# RoboCup Rescue Simulation League Team Description <Kherad (IRAN)>

< Saina Ghadiany, Saba Kiaei, Mahshid Eskandari, Mana Atarod,

Manzoumeh Kherad Institute
IRAN

{ghadianysaina, sabakiaei}@gmail.com,

eskandarimahshid@yahoo.com, mana.atarod@gmail.com

**Abstract**. Kherad team was formed in spring 2013, consisting of two members with some programming experience. To improve the level of the competitions, each year codes are being released in Rescue Simulation league. Therefore, Kherad team decided to use S.O.S base code as its base. Kherad team was ranked fifth at IranOpen RoboCup Competitions, 2014, Iran, and fourth at RoboCup Competitions, 2014, Brazil. In summer 2014, nine other members were added to the team. This combination caused new ideas to be added to the strategies, and therefore enhanced the team's performance greatly.

## 1.Introduction

Each year, millions of people die because of natural disasters. With a practical disaster management, this statistics can be decreased to the minimum. The purpose of the Rescue Simulation league is to provide a platform based on reality in which rescue operation strategies can be tested. This project is a new practical domain of RoboCup, found to provide emergency decision support by integration of disaster information, prediction, planning, and human interface. A generic urban disaster simulation environment is constructed on network computers in which heterogeneous intelligent agents such as Fire brigade, Ambulance team, and Police officer conduct search and rescue activities. Real-world interfaces such as helicopter image synchronize the virtuality and the reality by sensing data. For disaster researchers, RoboCup Rescue works as a standard basis in order to develop practical comprehensive simulators adding necessary disaster modules. This paper explains Kherad's work in this area.

## 2. Clustering

Clustering plays a sufficient role in how agents coordinate and do their tasks. The use of an efficient algorithm would enhance the performance generally. K-means algorithm was chosen for this purpose. [1, 2] This algorithm is used separately for each agent type, and therefore the number of clusters differs regarding the type of agents. The number of clusters for each type is equal to the number of agents, and is specified using the K-means algorithm. K-means clustering aims to partition observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. [1, 2]

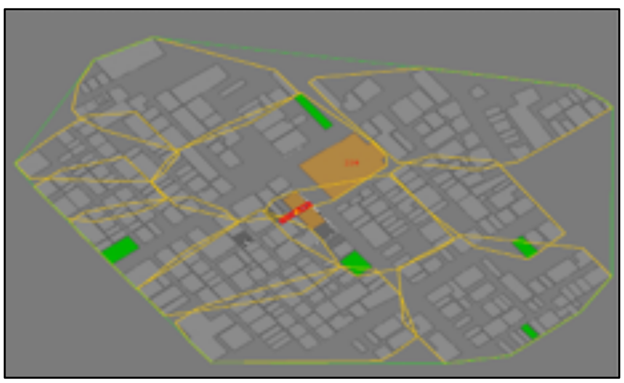

**Fig. 1.** Clustering a map using K-means algorithm

By the use of Hungarian algorithm, afterwards, each agent is linked to its cluster. [3] The Hungarian method is a combinatorial optimization algorithm that solves the assignment problem in polynomial time and which anticipated later primal-dual methods. [3]

## 3. Agents/Agent skills and action selection

### 1- Fire Brigade Agents

The fire brigade's responsibility is to extinguish the fire that may take place in the city during an earthquake and preventing it from spreading. This task is defined to reduce the severity of the damage done to the city and to provide the ambulance agents a more suitable situation so that they can do their duty more efficiently. Without having a good strategy, the city will most probably burn and get damaged, and this could result in a substantial number of losses.

### 1.1 Fire Zones

Sometimes, the spreading of the fire is so fast that the burning building's neighbors may be destroyed before the agents even get a chance to reach the building and extinguish it. In such cases, controlling the fire and preventing it from spreading is a better option. As a result, instead of targeting the building on fire, the high-temperature buildings around the area on the verge of getting burned are extinguished and their temperature is controlled. This will prevent the fire from spreading any further, and will protect the city from further damage. To reach this goal, the following steps are followed to generate a fire zone.

Firstly, a random on-fire building is chosen and its surrounding buildings get checked. If its neighbors are on fire or have a high temperature, they are added to the fire zone. When all of the respective building's neighbors are checked, the same action is repeated for added neighbors to the fire zone in the previous state. The cycle goes on until all the surrounding buildings are either added to the fire zone or have a low temperature.

### 1.2 Scoring

Scoring process is being done for Fire zone and it's buildings separately . An agent should choose the most efficient fire zone and start with the most significant building in that fire zone.

### 1- Fire Zone Scoring

The score is given to each fire zone according to some factors. These factors are as below:

1. **The existence of Gas station:** Gas stations are major threats during earthquakes. If the fire reaches a gas station, it could result in a tremendous explosion, and that means a lot of damage done to the city. So it is better to be more cautious with fire zones that are around a gas station.

**Fig. 2.** A Fire Zone

**Fig. 3.** Estimated Fire In That Fire Zone

2. **The number of the buildings:** A reason as to why this number is important is that in case the number is high, the city will get more damaged.

3. **Proximity to the center:** When a fire zone is closer to the center, it can spread from all angles and directions. Therefore, extinguishing it would be much harder and it would need more agents.

4. **Proximity to the corners:** When a fire starts in one of the corners, in case there are a lot of fire zones in the map, it would be considered less important since it spreads from few directions and will most probably cause little problem. This means that in the scoring, if the fire zone is in a corner, it gets negative points. For finding the corners, a Convex hull algorithm is used.

5. **The number of civilians:** As the life of the civilians is a significant factor, controlling the fire spread's direction must be based on that factor.

## 2.Building Scoring

1. **Extent:** Most of the time, the more the fire zone's extent is, more likely it is to cause damage. Therefore, it holds a lot of importance.

2. **Estimated fire spread:** The number of the buildings till the edge of the map or another fire zone or a long and wide road is being counted for each building and the building with the most neighbors is considered as a very important building since it would increase the spread of the fire greatly.

3. **Number of neighbors:** With more neighbors for a fiery building, more buildings are in danger; therefore, buildings with more neighbors weigh more than others.

$$\text{Weight Building} = \left( \frac{GasstationDis_b}{maxGasstationDis_b} + \frac{Area_b}{maxArea_b} + \left[1 - \frac{Temp_b}{maxTemp_b}\right] + \frac{\#Neighbour_b}{maxNeighbour_b} + \frac{1}{predictFieryness_b} \right) * \frac{CornerDis_b}{maxCornerDis_b}$$

4. **Temperature:** As the temperature increases, the building weighs more since its ability to put other buildings on fire increases.

5. **Fieriness and ignition:** Created fire zones may contain un-fired buildings for which the risk of ignition is high. Since the goal is the least damage, preventing them from getting ignited is the proper action.

6. **Lead of the corner fire:** when the number of the fire zones in a map is more than a specified number, the agents won't extinguish the corner fires. They just lead it to the end of the map with extinguishing some specified buildings in that fire zone. It's because corner fire zones can't spread the fire and they're less important than the other fire zones. Even a few agents can execute this action and as a result other agents could be prepared and free for more important fires

7. **Fieriness Prediction:** This factor is used for finding the connection between Fieriness and Temperature. For each Fieriness, considering an average primary temperate and using Moving Average approach, we get to an average temperature, and using each building's temperature, we map that building to its correlated group.

With building with fieriness f, temperature temp in cycle c:

$$Temp_{avg_{f,c}} = 0.8\ Temp_{avg_{f,c-1}} + 0.2\ temp$$

$$T_{avg_{f_1,t}} < temp < T_{avg_{f_2,t}} \Longrightarrow PredictFieriness_{temp,f} = f_1$$

Score formula is calculated as follows:

Each of the elements stated above has a specific coefficient that after performing the needed mathematical procedure on them, they define the net score.

## 1.3 Agent skills and action selection

### 1. States

A fire brigade's action selection depends on its current state. In order to find the right action for its current condition, it checks the defined states in the order they were prioritized.

### 2. Assigning

Fire brigade main duty is to check its cluster and in case of fire, extinguish it. When there is no fire in the cluster or the fire zone in the cluster is finally extinguished, the fire brigade starts checking the buildings in and around that area to make sure there is no more fire threatening the cluster. Afterwards, it will leave its cluster to help the rest of agents.

The number of fire brigades for each fire zone is decided according to it's score in comparison to others. In order to ensure about each fire's spreading, free agent is responsible for helping it's neighbors, so it will choose the nearest cluster to itself that is in need of more agents. [4]

### 3. Interrupts

Whenever a fire brigade faces a small fire when moving along the map, it will extinguish it to prevent it from getting bigger and forming a major fire that could lead to more serious damages to the city.

### 1.4.Predicting fire

In order to have the best results, sometimes it is better to predict the probable disasters that may occur to prevent them from happening or to at least be ready in case. For fire brigades, they do this according to the temperature of the fire zone's neighbors and the places where fire is more probable to happen, such as the areas around the gas station and where a fire had taken place in past cycles. This way, they can take precautions and decrease the possible damage done by the fire.

## 2-Ambulance Agents

The main responsibility of Ambulance Agents is to rescue damaged civilians. The most important part in Ambulance team function is to select the best civilian in order to be rescued. If the Ambulance Agents can decide properly, we can be sure that every important civilian in map will be rescued. The importance of each civilian is being calculated in each cycle according to it's health conditions.

### 1. Center-based decision

If the connection is in the high level, means that all agents can send message to others, there will be a center specified to makes decisions for other ambulance agents. This decision was made based on the fact that a center with all information would decide more efficiently than any other ambulance.

### 2. Cluster Assigning

If he communication is in the lowest level, means that agents' communication with each other is bound to trying for sending messages, each agent must decide for itself. In these situations, there will be a cluster specified by K-means and Hungarian algorithm for each agent.[1, 2, 3]

### 3. States

### 1. State Explanation

Kherad's strategy is state-based. In state-based strategies, a number of states' defined conditions are checked each cycle, and whenever a state's conditions are all verifiable, that state's actions get done.

### 2. Choosing civilian

The main strategy for choosing the best civilian is the same in self-agent-based decision or center-based decision. Nearly 10 cycles after the start of the run the decisions will be made. This delay is to ensure that at least a few civilians are identified and the best decision is going to be made. At first, each civilian will get a cost according to its specifications as its burriedness, death time, number of ambulances it needs to be rescued, etc. Then, starting from the most important civilian, each civilian will get some ambulances, according to how many ambulance it needs to be completely rescued, for itself. By this strategy, we can ensure the most important civilians will definitely be rescued.

This year, cost formula is much more accurate with the number of ATs needed being added to it. Last year we had the problem of not enough ATs needed for a Civ reaching it because of noise' existence or problems in communication. With this modification and adding this factor, Civs with less ATs needed are preferable, and the probability of this dilemma's happening is lessen. Moreover, all measures are translated into a number between zero and one, resulting in the formula's being accurate enough in each factor's influence on the cost.

$$Cost_{civ} = \frac{\#ATNeeded}{\#FreeATs} + Burriedness + (HP \times dt)$$

Other factors added to center's civ assigning include the number of civ's in a cluster. With this condition, cost's probability is calculated as following :

$$Cost'_{civ} = \sum_{i=1}^{\#CivsinCluster} cost_i$$

$$P_{cluster} = \frac{Cost'_{cluster}}{\sum_{j=1}^{\#clusterinMap} cost'_j}$$

For each civ, this probability of civ's cost is calculated, multiplied by above cost, and considered as the final cost. Afterwards the cost of this civ for the AT is calculated using following formula :

$$RescueTimeCost_{civ-AT} = MovetoCivTime + LoadCycle + \frac{Burriedness}{\#ATNeeded} + MovetoRefugeTime + UnloadedCycle$$

Each Civ's cost for each AT:

$$Cost_{civ} = \frac{\#ATNeeded}{\#FreeATs} + Burriedness + (HP \times dt)$$

Death time is a somewhat measures according to the civilian's burriedness, health, hp, etc. The death time is being calculated using the particle algorithm. A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best-known position in the search-space as well as the entire swarm's best-known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

## 1. Center-based Decision Making

In high-level communication, center has information about almost every civilian and ambulance agent; therefore, the best person to decide for every one is the center. Based on the cost given to each civilian, center starts by assigning ambulances to each civilian starting from the top of the civilians' values list. [4] By following the order of the list, it is guaranteed that most valuable civilians would get enough ambulances for being completely rescued.

Each ambulance has a specific free time specified by the time that ambulance is going to be free for doing another task. Whenever an ambulance is being chosen for a civilian, that civilian's rescue time will be added to that specific ambulance's free time. Therefore, for the next civilian, that ambulance will have a new free time. For choosing the best ambulance for each civilian, the value of that ambulance for that civilian will be calculated. This decision is provident because of the number of civilians that will be rescued according to this specific decision. Center by using the following formula lessens the probability of an inefficient decision.

$$\text{Ambulance value for each civilian} = \text{ambulance free time} + \text{distance to the civilian} + \text{burriedness} + \text{time to refuge}$$
$$(\text{this formula will give the time in which the civilian will be rescued})$$

One significant issue in high-level communication is the communication between ambulances and fire-bridge agents. Ambulances can inform fire-bridge agents by messaging them about fire bridges which might be dangerous for lots of civilians, or on the other hand, fire-bridge agents can inform ambulances about a filled-civilian place in the map. Center will decide if the ambulances should go to where they have been informed of or not.

## 2. Ambulance decision

### 2.1. Full-Level Communication

In full-level communication, though the center would assign a civilian to each ambulance agent, each ambulance would make a list consisting each civilian along with a cost calculated for it as well. When calculating the cost, however, the ambulance adds the factor of its distance to that civilian as well. This would prevent unnecessary moves to happen since the cost of each move in the map could be disastrous.

### 2.2 Low-level communication

In low-level communication, each ambulance will decide according to its cluster civilians until it has searched at least two third of the building of its cluster. One factor in low-level communication is the existence of Virtual Civilians. Virtual civilians are those whose information is not accurate enough for a reliable scoring. The ambulance would give a cost to each virtual civilian. However, this time it only chooses that civilian in case its value is clearly noticeable. Otherwise, choosing a civilian that its information might be wrong is not worth the time.

### 2.3 No-level Communication

In no-level communication, ambulances can only inform others about their situation by talking to them from a near distance. When an ambulance enters the refuge, it talks to other ambulances and shares its information. This would increase the chance of an ambulance being informed if any other ambulance is facing a hard condition in its cluster i.e. having lots of civilians at a place. The strategy is similar with the low-level communication strategy, with the difference of no virtual civilian's existence.

One of the risks in low-level communications is the same decision being made by different ambulances. Ambulances will send a message about whom they had chosen to rescue in case that the message will be received. Otherwise, each ambulance is aware of the others decision making algorithm. Therefore, they won't choose the same low valuable civilian to rescue.

## 3- Police Agent

The police agent's duty is to clear the paths and make the transportation between different points of the map available. This agent's strategy is based on clustering which has been explained above.

### 1. Police agent's Decision Strategy

### 1.1 Decision Making

Each police has it's own cluster and each cluster is divided to some parts by K-means algorithm which is used for clustering too- for a more accurate and better decision making. [1, 2] Police's decision making is state-based. However, some interrupts have been defined as well as to make sure significant situations are handled right away.

### 1.2 Interrupts

Interrupts are always prioritized to the states. Before any state's conditions being checked, an agent checks to see if any interrupt's condition is verified. Interrupts' specified actions get done even if an agent has a task defined by its states. After the interrupt's action is done, the agent continues following its state-defined task.

The most significant interrupt happens when an agent faces a closed refuge or a stuck agent. Another, but less important interrupt is opening stuck civilians. In this case if the final target is a civilian, the move time must be reduced from the death time of the final target. This interrupt's action is only acceptable if it doesn't put any other civilian in risk of running out of time

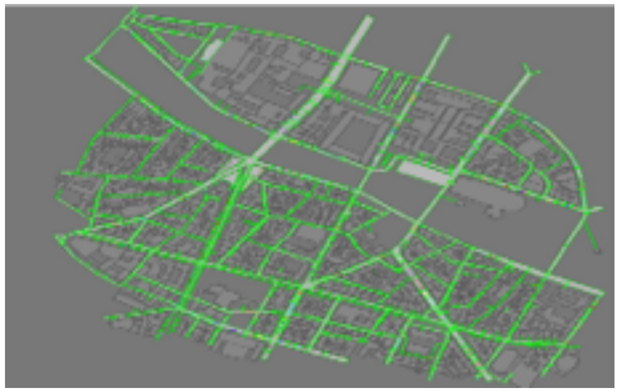

**Fig. 4.** The road graph between buildings which police agents use to clear the roads

## 2. Prioritizing/ States

The importance of paths and roads differs; as a result, they must be prioritized. Clearing the roads that lead to the refuges are of greater importance, and therefore each agent will make sure that any refuge in its cluster is open. The next action to be done is handling after shock situations. Whenever a shock happens in the middle of the run, police agents need to check every important cleared path again to make sure nothing has been damaged. Refuges' paths might get closed and agents might get stuck again after a shock occurrence.

Another task to be done is opening stuck agents. This could cause disaster since a stuck agent could never fulfill its duty. Police agents need to make sure no agent is stuck whatsoever. For choosing which agent to rescue first, each stuck agent gets a value according to the tasks it might do when released. Fire agents are crucial to be rescued since fire's spreading could result in the map's fully burning. This is a general fact; however, agents' priorities could change dynamically based on the situation.

Whenever an agent doesn't have any other task to do, it enters the search state where it is responsible for searching the map, clearing the remaining paths, and gathering information about the map.

## 3. Fire's Path

Fire agents cannot fulfill their duties unless they have a way for reaching the fire that is spreading. Each fire zone will be given a specified weight and that weight will be calculated by using the fire brigades' formula. According to that weight, the number of police agents needed for clearing the fire will be calculated. Police agents' fire assigning function is exactly the same as fire zone assigning used for fire agents. To make sure each cluster's fire gets a police whatsoever, each cluster's police has the duty of making sure the fire zone's area is clear. In case more agents are needed, the agent suitable for that fire zone's path opening will be chosen estimating the move time to that fire zone, the agent's priorities (whether it has a more important duty or not), and the cluster's location.

### 4. Choosing the Best Way

Police agents will choose the best way using the Dijkstra algorithm so that they can do the maximum task in the minimum time. [5] For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex.

### 5. Communication

Generally, no different strategy is used when the communication is lower than normal. Each police agent decides for itself, while considering other's probable decisions.

## 4. Search

An agent should search the map when it's free because most of the decisions are based on location of fires and civilians and the open roads.

Each agent starts the search from its own cluster. The search algorithm is like a circle maze that the agent starts the search from the center of it and eventually comes to the edges. The agent chooses the building that has the least cost of move for it to search.

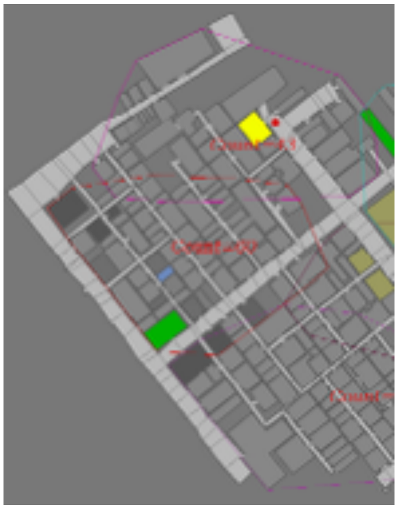

**Fig. 5.** The agent starts its search from the nearest house -which is shown with yellow color- in its own cluster , which is shown with pink color.

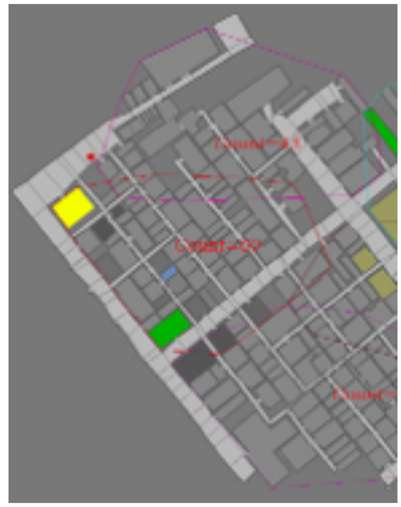

Fig. 6. After finish checking its own cluster, the agent searches the nearest building which is shown with yellow, in another cluster.In this example the other cluster is shown with red

## 5. References


1. Mogha, Priyanka, Nitika Sharma, and Satish Sharma. "CLUSTERING TECHNIQUE."
2. Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." Applied statistics (1979): 100-108.
3. Wright, M. B. "Speeding up the Hungarian algorithm." Computers & Operations Research 17.1 (1990): 95-96.
4. Carpaneto, Giorgio, and Paolo Toth. "Algorithm 548: Solution of the assignment problem [H]." ACM Transactions on Mathematical Software (TOMS) 6.1 (1980): 104-111.
5. Skiena, S. "Dijkstra's Algorithm." Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley (1990): 225-227.
6. "RoboCup Rescue Home". RoboCupRescue.org and RoboCupRescue.com. Retrieved 2009-06-26.
7. "RoboCup Rescue". RoboCup 2009. RoboCup. 2009. Retrieved 2009-06-26.
8. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)
9. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
10. Burkardt, John. "K-Means Clustering." Virginia Tech, Advanced Research Computing, Interdisciplinary Center for Applied Mathematics (2009).
11. Goodrich, Michael T., and Roberto Tamassia. "Algorithm design." (2002).