RoboCup 2016
Rescue Simulation League Team Description
GUC ArtSapience (Egypt)

Sameh Metias, Mohammed Waheed, Mohammed Farghal, Ahmed Osama, Patrick Attia, Salma Amr, Heba Amer, Ebrahim Elgaml, Maggie Moheb, Dina Helal, Ahmed Abouraya, Menna Bakry, Fadwa Sakr and Slim Abdennadher

German University in Cairo, Egypt,
[fadwa.elhussini, menna.nabil, dina.helal, slim.abdennadher]@guc.edu.eg

Abstract

This paper describes the contribution of the GUC_ArtSapience team in the Rescue Agent Simulation League in RoboCup 2016 competition. Each section describes the current approach and the changes that the team applied since last year [6]. The changes this year include enhancing the searching algorithm using Dijkstra as well as classifying the map into prioritized clusters. In addition, introducing a new supervised learning technique for predicting the civilian estimated time of death (ETD). Moreover, optimizing the agents task assignment problem using the Hungarian algorithm. Finally, using the centers for communication protocols.

1 Introduction

Rescue planning and optimization is one of the emerging fields in Artificial Intelligence and Multi-Agent Systems. The RoboCup Rescue Agent Simulation provides an interesting test bench for many algorithms and techniques in this field. The simulation environment provides challenging problems that combine routing, planning, scheduling tasks, coordination and communication.
The Robotics and Multi-Agent Systems (RMAS) research group at the German University in Cairo (GUC) was established in September 2010. The goal of the research group is to study and develop AI algorithms to solve problems in robotics and simulation systems. These fields include computational intelligence, machine learning, multi-agent systems, and classical AI approaches. The current research efforts investigate the following research directions:

- Enhanced cluster-prioritization technique using a many-to-many relation between agents and clusters.
- Supervised learning approach for estimating the time of death of civilians and an enhanced searching algorithm for Ambulance Team.
- Enhanced searching algorithm of targets for Police Forces and Fire Brigades using Dijkstra algorithm.
- Applying the Hungarian algorithm for the agents task assignment problem.

- Classification of Fire Brigades into static and dynamic roles.
- Enhanced clearing approach for Police Forces.
- Enhanced communication protocol with centralized decision making.

The paper is divided as follows, section 2 explains the enhancements in the agents' approach this year. Section 3 provides a summary of the communication protocol. In section 4, the routing approach is explained. Finally, section 5 summarizes the conclusion.

2 Approaches

In this section the different algorithms to tackle the different problems are discussed. In particular, approaches to solve the clustering problem and the assignment problem are presented.

2.1 Clustering

In the pre-processing phase, clustering is used to divide the map into regions and assign each agent to a certain region. K-means++ is used to calculate the initial cluster centroids which are selected from a uniform Gaussian distribution over the buildings/roads in the map. In last year's implementation, One-to-one mapping relation between clusters and agents existed [6]. This year, the relation changed to many-to-many, which means that each agent could be assigned to many clusters, and each cluster could have more than one agent assigned to it. Another enhancement to the clustering was assigning a priority to each cluster according to the average total area of the buildings in the cluster and their importance, that prioritization affects the agents assignment to clusters.

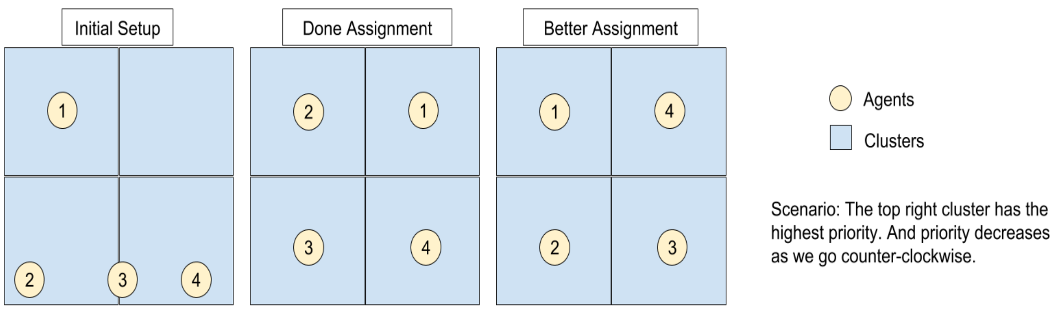

Figure 1: Assigning Agents to Clusters

However, this assignment technique is not an optimal one. As sometimes agents almost switch positions to reach their cluster, as shown in Figure 1. So some different approaches are currently being investigated in order to optimize the assignment phase. The first one is re-assigning agents among clusters upon the death of any of agent. Another approach for optimizing the assignment of agents to clusters, is minimizing the sum of the distances that agents would take to reach their assigned clusters. The advantage of the presented approaches is that other factors are considered beside the cluster's priority.

2.2 Hungarian Algorithm

One of the major tasks in the rescue simulation competition is the task assignment problem. It is a common situation, where multiple agents have several tasks to accomplish. However, assigning which agent to which task remains a problem. Thus, a high quality task assignment algorithm had to be implemented in order to ensure the best task assignment in terms of cost and efficiency.
The Hungarian Algorithm is a well-known task assignment algorithm that minimizes the total cost for assigning all agents to the different available tasks through minimizing the path required in the beginning to reach the target [3]. The generic algorithm can be described in the following steps:

1. As a first step, the cost matrix C is calculated where the rows represent agents, columns represent tasks and each entry C[i,j] in the matrix indicates the cost for agent "i" to accomplish job "j". These entries are calculated according to the assignment problem at hand.

2. The minimum cost value per row is subtracted from each element in that row individually, resulting in having at least one 0 per row in the cost matrix.

3. The minimum cost value per column is subtracted from each element in that column individually, resulting in having at least one 0 per column in the cost matrix.

4. The minimum number of rows and/or columns containing all zeros in the cost matrix are marked for optimal assignment. If that minimum number of rows and columns is equal to the dimension of the square cost matrix, then the assignment is straight forward: each agent will be assigned to the task where a zero cost is to be found. Otherwise, go on with step 5.

5. Additional zeros have to be found. This is done by subtracting the minimum cost value from each unmarked element in the cost matrix, yet adding it to the intersecting rows and columns from step 4, if this intersection is not a zero.

6. Go to step 4.

3 Agents

In the pre-processing phase, K-means++ clustering is used to divide the map into regions and assign each agent to a certain region [6]. After clustering, each agent is assigned to a list of tasks within its cluster.

3.1 Fire Brigades

Up till last year, every Fire Brigade agent followed the same procedure in order to obtain its list of tasks like all other agents and handeled those tasks in the same way like any other agent would handle them. In contrast to that, the main idea feature behind the Fire Brigade agents' code is to classify the agents into different classes: static and dynamic Fire Brigade agents. The main difference between them is described in the following sub-sections.

Static Fire Brigade agents are mainly concerned with the **fire detection** task. Therefore, they get assigned to clusters in which they keep patrolling for fast detection of fires. Since not all Fire Brigades are static, the number of clusters is less than the total number of agents. This makes each cluster bigger than the clusters computed using last year's code. In order to achieve fast patrolling over the whole cluster, a greedy set cover algorithm was implemented. The algorithm greedily yields the smallest set of roads from which all buildings inside a cluster can be seen and also can be extinguished. Thus, static Fire Brigade agents are in need of patrolling over this set only. According to figure 2 below, these critical points are marked in yellow. In order to achieve the minimum total traveling distance between the critical points of the cluster, the patrolling task is mapped to the traveling salesman problem and solved accordingly.

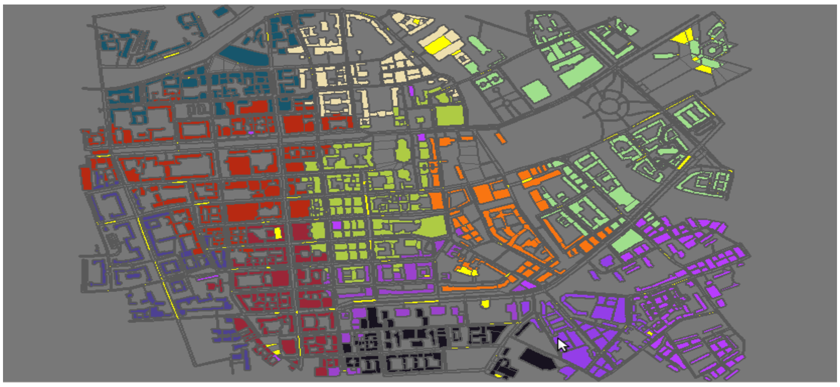

Figure 2: Critical observing points in fire brigade clusters

One application of the Hungarian algorithm is assigning the agents to their clusters. It also solves the problem of deciding whether an agent is static or dynamic. Instead of a random decision on whether an agent is static or dynamic, then optimizing the assignment problem for this subset of agents, the cost matrix is calculated for all agents, yet for reduced number of tasks/clusters. The calculated cost here is the BFS-distance from the initial position of the agent to the center of the cluster.

Dynamic Fire Brigade agents are mainly responsible for **fire extinguishing**. In contrast to static agents, dynamic agents are not assigned and hence are not committed to any cluster. They can move through the whole map in order to reach the fire at any place. Initially, they are spread through the map, also for faster fire detection. However, once a fire is reported they are committed to extinguishing it rather than detecting new fires. Communication is used in order to assign dynamic agents to different fires.

3.2 Ambulance Team

The main target for the Ambulance Team is to save the maximum number of civilians possible. This implies the maximum utilization of the time ahead of each agent, along with using all possible resources available such as the communication channels and the help of other agents throughout a coordination and cooperation plan.

This year the learning algorithm used for learning the estimated time of death of the civilians (ETD) is further explored. Last year Supervised learning techniques were used for this exact problem. However, this year after further investigations, Linear Regression was chosen as the classifier for this certain learning phase [5]. A training data-set was to be obtained for a further learning phase. This set was the result of exhaustive runs of multiple maps, where the behavior of the agents changed to fulfill this purpose. This was done by inducing a trainer mode for the agents through generated maps and scenarios. The final training data-set was obtained by running over 10 different maps each with multiple scenarios generating a data-set of total 41877 instances.

Given the training data-set, a relation is required between the input pairs (HP, Damage, buriedness) and the output (ETD). This relation was obtained first by training the data-set and then using the output learning model for future predictions. The output variable in the model often referred to as the target is the ETD, while there might be none in case the civilian is alive till the end of the simulation. The ETD still has a large number of possibilities and that is the main reason for choosing linear regression. Applying the classifier on the previously obtained training data-set generated the following model [2]:

$$ETD = -0.009 * hp - 0.9197 * damage + 0.2056 * burridness + 328.291 \tag{1}$$

As for the planning of agents' actions and decision making. The model takes as an input the parameters received from the server for each civilian in the agent's line of sight and outputs the ETD of the civilian. After comparison of civilians' ETD, the agent starts prioritizing the tasks accordingly, giving civilians with low ETD a higher priority than the ones with a higher ETD. As for the scenarios where the agent decides to rescue targets that are reported to be buried, the agent starts moving towards the reported civilian with the lowest ETD using the shortest path constructed through Breadth First Search (BFS). In order to combine the two approaches together, exploiting both the learning model and the shortest path for reaching targets, a search algorithm inspired by A^* search was the chosen search paradigm. The expansion of the nodes is dependent on the evaluation function $f(n)$, which is an estimation of the cheapest solution from node n [4]. The function is defined as the following:

$$f(n) = g(n) + h(n) \tag{2}$$

The $g(n)$ was defined as the length of the shortest path from the agent's location to the desirable target. This path was constructed using BFS algorithm, guaranteeing that it would be the shortest available path to the target.
Evaluation of the work took place to test whether the introduced learning model used for task prioritization and planning helped enhancing the performance of the Ambulance Team or not. For achieving this goal, multiple maps were tested using three different strategies. The first strategy was using the learning model to predict the ETD of the targets and prioritize them accordingly. Another strategy was prioritizing civilians according to the shortest distance from the agent. The final approach used the targets HP to plan and prioritize the agent's tasks. At the end of each run, the number of civilians rescued was extracted and used to produce the statistics presented in figure 3. The graph shows that in most cases the presented new approach outperforms the other two strategies in terms of how many civilians were rescued. Moreover, given the average number of civilians in each map, the presented statistics will show that using the first strategy 77% of the civilians are rescued. In comparison to 56% in the case of the second strategy and 64% in the third case.

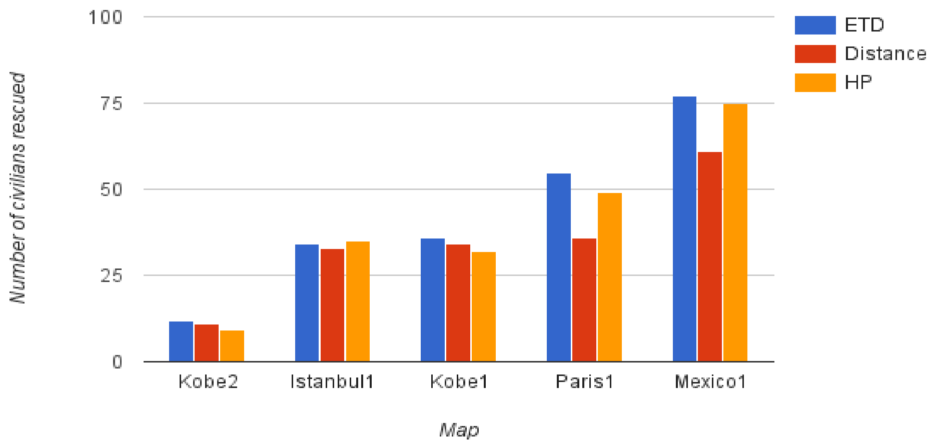

Figure 3: Graph showing number of civilians rescued per map

3.3 Police Forces

There are two important metrics that can be used to optimize the actions of the Police Forces, maximizing the area to be cleared or minimizing the time spent on clearing. The approach used this year optimizes the time the Police Forces spend on clearing and does not consider clearing all blockades, It clears only the necessary parts. The approach works as follows.

- Based on the line of sight of the police agent, it identifies all the seen blockades.
- The agent uses an algorithm that checks if it is possible to pass through the given set of seen blocked.
- The agent adds up all seen blockades to a cumulative set, then run the mentioned algorithm to identify the minimum number of blockades that are blocking a road.
- The agent sends clear signals parallel to the road. Thus, the agent clears the minimum path that makes it possible for agents to pass through the blockade.

The following two figures illustrate the difference in performance between the clearing approaches of last year against this year. Figure 4 shows the old behavior of the police agent in clearing [6] while Figure 5 is an example of the current behavior of the police agent after applying the approach explained above.

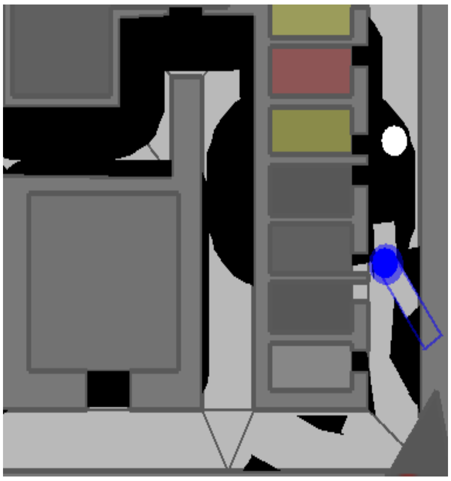

Figure 4: Irregular Clearing

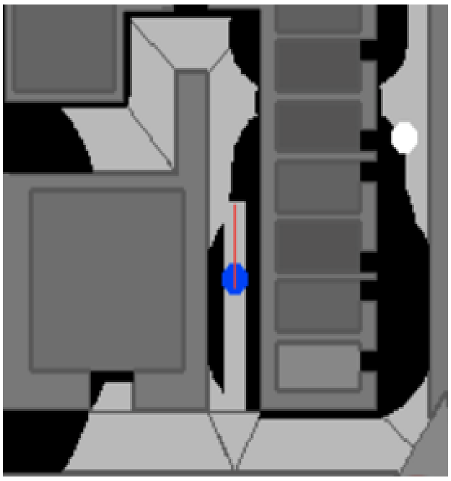

Figure 5: Straight Clearing

4 Routing

4.1 Dynamic Routing

This year Dijkstra algorithm is used instead of Breadth First Search (BFS) for path finding. Although BFS sometimes gives the same result as the Dijkstra algorithm, the Dijkstra algorithm gives better results in most of the cases, sometimes even reaching half of the cost given by the BFS. Table 1 shows some sample data from the logs of the agents showing the cost of the paths given by both the BFS and Dijkstra searches.

Agents	Dijkstra Cost	BFS Cost
Fire Brigades	826140	886554
Fire Brigades	887950	948364
Police Forces	576899	640347
Police Forces	887950	948364
Ambulance Team	412327	412327
Ambulance Team	78211	143793

Table 1: Total Distance performance measures Dijkstra Vs. BFS

4.2 Static Routing (Static Path Finding)

Dynamic routing search algorithm is computed each time step when an agent needs to calculate the path it shall follow. A different approach would be to statically compute paths between each two nodes on the world model graph in the pre-computation phase, which would be used as a look-up for the agents. The agent then looks up the path it is willing to take and check if it is clear, if so it traverses the path directly without any further computation, otherwise it will have to compute a path dynamically as in the usual case [1].

5 Conclusion

In conclusion, this paper presented the updated strategies adopted by the team since last year. The updates attempt to improve the agents performance. The explained enhancements include the following approaches. Cluster-prioritization technique using a many-to-many relation between agents and clusters. The Hungarian algorithm for the assignment problem of all agents to their different available tasks. Classification of Fire Brigades into static and dynamic roles. A Supervised learning approach for estimating the time of death of civilians. Moreover, An enhanced clearing approach for Police Forces, as well as an enhanced Communication Protocol with centralized decision making. Then eventually, an enhanced searching algorithm of targets using Dijkstra accompanied with results.

References

[1] Hayllander Blonski Hector Azp urua Jhielson Pimentel Omar Pino Rafael Colares Douglas Macharet Anderson Tavares, David Salda na and Luiz Chaimowicz. Rescue simulation league team description y-rescue. 2015.

[2] Kevin P Murphy. *Machine learning: a probabilistic perspective.* MIT press, 2012.

[3] SiavashMehrabi Mehdi Taherian Sam Mirzaee ramhormozi PooyaDeldarGohardani, PeymanArdestani and Mohammad Amin Yousefi. Rescue simulation league team description mrl (iran). 2013.

[4] Stuart Russell and Peter Norvig. Intelligent agents. *Artificial intelligence: A modern approach*, pages 46–47, 1995.

[5] Fadwa Sakr and Slim Abdennadher. Harnessing supervised learning techniques for the task planning of ambulance rescue agents. ICAART 2016, February 2016.

[6] Mina Sedra Ahmed Jihad Salman ElDash Maggie Moheb Mohab Ghanim Fadwa Sakr Ahmed Abouraya Dina Helal Sameh Metias, Mahmoud Walid and Slim Abdennadher. Rescue simulation league team description guc artsapience. Robocup, 2015.