position in response to the spreading of fires. In attempt to make this a more efficient process, we have said that fire brigade agents should decide on appropriate positions which should be extinguished with more urgency, so as to handle the rapidly spreading fires. In other words, we derived appropriate extinguishing positions taking into account the surrounding buildings with high temperatures, and the position of the agent in relation to these buildings. The main advantage of incorporating this strategy is that fire brigade agents are able to directly observe the spreading of fires, and extinguish them rapidly if taking the appropriate position.

Fig.1 shows our proposed method in more detail. Group A is the set of burning buildings in an extinguishable range for fire brigade agent, and Group B form a set of buildings which have a high temperature surrounding Group A. We connect the centre points of each of these buildings within each Group (such as we did making the convex hull) and make two figures. If any roads intersect the area which is in GroupB and not in GroupA, then this road locates a position outside of GroupA, and also in position to rapidly extinguish when fire spreads to GroupB. As a result of this, fire brigade agents will treat that area of the road as having the highest priority when preventin the spread of fires. If an agent sees more than one possible area where this applies, it is programmed to go to the nearest to itself.

Deriving the best positon for extinguishing allows fire brigade agents to minimize the expansion of damage caused by spreading of fire.

A problem yet to be resolved which was encountered in previous years with our team, was that when a fire brigade agent runs out of water, it leaves the spreading fire in order to

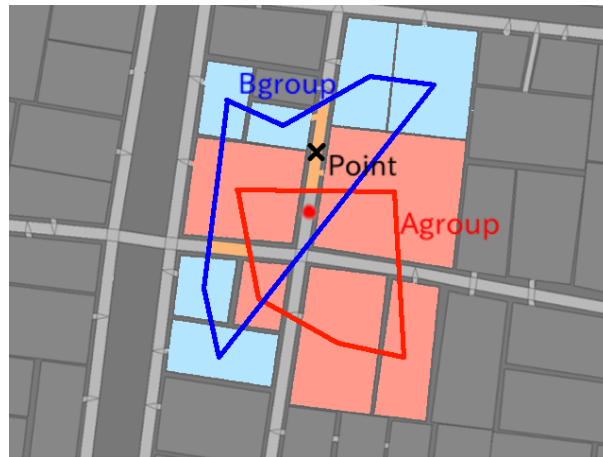


Fig. 1. Most appropriate fire extinguishing position

go to the water supply. This means that the fire carries on spreading rapidly while agents are gone, overall causing more damage than good.

This year, we have contrived a method which allows fire brigade agents to evaluate whether a fire is extinguishable by itself or if it would also need help from others. This refers to a situation when an agent catches sight of burning buildings. After making an assessment, if the result is that it is too difficult to extinguish by oneself, this agent shares the information of the target buildings to other agents, asking for help. When the other agents receive this information, they compare the priorities of the received operation, with that which they are carrying out at that moment. If an agent judges that the received operation has higher priority, it uses the information received and joins the original agent.

To perform this evaluation index contemplating whether it is extinguishable, we consider the following variables: the area of the building, the temperature, the material it is built of, the highest quantity of water the fire brigade agent can afford to use, and the residual quantity of water.

With the implementation of this method, we are presented with two major advantages. One is that fire brigade agents can extinguish fires on burning buildings with high priority, given that every fire brigade agent receives the information and makes their own comparisons. Another is that we are able to prevent fire brigade agents from excessive centralization to certain areas, by notifying each agent with the information of extinguishable buildings in advance.

4.3 Police Force

The main role of a police force agent is to clear blockades which are caused by disaster. The action of each police force agent influences all agents and civilians in the virtual city. All possible routes for every agent to take action, are determined by the range of passable roads. In order to efficiently perform the action which ambulance team and fire brigade agents execute, police force agents must clear the blockades efficiently.

The specification on how police force agents clear blockades was changed in 2013 from shrinking, to allocating a point and excavating in square shapes. This alternation meant that it was now better for police force agents to distribute its tasks amongst each other than to concentrate on clearing one blockade collectively. Taking this into account, this year we introduce a manner of distributing police force agents by clustering, which enables efficient

clearing upon the new regulation. For this we use the K-means++ algorithm once again by dividing the map into the number of police force agents, so that each agent has one area to patrol. This gives a good distribution of police force agents around the map, deleting many more blockades efficiently. As a result of this implementation, we succeed to enhance the efficiency of clearing.

An important fact which we have to bare in mind is that, each individual police force agent must remove blockades with greater accuracy than before, given that they are no longer shared tasks with other agents. Owing to this, we introduce a new way of clearing to solve this issue. Our team last year used the Point of Visibility navigation graph (hereinafter referred to as POV) (generating the shortest route to the target entity and appointing obstacles which interfered with this path). With last year's implementation, each agent computed the shortest path to its target entity, and evaluated whether or not there were obstacles in the way. In situations where they encountered blockades, the intersection between the POV line segment and the outline of obstacle was taken as point, and the agent would clear the blockage in the direction of the coordinates (Fig.2). Agents deleting blockades in this manner however, brought up obvious problems. since the shortest path is only considered with direct distance, it can be far different from the direction in which the agent would in fact like to walk in, which is along a path.

In order to resolve this problem, this year we took two points as S on the POV line segment (Fig.3), and implemented a method which cleared the blockade parallel to the line created by joining these two points together. This way we enable the agents to keep their line of direction without being influenced by their position relative to the blockades or shape of it.

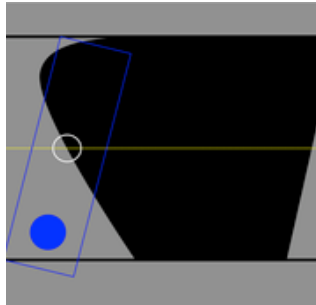


Fig. 2. Clearing towards the intersection of blockade and POV

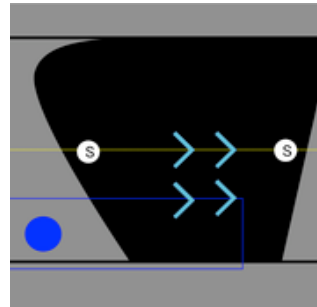


Fig. 3. Clearing parallel to the path

5 Result

Using various maps from the RCRS competition held in 2014, Table 1, 2, and 3 show the comparison between the damage made to buildings using our programs from 2014 and in 2015. Under the same conditions, Table 4, 5, and 6 show the comparison of the Civilian component.

This data allows us to draw the following conclusions. Regarding the damage made to buildings, the implementation this year do not show a lot of difference to the results obtained last year. The difference in points in the comparison of the Civilian component show great differences between maps, implying that the implementations are heavily dependent on the number and size of roads. Additionally, our implementation this year showed an increase in

points on comparatively larger maps. This is considered due to the new method of dispersing police force agents, resulting in clearing blockades with greater efficiency and discovering victims and the fires in wider areas.

Table 1. Comparison of Building damage on Preliminaries Day 1 and Day 2

	VC1	Eindhoven1	Berlin1	Paris1	Kobe2
Ri-one2014	0.280	0.122	0.942	0.976	0.923
Ri-one2015	0.347	0.127	0.892	0.972	0.900

Table 2. Comparison of Building damage on semifinals

	Berlin2	Paris2	Eindhoven2	Istanbul2	Kobe3
Ri-one2014	0.850	0.950	0.210	0.850	0.670
Ri-one2015	0.862	0.945	0.212	0.862	0.690

Table 3. Comparison of Building damage on finals

	Kobe4	Eindhoven3	Paris3	NY3	Istanbul3	Mexico2
Ri-one2014	0.963	0.220	0.990	0.577	0.180	0.813
Ri-one2015	0.960	0.230	0.990	0.540	0.180	0.790

Table 4. Comparison of Civilian Component on Preliminaries Day 1 and Day 2

	VC1	Eindhoven1	Berlin1	Paris1	Kobe2
Ri-one2014	206.680	221.335	131.366	200.672	82.777
Ri-one2015	198.742	216.902	127.188	191.622	81.360

Table 5. Comparison of Civilian Component on semifinals

	Berlin2	Paris2	Eindhoven2	Istanbul2	Kobe3
Ri-one2014	104.993	144.470	103.320	104.993	117.365
Ri-one2015	114.105	134.935	109.837	114.105	116.108

Table 6. Comparison of Civilian Component on finals

	Kobe4	Eindhoven3	Paris3	NY3	Istanbul3	Mexico2
Ri-one2014	159.810	29.235	83.687	61.280	27.047	205.973
Ri-one2015	158.215	29.573	74.060	56.765	28.060	197.615

6 Acknowledgment

We are deeply indebted to Chihiro Hatakeyama, Sho Okazaki, Ko Miyake and Tetsuaki Fujimoto for their advices.

References

1. K. Miyake, S. Oguri, M. Takashita, T. Fukui, T. Mori, Y. Takeuchi, and N. Sugi. RoboCupRescue 2014-Rescue, Simulation League. Team Description. Ri-one(Japan), 2014.
2. Ajith. Abraham, Crina. Grosan, Vitorino. Ramos, Satoshi. Kurihara and Kenichi. Fukui. Swarm intelligence in data mining, Japan, 2009. ISBN 9784501550905
3. 李志祥. (2011). Based on mobile agent Distributed Ant Colony Clustering Linearization method. <http://www.cs.is.noda.tus.ac.jp/~mune/thesis/li.pdf>