RoboCup 2017
Rescue Simulation League Team Description
S.O.S (Iran)

Kasra Darvishi, Amin Rashidbeigi, Reza Bayat, Nima Tavassoli,
Mehrnaz Payami Shabestary, Leila Azimi

Robotics Research Center, Department of Computer Engineering and Information Technology,
Amirkabir University of Technology, No. 424, Hafez Ave, Tehran, Iran

[kasra96.d, a.rashidbeigi, reza.bayat76, nima.tavassoli2, mn.payami] at gmail dot com
[l.azimi75] at yahoo dot com

Abstract

In this paper we will describe S.O.S team's general strategy and features which are implemented to increase agents performance. Our new approach to control fire spreading is introduced and ambulance team decision making process is described in detail.

1 Introduction

In the past few years S.O.S team developed efficient strategies on full and low communication. Since we are not allowed to use communication, for 2017 Robocup competition, the main focus of S.O.S will be on no-communication map. We developed no-communication strategies of previous years and tried to convert our previous methods and tools to new modular format (ADF). In addition to implementing recent algorithms on new server we designed strategies for Ambulance and Fire Brigade agents, which is described in agents section.

2 Modules

2.1 Clustering

In previous years, centroid-based clustering algorithms was used, such as k-means or star clustering. this year, we improved our clustering algorithm by Enhanced K-means approach[1]. This approach decreases time complexity and increases the accuracy of dividing map to regions. One advantage of this approach is to find the centroid of each cluster systematically. It results in calculating the position of centroids in less time and more accurate. This algorithm divides the map related to the number of agents.

2.2 Target Allocators

2.2.1 Ambulance Team

The main goal of the ambulance team is to rescue maximum possible number of humans until the end of simulation. In this year since communication between agents or central buildings was not allowed, agents do not have the ability of getting synchronized by messaging or central buildings. Therefore we focused on task allocating in no-communicating environments. We applied a self organized task allocation strategy in which no communication is exploited among agents and needs no central control to govern the agents behavior[4]. The allocation of agents to victims is performed based on particular specifications like burriedness of civilian, estimated dead time and the length of the shortest path between agent and civilian. Dead time estimation is performed by our teams recent method which presented in 2015 competitions. Main step in this strategy is finding the optimum number of agents that are needed to remove the burriedness of victim before a specific time that will be described in detail. After that, each agent decides which civilian should be ignored and which one should be rescued. In the last step each agent allocates itself to the appropriate victim. Assignment is performed as described below.

finding the optimum number of agents
In order for the allocation strategy to exploit the current number of robots available for allocation, the time between the start of simulation and the latest deadline, is divided into periods: $\pi_1, ..., \pi_m$ where: $\pi_i = D_i - D_{i-1} \forall i \in \{2, ..., m\}$ The first period has the length of the earliest deadline ($\pi_1 = D_1$). The allocation is done at the beginning of each period, so agents can make new decisions based on new informations about victims. The process of finding optimum needed number of agents starts with finding the probability of event $E_i(K_i)$. Ki is used to denote the number of burriedness units which could be cleared by a single agent from civilian C_i within a specific deadline D_i. The value of K_i is taken from zero to total

buriedness of civilian.

$$D_i = DeadTime - TimeToRefuge \tag{1}$$

And $E_i(K_i)$ is defined as an event that one agent(A_j) can clear K_i units of civilian C_i's burriedness before D_i.The probability of event $E_i(K_i)$ is calculated by the following formula:

$$P_{ij}(E_i(K_i)) = \frac{1}{2}[1 + erf(\frac{(D_i - K_i) - L_{ij}\mu}{\sqrt{2L_{ij}\sigma^2}})] \tag{2}$$

Which L_{ij} is the length of the path between agent and civilian , μ and σ are the mean and standard deviation of $\frac{1}{v}$ which v is the distance traversed by agent in each cycle and $D_i - K_i$ is the deadline for the agent to reach victim, since one unit of burriedness can be cleared per cycle. And the "erf" represents the "Error Function".
This probability is used to find the number of needed agents if we assume the path length of all available agents same as agent A_j.

$$N_{ij} = \lceil \frac{Burriedness_i}{P_{ij}K_i} \rceil \tag{3}$$

The total number of needed agents is the mean of calculated numbers by (3) for all available agents. For different values of K_i we choose the minimum value of $N_i(K_i)$ as the optimum one. This reduces interaction between agents and increases the chance of surviving more victims.

The total strategy of our ambulance team agent would be to :

1. Considering the victims and ignoring those ones which are impossible to be rescued.
2. Calculating the optimum number of agents for each victim.
3. If sum of the number of needed agents is more than available agents, ignore the civilian that needs the largest number of agents to be rescued, according to victims dead time. then repeat this step.
4. Agents due to the distance between civilian and other agents, Allocate themselves to their proper target.
5. After burriedness is removed one agent transports civilian to refuge and process repeats for other agents.

This approach will result in the rescue of each civilian by an individual agent in the initial cycles of simulation. and in the final cycles in which it is not possible to act as initial cycles, agents will co-operate with each other to save victims that were ignored before.

2.2.2 Fire Brigade Team

Our main strategy for the fire brigades, is similar to previous years, no communication. However this year we worked on some conditions in which the fire brigades dont have the ability to put the fire off completely and the only solution is to prevent the fire from spreading into the center of the map.

In some states, such as when the number of agents are not enough to extinguish the entire fire, it is easier to lead the fire to the corners of the map in order to prevent fire extension, it also causes less cost and damage compared with leading the fire to the other points of the map or acting similar to recent strategies.

The first step is trying to find the outer points of the map in the precomputing phase, the procedure was inspired from the Convex Hull algorithm [2]. For every three consecutive points from obtained polygon an area will be created (figure 1). We observe the starting position of the fire, and the direction of its spreading to the other regions. Based on these datas, we detect the corners that potentially can be used(around the fire zone). Appropriate corners have the following features.

1. Short distance between the center of intended fire
2. Having an area which is big enough to trap the fire.
3. The number of civilians is minimal.
4. gas stations, fire stations or refuges are not placed on the way of fires spreading direction.
5. the angle must be acute.

Fire brigades choose the appropriate corner based on the above features, then they try to lead the fire towards the chosen corner by standing at certain points which is described in next section. Afterwards, extinguishing the fire in a particular area is more controllable. This approach is used only if there are more than one corner near the fire.

Fire Brigades position

In previous years our team had problem finding suitable position of fire brigades, so the fire could not be extinguished efficiently.Clearly It is essential to prevent the fire from spreading into unexpected points, by minimum number of agents. To achieve this good we came up with an idea which was presented by the MRL team TDP 2016 before[3]. They used Maximal Covering to position their agents that reaching the next target with minimum movements would become possible. In this algorithm as inputs we have the position of the buildings around thefire zone, maximal covering selects the points at wich the maximum number of buildings are reachable by the fire brigades.

As you can see in the figure 1, the corner named as γ (the angle between AC and BC) that creates an area in a triangle with sides a, b and c.

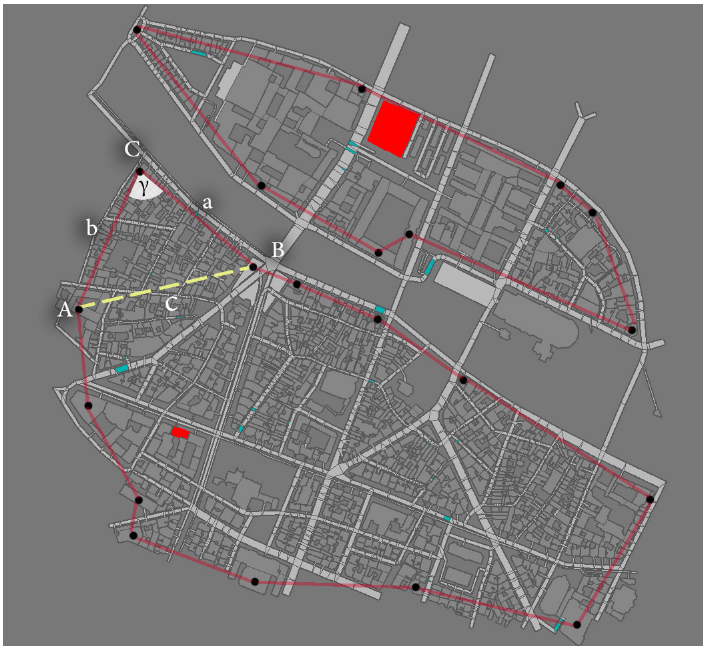

Figure 1: The procedure of detecting polygons and corners in pre-compute phase

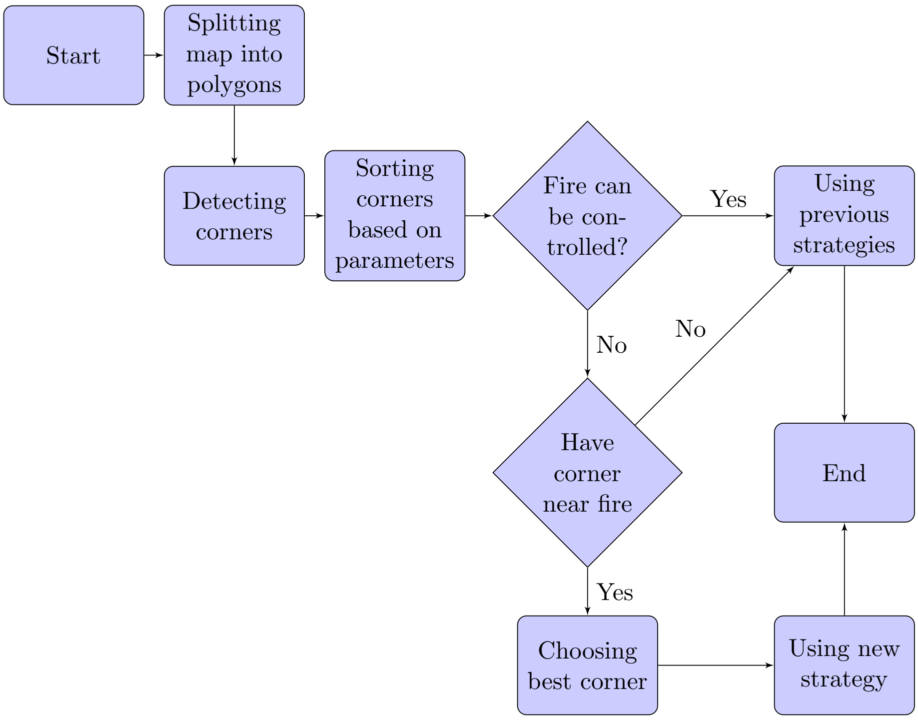

Figure 2: The procedure of fire extinguishing

2.2.3 Police Force

Earthquake cause destruction on roads. The main duty of police forces is to clear roads from blockades. So that the other agents will be able to move through the map faster via the shortest path. The priority in clearing operation is that fire brigades gain access to fire zones and ambulance team gain access to civilians. After pre computation, The highest priority is removing blockades in the roads that connect two clusters.

Make Fire Brigade reachable to Fire zone and Fire probable state Priority of Police Force's agents is to make a connection between fire bridge agents and fire zone. But if they want to clear all roads between fire brigades and fire zones they lose time and energy. So if a police force agent senses the fire, it tries to estimate the fire size. Based on the fire size if it is possible to control spreading fire, they would clear the road to that fire zone .Other strategy is that Police Force agents try to estimate the probable place of fire if they can sense the temperature of buildings. They select buildings with high temperature and if an agent is near to those buildings, it clears the road between them.

3 Preliminary Results

This year our agents performance, in no communication, is improved by new decision making process for ambulance agents. Fire brigades ability to prevent fire from spreading is increased. Our clustering method is faster and more accurate than last year. According to following table (figure 3), now we have achieved better results in comparison with MRL and Posiedon results from last year.

Result Table			
City	MRL 2016	Poseidon 2016	SOS (new strategies)
Kobe2	42.66	24.72	44.59
Paris1	127.36	98.99	131.19
Istanbul1	147.13	126.50	139.21
Berlin1	54.23	65.51	58.67

Figure 3: Comparison results between MRL 2016, Poseidon 2016 and new SOS strategies.

References

[1] G. Sathiya and P. Kavitha. An Efficient Enhanced K-Means Approach with Improved Initial Cluster Centers. Anna University, India, 2014.

[2] Discrete and Computational Geometry - 2011 Chapter 2, Satyan L.devadoss and Joseph O Rourke Princeton University Press, Princeton and Oxford.

[3] Pooya Deldar Gohardani, Siavash Mehrabi, Peyman Ardestani, Erfan Jazeb Nikoo, Sajjad Rostami, Mahdi Taherian. RoboCup Rescue, MRL, Iran, 2012.

[4] Yara Khaluf and Franz Rammig, Task Allocation Strategy for Time-Constrained Tasks in Robots Swarms, Heinz Nixdorf Institut, University of Paderborn, Germany, 2013.

[5] Angeh Aslanian, Aramik Markari,Faraz Falsafi Mohammad Ehsan Soleimani Yarandi, Salim Malakouti Navid Haeri, Reza Hoseini , Morteza Rezayi Khoshdarregi Nima Mokhtarian, Seyed Mohammad Reza Modares Saryazdi, RoboCup Rescue, S.O.S, Iran, 2010.

[6] Hesam AkbariBoshrabadi, Pegah Taheri, MohammadMilad Ameri, Marjan AlBouye, Morteza FaghaniLemraski, RoboCup Rescue, S.O.S, Iran, 2015.