RoboCupRescue 2014 – Rescue Simulation League

Team Description

<SEU_Jolly(P.R.China)>

Wei Niu, Shuang Cai, Jie Wu

School of Mechanical Engineering, Southeast University

niuweimaster@163.com

Abstract: In this paper, a task-state based decision making method will be introduced. Meanwhile, an algorithm "ant-tree" for ambulance team agent to search the injured civilians and police force to find the main roads will be covered. Then we build an effective communication system structure. Besides, researches on the multi-agent coordination has also been made , which is based on clustering analysis and Q learning algorithm.

1. Introduction

RoboCup Rescue Simulation System (RCRSS) is a large-scale Multi-Agent System (MAS) of urban disasters. In such a dynamic, partially observable environment, the action decision making is always the primary problems which needs to be effectively solved. Our code structure is shown in the graph below.

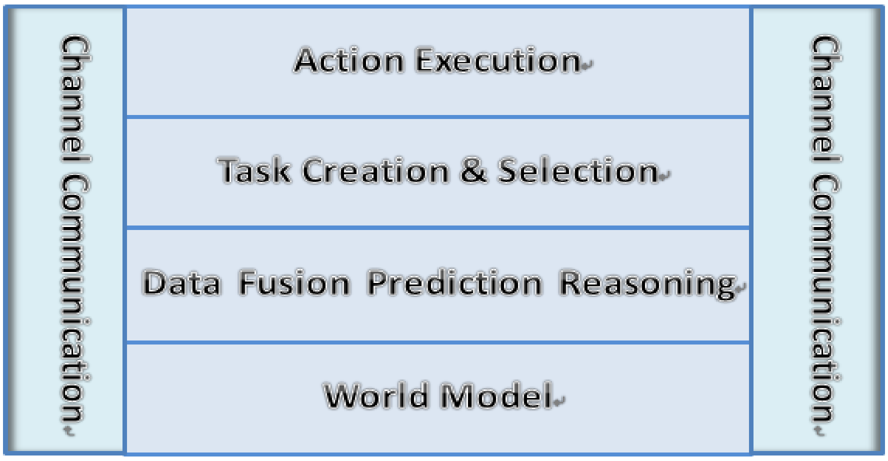

Figure 1. Code Structure of SEU_Jolly

The effectiveness of decision making needs a complete and accurate world modeling. So, we established different channel based communication models in diversified disasters for information sharing: the typical communication model and communication model under no center conditions. The latter model has some profitable characteristics such as adaptability, minimum time delay and virtually equally distributed channels. These characteristics especially enable us to build a more realistic world model under certain sharp conditions. As for decision making, we use task-state based decision method. A simplified decision process is shown in Fig.2. Basic low level action of moving is addressed to fulfill different needs of our agents in such a dynamic and uncertain system.

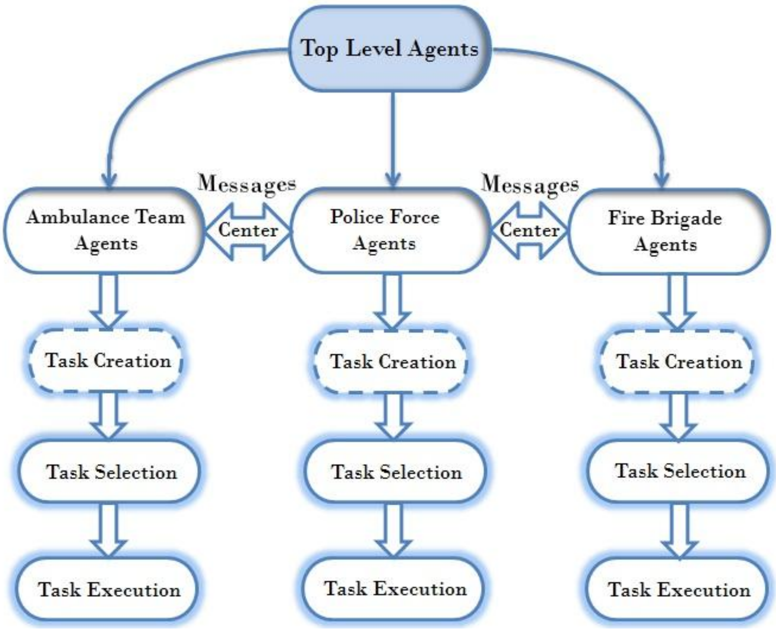

Figure 2 Module Structure of SEU_Jolly

Our main code structure is as Fig.3. There are 12 modules in our code. The most important modules include communication module, world model update module, path plan module and top agent module. They are the basic modules to construct all the code. The knowledge base of task-state decision module is updated by the world model update module. The communication module helps to update the world and execute the agents' command. Path plan module is a basic module that every kind agent must use it to get a path to the destination. The BFS method is low efficiency, so we do some efforts to improve it. We use traditional a star method to explore the path. The top agent model is the agent task manage center. The 3 kind agents' common task is done in this module and the world model update module is also called in this module.

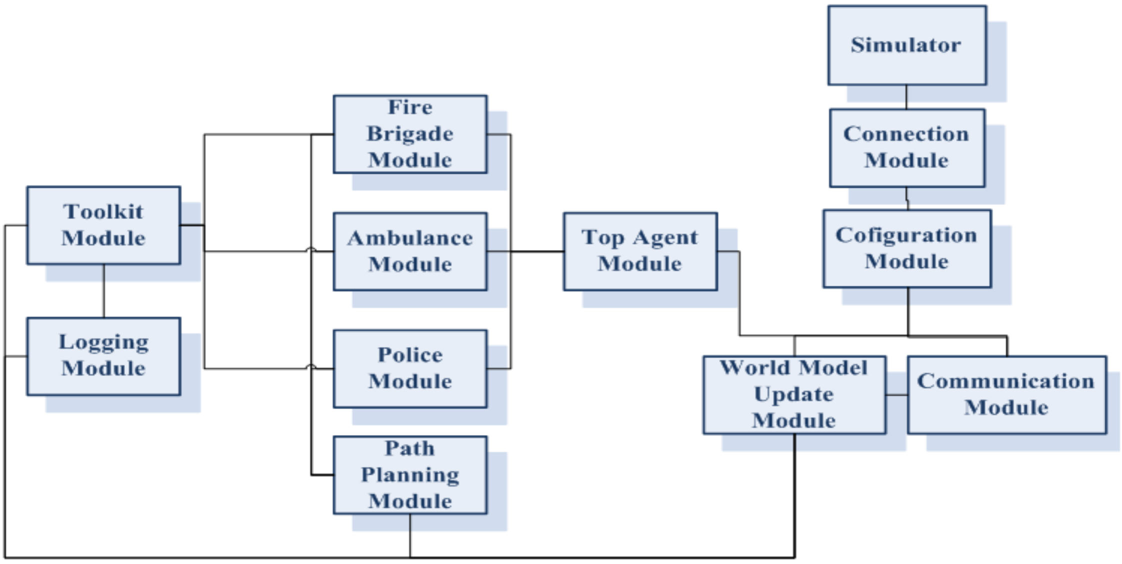

Figure 3. Basic Decision Process of Agents

2. Task-State Decision System [2]

We adopt a task state system to help us to do the decision of agents' activities. A task include task object and task flag. Task object is what to do the task. Task state is a kind of flag that indicate what kind of activity the agent wants to do. Task flag indicates the task's attribute.

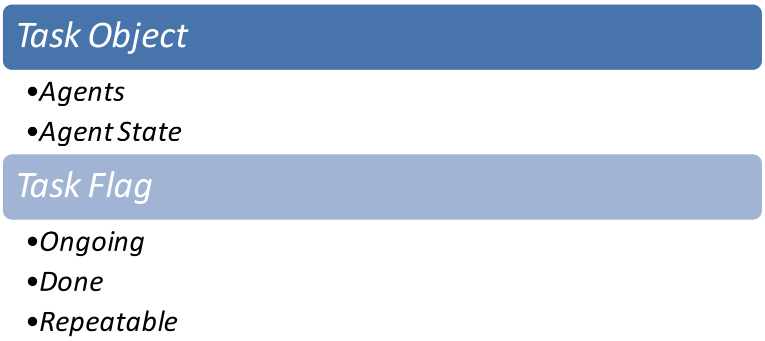

Figure 4. The Structure of a task

The task flow of task-state system is showed as fig.5 below.

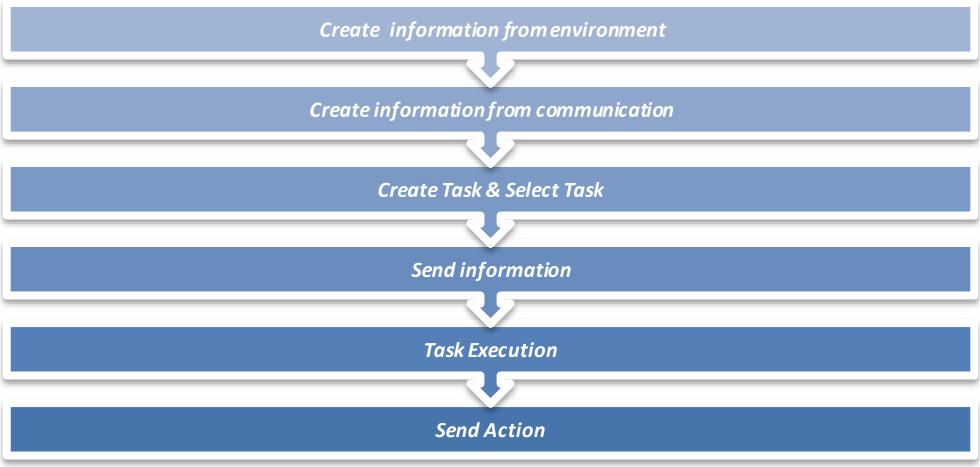
Create information from environment
Create information from communication
Create Task & Select Task
Send information
Task Execution
Send Action

Figure 5. The agent task flow

3. Agent Task Selection Method

3.1 Ant-Tree Searching

Although the ant colony optimization algorithm has had success in so many fields, there are also various combinatorial optimization problems in the application for difficult combinatorial optimization problems.

In this paper, we adopt the ant colony optimization strategy and present a solution method, called Ant-Tree, to provide an investigation of the ant colony optimization approach for coping with tree optimization problems. Ant-Tree is an improved ant colony optimization approach to the generalized minimum spanning tree problem. This paper presents the design and implementation of a multiple ant colony optimization algorithm, which utilizes an iteratively pheromone evaluation mechanism to tackle large optimization problem.

Ant-Tree (An Ant-Based Algorithm for Finding Degree-Constrained Minimum Spanning Tree) consists of a number of cycles. This algorithm can be used in solving many problems, such as find the main roads in the map, searching the important areas which may be many injured civilians around. Each period has two stages: exploration and construction. Then we adopt the algorithm elaborating how to find the main roads which is the important roads that agents travel frequently on them and the blockades on them should be cleared as soon as possible.

(1) Exploration stage--Ant Colony Algorithm with noisy signal

In the exploration stage of each cycle ants are used to discover a set of candidate edges from which a degree-constrained spanning tree is constructed.

As the statistical data tending to infinity, some potential problems are likely to happen. Just like traffic jam, the high probability for ants to choose this popular road results in the vacancy on other road.

$$\tau=(1-\eta)*\tau+\Delta\tau; \qquad \Delta\tau=\Delta\tau+Q; \tag{1}$$

In equation (1), η is an evaporate co-efficient. The larger η is, the faster the process of pheromone evaporation has and the sooner ants quickly forget previously selected edges.

Here we present an algorithm which take blocked road into consideration. We view the blocks as the noise. We define the length $\omega(p)$ of a path $p = (e_1, . . . , e_n)$ and evaluate the length of a path ρ. We do not get the real length $\omega(p)$,but a noisy length $\widetilde{\omega}(\rho)$ .Assume we are given a family $\eta(e_i)_{e\in E}$ of nonnegative random variables $\eta(e_i)\geq 0$. The noisy length $\widetilde{\omega}(\rho)$ of an edge $e \in E$ is defined and the noisy length $\widetilde{\omega}(\rho)$ of a path $p = (e_1, . . . , e_n)$ is defined as

$$\widetilde{\omega}(\rho)=\sum_{i=1}^{l}(1+\eta(e_i))/length(from\ \ p_1\ \ to\ \ p_i). \tag{2}$$

Once an ant with start vertex u has created a path, its best-so-far path ρ_u^* is updated with the newly constructed path ρ_u if the latter is not worse than the former. Note that this decision is made according to the noisy path lengths $\widetilde{\omega}(\rho_u^*)$ and $\widetilde{\omega}(\rho_u)$. Finally, the ant updates pheromones on the edges leaving its start vertex according to the best-so-far path ρ_u^* and the following formula (3):

$$\tau(\mathrm{u}) \leftarrow \begin{cases} min\{(1-\rho)*\tau(\mathrm{u})+\Delta\tau(\mathrm{u}),\tau_{max}\} & if\ \ e=(u,v)\in\rho_u^* \\ max\{(1-\rho)*\tau(\mathrm{u}),\tau_{min}\} & if\ \ e=(u,v)\notin\rho_u^* \end{cases} \tag{3}$$

Pheromones on an edge e are denoted by $\tau(e)$. Initially we set $\tau(u) = 1/\deg(u)$ for all $e = (u, \cdot)$ as well as $\rho_v^* = ()$ and $\widetilde{\omega}(\rho_v^*) = \infty$ for all $v\in V$. The evaporation factor $0\leq \rho \leq 1$ as well as τ_{min} and τ_{max} are parameters of the algorithm.The so-called pheromone borders τ_{min} and τ_{max} ensure that the pheromone for each edge is always bounded away from 0, so that there is always a chance of reverting a decision once made.

(2) Construction stage--Ant-Tree

The other question that should be addressed here is about local optimization and global optimization. The basically ant colony algorithm is not universal for a large system. Traditionally, heuristic algorithms start with some feasible solutions and continue to progress toward some local optimization which may not be global optimization. So, what we concern about is the combinatorial optimization problems.

With the help of the Ant Algorithm, we can find several high frequency nodes. The key stage for this algorithm is to use Kruskal Algorithm to choose the proper connection between two points. To combine the Ant colony optimization Algorithm with Minimum Spanning Tree Algorithm, we could find the main roads which can connect the entire map.

In the construction stage we build such a tree. Thus, each cycle produces a degree constrained spanning tree. The smallest cost tree returned by all cycles is returned by the algorithm. In each cycle, each ant explores a local section of the graph. The edges an ant travels across are chosen such that edges with a higher level of pheromone have a higher chance of being selected. The pheromone on an edge is increased when an ant travels across that edge, leaving an increasingly appealing trail for other ants to follow. It should be noted that for an edge (*i*, *j*), the initial amount of pheromone distributed at (*i*, *j*) is the same as the amount of pheromone an ant leaves on (*i*, *j*) when the ant traverses it. Furthermore, the lower the edge cost is the higher this pheromone amount is. Effectively, ants will have a tendency to travel along lower cost edges as well as edges that have been traveled more often by other ants. After all ants have explored their local section of the graph, meaning each ant has traveled across a set number of edges, the entire graph is examined. The edges with the highest pheromone levels, which are the edges ants traveled across most often, are selected as a candidate set of edges. A degree-constrained spanning tree is then built from these candidate edges in a greedy manner.

We use Ant-Tree Searching algorithm to find the map VC as the following figure .

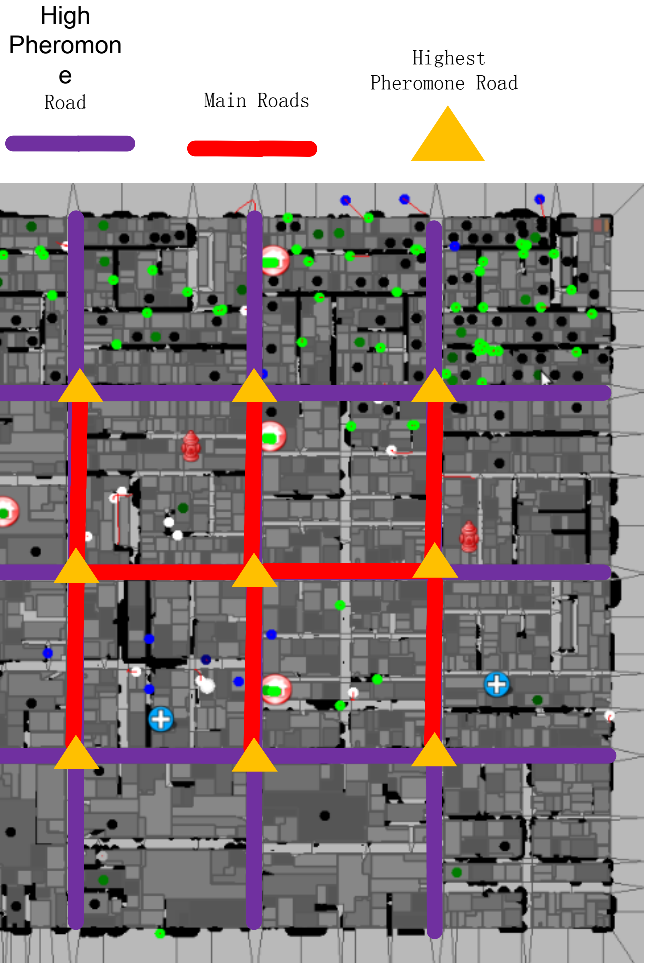

Figure 6. Main Roads of Map VC

3.2 Fire Brigade Agent

Prediction of fire spreading is crucial for fire brigade agents to control fires. By the estimation of buildings' temperature, we can get a more reasonable task allocation plan. We use the same algorithm as 2012[2] to predict the fire spreading and use convex hull algorithm to extinguish the fire [2].

3.3 Ambulance Team Agent

The performance of ambulances could greatly affect the final scores. We treat the process of rescuing civilians as a dynamic assignment problem: given *m* ambulances and *k* civilians. We use Particle Swarm Optimization (PSO) with ANNS to complete the allocation. The whole algorithm description is in our team's team description 2012[2].

How to find the injured civilians is critical to the rescue task. In our code, we use two algorithms to achieve it. Firstly, we use the center to locate the civilian's position. When agent hears the information which civilian shouts, it send a message "Civilian is found" message to report the position. The center collect the information and use entropy function to find out the most possible position and assign the rescue task to agents. Secondly, we use ant-tree searching algorithm like the previous description to get the most desired area to search the civilian with the information of fire and other messages.

3.4 Police Force

Police Force play an important part in connecting the whole disaster-stricken city and searching the whole city in order to identify the distribution of the residents. We use a method based on minimum spanning tree for police force agent to connect the city in the minimum time, which is described in [2].

This year, we divide police force into three types:

- Static Partition: Go to the assigned partition firstly and then work in the assigned partition.
- Dynamic Partition: No fixed assigned partition and work according to the other agents' request.
- Connected Partition: Clear the main roads firstly and then turn into dynamic partition type.

Connected Partition type of PF is very important. We use the ant-tree algorithm (previous description) to compute the main roads. Then the mission of this type of PF is to clear the blockades out to make sure the whole map is connected as soon as possible. Clearly, three types of PF have the different sensitivities to the other agents' request: Dynamic Partition> Static Partition> Connected Partition.

4. Communication

Communication module is a very important module to support world model update and the agent action execution. In the latest server, the RCRSS use variable channel count to replace fixed channel count. If the communication mode is fixed channel count, the distribution mode is based on the number of specific center. Now the channel count is variable, center is as an ordinary agent.

We can give a definition of the communication problem. To connect agents by P2P channel, to achieve message sharing and task allocation, minimize message quantity and maximize information sharing rate. The constraint condition include agent number, center number, channel count, available channel count, message size, utter number, false rate and dropout rate. This is presented in our 2012 TDP[2].

We have put large effort to promote our communication system. There are three issues in this domain.

Firstly, we design some coding algorithms to overcome the channel noise, aiming to get a relatively low error probability.

Secondly, our communication system is able to evaluate the channels and distribute different kind of messages among these channels in a proper way so that we can get a relatively high efficiency.

Thirdly, we develop a communication system which can adapt to different communication environments, form good condition with large bandwidth and low bit error rate to poor condition with small bandwidth and high bit error rate.

Here is the hierarchy of our communication system.

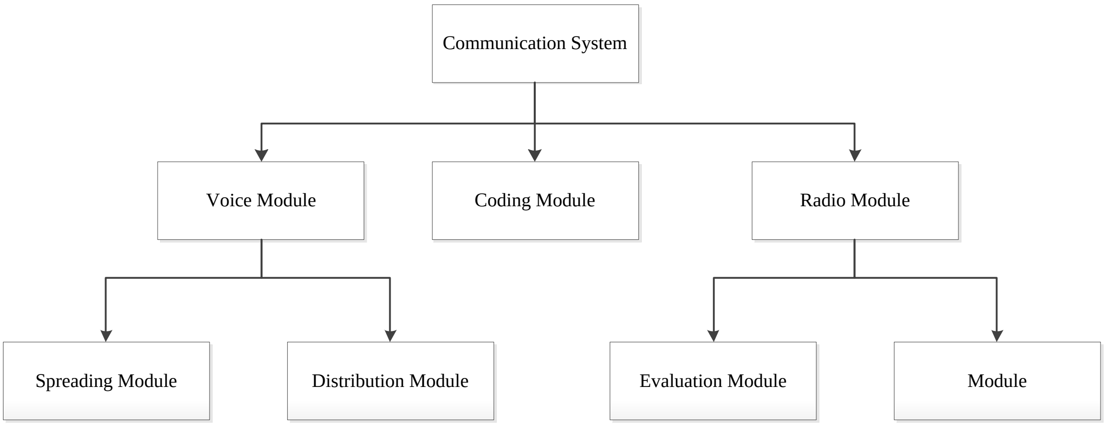

Figure 7. Communication System Structure

(1) Coding Module

Coding module is a general module which is used in both voice and radio communication. To get a balance between efficiency and reliability, we use different coding algorithms.

Coding Algorithm	Features	Available in Radio Module	Available in Voice Module
LEMPEL-ZIV coding	Source-coding algorithm for message compaction	YES	NO
Parity-check coding	Simple error-control coding algorithm	YES	YES
Hamming coding	Error-control coding with a large redundance and the ability to correct bit error	YES	NO

(2) Evaluation Module

The fundamental module is to evaluate the channels so that we can rank them and make a good distribution. Since there are two parameters, bandwidth and bit error probability, with each channel, we take a practical method to realize the evaluation.

We design a protocol which is only used at the beginning of the simulation. Each channel is allocated to two agents called e-sender and e-receiver. The e-sender is responsible to send a large number of channel-testing messages to the e-receiver and the latter evaluates the channel according to how many messages it receives.

(3) Distribution module

Distribution module is also available in both voice and radio channel, however, with several differences. Radio channel distribution is relative more complex. Given a set of channel resources and communication requirements, the function of the module is to distribute different kind of messages among these channels in a proper way.

The algorithm is to rank the channels according to the result of evaluation module. The communication requirements of messages are also sorted. We use greedy strategy to satisfy the requirements.

(4) Spreading Module

This module is the essential part of the voice communication, especially useful when the condition is so poor that radio is nearly not available. The spreading pattern of messages through voice channel is quite different because of limited transmission distance. It's completely dynamic, distributed and local.

We take advantage of communication network to model this spreading pattern. Each agent is a node of the network. The agent which is willing to send the message is active while others are inactive. The active node will continue to broadcast the message for several cycles and turn those adjacent nodes into active state. With this model ,we do so analysis and simulation and it shows that on average each voice message should repeat more than 30 cycles before it is wildly spreading.

5. Multi-agent Coordination

Focusing on rescue tasks, we try to realize multi-agent coordination in the following two steps, as is shown in Figure 8: dispersed tasks are combined into "task package" through clustering analysis as is shown in the big circle; then take "task package" as the input of coordination and make space for action and state with the assistance of Q learning algorithm. In Figure 8，the black dots are the known location of civilians, triangle represents the state and line is the action. Different actions can be chosen at different states, while the chosen action leading to another state. Our terminal goal is to reach a state that the "task package" is fully completed

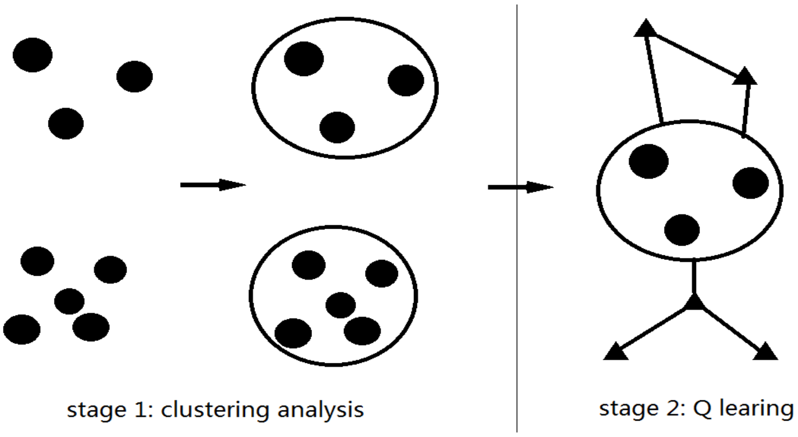

Figure 8. Steps to realize multi-agent coordination

5.1 Clustering analysis

The method this paper puts up with is based on improved artificial immune algorithm, and can be realized through the following steps, as shown in Fig 9. Take location of civilians as antigens, and with the help of the algorithm, we can combine related civilians into cluster or say “task package”, and antibody is the centre of the cluster.

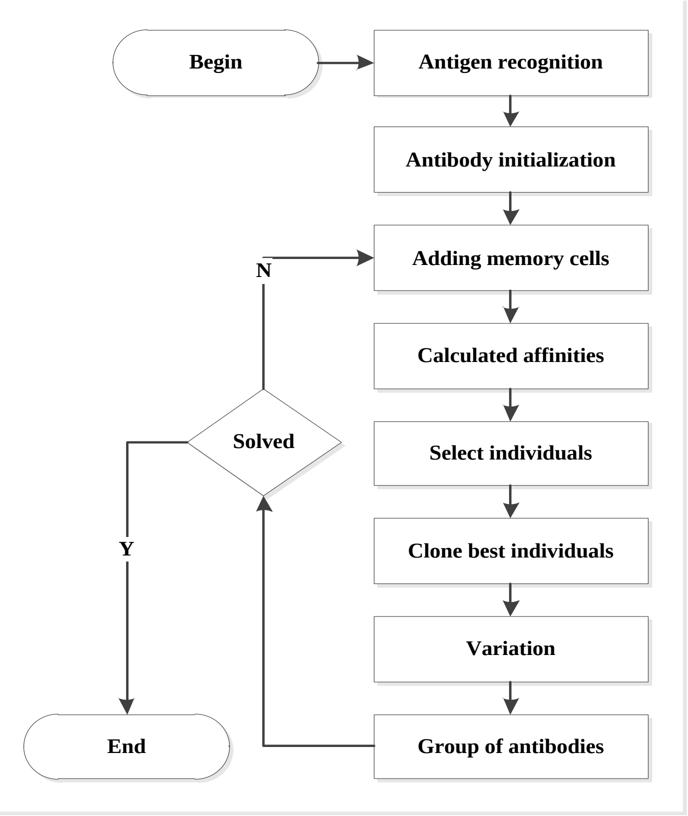

Fig 9. The flowchart of algorithm

Memory cells can help to store excellent solutions, which can be used as initial antibodies in the next round of clustering to make the whole process dynamical and at the same time speed up calculation.

5.2 Q learning

In Q learning, the function *Q(s, a)* to evaluate learning actions is defined as the biggest conversion of the cumulative returns starting from the state s and using action *a* as the first action, that is, the immediate return of performing action *a* at state *s*, plus the rewards for the following strategies.

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \max(s_{t+1}, a_t). \qquad (1)$$

(1) action space

(i) Common actions of all agents: Move, Search, Rest.

(ii) Actions of AT:

Rescue: SaveAgent, SaveCivilian, SaveMyself;

Load;

Unload.

(iii) Actions of PF:

Clear: ClearBlockedRoadToRefuge, ClearNearBlockade, ClearEntrance, ClearMainRoad;
Exploration.

(iv) Actions of FB:
WaterFill;
Extinguish: ExtinguishDangerousBuilding, ExtinguishNearFire, ExtinguishFromRefuge .

(2) state space in RCRSS

(i) Common states of all agents: MovingToRefuge, MovingToPartition, RandomMove, Exploring.

(ii) States of AT: MovingToHuman, RescuingHuman, LoadingHuman, TakingCivilianToRefuge, UnloadingHuman, SavingMyself.

(iii) States of PF: GoingToBlockedRoad, ClearingBlock, ClearingPath, ClearingRefugeEntrance, ClearingMainRoad.

(iv) States of FB: MovingToFireArea, Extinguishing, FillingWater, GoingToRefugeToRefillWater, GoingToHydrantToRefillWater.

(3) value functions

Use three functions of "fieriness", "buriedness" and "blockness" respectively to evaluate the work of FB, AT and PF. If the values of the functions shrink after the agent took an action, then reward the agent. Besides, the fulfillment of rescuing the wounded, putting out fire and other important tasks will get the bonus.

6. Conclusion

In this paper, we presented a brief overview of the structures and approaches designed and implemented in SEU_Jolly after RoboCup 2012. First of all, we want to build a complete and accurate world model via communications among different agents. Second, we adopt a new task-state decision system to improve our agents' decision. Then, various techniques have been tried or implemented in our code to deal with noisy, varied, real-time and dynamic disaster environments. Finally, we tried a new method to realize multi-agent coordination.

For the future, we plan to thoroughly test our code, modify minor bugs and use other Artificial Intelligence methods in order to establish an effectively cooperative team of agents in such a complex multi-agent domain to diminish the side effects of urban disasters.

7. Reference

1. RoboCup-Rescue Simulation Manual, RoboCup Rescue Technical Committee (2000).
2. Huang Zhourong, Guan Daqi, Shi Fabin. SEU_Jolly 2012 Team Description.
3. Guan Daqi, Ouyang Jin, Wang Huijun. SEU_RedSun 2011 Team Description.
4. Guan Daqi, Chen Ning, Jiang Yinhao. SEU_RedSun 2010 Team Description.
5. Meng Qingfa, Guan Daqi. SEU_RedSun 2009 Team Description.
6. Guan Daqi, Xu Yingqiu, Tan Yingzi. Particle Filter Based Study of Fire Forecasting in RCRSS.
7. Jiang Yinhao, Xu Yingqiu, Tan Yingzi. MST & SP Based Traffic Clearing and City Search in RCRSS.