

The Black Sheep Team Description

Cameron Skinner, Jonathan Teutenberg, Gary Cleveland, Mike Barley, Hans Guesgen, Pat Riddle, and Ute Loerch

The University of Auckland
New Zealand

`{cam,jteu004,gcle027,barley,hans,pat,ute}@cs.auckland.ac.nz`

1 Introduction

This paper describes the functionality of the Robocup Rescue agents built by Team Black Sheep¹ at the University of Auckland². We have implemented agents for each of the four components of the simulated environment: fire brigades, police forces, ambulance teams and emergency centers (fire stations, police offices and ambulance bases). Our mobile agents are called Fireman Sam[1], PC Plod and Dr Ropata[2].

We begin by describing the components that are shared amongst all agents: path planning, prediction of fire spread and communication. Following this, a detailed discussion of the strategies used by each agent is presented. Finally, some known issues are described and our plans for future work are illustrated.

2 Path Planning

All agents use an any-time path planner. This planner maintains a table of shortest paths between any two Node objects in the world. When an agent requests a path from node A to node B the planner looks up the table to see if a path already exists. If it does, then that path is returned immediately. If not then the request is queued and the planner returns a null path to signify that no path has yet been found.

The path table is updated asynchronously by processing the first request in the queue. The table is filled with tuples `<next node,cost>` representing the paths. To find the path from A to B we simply follow the links along `<next node>` until we find B. The algorithm for updating the table is shown in Figure 1.

In addition to the `<next node, code>` tuple in each table entry, we also include a tuple for the best path that is known to be free of blockages, and the best path that contains unknown road states. Thus, agents can choose to take the shortest possible path, the shortest path that is known to be clear, or the shortest path that may have blocked roads along it. Generally, police agents prefer to take the shortest possible path because they can clear roads along the

¹ <http://www.cs.auckland.ac.nz/~rescue>

² <http://www.auckland.ac.nz>

way and other agents prefer the shortest clear path for important movements (such as to a refuge), and the shortest unknown path when time is not so critical.

```
function updateentry(start,goal)
  N := neighbours of start
  Best := null
  Best cost := 0
  for n in N do
    cost := cost of moving to n + cost of path from n to goal
    if best is null OR cost < best cost then
      Best := n
      Best cost := cost
  fi
done
Entry := <best,best cost>
```

Fig. 1. The path planning algorithm

3 Prediction

A similar algorithm is used for predicting the spread of fires. Again, a table of paths is maintained that contains the shortest path a fire can take from one building to another. This is also updated asynchronously. The table entries contain the path that a fire can take and the expected time for the fire to spread that far. This time estimate is a pessimistic approximation of the fire simulator and provides a worst case estimate of when fires will spread.

4 Communication

All our agent implementations share a common communication protocol. Messages are encoded using a set of message types, a timestamp and zero or more bytes of message-dependant data. Agents queue messages internally during a timestep. When it comes time to send the messages, they are packed into 256-byte chunks. Up to four (the maximum number of messages allowed by the Robocup Rescue rules) of these chunks are then sent to the kernel.

Agents are only allowed to read four messages per timestep. This is enforced by each agent. Platoon agents only listen to messages from their center, centers listen to messages from their platoon agents and other centers.

The first byte of each message says how many bytes of data are in the message. Following this are zero or more message types. The type header is followed

by a byte describing how many entries of this type there are, followed by the data for each entry. The data for each of the types is shown in Table 1. Each piece of data is packed into the fewest number of bytes possible, for example timestamps are always two bytes, IDs are 4 bytes etc. The different types are as follows:

- Need rescue** This message is sent by agents that are trapped in collapsed buildings. This is the highest priority message, as it is vital that all our agents are freed as soon as possible. The message contains the ID of the buried agent, it's buriedness and it's location, as well as a timestamp for the information.
- Road request** A request to clear roads is sent when an agent cannot plan a path to it's preferred target because of known blockages. Only the ID of the road is sent.
- Road update** Agents send updates of road status every time they receive a KA_SENSE message from the kernel. Every time the blockedness of a road changes the agents broadcast this information so that the whole team can update their world model. The ID, blockedness and timestamp of the update are sent.
- Building update** When a building changes fieryness all agents broadcast the change in much the same manner as the road updates. The ID, fieryness and timestamp of the update are sent.
- Civilian update** Similar to the road and building updates, a civilian update is sent whenever an agent notices a change in a civilian's properties. The ID, buriedness, location and a timestamp are sent.
- Building searched** When agents enter a building they send a building searched message to notify other agents that the building has been searched and does not need searching again. This improves the efficiency of the search for unbound civilians trapped in buildings. The building ID alone is sent. This message is the lowest priority.

As an example, suppose in timestep 23 a fireman wants to report fires in buildings 1884,2046 and 1002, along with a blocked road at 927 and a trapped civilian (2050) in building 2046. He will send the message shown in Figure 2.

Messages are stored in a priority queue until the end of the timestep. They are then packed into up to 4 256-byte chunks and sent via an AK_TELL message. Any messages that are not sent this timestep are left in the queue and sent at a later time.

Once a message is received agents update their own world model. When an agent attempts to update a property it must specify a timestamp showing when the new information was received - if the update came from a KA_SENSE message then the current simulation time is used, if it came from a KA_HEAR then the timestamp included in the message body is used. The old and new timestamps are then compared and the property is updated to the new value if and only if the new value has a timestamp greater than or equal to the old one.

For example, in timestep 10 a fire brigade agent sees a civilian trapped in a building. It will update that civilian's buriedness and set the timestamp to 10. In

timestep 20 it receives a message from an ambulance saying that the civilian was partially unburied in timestep 18. The fire brigade will again update the civilians buriedness, but it will use a timestamp of 18 (not 20). If the fire brigade had seen the civilian in timestep 19 (before the message from the ambulance arrived) and updated buriedness then the update from the ambulance in timestep 20 would fail because the message data is too old.

Message type	Data
Need rescue	This agents ID, buriedness, position and timestamp
Request road clear	Road ID
Road update	Road ID, blockedness and timestamp
Building update	Building ID, fieryness and timestamp
Civilian update	Civilian ID, buriedness, position and timestamp
Building searched	Building ID

Table 1. Message data

Starting byte	Bytes used	Value	Meaning
0	1	1	Length of message
1	1	3	Road update constant
2	1	1	Number of road updates
3	4	927	The road ID
7	2	1000	The blockedness of road 927
9	1	4	Building update constant
10	1	3	Number of building updates
11	4	1884	First building ID
15	1	1	Building 1884 fieryness
16	2	23	Timestamp
18	4	2046	Second building ID
22	1	2	Building 2046 fieryness
23	2	23	Timestamp
25	4	1002	Third building ID
29	1	1	Building 1002 fieryness
30	2	23	Timestamp
32	1	1	Civilian update constant
33	1	1	Number of civilian updates
34	4	2050	Civilian ID
38	1	30	Buriedness
39	4	2046	Position
43	2	23	Timestamp

Fig. 2. Example “report a fire” message

5 Strategies

This section describes the strategies used by the various Black Sheep agents.

5.1 Fireman Sam

The fire brigade agents base their selection of which fire to extinguish next based on the concept of a *perimeter*. A perimeter is a set of burning buildings, each of which is adjacent to one or more unburnt buildings. A burning building is considered adjacent to an unburnt building if the distance between building centers is less than or equal to 30 meters. Each perimeter therefore encloses one block of burning buildings.

Each fire agent maintains a list of currently known perimeters. When choosing what to extinguish, the agent chooses the most easily extinguished perimeter it can reach. Once this is done, a single building must be chosen as the target. This is done by first picking a building with the greatest number of extinguished (or burnt out) neighbours. By choosing this as the first criteria the agents tend to extend existing fire breaks. This prevents wastage - if a single building is extinguished then it is easy for the fire to spread around the sides of the extinguished building and continue to cause damage. By building fire breaks the agents are better at containing large fires.

If there is a choice between two buildings with the same number of extinguished neighbours then the one with the most unburnt neighbours is chosen. If there is still a tie then the smallest building is extinguished first, with further ties broken randomly.

These rules give a total ordering over target buildings. Fireman Sam will attempt to plan a path to each target in order of priority until a successful path is found (i.e. one which does not attempt to cross any blocked roads). Whenever a good path cannot be found the shortest path via blocked roads is planned and a request to clear each blocked road along the path is made.

If no good paths can be found then Fireman Sam chooses a target at random and begins moving towards it in the hope that by the time he reaches the blocked roads the police might have cleared the blockage.

If there are no targets at all then Fireman Sam starts searching buildings for trapped civilians and notifying the ambulance teams of anyone he finds.

When choosing which perimeter is the highest priority Fireman Sam uses the prediction model described in Section 3 to estimate how much damage a fire will cause in the near future. This is used to weight the “easily extinguished” measure of the perimeter. An example of changing perimeters is shown in Figures 3 through 5

5.2 PC Plod

PC Plod is a fairly simple implementation of the Police Force. This agent maintains two lists of targets: roads that have been requested for clearing, and all

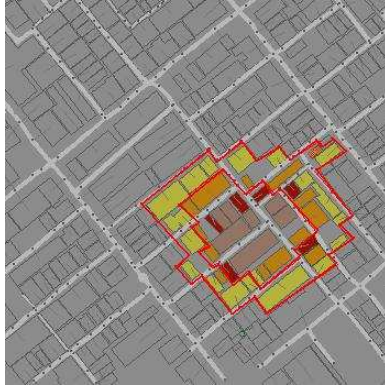


Fig. 3. Example of changing perimeters



Fig. 4. Example of changing perimeters



Fig. 5. Example of changing perimeters

known blocked roads. PC Plod will simply choose the closest request and attempt to move to it. Failing this, the closest known blocked road is cleared, and if there are no targets at all then PC Plod moves to a random road in the interests of exploration.

5.3 Dr Ropata

Dr Ropata is similar to PC Plod. He maintains a list of known trapped agents and known trapped civilians. Agents are rescued before civilians. Dr Ropata will move to the nearest target and attempt to rescue it, and take it to a refuge if it is a civilian. When there are no known targets Dr Ropata will begin searching nearby buildings.

5.4 Centers

Our three center implementations are simply message passers. When a message arrives the center updates its own view of the world - if the message data was relevant (i.e it resulted in a change to the centers world view) then it is passed on to the other centers and platoon agents, otherwise it is discarded. Messages such as requests to clear roads are always passed on.

6 Summary

We have broadly described how the Team Black Sheep agents have been implemented. Each platoon agent selects targets from a priority queue and attempts to plan paths that avoid blockages and traffic jams. The police agents will prefer to respond to requests to clear roads, the ambulance teams will dig out trapped platoon agents first, and the fire brigade agents will attempt to create fire breaks to contain the spread of fires. A simple messaging protocol has been established that allows sharing of information and requests for help.

References

1. Fireman Sam. <http://www.firemansam.co.uk> (2003)
2. New Zealand Film Archive.
http://www.filmarchive.org.nz/collections/collections_images.html (2003)