

RoboCupRescue 2010 – Rescue Simulation League

Team Description

<ZJUBase (P.R.China)>

Yuankai Wang, Junkun Li, Haijin Li, Yizhou Zhu,

Yingpeng He, Rong Xiong

State Key Laboratory of Industrial Control Technology,
Zhejiang University, P.R.China
romiohoo@yahoo.com.cn

Abstract. In this document, we describe features of ZJUBase rescue simulation team which is participating RoboCup 2010. Model building and algorithm implementation of the main aspect of Rescue Simulation such as civilian exploration, lifetime prediction, civilian rescue, blockades clearance, fire spread prediction and control are discussed. New ideas and approaches are introduced in this paper. Our contributions to the state of art in Agent Simulation are also depicted in detail.

1. Introduction

The RoboCup Rescue Simulation environment is of great social value and it provides a platform for the development of algorithm design, artificial intelligence, statistical learning and data mining. To achieve a better performance, a hierarchical agent structure is established in our implementation.

A real-time, buffered world model stores all the observed and predictive information of the whole disaster space. An effective, adaptive, buffered, channel-based communication system was also developed in order to make the world model more complete and accuracy. We implement A* algorithm and multi-source shortest path algorithm along with mode-based road length mechanism for path planning. For civilian exploration, a knowledge base is maintained and updated by the Bayes Law. Particle Filter is used to predict the accurate lifetime of civilians. For ambulance teams, a two-level greedy algorithm optimized by dynamic programming model was developed. For police force agents, strategies based on Graph Theory, Operational Research and A* algorithm are used. For Fire Brigades, the fire spread prediction evaluation system and a dynamic agents scheduling algorithm are implemented.

The simulation environment has changed a lot since last years' competition. We have redesigned some of our system and tried some new approaches to achieve better performance. A reinforced communication network is developed. Some more dynamic

online algorithms for fire extinguish are implemented. The rest of the paper will describe these specific features and show our performance along with proof.

2. World Modeling and Communication

Each agent maintains a buffered, multi-level world model as the lowest level of the agent structure. Observed and predictive information is stored and updated each cycle. A stable, effective and self-adaptive communication system is established in order to make the world model more complete.

Since the number of platoon agents and the number of center agents increase in RoboCup, it is necessary to build a new communication network. Fortunately, our previous adaptive channel distribution communication network shows its strong generalization ability, so only little changes are made to fit the new communication model.

The whole communication system is treated as a graph show in Fig.1. The key vertexes are three virtual centers. In previous design, the virtual centers are either center agents or some selected platoon agents when there are no corresponding centers. Now, the virtual center consists of a cluster of centers of certain type. All the centers exchange messages with other virtual centers with only one of them responsible for fusing all the information and communicating with platoon agents. When there is no enough center agents to be the virtual center, some platoon agent takes the place as our previous strategy.

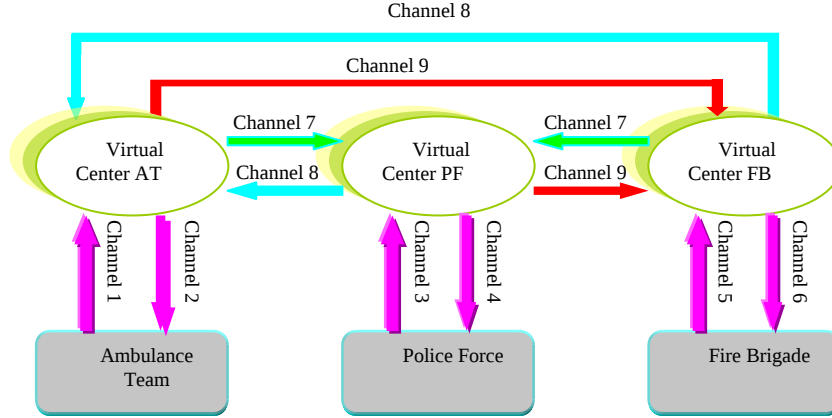


Fig.1. reinforced communication network

3. Region Based Strategy

One kind of strategy we have tried and implemented is the region based strategy. The disaster space is divided into different partitions and every platoon agent is as-

signed to a partition to accomplish some tasks in its specific partition. In our implementation, different partition algorithms are used in different scenarios and for different purpose. We found these algorithms graceful and efficiency, which help a lot in our strategy design.

In our system, some of the Police Forces are working in different regions in the disaster space. To ensure the efficiency of whole Police Force strategy, the number of this kind of Police Forces varies in different cycle. In order to robustly make the assignment, a Dynamic Partitioning Algorithm is implemented. The Dynamic Partitioning Algorithm divides the map into different parts following the basic principles of creating a KD Tree in two-dimensional area. The algorithm is also used when Ambulance Teams searching for civilians and widely applied in our no-communication strategy.

It is widely accepted that fire spread quickly and strongly when buildings are closely connected and spread slowly and weakly when buildings are far apart. To model and evaluate the fire areas more correctly and accurately, we apply density-based cluster algorithm to divide the city into different fire clusters. Each fire cluster is a bunch of closely connected buildings which fire spread quickly between them.

Fig.2 shows the results of these partition algorithms.

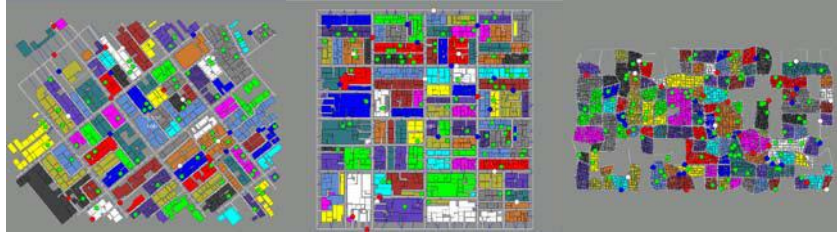


Fig.2. typical buildings partition results

4. Online Learning and Estimation

One of our ambition to join the rescue simulation league is to design online learning and estimation algorithm which makes AI applied to real-time scenarios. In our system some of these methods are successfully implemented which improve our performance significantly.

Civilian lifetime prediction

Accurate lifetime prediction of civilians is one of the most important factors when planning appropriate rescue sequence for ambulance team agents. Many methods have been developed to achieve an acceptable prediction of civilian lifetime, such as CART regression, ADABOOST, multi-layer perceptrons, Bayes classifiers, and neural networks. However, these methods are both simply based on observations of agents. In other words, only explicit observations, say, current HP and damage of the civilian, are taken into account and served as the inputs of classifiers. Unfortunately,

there are several implicit factors which contribute a lot to the lifetime of civilians. Two main aspects of implicit factors are listed below:

(a): Quantization error: the damage value observed by agents is rounded to the nearest 10 by server and the HP value is rounded to the nearest 1000.

(b): Different growth rate of damage: The growth rate of damage also differs a lot even though the original damage values are exactly the same.

To deal with these implicit factors which can not be obtained by the direct observation, Particle Filter is introduced in our implementation.

The procedure is described in detail as below:

Step1: Generate the first generation of particles based on observations. Each particle is a triple (Hp, Dmg, Rate), where Hp and Dmg are possible values given observations and Rate is a value of range 0 to 1 which denotes the growth rate of damage.

Step2: Every cycle we update the particles by applying the damage model to each particle:

$$[Hp \ Dmg \ Rate]_{n+1} = f([Hp \ Dmg \ Rate]_n) \quad (1)$$

Step3: Whenever new observations are generated, we check each particle to see whether the values of Hp and Dmg are within the quantization range of observations. Illegal particles are removed from the collection and we expand the collection of particles to its original size by randomly copy legal particles.

Step4: If all the particles are illegal, go to step 1.

Fig.3 shows the predictive result of ZJUBase, ResQ04 and MRL06.

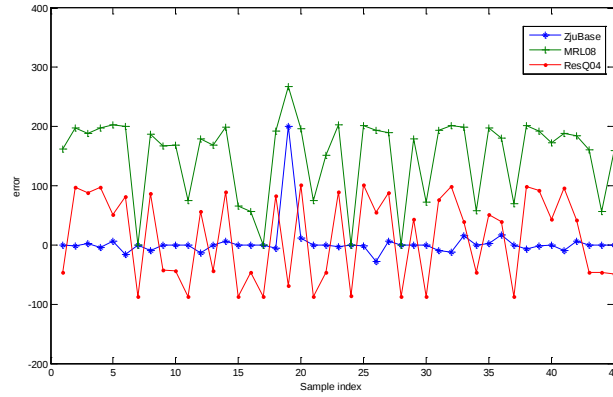


Fig.3. Civilian Life Time Prediction Results of ZJUBase, ResQ04 and MRL06

Fire Brigade scheduling

One important aspect of fire extinguish is to allocate each fire site a proper number Fire Brigades. We introduce a dynamic allocation strategy based on the conception of feedback to carry out the fire brigades scheduling.

The dangerous degree of a fire site is characterized by a characteristic function S in our system. The function is a combination of various aspects of the fire site, which include area, burning time and affected area of the buildings in the fire site.

We regard fire fighting as a system, so we can introduce the conception of control. Other Teams mostly use the open-loop control from the view of control theory. They regard the fire site as control object and the number of fire brigades needed as the actuator. The combustion degree of a fire site is sensed and characterized somehow, which is the guideline of the fire brigades allocation. The effect of this kind of method would be influenced significantly by estimation error and some accidents like road block or traffic jam. By the way, it's not easy to design a general algorithm to predict the number of fire brigades accurately in the open-loop system.

So we introduce the closed-loop control to the system of fire extinguishing. Our system is represented as follows:

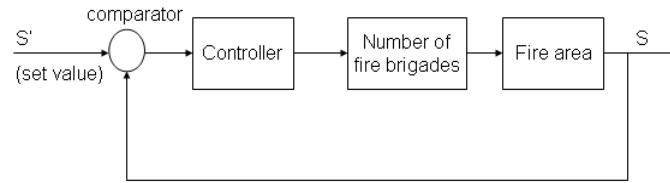


Fig.4. Diagram of fire fighting system

In our decision system, the characteristic function S of fire site is compared with set value S' and the difference is taken as the input of the controller. The output of the controller adjust the number of fire brigades assigned to the fire site, which will cause the variance of the characteristic function until S equals to S' . If we set S' as the threshold of safety, we can control the fire site to a safety state with appropriate number of fire brigades.

The design of the controller is the key problem. The PI control algorithm is used in our system. The expression of PI controller is as follows:

$$y(t) = P * x(t) + I * \sum_{i=0}^t x(i) \quad (2)$$

Where $x(t)$ and $y(t)$ denote the input and output at the time t . P is the proportion coefficient and I is the integration coefficient.

Tab.1. shows the statistical comparison of the percentage of unburned buildings between our previous approach and the one with dynamic scheduling.

	FL-3	FL-5	Kobe-2	Kobe-4	VC-2	VC-4	RS-6	RS-5
previous	0.799	0.989	0.783	0.678	0.390	0.436	0.903	0.510
New	0.825	0.996	0.835	0.732	0.348	0.446	0.888	0.506

Tab.1. Percentage of unburned buildings

5. Civilian Exploration

Civilian exploration is the first step in the whole rescue operation. Probability based civilian search algorithm is widely applied in rescue simulation league, so we

omit that part. What we want to emphasize here is our exploration strategy which is quite different from others, which illustrates our understanding of the task and the whole rescue simulation system.

We do not treat exploration as a task with equal level to the tasks such as civilian rescue, fire control and blockade clearance. In our implementation, **exploration task is merged to path planning** which named as “By The Way” strategy.

The length of a route is magnified by the formula below when considering exploration:

$$L = k * l_{org} - (k - 1) * l_{org} * \frac{E_{civ}}{E_{org}} \quad (3)$$

In this formula, l_{org} is the original length (geometric length) of the route, E_{civ} denotes the mathematical expectation of the number of civilians which can be seen from at least one location of the route and E_{org} is the value of E_{civ} in the first cycle of the simulation. k is the key value which denotes the priority of exploration.

When a route is chosen, we use the following evaluation formula for each node along the route:

$$Score = V_{avg} + E_{ep} \quad (4)$$

In this formula, V_{avg} is the average speed of platoon agent and E_{ep} denotes the mathematical expectation of the number of civilians which can be seen from that node. We choose such a node as the destination that this node is the first node from the starting point with a score higher than a constant threshold.

6. Agent Strategies and Actions

Ambulance Team

We are trying to build a strict, effective, adjustable, comprehensible and time-saving model to achieve the global optimal approximation of the rescue sequence. There are two interactional factors which affect the performance of the rescue operation:

- (a) The rescue order of civilians.
- (b) Time elapsed on the road.

These two factors are interactional and sometimes contradictory and both factors affect the coarse rescue sequence. We implement the **Optimized MUMI (More urgent, more important) strategy** to maximize the Rescueable Collection which is the set of civilians that all elements in it can be saved by Ambulance Team.

The main procedure of this strategy is:

- (a): Sort the civilians by their lifetime and assign each civilian a priority level based on their lifetime.

(b): For each civilian, calculate the minimum number of ambulance teams needed for their security.

(c): For each free ambulance team (An ambulance team is free if there is no task assigned or the previous task is just finished), assign such a civilian that this civilian is of the highest priority among those civilians which don't have enough number of ambulance teams rescuing them.

A two-step algorithm was developed to decide the rescue sequence for each agent:

- 1) Maximize Rescueable Collection given information of all the civilians.
- 2) Decide the rescue sequence for each agent.

All the rescue sequence layout algorithms are just seeking a balance between the two main factors mentioned above if we only pay attention to the outputs of the algorithms, not how the outputs are generated. This statement directly leads to our strategy.

Each cycle, a current task will be assigned to each ambulance team. We define **“Initial Energy”** as the total distance between each ambulance team and its current destination. For a **“Rescueable Collection”**, if we adopt the Optimized MUMI strategy, the rescue operation will definitely success if the time elapsed on road is not taken into account (See Proposition 1). If greedy algorithm is adopted, the **“Initial Energy”** will be the lowest but the **“Rescueable”** property may not be ensured for a **“Rescueable Collection”**.

Our strategy is **to minimize the “Initial Energy” of the assignment on the premise of keeping the “Rescueable” property for the “Rescueable Collection”**. In other words, we must ensure that after all the ambulance teams complete their initial task there is still at least one possible rescue sequence for ambulance teams to save all the civilians of the **“Rescueable collection”**.

In our implementation, a greedy assignment is generated and adjusted by solving a knapsack problem with dynamic programming.

Fire Brigade Agent

The main task of fire brigade agents is to fight with the fire, either put out all fire areas or control the spread of fire. Since this year, the **“save the city”** challenge will show its face in the competition, which means fire extinguish strategy should be designed more carefully and more thoughtful.

We follow the idea which is evaluating fire sites by modeling fire spread prediction. The main difference is that we combine our previous methods with the region based strategy mentioned in early section.

The overall strategy is a two level hierarchical decision-making architecture. Firstly, each fire brigade evaluates every fire area to obtain the order of importance of the fire areas. Then the novel distributed online dynamic scheduling algorithm described in section 4 is implemented to assign proper number of agents to each fire site. Also the algorithm tunes the assignment in real-time to ensure the efficiency of fire extinguish.

Several factors are considered to give the importance of each fire area which are:

- (a) The importance of the fire clusters which the fire site stands
- (b) The distance between the fire area and the fire brigades
- (c) The burning ratio and burning area of the fire clusters

(d) Number of civilians around the fire area.

Police Force Agent

A semi-centralized-semi-distributed architecture is implemented to develop the Police Agents system. The disaster space is modeled as a graph. We divide the graph into many connected components which are connected by blocked paths. An off-line statistical method called Region Force Power Evaluation Algorithm is used to estimate the Rescue Power Force (namely Ambulance Teams, Fire Brigades) and brokenness, also the gap between these two, in the connected components. The gap along with other factors which is calculated by A *algorithm constitute the weight of the paths.

The Police Office selects “most important paths” based on their weights and assigns these most important paths to Police Forces. A simplified Hungarian Algorithm is implemented to solve this Assignment Problem.

Rest of the Police Forces, with time-varying numbers, are distributed into different partitions of the map by the algorithms mentioned in section 4. These Police Forces clears blockades and search civilians according to a linear score function which considers the burning degree and search score of buildings.

7. Acknowledgements

At last, we would like to thank all of the server developers’ team for maintaining the rescue simulation server and for their technical support; thank ResQ Freiburg 2004 Team for their TDP and source code; thank MRL 2006 Team for their code and also thank developers of package rescuecore because this package is the base code of team ZJUBase.

8. References

1. Morimoto, T: How to Develop a RoboCup Rescue Agent, (RCRSS manual), <http://robomec.cs.kobe-u.ac.jp/robocup-rescue>.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of statistical learning. Springer-Verlag, 2001.
3. Mark Allen Weiss: Data Structure and Algorithm Analysis in C. Pearson Education Asia LTD. 2005.
4. Alexander Kleiner, Michael Brenner, Tobias Bräuer, etc.: ResQ Freiburg: Team Description and Evaluation. <http://www.infomatic.uni-freiburg.de/~rescue/>. 2004.
5. J.H.Holland: Adaption in Natural and Artificial Systems. University of Michigan Press, 1975.
6. T.Cormen, C.Leiserson, R.Rivest: Introduction to Algorithms. MIT Press, 1992.
7. Rong Xiong, Yikun Tao, Yuankai Wang, Shunhong Zhong, Jie Zhang, Wentong Li Baixin Wang, Ye Tian: ZJUBase RoboCup 2008 Team Description
8. Wang Yuankai, Hao Baoping, Zhu Yizhou, Li Junkun, Li Haijin, He Yingpeng, Xiong Rong: ZJUBase RoboCup 2009 Team Description.