Gemini-R Team Description

Masayuki Ohta

Cyber Assist Research Center,
National Institution of Advanced Industrial Science and Technology,
AIST Tokyo Waterfront 2-41-6 Aomi Koto-ku Tokyo 135-0064 Japan
ohta@carc.aist.go.jp

Abstract. We implemented Gemini-R for the purpose of acquiring a sub-optimal rescue policy automatically with using machine learning techniques. We have dealt with agent distribution problem and problem to decide fire extinguishing order as sub-problems of rescue operation. We will put these subtask together and build a system as the whole.

1 Introduction

RoboCupRescue project[1] is aim at disaster prevention and mitigation from a standpoint of computer science. Along this project, we implemented a set of rescue agent program Gemini-R which can run on a disaster simulator RoboCupRescue Simulation System[3]. Our main goal is to acquire one of a sub-optimal rescue policy automatically for given disaster information, using machine learning technology, and to make use of it to the rescue operation in the real world. This domain includes a lot of problems such as simple adjustment of variable, agent dispatch problem, leader selection, cooperation between different kind of agents etc. Because this contains various factors, we are taking bottom-up approach starting from a small part and then combine all together. In this paper, we introduce our approach to these sub-problems and introduce how to combine them.

2 Features of Gemini-R

Gemini-R includes three kind of rescue agents: fire brigade, ambulance team, and police force. It does not have long term strategy at present, and behaves reactively for perceptual information. We have not implemented communication between agents, so agents' actions are independent each other. Police forces and ambulance teams search for disaster-stricken area randomly, and execute their own rescue command as the need arises. Fire brigades calculates priority of fire extinguishing operation with neural network from the layout pattern of buildings. The building which has the highest priority is selected as the next target. The neural network was trained off-line. In the next section, we show the details of this fire brigade.

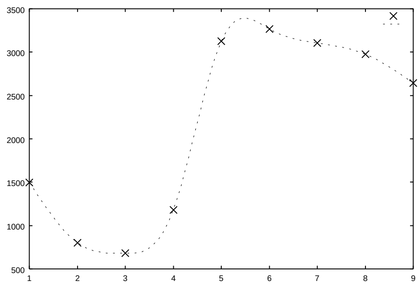

Fig. 1. How many fire brigade to dispatch for one fire site

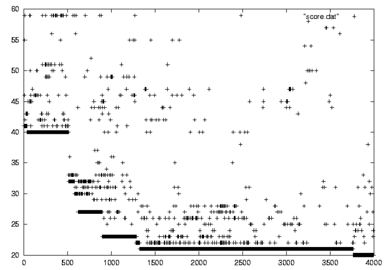

Fig. 2. Learning curve of priority

3 Learning Approach

Results of learning are applied to the action rule of the fire brigade in Gemini-R. We have dealt with "agent dispatch problem" and "action priority problem" for experiment. We introduce these two briefly, and show a method to integrate them.

3.1 Agent Dispatch Problem

First, we investigated the relation between the number of fire brigade and efficiency when a fire brake out. The result is shown as Fig.1, in which we can see that there is a point which maximize the efficiency. It shows that dispatch 5 or 6 agents per one fire site gives the best efficiency in this case. In this experiment, we showed fire fighting is such kind of operation that cooperation can make the efficiency per agent be much improved[2]. This means damage can be reduced by deciding the distribution ratio of fire brigade for the fire site properly. Using this result, we can decide how many agent to dispatch for each fire.

3.2 Learning Priority of Actions

Second, we make agents learn priority of each building to extinguish fire, and decide the optimal order of fire extinguishing. The agents did random search with their neural network, to find a policy which minimize the damage. For the input of the neural network, we use the distance to the building and the average of fieriness of neighbor houses of the building in eight directions. The learning curve for this experiment is as Fig.2, where the x-axis is the number of episodes and the y-axis is the number of burned buildings. The result of fire fighting before learning was as Fig.3, but after 4000 episodes, the result was as Fig.4. The agents after learning could show almost the same result as a hand coded program. In this experiment we use random search because the continuity of the problem space is low and then methods like hill climb is not suitable.

Fig. 3. Before learning

Fig. 4. After learning

3.3 Integration Using Auction Mechanism

We are planning to integrate above two using auction mechanism. In this case, fire office is the auctioneer, burning buildings in the agent dispatch problem can be considered as items which are bid for, and each fire brigade can be considered as a bidder in auction. As the price, fire brigades bid the number which indicate how well it suit for the task. It make bids with a set of item and price pair such as "((A, 5),(B, 3))" to the auctioneer, and the auctioneer searches the combination maximizing the total price taking the change of price by cooperation into account.

There are some researches applying auction mechanism into cooperation problem on multi-agent domain[4]. But it is sometimes difficult for bidder to estimate the price of items. Moreover, because efficiency per agent will change as the number of the cooperator, this problem becomes much more difficult. In our approach, proper price is acquired automatically, and the influence of cooperation is also considered. A disadvantage of our method is the lack of the way to select the bids by the bidder. Now the bidder make bids for all burning buildings, and it takes long time. We should select the bid and change it to calculate sub-optimal solution in shorter time.

4 Conclusion

Gemini-R carry out rescue operations according to a sub-optimal policy acquired with off-line learning. We are taking bottom-up approach, and have dealt with "agent dispatch problem" and "action priority problem". Auction mechanism is one of an available candidate to integrate these smaller problem solver, but we should find the way to reduce the computational complexity to use this.

References

1. Hiroaki Kitano, Satoshi Tadokoro et al. RoboCup-Rescue: search and rescue in large-scale disasters as a domain for autonomous agents research, Proc. of IEEE SMC, 1999.
2. Masayuki Ohta, Tetsuhiko Koto, et al. Design and Implementation of the Kernel and Agents for the RoboCup-Rescue Proc. of The Fourth ICMAS pages 423-424, July 2000.
3. http://kiyosu.isc.chubu.ac.jp/robocup/Rescue/manual-English-v0r4/manual-v0-r4.ps
4. Ranjit Nair, Takayuki Ito, Milind Tambe and Stacy Marsella, "Task Allocation in the Rescue Simulation Domain: A Short Note", RoboCup 2001 Book, Springer Publication