Sepanta Rescue Simulation 2004 Team Description

Sepanta Robotic Research Foundation
Rescue Simulation Project Team
Reza Hesamifard, Reza Zakery

rescue-sim@srrf.net
{r.hesami, r.zakery}@srrf.net
http://www.srrf.net/projects/rescue-sim

Abstract. This paper describes the main features of the *Sepanta Rescue Simulation 2004* Team. Main features will be addressed briefly, including our redefinition of the urban disaster problem, partitioning approach, bisectional mission regions, etc. Finally we will describe our future research directions.

1 Introduction

RoboCup Rescue Simulation Project subject is to develop an emergency decision support system by integration of disaster information, prediction, planning, and human interface [1]. To do this, a generic urban disaster simulation environment is created that simulates the urban environment and several disasters that may happen after an earthquake.

Besides the unique properties of this simulation environment as a multi-agent system that can be used as a good research problem in this area, the possibility of extending the system by adding necessary disaster modules is the next important feature that must be mentioned. Based on these two features, Sepanta Robotic Research Foundation has decided to define a project on this subject. The main goal of Sepanta rescue simulation project is building a real urban disaster decision support system in future, and RoboCup Rescue Simulation Project has been selected as a good starting point for this project.

In this paper, Section 2 explains our redefinition from the problem which has influenced the way we seek for a solution; Section 3 describes our partitioning approach and explains how we segment the urban environment into mission regions; Section 4 explains our agents' general behavior; finally, Section 5 includes conclusions and describes our future research directions.

2 Problem Definition

We have redefined the rescue simulation problem in a new way. In this way, the problem is defined as follows:

> *"We have three types of requirements in an environment. Each requirement is affecting other types of requirements. Also we have three types of facilities which of them responsible to resolve one of requirements. The problem is to find the best possible association between requirements and facilities which aren't fixed during the time."*

In other words, we have mapped each type of disaster to a requirement and each type of rescue simulation mobile agent to a facility. This problem has led us to a multi-agent system that can be viewed in figure 1. Based on this statement, each type of mobile agent has a separated view from the disaster environment and a set of requirements defined in this view.

The police agent just sees the roads and knows which of them is collapsed, but has no information about victims or burning buildings. By default the collapsed roads are requirements which their rank may increase when other agents need to use them. Similarly, the firefighter agent sees the roads and the burning buildings and knows nothing about the victims, and again by default burning buildings are requirements which their rank may increase when other agents need to use them. In the same way, the ambulance agent sees the roads and the victims. Also ambulance agents must see the burning buildings, because entering them destroys the ambulances.

Any changes in any of these three views affect the other views too. This can be done by announcing the requirements by each agent. Mobile agents can communicate just through their respective stations. For example, assume that an ambulance needs a way to be opened by the police. This ambulance must send a message to ambulance station and inform it about its requirement. The ambulance station receives the message and sends it to the police station. Then the police station defines a new requirement in its view and sends it to the police agents.

3 Partitioning Approach

Finding suitable positions for a set of facilities to resolve a set of requirements with the least total distance is a classic optimization problem. One of the most famous solutions to this problem is the “electric charge plate”. In this solution the requirements are assumed as negative electric charges that are placed on the plate border and facilities are assumed as positive electric charges that are spattered on the plate surface. The final stable position of positive electric charges determines the optimum solution.

Generally speaking, in Sepanta rescue simulation team we look at the problem of agents’ dispersal in disaster environment as a generalization of this problem. Based on this view we have three types of negative charges and three types of positive charges and the dispersal problem becomes a combination of three electric charge distribution problems.

After analyzing the disaster environment and finding the suitable dispersal pattern, each station must define some mission regions for its agents. Defining mission regions prevents agents from duty interferences and helps the rescue agents to cover the entire urban

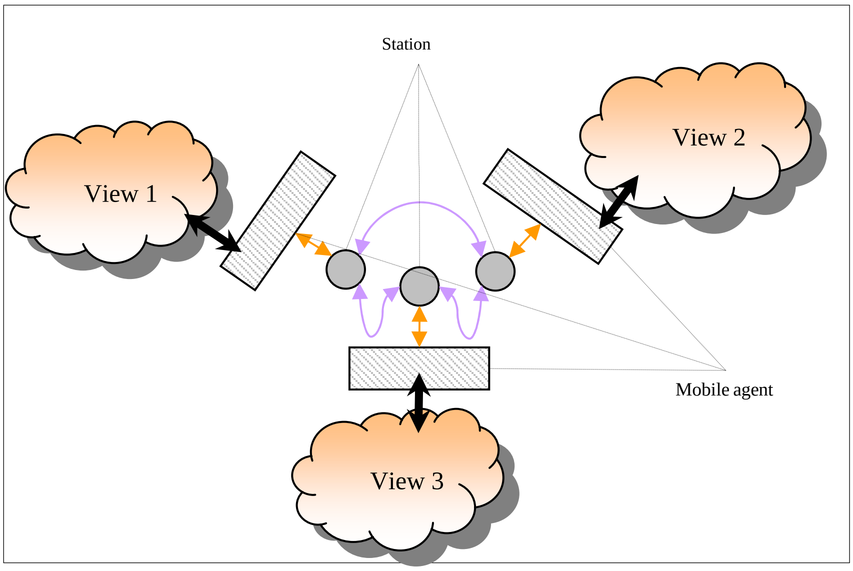

Figure 1: Each type of mobile agent has a separated view from the environment and a set of requirements defined in this view.

environment[1]. These regions can not be accurate because using accurate regions may cause ignoring some victim or burning building just because it is out of the mission region and leads to reduce the team performance.

At least two solutions exist for this problem. The first is using bisectional mission regions. The first section is named "primary region" and the second is named "border region". Every point in primary region has a membership weight equal to 1.0 and other points' membership weight is some value between 0.0 and 1.0. A distribution function determines the membership weight for points outside the primary region. Any distribution function can be used and of course results in different agents' behavior. Border region is created from points which are outside the primary region and their membership weight is greater than zero.

Border regions must be big enough to let the agents cooperate to resolve a high weighted requirement. There is no explicit limitation on the number of ambulance or fire fighter agents that may cooperate to resolve a requirement. Police agents have a restriction that at most two agents are allowed to open a collapsed road from two sides. When an agent decides to resolve a requirement, it announces its decision to other agents from the same type. By being informed that some body has decided to resolve a requirement, other agents decrease the weight of that requirement. In this way number of agents that cooperate on a requirement is controlled.

The next solution which we haven't implemented yet is using fuzzy mission regions. In this approach there will be no exact border for the mission regions. Of course this idea is currently under development and we must work more on it.

[1] The approach used by real human rescuers is something very close to this. They first partition the collapsed urban space into regions and then associate each rescue team to a region. Of course these regions usually are nearly separated and have no overlapping.

4 Agent Behavior

Based on the problem definition mentioned in Section 2, the general behavior of rescue agents is based on the requirement concept. As mentioned before, every view has some requirements defined in it. If an agent has several requirements in its mission region then it must choose between them. The priority of a requirement is a function of some parameters such as its weight in view (first specified by respective station, but later may be decreased by the agent itself as described in previous section), requirement position membership weight in agent's mission region, and the feasibility of requirement resolution (for example a firefighter may decide to attack a burning building that the roads to this building are all collapsed). This function currently is a combination of some hand written heuristics, but it is the first subject of using machine learning methods in our agents. The probable candidate approaches that we guess may outscore others are using GP or ANN. As reported in [2] GP methods has been used successfully in such an environment, and we guess ANNs as universal function approximators must do well too.

When there is no requirement in agents' views, they do the default behavior. The default behavior for all the agents is traversing their mission region. Trivially if the agents see something that is new for them (for example a police agent that notes a burning building that based on its map must be safe) must report it to their station.

All the agents use A* for finding the shortest path between two points. The straight line distance is used as the A* heuristic function.

5 Future Works

From this point forward, we first focus on using machine learning methods to improve the requirement priority function. The first candidate method is GP and we like to test ANNs too. The next idea we like to implement and use in our agents is the concept of fuzzy mission regions that we expect it to improve the dispersal of agents in the environment.

References

1. Takeshi Morimoto, How to Develop a RoboCupRescue Agent, http://robomec.cs.kobe-u.ac.jp/robocup-rescue.
2. Mazda Ahmadi, Multi-Agent Hierarchical Reinforcement Learning, Thesis, Sharif University of Technology, Iran, 2003.