Harnessing Supervised Learning Techniques for the Task Planning of Ambulance Rescue Agents

Fadwa Sakr and Slim Abdennadher
Faculty of Media Engineering and Technology, German University in Cairo, Cairo, Egypt

Keywords: Multi-agent Planning, Learning, Supervised Learning Algorithms, Classification, RoboCup, Recsue, RoboCup Rescue Simulation, Task Planning.

Abstract: One of the challenging problems in Artificial Intelligence and Multi-Agent systems is the RoboCup Rescue project that was established in 2001. The Rescue Simulation provides a broad test bench for many algorithms and approaches in the field of AI. The Simulation presents three types of agents: police agents, firebrigade agents and ambulance agents. Each of them has a crucial role in the rescuing problem. The work presented in this paper focuses on the task planning of the ambulance team whose main role is rescuing the maximum number of civilians. It is obvious that this target is a complicated one due to the number of problems that the agent is faced with. One of the problems is estimating the time each civilian takes to die; the Estimated Time of Death (ETD). Realistic estimations of the ETD will lead to a better performance of the ambulance agents by planning their tasks accordingly. Supervised learning is our approach to learn and predict the ETD civilians leading to an optimized planning of the agents tasks.

1 INTRODUCTION

The RoboCup Rescue project was initially triggered after the Hanshi-Awaji hit Kobe City on the 17th of January in 1995 causing enormous number of casualties and damage. The aim behind the project was to use an environment for research and development. The main target would involve multi-agent planning and team coordination, physical robotic agents for search and rescue, information infrastructures, personal digital assistants, a standard simulator and decision support systems, evaluation benchmarks for rescue strategies and robotic systems that are all to be integrated into a comprehensive systems in future (Kitano and Tadokoro, 2001). Teams from all over the world participate each year to compete through proposing new solutions and tactics for overcoming these disastrous scenarios with minimal losses.

The RoboCup Rescue simulation is a league driven from the Rescue project. The simulation models an earthquake in an urban centre presented in the form of a map. The simulated earthquake causes building to collapse, roads to be blocked, fires ignitions, and civilians to be trapped and buried inside collapsed buildings. Typically, specialized rescuing forces would be needed for damage control. In the simulation world, there are three teams that are responsible for all rescuing purposes; the ambulance team, fire-brigade and police forces. The role of the ambulance team is to rescue and save buried civilians and safely get them to refuges. As for the fire-brigades, their task is to extinguish building that catch fires. Finally, the police forces' main concern is to clear all roads from rubble and other debris. Thus, clearing the way for the other agents to move freely. The research question and motive for all participating teams in the league is to find a utilized plan for each team to reach the maximum possible damage control. Moreover, the plan has to be a generic one, that would efficiently deal with all possible scenarios. Furthermore, we need to have a dynamic plan that would fit all possible scenarios enabling the ambulance team to save the maximum number of civilians, the fire-brigades to extinguish the maximum number of fires, and the police forces to clear the maximum number of roads. Such aim can only be achieved through agent coordination and team work. A wide bench of research and application of different techniques and strategies would be needed to solve a multi-agent planning optimization problem such as the rescue problem.As described in (Hussein et al., 2012), the rescue problem can be divided into three main tasks: extinguishing buildings on fire, saving civilians, and clearing blocked roads. In the scope of

Sakr, F. and Abdennadher, S.
Harnessing Supervised Learning Techniques for the Task Planning of Ambulance Rescue Agents.
DOI: 10.5220/0005692001570164
In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016) - Volume 1*, pages 157-164
ISBN: 978-989-758-172-4

this paper, we are mainly focused on the tasks of the ambulance team, which consists of saving the maximum number of civilians.

The ambulance team is faced with a large number of obstacles and challenges in order to achieve the above mentioned goal, thus, increasing the need for optimized planning and coordination. Predictions of the ETD of civilians is one of the steps for achieving this goal. Harnessing supervised learning techniques for task planning has been the new approach for tackling planning problems such as the problem of the ambulance team. Consequently, many of the teams participating in the competition have been working in similar scenarios and techniques, along with multiple participation for the team motivated us for exploring such options.

In (Guan et al., 2010) ZJU team used particle filters to tackle the problem of implicit parameters provided by the simulator, thus predicting the ETD of the civilians. They first generate a set of particles, where each particle in a triple of (Hp, Damage, Rate). Hp and Damage are obtained by observations of the civilians behavior, while Rate is a value randing from 0 to 1 indicating the growth rate of damage. After each time step particles are updated using a given damage model. Moreover, they introduced a quantization range used for classifying new observations and deciding whether or not the values of old particles remained legal or not.

In 2013, team Poseidon modeled the ambulance problem using the Knapsack algorithm (Afzal et al., 2013). The Knapsack problem is defined using a set of items each with a given value and cost. The aim is to choose the item of the minimum cost and maximum value, while not exceeding some given cost, capacity and ensuring that the total value is as large as possible. With regards to the ambulance problem maximizing the value means maximizing the number of rescued civilians. Moreover, each civilian has a value and a cost that is determined based on multiple values, one of which is the estimated death time of the civilian. For the ETD they used the learning model presented by ZJU team (Guan et al., 2010).

In previous years, the S.O.S team depended on particle filters for estimating the death time of the civilians. However, after evaluating the approach during the competition, some mistakes were discovered and the team switched to decision trees for predicting the ETD (Hesam et al., 2015). The team used a state space that consisted of 15 states. Moreover, the team decided to not only depend on the ETD for task allocation, but also include the health points of the civilians to do so.

Even though learning has been used previously for tackling multi-agent planning problems, to our best of knowledge supervised learning has never been used for this particular problem so far. The motivation behind the choice is that given the simulation environment, it is convenient to have a training phase based on observations of the environment to have an accurate outcome. The outcome would be later used for learning and future predictions.

The paper is organized as follows. In Section 2, the supervised learning approach is discussed thoroughly. An evaluation of the work is presented in 3. Finally, in Section 4 we conclude and give some directions for future work.

2 OUR APPROACH

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. First to tackle the problem of time we needed to make sure that no time will be wasted on targets that will die either during or after rescuing. That was a problem in previous implementation in which civilians were prioritized according to the shortest distance from each agent. Our approach introduces a new solution for the problem through learning the ETD of each civilian; using which we were able to utilize the planning of each agent during rescuing. This was achieved using a supervised learning algorithm.

2.1 Learning

In each time step of the simulation, the state of any given civilian changes according to a set of parameters determined and rapidly changed by the simulator. This set consists of the following parameters:

- HP : Health points of a civilian stating how healthy the civilian is. It starts with an initial value, when it reaches zero it means the civilian is dead.
- Damage: The points that will be deducted from the civilian HP each time step.
- Buriedness: The level of buriedness of a civilian within a building.

Typically having these parameters through arithmetic calculations, we could know when exactly will the civilian die i.e. the ETD. Unfortunately, the damage value provided by the simulator is not accurate but rounded to the nearest 10 and the HP value is rounded to the nearest 1000. Moreover, the growth rate of damage also differs a lot from one civilian to the other even if their original damage values were the same. In order to overcome these implicit values and tackle our problem a learning approach had to be introduced. Supervised learning is used when some correct input-output pairs are known. That was the case here since the simulator environment already provides some parameters indicating whether or not a civilian is alive, i.e.the HP through each time step along with other multiple parameters. In other words, the feedback available from the server or the simulator was our motivation towards a supervised learning approach.

2.1.1 Training Dataset

A training dataset 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. Instead of rescuing the civilians the agents were to observe and log their state during each time step. Some further changes were done to the simulator for the training phase. All obstacles that normally would face the agents were disabled. For example, maps were run with no blockades, both the fire simulator and the ignition simulator were disabled as well, which led to fire-less scenarios for all the maps we ran. This training dataset represented the history of each civilian ever since the simulation started till the civilian is dead. That would be the base of the learning model that agents use to learn the ETD of a civilian when planning their actions.

The training dataset could also be modeled as a decision tree in which given the values of some attributes a certain goal predicate is evaluated to either true or false. The goal predicate defined in our tree model is the health state of the civilian. The predicate outcomes true if the civilian is dead and false otherwise. A is a set of attributes $A_1, A_2, \ldots, A_m$ of the decision tree. A set of arity 4 was used in our model. The four parameters of the set are the following:

- Unique ID of the civilian
- HP of the civilian
- Damage rate
- Buriedness level

Each pair of the dataset is called an example. An example is a pair $((v_1, v_2, v_3, v_4), f(v_1, v_2, v_3, v_4))$ where v_i is the value of A_i. A positive example is an example where $f(v_1, v_2, v_3, v_4)$ is true. Otherwise it is a negative example. In our decision tree a positive example means the civilian is dead. In that case, an extra attribute is added to this example and that is the time when this civilian died. This additional attribute will be used later on for labelling the training dataset. Certainly this decision tree model of the dataset is not sufficient enough for a supervised learning classification. Since not all examples are of the same input and output format needed for a supervised learning classifier, only positive ones have the ETD of the civilian added. This enforced some further preprocessing of the training dataset.

2.1.2 Data Preprocessing and Labelling

For a supervised learning algorithm to be applied on a training dataset, each example of that set has to have an output attribute, which is the output value of the classification. In the proposed model this attribute is the death time of the civilian. This desired formatting of the dataset is not yet present, due to the fact that the only labeled or positive pairs are the ones that belong to a dead civilian leaving all negative examples unlabeled.

The preporcessing of the data focused mainly on matching every positive labelled example with all the negative unlabelled examples that belong together, in the sense of belonging to the same civilian. The unique identifier that each civilian has and which was previously included in the set of attributes of the decision tree was used for the matching. For each negative example, if it has a matching positive example, then it will be labelled with the same value of ETD. If no match was found, it means that the civilian was alive till the end of the simulation. In this case, the pair will be given the value of the time the simulation ends. After the matching of all pairs the identifiers of the civilian shall be removed from the training set since there will be no use for it anymore.

So far what we have is a training dataset that contains pairs of values for each attribute in the set and a value for the time where this civilian will die even if it is not dead yet. This dataset will be used as an input for a learning classifier, that will be applied later to classify the time of death of each civilian according to the given training set.

2.1.3 Learning Classifiers

Given the training dataset, we would like to learn the relation between the input pairs (HP, Damage, buriedness) and the output (ETD). This relation was obtained first by training the dataset and then using the output learning model for future predictions. Thus, a classifier is needed to achieve both goals. The output being a numeric one discarded some classifiers from the large pool of classifiers now present. After further filtration enforced by restrictions of our problem, we chose linear regression as our classifier here. The Weka tool [1] was used for all learning purposes.

In our model the output variable 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 was the main reason for choosing linear regression. In linear regression the output variable or the target can have a value from an infinite number of possible values (Lane, 2015). A learning model can be used for predictions of the ETD of any given input. This prediction is done using the following equation:

$$y(x_{n+1}) = \theta_1 + x_{n+1} * \theta_2 \tag{1}$$

Where y is a function used for evaluating the line at point x_{n+1} and finding a label for it, θ_1 is the slope of the linear line and θ_2 is the intercept of y. Now for the prediction to work we need to learn both θ_1 and θ_2 (Murphy, 2012). Applying the classifier on the previously obtained training dataset we were able to do so resulting in the following model:

$$\begin{aligned} ETD = &-0.009 * hp - 0.9197 * damage \\ &+ 0.2056 * burridness + 328.291 \end{aligned} \tag{2}$$

When using machine learning classifiers some testing need to be done to ensure smallest range of error possible. There are various methods to do so. However, here we chose cross validation for testing. The idea of cross validation is to estimate how well the current dataset can predict an output value for any given input instance. This is done by using a fraction of the dataset and testing the prediction of it using the rest of the dataset. The same process can be repeated using different subsets of the data. A 10 fold cross validation was applied to the training dataset leading to the following results:

```
-Correlation coefficient     0.5189
-Mean absolute error         32.9997
-Root mean squared error     39.6019
-Relative absolute error     85.2929%
-Root relative squared error 85.4812%
```

After the learning of the training dataset, we were able to do some predictions of the ETD of any given instance provided by the agent. Moreover, we were able to learn more about the direct relation between each of the model attributes and the ETD of the civilians. This relation was a bit unclear before due to the implicit attributes provided by the simulator. Presented in Figure 1 a plot for the damage rate as y-axis against health points as x-axis, in which civilians having the same health points are classified into different classes mainly depending on the damage. Similarly, civilians of high damage tend to belong to classes of low ETD (blue instances), while the ones with low damage belongs to classes of high ETD (orange instances). Moreover, there are instances of the same HP and damage, however, belong to different classes. Thus, showing that depending on only the health points or the damage to determine the priority for saving civilians would be highly inaccurate and unreliable. Which was the strategy followed in previous approach before using the new learning model(Abouraya et al., 2014). After finalizing the learning model, the next phase was to integrate this model with the task planning of the agents.

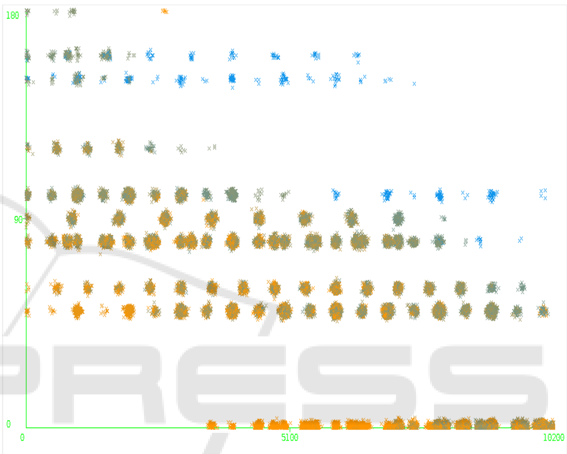

Figure 1: Damage vs health points.

2.2 Planning

Rescue agents need to plan their decisions to reach their restricted goal that is saving the maximum number of civilians possible. The main source of information for the agent is the world model maintained from the simulator each time step. The world model is based on the change set of the agent; what the agent sees and hears formulate the world model of the agent. In addition to a communication model based on a decentralized plan, an approach used for multi-agent planning (Abouraya et al., 2014), where all agents communicate together through communication channels. The communication model helps the agent receive information about parts of the map outside the agent's world model. A third source of information now added to the agent's decision making process is the learning model previously obtained. As shown in Figure 2 the three models are the input to the planner according to which the agent gain an extra set of information namely the ETD about the target. Subsequently, a decision making process to evaluate all possible actions and chose one to perform had to follow.

[1] http://www.cs.waikato.ac.nz/ml/weka/

Analogously, ambulance agents plan their actions and start acting accordingly.

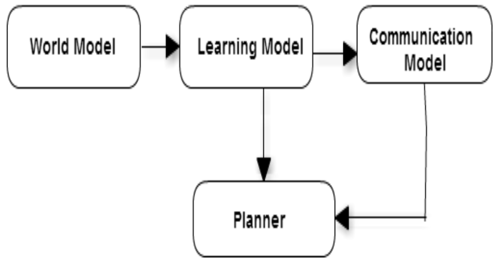

Figure 2: Model of the implemented approach.

2.2.1 Decision Making

Ambulance agents have a set of actions to perform: load/unload a target, rescue a target or move towards a target. In the scope of this section we are mainly concerned with the last two actions. Based on the three models that provide information to the agent, the agent decides upon which action to perform at each time step. Previously, the world model was the most important one of them all. Accordingly, the agent based all decisions regarding which civilian to save first, either on the shortest distance or the HP of the civilians. In addition, agents only decided to move towards the target only if it is still in line of sight. Meaning that if a target is reachable but not in the agent's line of sight, this target would have less priority than the one who is currently seen by the agent regardless if which is in more danger. As discussed in Section 2.1.3 that was not the most accurate or reliable decision. Then the communication model is next in the order of priority, through which agents receive information about buried civilians outside the world model. If the agent has no reachable targets within the world model, then it starts choosing targets from the reported ones. Similar to the seen targets of the world model, civilians were also prioritized according to distance and HP as well.

The introduced learning model now acts as an intermediate level between what the agent perceives from the world model and deciding which action to do. 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. Subsequently, if the agent is faced with two targets, one with a high priority but not in line of sight and the other one with a lower priority and in line of sight, then the agent will decide to move towards the former target instead of the latter. The task prioritizing process does not intervene with the actions the agent decides to do, as it takes place at the beginning of each time step, when the thinking process of the agent is just starting. By the time the agent is ready to make a decision all tasks should have the appropriate priority. Whenever the agent decides to rescue a civilian, the estimated time of rescue would be calculated and added to the current time step, if it exceeds the ETD of the civilian then the agent would have to call out for other agents to help.

Similar to the previous planner, in cases where the agent has no decisions to make regarding all seen targets -if none of the targets are reachable- the agent moves on to targets that are reported as buried by other agents. However, some adjustments had to be done to the communication model in order to be synchronized with the new planner. One of which is that all agents in the map either ambulance, police or fire-brigades were to use the new learning model before reporting any buried civilians. With the use of the new model, agents are able to predict the ETD of the seen targets and add them to the reported message delivered to other agents. This requires the learning model to be generic enough to be used by any type of agent. Upon receiving a reported message, the ambulance team agents proceed with their task prioritizing. However, in this case agents have only one decision to make which is moving towards the target. Once a target is reported to an agent, it means that this target is outside the agent's line of sight. Thus, the decision making process here is different than the one of the targets extracted from the world model.

2.2.2 Time Versus Distance

The message for reporting buried civilians previously consisted of the target's location, since the target is not in the agent who is receiving the message line of sight. Currently after the new approach, the ETD of the target is added to the message along with the location. For an agent to move from one point in the map to another a path has to be provided for the agent to follow. This path is a set of nodes, where each node is an entity on the map. An entity could either be a road or a building. For a more formal presentation, the nodes of the map were structured in the form of a graph. Some factors related to the rescue problem had to be taken into consideration, one of which is that blocked roads are constantly removed from the graph. Similarly, cleared roads are restored once they are reported to be clear. The search algorithm used in previous work based on (Abouraya et al., 2014), for searching the graph and constructing a path is Breadth First Search (BFS). The aim of the algorithm is to find the shortest path to the target node. If the agent had no way to reach any of the seen targets, the agent would search the graph for the closest reported target, followed by a movement towards that target.

Initially, we followed the same approach for planning the task of rescuing reported targets the same way as we did with seen targets. For any given agent, all reported civilians are sorted and prioritized according to their ETD. When 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 BFS. However, the time an agent takes to reach a certain location on the map; namely the Estimated Time Of Arrival(ETA) rapidly changes according to different factors, e.g. the number of nodes in the path, along with the speed of the agent, percentage of blockades in each road and some other factors. This means that the ETA of an agent might exceed the ETD of the civilian with the highest priority, resulting in the civilian's death. Additionally, wasting time on a task that was de-prioritized and overestimation of the cost to reach a certain target, which happened to be the case during evaluation.

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. A^* search is the best-first search algorithm. It is also known to be the only optimal algorithm for expanding the minimum number of nodes in any search space, besides being complete (Russell and Norvig, 1995). The expansion of the nodes is dependant on the evaluation function $f(n)$, which is an estimation of the cheapest solution from node n. The function is defined as the following:

$$f(n) = g(n) + h(n) \tag{3}$$

where

- $g(n)$ is the path cost from node n to the goal.
- $h(n)$ is the heuristic function.

For the algorithm to be an A^* search $h(n)$ has to be an admissible heuristic $h^*(n)$, which was not the case in the chosen heuristic here. So an approximation of the optimal heuristic function $h(n)$ will be used here to estimate the cost of the optimal function $h^*(n)$ (Bonet and Geffner, 2001). In our example, 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. The heuristic function $h(n)$ is the ETD of the target. The combination of these two functions led to the following prioritization of the agent's tasks: targets that are closest to the agent with a low ETD will have higher priority than further targets with a higher ETD. In other words, if an agent is faced with two targets t_1 and t_2, the length of the path to both targets is l_1 and l_2 and the ETD is e_1 and e_2 respectively. First, in the case of both targets being located at the same distance from the agent, then the evaluation function will choose the target with a lower ETD. Additionally, the case of one target with a closer location to the agent than the other one, assuming that $l_1 < l_2$, the agent will be faced with one of two decisions:

1. if $e_1 \leq e_2$ then t_1 would be of higher priority than t_2.
2. if $e_1 > e_2$, t_2 would be chosen, if and only if, $e_1 - e_2 > l_2 - l_1$. Meaning that if the agent decided to move towards t_1 by the time it reaches t_2 it will most likely die before then. However, this will not be the case using the evaluation function presented above. Otherwise, t_1 will be of higher priority as well.

3 EVALUATION

The goal of our solution for the rescue problem is to maximize the number of rescued civilians by utilizing the task planning of the agents. Thus, the number of rescued civilians was chosen to be the evaluation measurement of our approach. Another measurement of the overall performance of the agents would be the final score, which is calculated by the simulator kernel for each running scenario and map. However, we couldn't highly depend on this score for testing our approach, since the rescue simulation depends on a number of components and factors as mentioned in section 1. Meanwhile, our solution only targeted the ambulance team. That's why to ensure an accurate evaluation of our approach, the same environment mentioned in section 2.1.1 for obtaining the training dataset was used for testing as well. The aim of the evaluation is to test whether the introduced learning model used for task prioritization and planning helped enhance the performance of the ambulance team or not. Moreover, we also wanted a confirmation that it is better than previous work such as (Abouraya et al., 2014).

For achieving this goal, we ran multiple maps that were released and used in the RoboCup 2015 world finals [2]. Each map was 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

[2]http://goo.gl/IXI31O

plan and prioritize the agent's tasks. Even though, the three strategies followed different methods for task prioritization, they all used the same new planner introduced in this paper for completing their 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. Certainly, different scenarios might lead to different outcomes, as the simulation environment is highly non-deterministic. Moreover, the initial distribution of both agents and civilians in the map also contributes to how the rest of the simulation proceeds. As was the case in some of the maps used for testing such as Istanbul1.However, given the presented statistics, the performance of the new approach will out perform any of the other two approaches if used alone by minimum 10%.

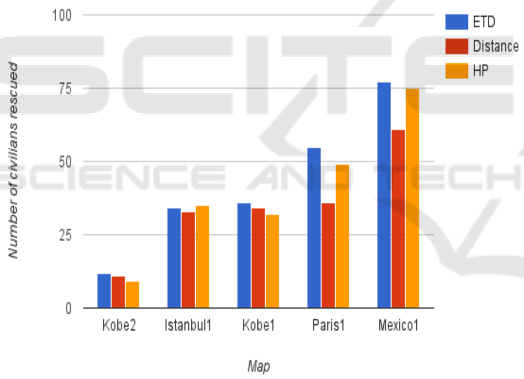

Figure 3: Graph showing number of civilians rescued per map.

Another evaluation measure is to use the final score of the map. However, not all maps are fully dependant on the ambulance team, so we had to pick a map that would only test the performance of the ambulance agents. In such maps the score will be directly affected by the number of civilians rescued, since no other factors have an equivalent high effect. The Paris1 map that was presented in RoboCup 2014 world final competition[3] was chosen for this task. Formerly, during the competition, the new approach was proposed but was not yet finalized or integrated with the work. Task planning was still mainly dependant on calculating the distance between the agent and the targets. After the new approach was finalized, the new implementation was tested against the old one using the mentioned map for score comparison.

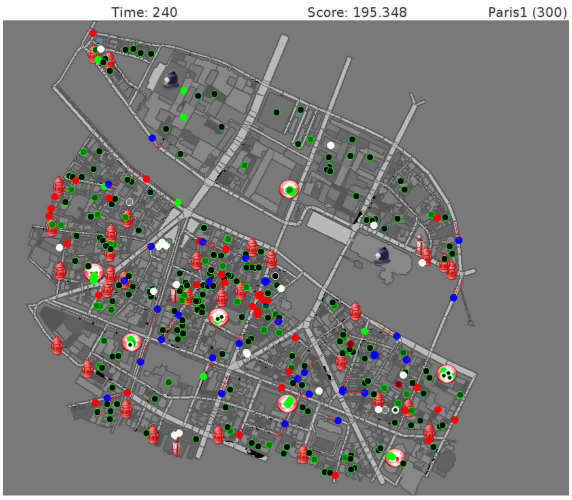

Figure 4: Paris1 using 2014 approach.

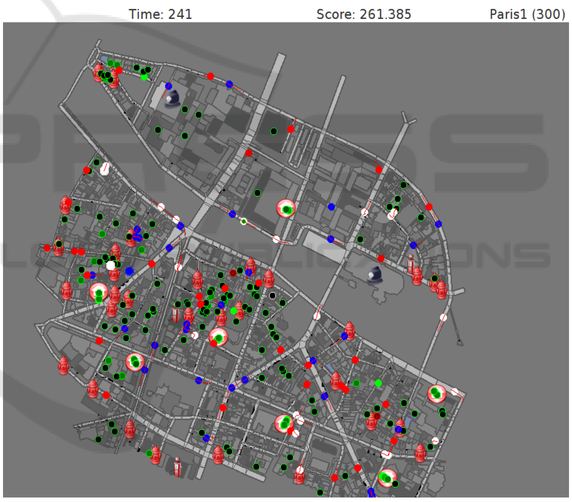

Figure 5: Paris1 using 2015 approach.

Figures 4 and 5 are two screen shots after running Paris1 map using both the old approach and the new one respectively. The difference in scores between the two maps is almost 70 points which is immense. Moreover, the figures show a big gap between the two approaches in terms of rescued civilians. Similar to the previous evaluation phase, this comparison lead to the conclusion that when faced with a scenario fully dependant on the ambulance team, the new approach clearly outperforms the old one. Furthermore, in Figure 4 the resulting map shows that some of the previously rescued civilians died in the refuge. As civilians then were chosen according to their distance from the rescuing agents, so the possibility of a civilian to die

[3] http://roborescue.sourceforge.net/2014/results/

while or even after rescuing is high. Whereas, using the new approach in Figure 5 this is not the case. Since civilians were chosen and prioritized according to their ETD, decreasing the possibility of their death after they have been rescued.

4 CONCLUSION

Including a new learning model in the ambulance team agents thinking process helped utilize the agents time while carrying out with their rescuing duties. This model was the outcome of a training data set that was trained using linear regression algorithm. Subsequently, the model was used for serving the planner by allowing agents to predict the ETD of the civilians. The planner then uses the ETD for task prioritizing and planning. Moreover, the ETD was also used for optimizing the search algorithm that constructs paths for the agents to move from one location on the map to another. This was done by replacing the old traditional breadth first search by a heuristic search, which includes the ETD as a heuristic for the evaluation function of expanding nodes. According to the exhaustive evaluation performed as mentioned in section 3, both the learning model and the new planning helped increase the number of rescued civilians by more than 10% compared with other strategies, such as depending on the HP of the civilians or the distance for tasks planning.

However, during evaluation there were some scenarios where the new approach performed almost similar to the other two approaches. This is likely to happen since the two parameters used for the other two approaches are practically a part of the new learning model and planner, especially, the one with the civilians sorted according to their HP. In future work, we are planning to use a more weighted classification for further enhancements of the results. For example, the HP in the used training dataset could have a larger weight than the other parameters in both the training and prediction phases. Additionally, when the approach was tested against previous approach using a map highly dependent on the ambulance performance, better scores were achieved.

The new proposed solution did not only help optimize the task planning of the agents and achieve better results. It also helped overcome the obstacles enforced by the inaccurate values retrieved from the simulator regarding civilians. In other words, having the training dataset was the reason the relation between these parameters was finally revealed and understood. As mentioned in Section 2.1.3, the presented graphs showed that neither the HP nor the damage can determine the ETD of the civilian if used alone. This strategy was previously used for rescuing civilians. This explains why having a training dataset that consists of multiple parameters was highly effective to determine and predict the ETD. Moreover, this helped clarifying what are the parameters that mostly affect the state of the civilian at each time step.

REFERENCES

Abouraya, A., Helal, D., Sakr, F., Khater, N., Osama, S., and Abdennadher, S. (2014). Clustering and planning for rescue agent simulation. In *RoboCup 2013: Robot World Cup XVII*, pages 125–134. Springer.

Afzal, A., Alizadeh, P., Nezhad, M. A., Ghaffari, K., Hasanpour, G., Jahedi, K., Kaviani, P., and Omidvar, G. (2013). Poseidon team description paper robocup 2013, eindhoven.

Bonet, B. and Geffner, H. (2001). Planning as heuristic search. *Artificial Intelligence*, 129(1):5–33.

Guan, D.-q., Chen, N., and Jiang, Y.-h. (2010). Robocuprescue 2010-rescue simulation league team description. In *RoboCup 2010 Symposium Proceeding CD. Singapore: RoboCup Foundation*.

Hesam, A., Taheri, P., Ameri, M., Al-Bouye, M., and FaghaniLemraski, M. (2015). Robocup rescue 2015 rescue simulation league team description paper.

Hussein, A., Gervet, C., and Abdennadher, S. (2012). Multi-agent planning for the robocup rescue simulation-applying clustering into task allocation and coordination. In *ICAART (2)*, pages 339–342.

Kitano, H. and Tadokoro, S. (2001). Robocup rescue: A grand challenge for multiagent and intelligent systems. *AI magazine*, 22(1):39.

Lane, D. M. (2015). *Online statistics education: An interactive multimedia course of study*.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Russell, S. and Norvig, P. (1995). Intelligent agents. *Artificial intelligence: A modern approach*, pages 46–47.