RoboCup 2014
Rescue Agent Simulation Competition
GUC ArtSapience Team Description Paper

Dina Helal, Menna Bakry, Kareem Yousri, Ahmed El Safty, Ahmed Badr, Ahmed Jihad, Mina Ezzat, Fadwa Sakr, Ahmed Abouraya and Slim Abdennadher

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

Abstract

This paper describes the contribution of the GUC_ArtSapience team to the Rescue Agent Simulation; in terms of the current research approach implemented in the RoboCup 2014 competition. This year we are extending last years approach which was modeling of the problem as a multi-agent planning problem. Task allocation is handled mainly through the use of clustering to divide the map among the agents. Coordination depends on the use of communication, if available, but also can be done dynamically without the use of communication. We enhance our Ambulance agents' performance this year by calculating the expected civilians' death time using supervised learning. We modify our Police agents' prioritization of tasks by identifying critical roads. Moreover, we modify our Fire Brigades technique by grouping fires and prioritizing the Fire Brigades tasks. Finally, we improve our resources utilization among the agents using our modified communication model.

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 optimization (routing, planning, scheduling) and multi-agent systems (coordination, communication, noisy or missing communication)[3].

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:

- Dynamic clustering for task allocation and coordination between Ambulance agents.
- Predicting civilians death time using supervised learning.
- Determining critical roads to speed up the process of clearing the map from all blockades.

- Enhancing our communication model for better resource utilization.

The GUC_ArtSapience team became the champions in the Rescue Agent Simulation in 2013 by ranking first place in our third participation in the competition. Our first participation (as RMAS ArtSapience) was in 2011 where we ranked 3rd place in the final round. This paper describes the current teams achievements in tackling the RAS problem. Section 2 describes our clustering approach. Section 3 describes the enhancements in our agents' approach this year. In section 4, we explain our modifications in last year's communication model. Finally section 5 shows our updated results compared to last year.

2 Clustering

In the rescue simulation environment, there are a lot of information that are obscure, for example, in disaster scenarios, the initial locations of fires, buried and injured civilians, and blockades are unknown. Moreover, tasks that are assigned to an agent don't specify where exactly that agent should carry out these tasks. So agents have no other choice but to traverse the whole map and search for them. And since it is impractical for a single agent to cover the whole city map (all buildings and/or roads), a clustering approach was developed to divide the map into smaller parts. We continue to use our clustering technique from 2012 which is to divide the map into regions as shown in figure 1. Each agent gets assigned to a certain region where it traverses all of the buildings and roads in that region searching for events that require rescue actions. We use K-means ++ clustering algorithm [4] to calculate the initial centroids which are selected from a uniform Gaussian distribution over the buildings/roads in the map. Using K-means++ algorithm, we are able to compute clusters equal to the number of agents during the preprocessing time and to assign initial tasks to each agent within its cluster.

3 Agents

Each agent has a queue of tasks that are prioritized according to different factors. In this section we discuss the changes in the agents' tasks this year to improve their performance and achieve better scores.

3.1 Ambulance Team

The Ambulance team's main task is to rescue buried civilians and carry them to refuges. Last year each Ambulance agent was assigned to a cluster in the preprocessing phase. However, we noticed that in many maps the civilians are cluttered in certain areas in the map where other areas have very few or no civilians at all. Based on that, we modified our clustering technique for the Ambulance agent to be done dynamically based on the reports from all other agents. In the scenarios with communication, the Ambulance

Figure 1: K-means ++ clustering

agents listen to reports about buried civilians and their locations from all agents. During the first 20 time steps the Ambulance agents form statistics from these reports to detect the areas where there are many civilians. Based on these statistics, the clustering is done such that the number of clusters doesn't have to be equal to the number of Ambulance agents in the map as we did last year. Each Ambulance agent searches for buried civilians within its cluster then moves on to other clusters. Each Ambulance agent keeps track of the buried civilians it came across but couldn't rescue because of blockades or fires. Accordingly, Ambulance agents divide their tasks into three groups. The highest priority is assigned to the currently seen buried civilians followed by the ones it previously passed by but could not reach followed by the reported targets seen by other agents.

3.1.1 Estimating Death Time

This year we propose using machine learning to estimate the civilians and agents death time. In the previous years, we faced a problem with that estimation due to the fact that the damage and health points values were rounded up and down by huge factors resulting in inaccurate readings from the simulator. This inaccuracy made it hard to find a realistic formula to estimate the expected death time as we were not getting their real heath status from the simulator. This year we will be using a support vector machine (SVM) classifier to tackle this problem. Classification which is also known as pattern recognition, is an important part of machine learning. In classifiers, machines learn to automatically recognize complex patterns and make intelligent decisions. This learning process is based on training the machine with a sample dataset. Based on this training the machine learns how to behave. An SVM classifier can be trained with a set of examples, where each sample in the data set is labeled with the class it belongs to. In our case we have two classes which are dead and alive. The features used as input to the classifier are the civilians' burriedness, health points, damage and number of time steps. The classifier then determines if after a certain amount of times steps the civilians will be alive or dead based on the burriedness, health points and damage parameters. A lot of training data is collected from the simulator by running different maps multiple times and recording the civilians health status throughout the simulation. Then an internal SVM training algorithm builds a model which represents the training examples as points in space, where the examples belonging to different classes are divided by a clear gap that is as wide as possible as shown in figure 2. Based on this model, new examples are then mapped into that same space and predicted to belong to a class based on which side of the gap they fall on.

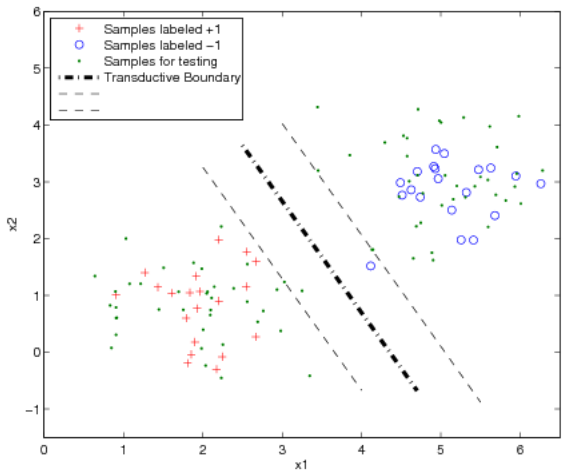

Figure 2: SVM Model with two features

Using the SVM model, each civilian within an agent's range can be labeled as dead or alive at a certain time. Based on that prediction, civilians that are close to dying and won't have enough time to make it to the refuge are ignored as they probably die after being loaded before reaching the refuge. Civilians that are close to dying but have enough time to make it to the refuge are assigned a higher priority than those who still have more time. Moreover, knowing the estimate death time of a civilian will help us determine of the number of Ambulance agents needed to rescue this civilian before it dies.

3.2 Police Agents

The Police agents are distributed across the map, whereby, each agent is assigned a certain cluster consisting of a set of roads and buildings. Each agent is mainly concerned with clearing blockades in its cluster so as to facilitate the movement of other agents, Fire Brigades and Ambulance agents, to their respective targets. Such a process requires prioritizing the tasks received by the Police agent, based on the severity of each task. The highest priority is to clearing blockades on paths leading to buildings which have been reported on fire in order to facilitate the movement of Fire Brigades to contain the fire before spreading. Furthermore, in the case of absence of any high priority tasks in its cluster, the agent would undertake any fire report in the two clusters in its immediate proximity. This is due to urgency of fire, and the need for rapid extinguishing. Second is clearing blockades on paths leading to agents jammed in blockades, and hence, incapable of fulfilling their tasks. Third is clearing blockades on paths leading to refuges, satisfying the needs of Fire Brigades for water, and that of civilians for shelter. Fourth is clearing blockades on paths leading to civilians which were reported buried. In case of the absence of any tasks, the Police agent patrols the whole cluster in search for any blockades that may cause any future problems.

3.2.1 Defining Critical Blockades

It is vital for the Police agents to be able to clear blockades in the least possible time, in order to enable the fulfillment of the tasks of the other agents, hence, the use of critical blockades. Critical blockades are those blocking roads or buildings entrances, thereby, restricting the access of other agents to such entities. The approach adopted considers the degree to which a road is blocked. If the blockades are on the same side of the road, and the distance between each of the blockades and the side of the road is greater than agent's diameter, the road need not be cleared as the agent can still move in the road between the blockades. Otherwise, if they exist on opposing sides of the road, we need to ensure the existence of a path between the blockades such that the path can accommodate the agent. This is done by comparing the edges of the blockades and the edges of the roads and where they intersect to determine if a road is completely blocked or partially blocked.

3.3 Fire Brigades

As mentioned earlier K-means++ algorithm is used to partition the map into clusters of buildings in the preprocessing phase. Each cluster is then assigned to the nearest Fire Brigades agent. The main objective of the Fire Brigades is to extinguish and contain any building on fire. Each agent iterates over all buildings in the assigned cluster and report any building ablaze using radio communications, in order to acknowledge other agents for the need of extra assistance.

3.3.1 Grouping Fires

Due to the fieriness (a parameter to measure the degree of fire in a building) rate, tank size and other constraints, a single agent can not handle extinguishing a building in its cluster all by itself. Rather, Fire Brigades agents from other clusters contribute to fire notifications once found in order to help in the extinguishing tasks. The main limitation in this technique is that agents responding to the fire report leave unprotected parts in the map where another fire can start with no one available to extinguish it. To overcome this dilemma, which resembles in sending a portion of our agents to buildings ablaze, close fires are considered to be belonging to one group with a center C and radius R. Agents are distributed among different groups of fire such that an agent is assigned to the closest group. Thus, the agents can handle more than one fire simultaneously.

3.3.2 Prioritizing Fires

The main idea when prioritizing the extinguishing list is containing the inferno, rather than putting off the center of the fire. Due to few reasons, it is much faster to extinguish warm buildings - not yet on fire - than buildings ablaze. Warm buildings are reported by agents and considered to be on fire until their temperature is below a certain threshold. Extinguishing a building if it is warm is easier than extinguishing them if they are on fire. Extinguishing warm buildings helps in containing the fire because warm buildings are usually at the outer part of a fire group. Based on that, the highest priority is given to the warm buildings, then to buildings with lowest degree of fieriness then to buildings of the highest degree of fieriness. This way, the outer part of the fire group is targeted first which increases the possibility of containing the fire.

4 Coordination and Communication

Agents and centers subscribe to the communication channels according to available number of radio channels, maximum number of channels an agent can subscribe to, and the maximum number of channels a center can subscribe to. Agents exchange messages between each other using the communication channels to report fires, blockades and buried civilians. We use the same message compression and noise handling techniques we used in 2013[1].
This year we use communication not only to report tasks to the agents but to allow the agents to notify each other with their tasks. For example, when a group of Ambulance agents receive/listen to a report about a certain civilian, each agent who is going to go to rescue that civilian will send a message notifying the rest of the agents with the civilian ID that it will attempt to rescue. Using the Ambulance center, only the closest agents to the civilian will be assigned to save it and will acquire a lock on that task so that no other Ambulance agent tries to execute the same task unless the civilian requires more than 1 agent to rescue him. In this case the closest 2 agents will be assigned to that task then it will be locked. The same technique applies to Police agents going to blocked roads and Fire Brigades going to buildings. This ensures that each task is only executed by the closest agents to it and the appropriate number of agents which saves a lot of wasted time where more agents than needed were attempting to do the same task.

5 Results and Conclusion

To evaluate our modifications this year, we used last year's finals maps as a test bench. Table 1 shows a comparison between our current scores in some of 2013 maps and our scores in the competition.

Table 1: Current scores compared to 2013

Map	Kobe4	Eindhoven5	Mexico3	Berlin3
2013 Scores	169.57	169.22	145.42	79.01
Current Scores	173.24	172.13	150.84	86.39

Changing the clustering of the Ambulance agents by making it dynamic depending on the other agents reports lead to improvements in rescuing civilians. The Ambulance agents are now distributed in a better way heading towards the regions with more civilians. In addition, determining the critical blockades for the Police agents lead to clearing the map faster from all the major blockades at the beginning of the simulation so that all agents can move freely and execute their tasks. Finally, using the communication channels to acquire locks on tasks that are being executed by agents improved their coordination together and saved a lot of wasted time where many agents were heading to execute the same tasks leaving other tasks completely unhandled. Further tests are being carried out now to evaluate using machine learning to predict the civilians death time.

References

[1] Guc artsapience team description paper 2012.

[2] Guc artsapience team description paper 2013.

[3] Robocup rescue website. http://sourceforge.net/apps/mediawiki/roborescue/.

[4] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. *Proceedings of the eighteenth annual ACMSIAM symposium on Discrete algorithms*, 8(2006-13):1027–1035, 2007.