CS886 – Assignment 3

Stefan Buettcher
#20161244

The RoboCup Rescue League

The Rescue League of the international RoboCup competition was created as a reaction to the disastrous earthquake that hit the city of Kobe in January 1995. The earthquake killed several thousand people and virtually destroyed an entire city. Agencies responsible for civil protection and disaster response were unable to cope with the situation; fire brigades did not have water to fight the fires that were spreading throughout the city, ambulances could not reach injured people because roads were damaged and obstructed by collapsed buildings, and the coordination of the available forces was very poor. [1]

The ambitious goal of the RoboCup Rescue League is the development of a system of cooperating agents (humans, computers, robots) that is able to deal with urban disasters like the one that struck Kobe in 1995. The Rescue League is divided into

- the Robot League, which faces the robotic problems arising when rescuing injured people from disaster areas, such as moving through collapsed buildings;
- the Simulation League, which targets the problem of coordinating the efforts of various participating agents from a high-level perspective.

This essay focuses on the 2001 RoboCup Rescue Simulation League and its winning entrant, YabAI.

The RoboCup Rescue Simulation League

In the Simulation League, several types of agents have to cooperate in order to extinguish fires in a city hit by an earthquake and to rescue injured civilians. The simulation proceeds in discrete time steps, each time steps represents 1 minute in real time. In each step, the simulation system informs each agents in the simulation of the status of the part of the city that the agent can have knowledge of (with respect to perception rules and such) and collects actions that the agent has decided to perform during the next step. A game instance consists of 300 steps and thus represents 5 hours of real time.

Each agent type has different capabilities: ambulances can rescue civilians, fire brigades can extinguish fires and police forces can clear roads from lingering civilians and debris from collapsed buildings. A headquarters agent that may support coordination and communication between the agents is associated with each agent type, i.e. there is a special ambulance headquarters agent, a fire HQ, and a police HQ agent.

All mobile agents (ambulance, fire brigade, police) can use roads to move through the city and can sense the surrounding area (e.g. look for civilians that are buried in buildings) and see everything within a distance of up to 10 meters. They can communicate with all nearby agents by vocal communication (up to a distance of 30 meters) and communicate with agents of the same type by electrical communication. If, for instance, a fire brigade needs to communicate with a police car that is not in the closer neighbourhood of the fire brigade, it cannot do so directly but has to send a message to the fire HQ, which relays it to the police HQ, which passes it to the intended recipient of the message. In other words, mobile agents of the same type share the same communication channel, while agents of different types do not.

Electrical communication is restricted: every agent may send and receive up to four messages per minute. If there are more than four messages within a minute on the same channel, every agent listening on the channel received a list of senders and may select which messages it wants to receive.

Cooperation between agents of the same type as well as between agents of different types in often helpful. A single fire brigade, for instance, is sometimes not able to extinguish a fire alone. Instead, it relies on the help of other fire trucks to help. Similarly, two ambulances can carry out a rescue action more quickly than a single ambulance. While this is often only a quantitative difference, things change when the civilian to be rescued is about to die, in which case even a few minutes may be very precious. Cooperation between different types of agents is necessary if, for example, a fire brigade cannot reach the fire it wants to extinguish because a road is blocked (in which case police has to clear the road first) or if an ambulance is surrounded by flames (in which case fire brigades should extinguish the fire very quickly).

Altogether, 10 fire brigades, 5 ambulances, 10 police agents, and one headquarters agent of each type had to be controlled in the 2001 Rescue Simulation League. In addition to these agent types, whose behavior is defined by each participating team, there are several other agents that define the environment that the entrants have to deal with. The most important are civilians, who may be injured or not, and who may cause traffic jams and thus block roads. The dispersion of fires in the city is calculated by a special type of agent, the fire simulator, which is also responsible for calculating the intensity of a fire and the effect of an extinguish action conducted by a fire brigade. An overview of disaster simulators is given by Tadokoro et al. [3].

Artificial intelligence and multi-agent system issues

There are two major typical AI problems that arise in the context of the Search and Rescue simulation and that have to be considered when designing a set of agents for the competition:

Planning Usually, there is more than just one task an agent can perform at any given time: since there are many fires in the city, the fire brigade agents have to decide on an order in which they want to fight the flames; an ambulance that knows of more than one injured person has to decide whom to rescue first; a police car has to decide on an order in which roads should be cleared, or it can even decide not to clear any road at the moment but instead check roads whose status is unknown and who are likely to be used by fire trucks or ambulances on their way through the city.

Uncertainty The only things that a certain agent knows for sure are the positions and intensities of the fires and the status of its closer environment within a radius of 10 meters. The most important fact is that the agents do not have definite knowledge about is the status of the roads. They can either be clear and thus usable by the agents, or they can be blocked by crashed buildings or hysterical civilians. Unfortunately, the status of the roads has to be known by an agents in order to determine the optimal route to a certain point in the city, e.g. a fire. It is even necessary for finding a solution to the planning problem because trying to fight a fire that cannot be reached is pointless. So, some kind of probabilistic modelling has to be developed.

In addition, to these standard AI problems, there are additional challenges that result from the special multi-agent scenario. The main reason that the Search and Rescue competition has to be considered a multi-agent scenario is the restricted communication that is allowed between agents. If unlimited communication were allowed, it would be possible to have a single agent that does all the calculation and develops plans for all other agents and then simply tells them what to do. However, the fact that each agent may only send up to four messages per minute creates the necessity to address certain multi-agent issues.

Communication Messages transmitted by one agent shouting in the direction of another agent are not too problematic, but radio communication is. Since there are only four messages allowed per minute per channel, it is not only necessary that the agents do not send unnecessary or unimportant information over the channel, but also that every agent has a strategy to decide what messages to receive, which it has to do with the identity of the senders as the only knowledge.

Cooperation If the above general AI problems have been addressed, every agent should have a plan defining the order in which different tasks are executed. However, it might be the case that an agent needs the help of another agent, which is for instance the case when a single fire brigade that needs to extinguish a fire cannot do so without the help of other fire brigades. Under what circumstances should an agent abolish its former plan if it receives a request to help another agent?

Reasoning about other agents Since the communication between agents is so severely restricted, an agent should take the probable plans of other agents into account when it is developing its own plan. If, for example, a street needs to be cleared and if that can be done by a single police unit, then all police units should be able to estimate the likelihood of other police cars doing the job so that not all police cars are going to the same place at the same time.

YabAI, the 2001 champion

When gathering information on implementation details of YabAI, I faced the problem that the strategy description by Morimoto et al. [2] is very unspecific. The authors point to a web address from where the source code of their implementation could be downloaded, but the HTTP object does not exist any more. I have been able to download a later version of YabAI ([4], from April 2002), which I use as the point of reference whenever the necessary information cannot be extracted from [2].

Ambulances

According to Morimoto et al., "is it nearly impossible to optimize a rescue plan", and "Ambulance teams search for injured persons in a distributed manner". This is a different way of expressing that ambulance agents are driving through the city in an almost random fashion without any kind of cooperation. If an ambulance agent knows the position of a civilian but does not know its health status, it heads towards the civilians and checks whether the person has to be rescued. If there is no knowledge on the positions of civilians, the ambulances go to special refuge positions where they expect to find persons that have to be rescued.

If the intended route of an ambulance agent to an injured civilian is blocked, it sends a message to the ambulance headquarters, which pass it on to the police so that the blocked road can be cleared. Although the authors claim that there is also communication between the ambulances, evidence supporting this claim could not be found in the YabAI source code.

Fire brigades

YabAI's fire brigade agents essentially consist of a set of rules that defines which fire should be fought first. Priority is given to early fires, which can be quickly extinguished and prevented from spreading easily. If no early fire with top priority can be found, the fire trucks try to extinguish fires that have unburned buildings in the neighbourhood. The rationale behind this is, of course, that it is worse if the flames reach a completely intact building than if they spread onto an already partially destroyed building.

There is no explicit communication between different fire brigade agents. Like the ambulances, fire trucks report blocked roads to the police.

Police forces

The police forces can repair roads that have been reported to be unpassable by other agents. Two strategies are employed: If any road is reported to be blocked, police forces go there and clear the road. If no road is known to be unpassable, police agents drive through the city and try to find blocked roads which can be cleared, with the obvious intention to allow ambulances and fire brigades to go to their target destination using the shortest route.

If a police unit has cleared a road, it reports the new status of the road to police headquarters agent, which propagates the message to the other headquarters. This way, all agents in the system are getting an updated view of the status of the roads in the city, which they can use to compute shortest routes. There is no special communication between police agents.

AI and multi-agent issues addressed

The following issues, as introduced above, are or are not addressed by YabAI's approach to the Rescue simulation problem:

Planning YabAI does not make any use of long-term planning. This holds for all agent types: Ambulances select one civilian, perform a rescue operation, then select the next civilian. Fire brigades select one fire at a time, according to the priority rules defined, extinguish it and then move on to the next fire. However, fire brigades may be interrupted in their current extinction action if a new fire with higher priority emerges. YabAI's police agents most of the time do not even decide what to do but simply do what they are told by other agents.

Uncertainty Ambulances and fire brigades have to deal with uncertainty if they compute the optimal from their current position to their current target destination. It is not always known if the roads that a route is composed of are blocked or passable. Costs are assigned to routes using the cost function: [2]

$$costFunction = actualDistance \times \begin{cases} 1 & \text{movable} \\ 10 & \text{not confirmed} \\ 100 & \text{unmovable} \end{cases}$$

Thus, the cost of using a road whose status is unknown is higher than the cost of using a road that is known to be passable, but lower than the cost of a road that is known to be blocked.

Communication Because communication is expensive, it is reduced to a minimum in the YabAI implementation. Messages sent are restricted to the status of roads (road unpassable/cleared).

Cooperation Although Morimoto et al. claim that there is cooperation between agents, they describe it in such a general way that it is not clear at all how this cooperation could look like. The YabAI source code shows no cooperation between agents other than that mentioned above (road unpassable/cleared).

Reasoning about others Possible intentions or plans of other agents are not taken into account when an agent decides what to do next.

In general, almost none of the main issues arising in the context of the Rescue and Simulation league are addressed in an appropriate way by YabAI. It is surprising to me that it could become the 2001 world champion.

Shortcomings and improvements

There are several scenarios in which YabAI would perform very poorly. For instance, if there are no blocked roads, police do nothing; if there are no fires, fire brigades do nothing. Since police agents and fire brigades can sense their environment, they could help the ambulances to find buried civilians instead of doing nothing.

What happens if an ambulance agent is surrounded by flames and is getting damaged by the fire? According to YabAI's strategy, it should go to the next refuge. A better solution would be if the ambulance sent an SOS message which made any nearby fire brigade help the ambulance and fight the flames surrounding it in order to minimize the additional damage that would be caused by cross the flames by the ambulance's way to the refuge.

The shortcoming that I am focusing on is the way that YabAI's agents deal with uncertainty. The cost function used to find a shortest path to a given destination is a pre-defined, static function, which is unnecessary. There are two ways in which this function can be improved:

1. The constant cost factor 10 for a road R whose status is unknown can be replaced by

   $$cost_{R,unknown} = cost_{R,clear} \cdot Pr(R \text{ is clear}) + cost_{R,blocked} \cdot Pr(R \text{ is blocked}),$$

   where $Pr(R \text{ is blocked})$ and $Pr(R \text{ is clear})$ are the respective probabilities. $Pr(R \text{ is blocked})$ can be estimated by an agent either from its own experience or from the experience of other agents. Since the police agents do nothing but travelling through the city and looking for blocked roads which can be cleared, it is reasonable to use their experience in order to estimate a road's probability of being blocked:

   $$Pr(R \text{ blocked} \mid R'\text{s state unknown}) = \frac{1}{|A|}\sum_{a\in A}\frac{\#\text{blocked seen by } a - \#\text{blocked cleared by } a}{\#\text{roads seen by } a},$$

   where A is the set of all police agents. Of course, the formula still needs to be adjusted because roads can be blocked in one direction or in both directions, and the probability of being blocked certainly depends on the length of a road, too.

2. The cost factor 100 for a road that is known to be blocked is completely arbitrary. It would increase the performance of the routing algorithm if the proximity of the closest available police agent were taken into consideration. Assume an agent a decides to take a route along which exactly one road segment R of length 1 is blocked. Let $cost(A, B)$ be the time it takes

to get from A to B and $repair(R)$ the time it takes a single police agent to clear R. Then it is possible to approximate the cost of the entire route from X to Y via the R:

$$cost(X,Y) = cost(X,R) + length(R) + cost(R,Y) + \begin{cases} 0 & \text{if } \Delta < 0 \\ \Delta & \text{otherwise} \end{cases},$$

where $\Delta = cost(P,R) + repair(R) - cost(X,R)$, and P is the position of the closest police agent. Again, this formula needs some adjustments that take into account the time it takes for the message sent by a to arrive at the next police agent and the movement of the police car that can happen in the meantime.

Cost values for routes that contain more than one blocked road segment can be calculated accordingly.

Remaining problems

The second improvement suggested can be problematic if many roads have to be cleared because

- police agents have to report their position periodically;
- any agent computing an optimal route has to know the positions of the police agents.

This causes additional communication overhead, which might in some cases exceed the limit of 4 messages per channel per minute that is part of the rules. However, it is possible to determine if there is a jam on the communication channel, in which case an agent should switch back to the old routing algorithm and thus reduce the number of messages sent.

The problem that there is no long-term planning in YabAI would still remain even if the above changes had been applied. However, this does not seem to be a dramatic problem because, after all, YabAI finished first in the 2001 Rescue Simulation League, which suggests that long-term planning is not absolutely necessary in the RoboCup Rescue Simulation League.

References

[1] H. Kitano. *RoboCup Rescue – A Grand Challenge for Multiagent and Intelligent Systems.* AI Magazine 22, 2001.

[2] T. Morimoto, K. Kono, I. Takeuchi. *YabAI. The first Rescue Simulation League Champion.* RoboCup 2001.

[3] S. Tadokoro, H. Kitano et al. *The RoboCup-Rescue: An International Cooperative Research Project of Robotics and AI for the Disaster Mitigation Problem.* Proceedings of SPIE 14th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls (AeroSense), 2000.

[4] T. Morimoto. *YabAI source code.* http://ne.cs.uec.ac.jp/˜ morimoto/rescue/yabai/