RoboAkut Rescue Team

Bilge Talaysüm and H. Levent Akın

Department of Computer Engineering, Bogazici University TR-80815 Bebek, Istanbul TURKEY

talaysum@boun.edu.tr, akin@boun.edu.tr

Abstract. RoboAkut is a multi-agent rescue team that combines autonomy and hierarchy. The members of the rescue team are dispatchers and platoons, where dispatchers are responsible of coordinating the platoons which do the actual rescuing. The dispatchers suggest target possibilities to the platoons, but the platoons are free to rescue targets that they have decided by themselves. The degree of autonomy versus hierarchy is learned via reinforcement learning by the agents. Also, the dispatchers learn the best options in suggesting targets to the platoons and the platoons learn the best targets to rescue via reinforcement learning. A variation of temporal difference of Q-values is used as the reinforcement learning method. The other main modules of the rescue team are the world model and the route finder. The agents keep both one-to-one and abstract world models, where first helps in navigation and the second helps in learning. Routes are found using breadth first search based on hop count. The agents can do greedy actions and can explore. Exploration is used during training, greedy actions are used in competition.

1 Introduction

RoboAkut is one of several projects carried out in the Artificial Intelligence Lab of the Computer Engineering Department of Bogazici University. The multi-agent rescue team involves both hierarchy and autonomy, the degree of which is determined via reinforcement learning. RoboAkut has participated in the RoboCup Rescue Simulation League 2002.

2 The General Structure

Both the dispatcher agents and the platoon agents have the following components: World model, learning module, I/O module and performance module. In addition, the platoon agents have a route finder component. These components are described in the following sections.

2.1 The World Model

Each agent keeps both a one-to-one world model and an abstract world model. One-to-one world model contains all the information provided on the environment by the simulation kernel. This helps in locating targets, calculating distances and routes. Each agent also keeps an abstraction of the world in order to describe the environment's state by a single index. Each time the world information of an agent is updated, each agent calculates three indices, which separately describe the severity of the conditions of the buildings, the civilians, and the roads in the environment. The indices are in the range [0, 1], where zero describes an ideal state where everything is under control, one describes a disastrous state. These indices are combined to form an overall index for describing the state the agent believes to be in. This index is used as the state description when applying reinforcement learning.

2.2 The Learning Component

Dispatcher agents learn which method is best in assigning platoons to targets in a given state. Rescue agents learn thich target is the most suitable one to save in a given state. Also, a rescue agent learns whether to obey the dispatcher's suggestion or to go after its own target decision. Both the platoon agents and the dispatcher agents use reinforcement learning as the learning method. A variation of temporal difference for updating Q-values is used as the reinforcement learning method[1], [2].

The options for a dispatcher agent when assigning agents to targets are to assign each agent to the closest target, to assign agents to targets starting from the target in best condition, or to assign them starting from the target in worst condition. Agents can be assigned to targets one by one, as couples or as triples. Therefore, in overall, there are nine methods of assignment for a dispatcher agent to choose among.

The options for a platoon agent in deciding on a target on its own is to go to the closest target, to the target in worst condition or to the target in best condition. If more than one target falls into the same category, the agent goes towards the closest of them.

The agents can be run and decide on their actions in two different modes: greedy or explorer. If the agent is running in greedy mode, it always tries the action with the best Q-value so far. If the agent is running in explorer mode, it can accept actions with lower Q-values with a predefined probability, which is the exploration rate. The explorer mode is to be used during training, and greedy mode is to be used during real action.

2.3 I/O and Cooperation

The I/O module contains the raw functions for sending and receiving messages to the kernel, and wrapper functions used for sending special messages.

The agents communicate with each other when the dispatcher assigns agents to targets, the platoons request targets or send results of actions to their dispatcher. Also, the agents report injured civilians, blocked roads and buildings on fire to each other.

The communication is structured so that each message destined to a platoon is first received by its dispatcher. In this manner, each platoon can listen only to its dispatcher and get the messages. This allows more messages to reach their destination when compared with the option of agents communicating directly with one another. This structure also allows for messages that cannot be sent in a cycle to be buffered to be sent in following cycles.

2.4 The Route Finder Module

The platoon agents are equipped with a route finder module. The information provided by kernel during the first two cycles of the simulation provides the information necessary for finding paths to any target in the environment. The route search is done as a breadth first search and loops are prevented by not adding a node that has been expanded before to the list of nodes to be expanded. Since the agent cannot always move to the target in one turn, at each loop, if the agent has not arrived, the route is re-calculated.

2.5 The Performance Module

The performance module of the agents executes in a loop and controls the flow of actions. The loop starts with receiving messages from the kernel, and making actions upon the received information. Two types of messages can be received by each agent "sense" message or "hear" message. Each "sense" message sent by the kernel to the agents indicates the start of a new cycle. The agents keep track of the elapsed time and the available number of messages that can be heard or said in a cycle by keeping track of the received "sense" messages.

Platoon Agents. The platoon agents start by connecting to the kernel, getting their identification numbers and sensory information. In each cycle the world information, abstract world information and the internal state representation is updated. If the agent does not have a target, decides whether it will choose a target suggested by the dispatcher or choose the target by itself. If it does have a target, it checks whether there is a more prior target in the environment or suggested by the dispathcer. When the agent decides on the target for the current cycle, it acts to save the target. When there is no target, the agent goes to random targets to explore the environment.

Dispatcher Agents. In each cycle the dispatchers assign their platoons to targets. Upon receiving feedback from the platoons, they update their Q-values and learn about the performance of their assignment methods. Also, the dispatchers act as a communication center, buffering and forwarding each message to the appropriate receivers.

3 Conclusions

The rescue team is trained so that the decisions yield good target choices. It is seen that the decisions of the agents in different disaster situations yield suitable results. Also, the design is independent of maps and initial conditions used.

References

1. Norvig, P., Russel, S.: Artificial Intelligence: A Modern Approach, Prentice-Hall, 1995.
2. Sutton, S. R., Barto, A. G.: Reinforcement Learning I: Introduction, MIT Press, Cambridge, MA, 1998.

This article was processed by the author using the TEX macro package from Springer-Verlag.