RoboCup Rescue 2013 – Rescue Simulation League Team Description MRL (Iran)

PooyaDeldarGohardani, PeymanArdestani, SiavashMehrabi, Mehdi Taherian, Sam Mirzaee ramhormozi, Mohammad Amin Yousefi

[1]Mechatronics Research Labratory, Islamic Azad University, Qazvin Branch, Qazvin, Iran

{Pooya.Deldar, Peyman.Ardestani}@Qiau.ac.ir
{Siavash.Mehrabi, Sam.Mirzaee, M.Amin.Yousefi}@gmail.com
http://www.mrl.ir

Abstract. In this paper we are going to introduce what we have done in MRL Rescue Simulation Team to participate in RoboCup 2013. For this upcoming event we utilize Hungarian algorithm to enhance assignments, this algorithm is applied to Police Force agents and Fire Brigade agents. Learning Automata is used in Ambulance team agents instead of Q-learning which shows improvements in tests. In partitioning we've changed our method from Voronoi to K-means. K-means guarantees the balance in partitioning. This partitioning is used to improve Police Force distribution in map.

Keywords: RoboCup Rescue Simulation, Hungarian algorithm, Learning Automata

1. Introduction

We have made major improvements on our code after 2012 competitions [1]; one of the most important improvements is utilization of Hungarian algorithm in our assignment problems [2]. Replacing former Q-learning methods with Learning Automata is another improvement in Ambulance Team Agents. The other major change is utilizing K-means [4] in our partitioning algorithm. In our new strategy for Police Forces we first divide the map into several partitions and then we use Hungarian algorithm to assign our functional agents (agents without buriedness) to the partitions; we also use a market based algorithm enhanced with a Learning Automata [5] instead of Q-learning for coordination in Ambulance Team Agents which results in faster and better performance. Our Fire Brigade agents are using a lightweight fire simulator and a new decision making algorithm to select buildings according to fire spread direction. Hungarian algorithm is used to assign agents to buildings on fire more effectively.

We are going to briefly introduce Hungarian algorithm and then explain how we utilized this algorithm in rescue agents, afterwards we are going to explain our new developments on Fire Brigade, Ambulance Team and Police Forces.

2. Assignment Method

One of the most fundamental problems in rescue simulation system is the assignment operation. In this system we always encounter a situation in which we have to assign limited number of agents to several targets and gain the maximum utility or the minimum cost. Thus this subject is considered as an assignment problem and we are seeking an efficient solution, compatible to rescue simulation characteristics which are shortage of calculation time and shortage of memory.

The assignment problem is a special case of the transportation problem, which is a special case of the minimum cost flow problem, which in turn is a special case of a linear program. While it is possible to solve any of these problems using the simplex algorithm, each specialization has more efficient algorithms designed to take advantage of its special structure.

Informally speaking, we are given an n×n cost matrix C=(c ij) and we want to match each row to a different column in such a way that the sum of the corresponding entries is minimized. In other words, we want to select n elements of C so that there is exactly one element in each row and one in each column and the sum of the corresponding costs is a minimum.

There are a lot of different solutions to this problem with different complexity orders [6] we have used Hungarian algorithm because of its simplicity of implementation and relatively low time complexity.

2.1. Hungarian Algorithm

It was developed and published by Harold Kuhn in 1955, who gave the name "Hungarian method" because the algorithm was largely based on the earlier works of two Hungarian mathematicians: Dénes Kőnig and Jenő Egerváry. And the time complexity of this algorithm is $O(n^4)$. The algorithm is consist of following steps**[7]**:

1. Arrange your information in a matrix with the "people" on the left and the "activity" along the top, with the "cost" for each pair in the middle.
2. Ensure that the matrix is square by the addition of dummy rows/columns if necessary. Conventionally, each element in the dummy row/column is the same as the largest number in the matrix.
3. Reduce the rows by subtracting the minimum value of each row from that row.
4. Reduce the columns by subtracting the minimum value of each column from that column.
5. Cover the zero elements with the minimum number of lines it is possible to cover them with. (If the number of lines is equal to the number of rows then go to step 9)
6. Add the minimum uncovered element to every covered element. If an element is covered twice, add the minimum element to it twice.
7. Subtract the minimum element from every element in the matrix.

8. Cover the zero elements again. If the number of lines covering the zero elements is not equal to the number of rows, return to step 6.
9. Select a matching by choosing a set of zeros so that each row or column has only one selected.
10. Apply the matching to the original matrix, disregarding dummy rows. This shows who should do which activity, and adding the costs will give the total minimum cost.

Occasions of this algorithm usage in our code, is mentioned in fire brigade and police force explanation section.

3. Fire-brigade Agent

Due to the important role of fire brigade agent in rescue simulation system, during recent years including 2012 [1] we have developed some utilities for predicting the fire spread model. Equation 1 and 2 were used to simulate the fire model in a lightweight simulator.

$$\text{Energy} = \text{building-temperature} * \text{building-size} \tag{1}$$

$$\text{Radiation} = \text{building-temperature} * \text{wall-overlap} * \text{beta}^{1} / \text{distance} \tag{2}$$

After predicting the burning area we surround the area with a convex hull and mark the outer buildings in the convex hull as important buildings. In the next step we try to find the best direction to approach and put off the fire.

In 2012 greedy method was used for fire cluster and target building selection. This method demands a very short CPU time and assigns nearest target to the agent; but it has some serious drawbacks, for example balancing the number of agents assigned to the clusters. To overcome this problem we have used Hungarian method; as described in first section of this report, a matrix is used for assignment.

Matrix weighting model is determined according to buildings fieriness, buildings dimensions and distance between the agent and buildings. Thus for each agent there is a list of targets with different weights, and then using Hungarian algorithm we can find an optimized assignment. This assignment can be centralized e.g. fire station and leader; or distributed on each agent (In scenarios with large enough communication bandwidth and almost synchronized data between agents, all agents' decisions will be equal).

$$\text{Weight} = (\text{building area} * \text{temperature} * f(\text{building fieriness})) / (\text{distance}) \tag{3}$$

$$f(\text{building fieriness}) = \begin{cases} 100 & if\ firiness = 1 \\ 10 & if\ firiness = 2 \\ 1 & if\ firiness = 3 \\ 0 & otherwise \end{cases} \tag{4}$$

[1] A coefficient related to Boltzmann Constant

In case of no radio communication between agents, fire brigade agents could collaborate in small working groups and utilize this model based on local data. However results will be almost equal to greedy methods.

4. Ambulance Agent

In recent years we have used different methods for ambulance agents' coordination. Since 2010, market based method [3] known as LIA[1], Market based method with obstinate agents known as MOA[2], and Q-Learning with obstinate agents method known as QMOA, are used. In upcoming challenges we will use market based method with obstinate agents and learning automata. Our methods in past challenges have some problems which resolved gradually by some changes. In table 1 a short description of drawbacks in each method are mentioned.

Table 1.perevoius approaches weakness and problems

Approach Name	Problem
LIA	Cause heavy processing on messaging system
MOA	Wrong number of agents assignment to targets
MOA+Q Learning	Taking too much time for learning, too many different conditions and a huge Q-table

According to above descriptions we were looking for a method which has advantages of reinforcement learning, but quicker and more dynamic in state selection. Thus we have combined market based method with obstinate agents and learning automata.

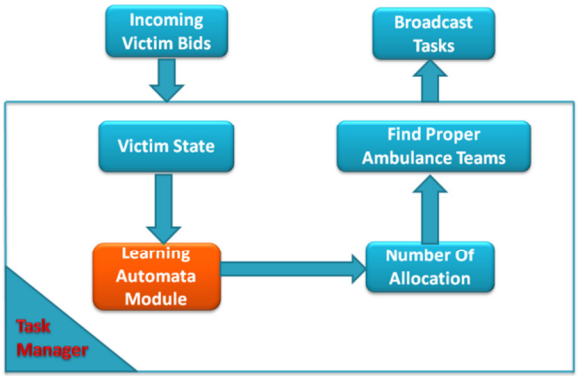

Fig. 1 Structure of Ambulance strategy with new learning method

[1] Leader Initiator Auction
[2] Market with Obstinate Agents

Learning automata has three different models which are P, Q and S. These models change based on continuous or discrete and conditions of results. Although different schemes of LRP, LRI and LReP based on used punishment rate in learning process are available for using. As shown in Fig. 1 location of learning Automata is as same as Q-learning, but learning method and updating and used structure in this method will be different. In learning Automata structure for each defined condition in environment we have a vector of available action in that condition, if we consider the amount of these actions as 'n' in pure chance condition probability of each condition will be 1/n. this structure can be seen in Fig. 2.

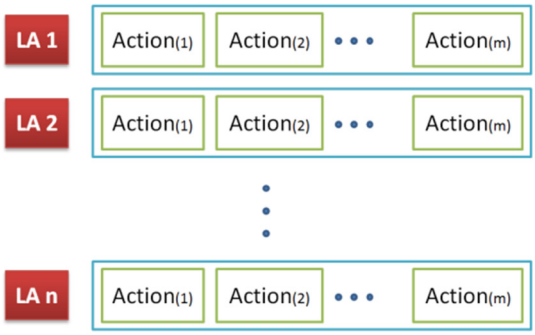

Fig. 2 Used structure for learning automata

For updating learnt values, equation 5 and 6 are used:

In the case of success, $\beta = 0$

$$\begin{cases} P_i(n+1) = P_i(n) + a \times (1 - P_i(n)) & \text{if i=j} \\ P_j(n+1) = (1-a) \times P_j(n) & \forall i \neq j \end{cases} \quad (5)$$

In the case of failure, $\beta = 1$

$$\begin{cases} P_i(n+1) = (1-b) \times P_i(n) & if \quad i=j \\ P_j(n+1) = \frac{b}{r-1} + (1-b) \times P_j & \forall i \neq j \end{cases} \quad (6)$$

In these formulas, β is environment output, p is probability of action selection, a and b are coefficient in range of [0, 1] and r is amount of authorized actions in machine. Used parameters in learning are shown in table 2.

Table 2: Different kinds of LA based on different parameters

Parameter Method	State Type	Scheme	A
LA1	TTD_BRD_Time	LRP	0.05
LA2	TTD_BRD	LReP	0.05
LA3	TTD_BRD	LRP	0.05
LA4	TTD_BRD	LeRP	0.05
LA5	TTD_BRD	LRP	0.01

Learning process during 1000 runs is shown in Fig. 3:

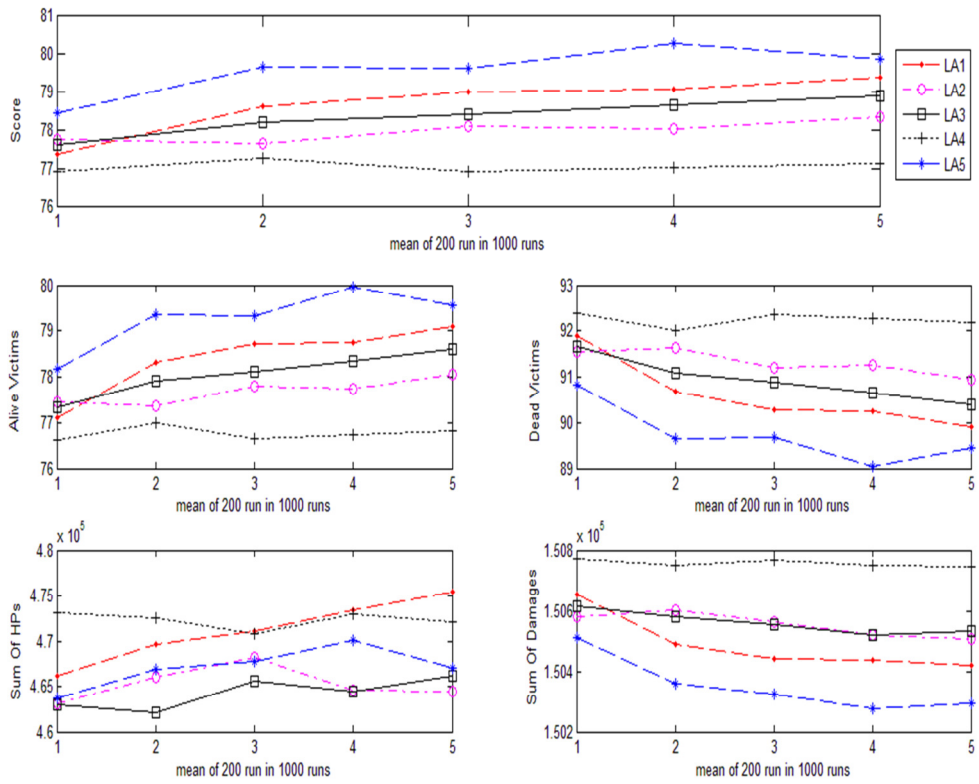

Fig. 3 Learning processes with different parameters

Table 3: Comparison of different methods on different maps based on score parameters

SCORE	Method Map	RiOne	roboAKUT	LA1	LA2	LA3	LA4	LA5
	Kobe	63.23	67.29	80.29	79.28	81.28	75.28	82.28
	VC	40.15	41.95	41.69	53.31	49.72	49.92	50.72
	Berlin	44.09	45.34	35.69	49.12	54.13	53.94	48.53
	Paris	41.48	44.12	36.67	43.09	54.10	48.90	46.10

5. Partitioning

We used Voronoi algorithm[8] for partitioning in past year. In this method some important points were specified on map as input for Voronoi algorithm to do the partitioning. But in new method we use K-means clustering. The problem of previous method was that partitions were not created based on number of buildings in each partition but they were based on geometric clustering, and this could lead to unfair partitioning and inefficient assignment of healthy agents to partitions. So we decided to create variable number of partitions based on count of healthy agents(agents with zero buriedness) then center of each partition will be determined using K-means algorithm, the similarity measure for K-means is a function of number of buildings in a partition and distance between each building and center of the partition. This guarantees that there is at least one healthy agent available for each partition. In Fig. 4 the output of new model is shown. Green circles show the center of each partition and lines show each partition's boundary.

5.1. Partition assignment

One of the problems that occur in using partitions is assigning agents to partitions. Greedy assignment may result in assigning an agent to a very distant partition that takes a lot of time to reach, sometimes the whole simulation. To solve this problem we use Hungarian algorithm. This algorithm results in relatively fast and optimized assignment.

Fig.4 Partitioning by K-means algorithm

References

[1] Deldar Gohardani, Pooya, Peyman Ardestani, Siavash Mehrabi, Mahdi Taherian, Sam Mirzayee Ramhormozi, and MohammadAmin Yousefi. *RoboCup Rescue 2012 – MRL Rescue Simulation Team Description.* Team Description Paper, Mexico city: International Robocup, 2012.

[2] Kuhn, H. W. "The Hungarian method for the assignment problem." *Naval Research Logistics Quarterly*, 1955: 83-97

[3] Deldar Gohardani, Pooya, Peyman Ardestani, Behrooz Masoumi, MohammdReza Meybodi, and Siavash Mehrabi. "Coordination of Ambulance Team agents in Rescue Simulation Using Auction Strategy." *International Conference on Affective Computing and Intelligent Interaction.* Taipei: Information Engineering Research Institute, 2012. 418-425.

[4] MacQueen, J. "Some methods for classification and analysis of multivariate observations." *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability.* University of California Press, 1967. 281-297.

[5] Narendra, Kumpati S., and Mandayam A. L. Thathachar. *Learning Automata: An Introduction.* Prentice-Hall, Inc, 1999.

[6] Rainer , Burkard, Dell'Amico Mauro , and Martello Silvano . *Assignment Problems, Revised Reprint.* Philadelphia: SIAM, 2012.

[7] Queen, Jirachi, Maluniu, and Krystle. "Use-the-Hungarian-Algorithm." *wikihow.* n.d. http://www.wikihow.com/Use-the-Hungarian-Algorithm.

[8] Aurenhammer, Franz. "Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure." *ACM Computing Surveys.* 1991. 345-405.