RoboCup Rescue 2014 – Rescue Simulation League Team Description
MRL (Iran)

Pooya Deldar Gohardani, Peyman Ardestani, Siavash Mehrabi, Mehdi Taherian, Mostafa Shabani

1 Mechatronics Research Laboratory, Islamic Azad University, Qazvin Brach, Qazvin, Iran

{pooya.gohardani, peyman.ardestani, siavash.mehrani, mustafa.shabani}@gmail.com

http://www.mrl.ir

Abstract. In this paper we will describe the preparations we have made to take part in RoboCup 2014. In this competition we are using K-means and convex-hull algorithms for clustering and partitioning. Hungarian algorithm is also used for assigning agents to the clusters. Using this algorithm guarantees that each agent's travel distance is minimized and the tasks are well distributed. For optimal assignment of ambulance agents to trapped civilians we are using a market based algorithm enhanced with learning automata, and to find the optimal positioning of the agents while performing searching tasks we have used Maximal Covering Location Problem.

Keywords: RoboCup, Rescue Simulation, K-means, Hungarian, Set Covering

1. Introduction

In 2012's competitions [1] we have used some algorithms that had acceptable results so we used them again as a base for our new code. In 2013 [2] we replaced the Q-learning for ambulance team agent with Learning Automata which resulted in minimizing the learning time and better performance on ambulance team agents. K-means and Hungarian algorithms also improved the performance on assigning different agents to different tasks. K-means algorithm provides a good clustering for resource distribution all over the map, and Hungarian algorithm helps us assign agents between these clusters so that the travel time is minimized. The Police agents use these algorithms to cover the map.

In addition to the above, we have made some changes in messaging system and our communication model. In 2013 we redesigned the communication model and managed to reasonably reduce the message size but there was still some issues in channel selection and bandwidth utilization, therefore we put more effort on optimizing the bandwidth utilization in 2014.

The old *clear* method was being used by our algorithm in 2013, due to the late release of the update on clear method and unforeseen problems that may have occurred by using the new clear method, which caused a low performance on clearing the roads by police force agent in comparison with other teams. For upcoming competitions we are using the new clear method which greatly enhances the performance of police agents. Also we have made some improvements to fire brigades' decision making and fire simulation model that we describe later. The new feature of our code is on search algorithm, we are using *Maximal Covering Location Problem* to enhance the performance of searching through the map by finding the best locations on the map for maximum area coverage and minimum travel distance. Therefore we can search more buildings with less movement instead of looking into buildings one by one. The described method is utilized by fire brigades to make sure that a fire site is thoroughly put off.

In this paper, first we are going to describe the noted algorithms which are K-means and Hungarian, then we are going to define the **MCLP** (Maximal Covering Location Problem) and then we will describe the changes on decision making algorithms for fire brigade and police force agents; after that we will explain how the communication works.

2. Assignment and Optimization

The optimal assignment of the agents to the different tasks is one of the major problems in rescue simulation. We always face this problem that each agent goes where on the map or attend to which task in a specific time. To tackle this problem we applied clustering for optimal distribution of agents in the map. Then using Hungarian algorithm we assign agents to the clusters.

Also we use Maximal Covering Problem to enhance our search algorithm; this method enables us to find the optimal points over the map to cover the whole map with minimum overlap on visited areas.

2.1.K-means

We first used K-means [3] on 2013[2]. This algorithm is very desirable because of its high performance and optimal result. After finding the clusters we use convex hull to define the limits of each cluster.

Basic K-means algorithm for finding K clusters
1. select k points as the initial centroids 2. assign all points to the closest centroid 3. recomputed the centroid of each cluster 4. repeat step 2 and 3 until the centroids don't change

2.2.Hungarian Algorithm

Hungarian Algorithm [4] is used for optimal one to one assignment of tasks to agents [5] and its complexity order is $O(n^4)$. The algorithm is based on an $n \times n$ matrix which is called Cost matrix = C(c ij).

Hungarian Algorithm
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.

Formerly we were using greedy algorithms and Euclidean distance for assignment by sorting the distance between all agents and all clusters, and then assigning the first agent with minimum distance to a cluster and so on. With this approach some agents will have a very short travel distance to their clusters but the others may be assigned to very distant clusters which results in an inefficient total travel distance. By using Hungarian algorithm we can optimize the total travel distance and all agents can reach their cluster in a fairly equal time span [2].

3. Maximal Covering Problem

One of the issues we were facing in fire brigade agent was that the agent should check for buildings on fire around a recently put off building and make sure that the building will not reignite by other fires. To accomplish this task we need to find the spots that, with the minimum number of this spots, we can check the maximum number of buildings, and minimize the time we need to check the buildings.

This problem maps to a general *Covering Problem* and specifically this is a *Maximal Covering Location Problem*. The maximal covering location problem was first introduced in 1975 in a paper with the same name by Richard Church and Charles Revelle.[10]

In combinatorics and computer science, covering problems are computational problems that ask whether a certain combinatorial structure 'covers' another, or how large the structure has to be to do that. Covering problems are minimization problems and usually linear programs, whose dual problems are called packing problems. The most prominent examples of covering problems are the set cover problem, which is equivalent to the hitting set problem, and its special cases, the vertex cover problem and the edge cover problem [6].

The maximum coverage problem is a problem that is widely taught in approximation algorithms. As input you are given several sets and a number k. The sets may have some elements in common. You must select at most k of these sets such that the maximum number of elements are covered, i.e. the union of the selected sets has maximal size. The maximum coverage problem can be formulated as the following integer linear program [7].

$$\sum_{e_j \in E} y_i \ , maximizing\ the\ sum\ of\ covered\ elements \tag{1}$$

The **set covering problem** (**SCP**) is a classical question in combinatorics, computer science and complexity theory. It is a problem "whose study has led to the development of fundamental techniques for the entire field" of approximation algorithms. It was also one of Karp's 21 NP-complete problems shown to be NP-complete in 1972 [8].

Given a set of elements {1, 2, …, m} (called the universe) and a set S of n sets whose union equals the universe, the set cover problem is to identify the smallest subset of S whose union equals the universe. The decision version of set covering is NP-complete, and the optimization version of set cover is NP-hard [8,9].

3.1.Greedy algorithm

The greedy strategy applies naturally to the set cover problem: iteratively pick the most cost-effective set and move the covered elements, until all elements are covered. Let C be the set of elements already covered at the beginning of an iteration, during the iteration, define the cost-effectiveness of a set S to be the average cost at which it covers new elements, i.e. c(S)/|S-C|. Define the price of an element to be the average cost at which it is covered. Equivalently, when a set S is picked, we can think of its cost being distributed equally among the new elements covered, to set their prices [8].

Greedy Set Cover Algorithm
1. $C \leftarrow \emptyset$ 2. While $C \neq U$ do Find the most cost-effective set in the current iteration, say S. Let $= \frac{costs(S)}{\lvert S-C \rvert}$, i.e., the cost-effectiveness of S. Pick S, and for each $e \epsilon S - C$, set price(e) = α. 3. Output the picked sets.

4. Fire Brigade

By reviewing our performance in the past we have noticed that the logic of fire brigade agent has problem in selecting its strategy to contain the fire. There are two different approaches defined for our fire brigade agent, greedy strategy and direction based strategy. The first approach is useful when the agent approximates that the fire is containable by only a few agents. The second approach is used when the agent finds out that the fire is big enough to spread very fast, and the best approach is to block the fire from one direction.

4.1.Greedy

This approach is inspired by ZJU's fire brigades. We first evaluate each cluster by parameters like: distance from refuge, energy, fieriness, total burning area and the distance from the agent which the latest has a bigger value than the others. After this evaluation the building with the highest value will be selected to put off.

4.2.Direction based

In this approach we split the map into 9 sections in a way that the fire site locates on center with another 8 sections surrounding it. This is done with 4 infinite lines crossing each other and forming the smallest possible rectangle around the fire site. Then we give a value to each cluster based on its total area of unburned buildings, number of refuges and number of gas stations. The line between center of the fire-site and center of the most valuable section is the starting direction to put the fire off. The agents focus on the buildings that are closer to the direction line. When all the buildings in that direction are put off we split the agents in two groups so that the group one continue to extinguish buildings in clockwise order and the group two will do the same in counterclockwise order. You can see the sample in figure 1.

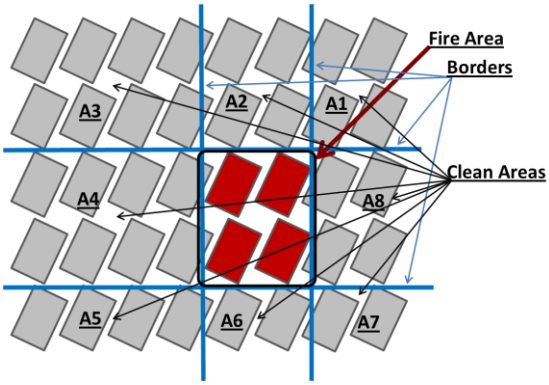

Figure 1. The whole map is split into 9 parts and the fire site is in the center

The other improvement on fire brigades is a procedure to make sure that the fire is completely put off. In the past, agents would change their target fire site, after putting off the first burning building they have found and randomly search somewhere else, while there are more buildings on fire just one block away that will reignite the fire. And because the agent thinks that there is no fire in that location anymore there is a good chance that the fire gets incontrollable in a few cycles. To avoid this, the agent will search around a recently put off fire and makes sure there is not any other building on fire in that area. But it can take a long time to visit all the buildings one by one, here the Maximal Covering Location Problem comes in handy to find the optimum locations on the map to see all the buildings from them so that we can check all the buildings around the fire as fast as possible (Figure 2).

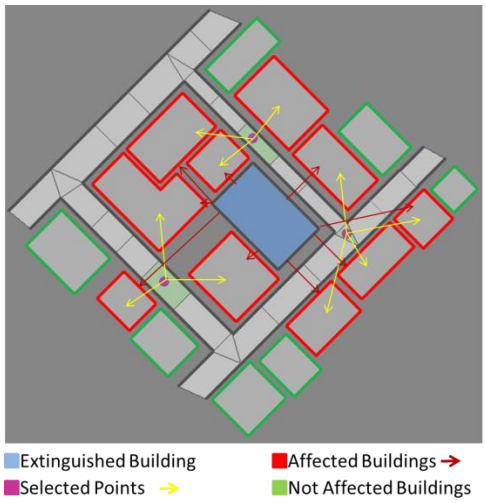

Figure 2. The spots on the roads are found by MCLP

5. Police force

Police force agents were using the old clear method to clear the blockades from the roads, last year. The old clear method removes a limited part of a block each cycle, which is very slow comparing with the new clear method. The teams that were utilizing the new method had the upper hand, so we decided to use the new clear method for the next RoboCup. According to how the new clear method works, we are using imaginary lines, called guidelines.

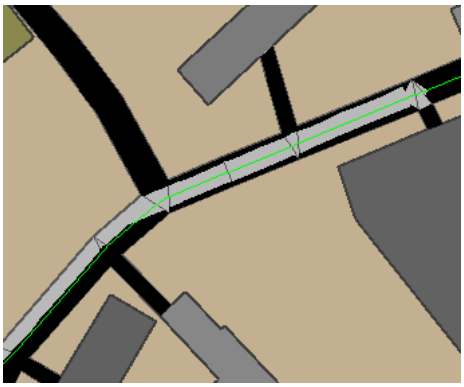

Figure 3. Clearing the roads using the guideline

We clear the roads along the guidelines to smoothly clear the road without any residues so that agents can move along faster, without getting stuck.

6. Messaging

Formerly we were using only one channel (the channel with the maximum bandwidth) for communication; this was limiting our bandwidth and communication. Our recent improvement on communication is that we reserve multiple channels for agents based on their priority, so that ambulances, fire brigades and police forces have 4, 3 and 2 shares from bandwidth, respectively. Then based on available channel subscription limit, we split the channels between the agents so that the agents with highest priority subscribe to the biggest channel available and so on, until all agents subscribe to the maximum number of the channels they are allowed to.

7. References

1. Deldar Gohardani, P., Ardestani, P., Shabani, M., Mehrabi, S. and Hooshangi, V., RoboCup Rescue 2012, rescue simulation league, team description, MRL (Iran), (2012).
2. Deldar Gohardani, P., Ardestani, P., Mehrabi, S., Taherian, M., Mirzaei Ramhormozi, S. and Yousefi, M.A., RoboCup Rescue 2013, rescue simulation league, team description, MRL (Iran), (2013).
3. Steinbach, M., Karypis, G. and Kumar, V., A Comparison of Document Clustering Techniques, University Of Minnesota, Technical Report, pp. #00-034, (2000).
4. Kuhn, H. W., The Hungarian method for the assignment problem, naval reaserch logistics quarterly, pp. 83-97, (1955).
5. Burkard, R., Mauro, D. and Silvano, M., Assignment problem, revised, reprint. Philadelphia: SIAM, (2012).
6. Wikipedia, Covering Problem, http://en.wikipedia.org/wiki/Covering_problem, (2013).
7. Wikipedia, Maximum Coverage Problme, http://en.wikipedia.org/wiki/Maximum_coverage_problem, (2014).
8. Vazirani, V. V., Approximation Algorithms, Springer, ISBN: 978-3-662-04565-7, (2001).
9. Wikipedia, Set Cover Problem, http://en.wikipedia.org/wiki/Set_cover_problem, (2014).
10. Church, R. and Revelle, C., The Maximal Covering Location Problem, Journal of Regional Science Association International, Vol. 32, pp. 101-118, (1974).