RoboCupRescue 2009 - Rescue Simulation League Team Description CSU_Yunlu(China)

Jun Peng, Fu Jiang, Ya Liu, Xiaoyong Zhang, and MinWu

Central South University, School of Information Science and Engineering
22# South Shaoshan Road, Tianxin District, Changsha
Hunan 410075,China
jiangfu0912@gmail.com
http://robocup.csu.edu.cn/en

Abstract. On the basis of the former optimized cooperation algorithm based on information process and state prediction used by our team last year, a new multiple intelligent agents' coordination model based on fuzzy Qfunction is proposed for CSU_Yunlu RoboCupRescue simulation team 2009. This model uses fuzzy logic to generalize the agent's continuous state space. Every agent, when making decisions on its actions, needs to consider the influences of other agents to the environment. The agent first evaluates the actions they select, then, uses the fuzzy Q-learning to learn their action strategy. In the process, the action keeps improving and the conflicts among agents can be resolved. This technique has been successfully applied in the RoboCupRescue simulation team competition, and, in the process, validated of the effectiveness and generalization of the cooperation algorithm.

1 Introduction

CSU_Yunlu Rescue team started participating in the RoboCupRescue competition from 2006. We gained numerous of precious experience in the RoboCup 2006 in Bremen, 2008 in Suzhou, and China Open 2006, 2007, 2008 in Suzhou, Jinan and Zhongshan.

RoboCupRescue simulation system (RCRSS) is composed of multiple autonomous agents so that, through interactions among agents, a collective goal can be achieved. For a system to function properly each agent needs to have the ability of rapid response to the environment; in addition, the system has to have an efficient coordination algorithm for the whole system to operate intelligently. And the importance of its role in the coordination and inter-system information sharing cannot be ignored. We studied the dynamic coordination of MAS through information processing and prediction using the information extracted in the process[1].

To achieve the system mission as quickly as possible, the agents need to coordinate their actions through interactions with each other to eliminate conflicts in their actions and reasonably distribute their goals and resources, so that they can achieve their individual goals in a consistent and harmony manner[2].

In agent skills and action selection, we addressed advanced algorithms for path planning and civilian exploration[3]. And in the disaster prediction, mixed forecast method is adopted to predict the fire and victims' lifetime. And in agent coordination and communication, the multiple intelligent agents' coordination model based on fuzzy Q-function is proposed. Because of these new approaches, our team has greatly improved in the aspects of disaster prediction and cooperative fire-fighting.

2 Agents

2.1 Ambulance Team

Suppose that accurate outputs can be got in the implementation of civilian exploration and lifetime prediction, that the factors which would affect AT's behaviors are: The time AT takes when moving on the road; the rescuing order of civilians.

To address these two issues, we defined a set of rescue sequence collection, and at least one victim in this collection. And the HP, death time, damage of victims are also stored, so that we can get a rescue sequence, which defined as rescuable coefficient. Through this rescuable coefficient, ATs can be assigned suitably, and this strategy is called the-most- exigency-the-most- priority mechanism.

But the moving time has a greater impact on this strategy, if AT spends too much time moving, then in the next cycle, the priority of this task which determined by rescuable coefficient will be lower, and AT is likely to move in two non-stop round-trip instead of carrying out rescue missions, this is obviously very unfavorable.

Therefore, we improved this strategy, and the collection was changed into a two layer rescuable collection[4], that the second layer is calculated based on the first layer and added with moving time. Then the rescuable coefficient is calculated once again, and compared with the rescuable coefficient of the latest civilian. If the new rescuable coefficient is higher than the rescuable coefficient of latest civilian, then it won't be added to the second layer collection, otherwise it will be added to the second layer collection. And greedy algorithm is adopted in the second layer collection, in order to save civilians as soon as possible. Then back to the first layer rescuable collection, and comparing once again.

2.2 Police Force

The main task of PF in the entire rescue simulation system is clearing roads to help other agents. We designed a semi-distribution and semi-centralized task distribution mechanism for PF. PFs Play a great important role in the early stage, so that ATs and BFs will be released quickly, and strategies of AT and FB will running efficiently in the simulation. First the whole map was divided into several sectors with distributed rules, and there is no overlap in each division of the region.

Therefore, we need an efficient and time-saving data structure. So that the smallest Binary Search Tree was chosen, and the connecting of two parts is defined as vertex (connect part is divided by blockades in map), and the road which was blocked by blockades is defined as edge, and finally formed the rudiment of smallest Binary Search Tree. Agents in each zone were managed centralized, the PF who reponsible for this sector is defined as the leader of this zone, and tasks distribution was determined by PF in the manner of centrality.

The behavior of those PoliceForces without a sector are quite simple, they have only two tasks. The first one is to clear roads on the path from a fire to a refuge. Unlike the other type of PoliceForces, they are not restrained to a specific sector, so they choose the closest fire. By clearing the path from a fire to a refuge, they are clearing the roads around fires and around refuges. This is interesting since they are really important spots to clear to help the other agents to move freely in the city. When there are no more roads to clear from a fire to a refuge, they will search for buried civilians by visiting all unexplored buildings.

When information of world model is highly incomplete and uncertain, there may be fail to divide the region. Therefore, for those PFs who don't have sector, greedy algorithm will be adopted, and those PFs can clear roads nearby. If division is successful, a Particle Swarm Optimization algorithm will be adopted.

We also adopted Floyd algorithm for calculating the shortest path among multi-sectors. And in the calculation of partition, we can pick up the most important path. According to our design, the assignment of sector and PF are cycled again and again, which formed the dynamic sector assignment algorithm.

2.3 Fire Brigade

FB is the most critical agent to control the disaster space. Thus a good and effective fire prediction model is essential, while the choice of targets is also very important, in the early stage of simulation; the choice order of different points has different effects on rescue results. So, for fire prediction model, we mainly concentrate on the construct of middle level. So that even in exceptional circumstances, fire prediction can also be got.

In the selection of firezone, we construct a distinguish collection according to the fire prediction model. Distinguish priority which calculated by Takagi-Sugeno inference theory was stored in the distinguish collection. According to different firezone, different approaches will be adopted; for fire points, the inputs are the distance between FB and fire point, the relative position of fire point, the center of the world model, and the degree of the fire; for fire zone, the inputs are the distance between fire point in fire zone and this fire zone, the relative position of fire point and the center of the world model, the burning area, and remnant area; for buildings, the inputs are the property of building, burning time, and the total area.

3 Agent skills and action selection

Aim at the characteristic of complex and uncertainty of simulation state space and disaster environment; we need to do some matting for varieties of strategies.

1. Path planning

This feature should be the most basic ability for rescue agent.

i. First we abstract the disaster model to a structure of graph, and optimize this structure of graph. To simplify this structure, some repeated information will be filtrated;

ii. In each cycle, before the high-level of decision-making, multi-source shortest path algorithm is used to find the shortest path for mobile agents to reach their destinations;

iii. A-STAR algorithm is adopted to compute the paths for two objectives.

2. Civilians exploration

For a better search of the whole map, and to search that area in the shortest time, we devised the algorithm of exploration. And we adopts off-line search mode. Those agents can be seen in the map will focus on the smallest point, on the basis of this focus of point, to find all points focus on this point in shortest time. This is a control set problem, because it belongs to a NP problem. So in the process of ever-changing, we calculate the approximation continuously. At several changes of point collection, we can identify all of the visual point, so that to realize the entire map search. Exploration is a lower level task, when and only when agent had been completed the more important work, it may use this feature.

4 World modeling

We have described at the beginning of this paper about the state space of simulation, that the information has characteristics of massive, high-frequency of being visited, uncertainty, imperfection, so that we need to process those information before used in our system, in order to improve the utilization of information.

We set priorities to perceived information, as to Kernel, the priority is based on the connecting order, that is the agent who connected with Kernel first will have the highest priority. However, this will be resulted in delay when dealing with important tasks. So it is necessary to adopt efficient data structure on the processing of perceived information.

Time division multiplex and buffer function are also used in the processing of information. That is, when there are too many agents connect to the same center (or centers), in order to prevent overloading of the communication systems, time division multiplex is adopted, and important information was selected and processed first. However, in the world model of great state space, there will be inevitable to have delay in the process of information transmission and processing, especially in the circumstances of center collapse, or lack of communication. So, we use a simple approach to treat this conduction, that is a Leader is set in every several agents, and let it as a small-scale relay station of information processing.

5 Disaster Prediction and parameter learning

In RCRSS, the update of world model's state information is obtained through the sensing device of the intelligent agent. However, the updated information usually arrives a few cycles behind real time[5]. To compensate the delay of the updates, state prediction approach is used to estimate the current states of the world model.

1. Lifetime prediction

The value of life time and survival time prediction is a crucial factor in the design of rescue order. In this regard, we use a mixed of variety of methods to predict, such as regression trees, AdaBoost, C4.5, Bayes Classifier, etc. C4.5 algorithm is adopted to sort the tasks, and AdaBoost is for convergence of simulation. The use of regression trees is to evaluate the impact of surrounding environment; Bayes Classifier is for final convergence. However, results calculated by these algorithms greatly influenced by the veracity of observed information. If the observation of information is inaccurate, the results will have a greater error, for which we use a particle swarm optimization algorithm with shrinkage factor called particle filter to optimize.

Prediction function is as follows:

$$Eva_c(Civ_i) = movetime + A * \frac{max(1, num_{AT})}{e^{at}} \tag{1}$$

Here, A is the initial value of the civilian life value, $Hp(0)$. The ones with lower life values have higher urgencies to be rescued. α represents the degree of damage of the civilian; t represents the number simulation cycles; num_{AT} represents the number of active rescue intelligent agents; movetime represents the time needed to move the injured from the current location to the nearest shelter. The less the $Eva_c(Civ_i)$ value is, the sooner the victim will be rescued.

2. Fire prediction

The forecast of fire spreading can effectively prevent the fire zones from connecting to each other, so that we need to reduce chain-reaction and reduce possibility of regeneration of a certain point. It can help us to better set the extinguish order. So, a fire development model is set up. First of all, properties of buildings around to the collection are added, with different properties, the degree of fire will be different. So, we can find the direction, which will spread faster through single-source shortest path algorithm. Then neighboring fire point, and the situation in the fire zone are added into the collection. Through the two disjoint collections, we can calculate the relevance degree of surround buildings, so as to determine the possibility of connected time or the connected time. In the forecast of linked time may have larger errors, as the uncertainty of changes of surroundings' situations. To solve this problem, we need to calculate a number of targets (usually between 2 and 4) in different weights, in order to find the optimal solution of PARETO. Finally, we should consider the factors of multi-faceted fire zone, the arrangement of fire points, and so on.

In addition to the deterministic sensing of ignition times, agents can also calculate the Eva_f of each fire zone [7], which related to the area of buildings in danger, number of neighbor buildings and fireness. Eva_f is defined as follows:

$$Evaf = area * Neighbor.No * fiery. \tag{2}$$

6 Agent coordination and communication

This paper suggests using the improved fuzzy Q-learning[6] algorithm in the MAS coordination and establishes a coordination model for this purpose. The model has a two-layer structure, the coordination layer and action layer, as shown in Figure 1. The objective is to achieve a coordinated and autonomous action for the intelligent agent.

Definition 1: Consider agents i and j. The fuzzy Q-function of agent i with

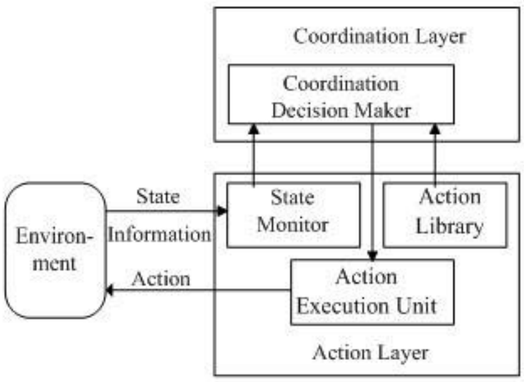

Fig. 1. Multiple Intelligent Agents Coordination Model Based on Fuzzy Q-Function.

state s_k^i taking action a_i is $Q(s_k^i, a_i, a_j)$. Here a_i,$a_i \in A_i$, and a_j,$a_j \in A_j$, are the actions of agent i and j, respectively, in their current states, and A_i and A_j are the action spaces of the two agents, respectively. The update of the fuzzy Q-function is given by

$$\begin{aligned} Q(s_k^i, a_i, a_j) = {} & [1 - B \cdot u(i) \cdot u(j)] \cdot Q(s_k^i, a_i, a_j) \\ & + \beta \cdot u(i) \cdot u(j) \cdot [r + \gamma \max_{a_i \in A_i} Q({s_k^i}', {a_i}', {a_j}')] \end{aligned} \tag{3}$$

Where ${a_i}'$ is the action of agent i after entering the next state ${s_k^i}'$, ${a_j}'$ is the predicted action of agent j with state according to the evaluation function, i.e.,${a_j}' = \arg\max_{a_j \in A_j} F({s_{max}^i}', a_j)$

The action layer is in charge of the action learning of the intelligent agent and executes the action decisions from the coordination layer. It consists of three parts: the state monitor, action execution unit, and action library. The state monitor senses the agent's state information from the environment and provides that to the coordination layer. The action library is a collection of the intelligent agent's possible actions, which will be provided to the coordination layer to process. The action execution unit executes the action handed down from the coordination layer, which makes decision based on the current environment states.

The major part in the coordination layer is the coordination decision maker. It analyzes and makes decision of the action of the agent, and then hands down to the action layer to execute and change the environment. After receiving the intelligent agent's states and possible actions from the action layer, the coordination decision maker first fuzzifies the states to obtain the fuzzy space of the agent. Then, it uses the Q-learning algorithm to evaluate the actions of other agents and learn their action strategy in order to modify its own action, resolve conflicts, and reach a better action decision.

7 Performance

All the strategies were used in CSU_Yunlu RoboCupRescue simulation team, and the resultes were validated the validity of these strategies. And some results are showed in figure 2 and figure 3 when running in the map of Kobe.

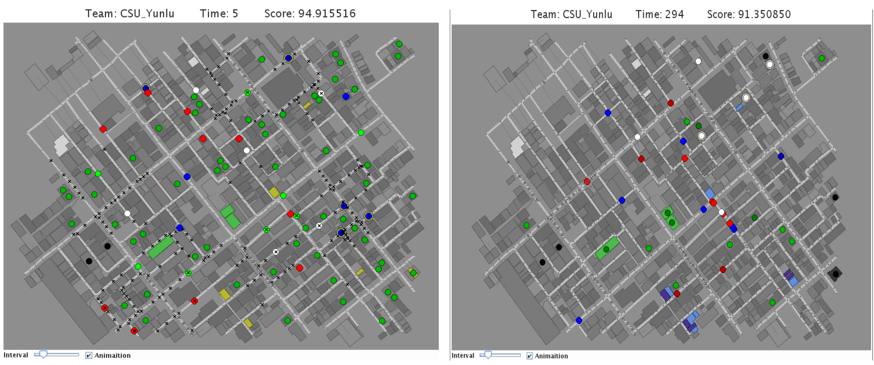

Fig. 2. In the beginning of simulation At the end of simulation

8 Acknowledgements

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

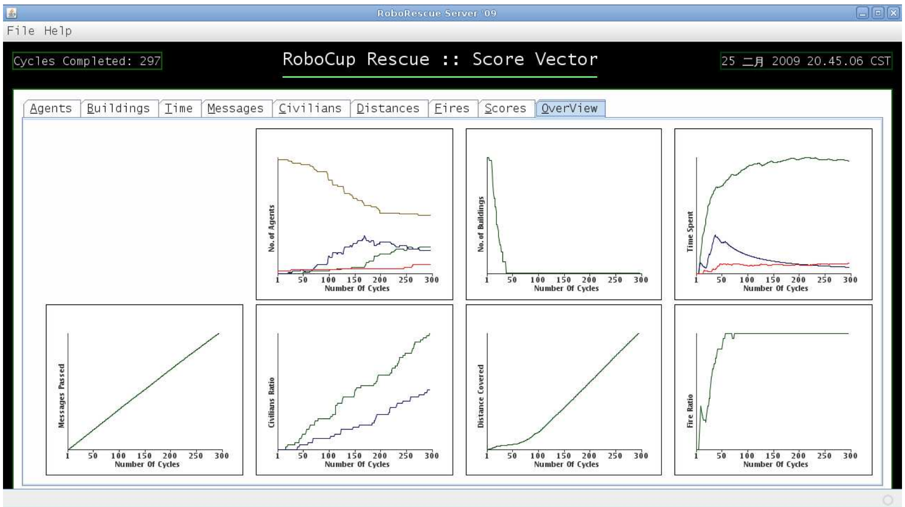

Fig. 3. Analysis of results

Particularly, the author would like to acknowledge the support of Professor Weihua Gui and Min Wu.

Thanks Yabapi for its source code.

Thanks Capsion2005 Team and ZJUBase Team we made modification based on their codes.

Thanks MRL Team and Impossibles Team for part of their source codes, which gives us a lot of inspiration.

Thanks IUST Team and SBCe_Saviour Team.

References

1. Peng J., Liu Y., Wu M., et al.: State Prediction based Dynamic Cooperation Algorithm in Multi-agent System. Journal of System Simulation. 20(20), 5511-5515(2008)
2. Chen, W., Decker, K.: Managing Multi-agent Coordination, Planning, and Scheduling. in: The Third International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1360-1361. New York (2004)
3. 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. In: Proceedings of the 2006 International Symposium on Collaborative Technologies and Systems, pp. IEEE, (2006)
4. Kloder S., Hutchinson S.: Path planning for permutation-invariant multirobot formations. J. Mol. Biol. 22(4), 650-665 (2006)
5. B. Fan, et al.,A Multi-agent Coordination Method Based on Markov Game and Application to Robot Soccer.:Robot. 27(1), (2005)
6. Peng J., Liu M., Lin K.Chi.: A Coordination Model Using Fuzzy Reinforcement Learning for Multi-agent. In: Proceedings of the 2007 International Symposium on Collaborative Technologies and Systems, IEEE Computer Society, USA (2007).