RoboCupRescue 2009 – Rescue Simulation League Team Description <BonabRescue2009 (Iran)>

Farshid Faraji, Hekmat Hajizadeh, Mohammad Reza Khojasteh, Mohsen Bayat, Abbas Abdolmaleki

AI & Robotics Laboratory, Computer Group, Azad University of Bonab
Iran
{faraji, khojasteh, hajizadeh, bayat}@bonabrobotics.ir
http://www.BonabRobotics.ir

Abstract. This paper describes the main features of the BonabRescue 2009 rescue simulation team. Our major goal is to investigate different aspects of the Reinforcement Machine Learning Methods in RoboCup Rescue Domain. We have proposed a cooperation method by implementing a control structure based on behavior. We described how we taught our agents to cooperate without communication using Q leaning technique. Here our emphasis is not on learning a behavior by an agent; but the goal is choosing the best behavior in different situations and then to get cooperation in a group.

1. Introduction

RoboCup Rescue Simulation is an excellent multi-agent domain which includes heterogeneous Agents that try to cooperate in order to minimize damage in a disaster. These Agents must have effective cooperative behavior despite incomplete information.

Much research in to agents has focused on building agents that can act robustly in a real world environment. To act in the real world requires that an agent be able to handle complex and dynamic situations. Complex environments are difficult to handle because they are too complicated to model completely and have too many unknown factors to reliably predict the outcome of a series of actions. Dynamic environments present difficulties because they are changing rapidly and unpredictably. To perform useful functions in such environments, agents must be autonomous. It means that they must autonomously perform both pro-active and reactive actions.

In this paper we have proposed intelligent behaviors that are done by autonomous agents in different type of situation. We believe that in distributed systems, complex overall intelligent behavior of the agent emerges from local interactions based on simple rules. Here our emphasis is not on learning a behavior by agent; but the goal is choosing suitable behavior in different situations and then to get cooperation in a group by above mentioned rule.

In our proposed model, agents act autonomously and learn to do the best behavior in each time as regards to its behaviors, effects of these behaviors on environment and

environment's changes. This decision is based on local information and choosing the best team's behavior leads not to require communication among agents and leads to suitable team work. The important point in this structure is to present a mechanism to choose behaviors and the agents should learn how to treat in different conditions to benefit the team more.

Control mechanisms which are used in most Behavior-Based Systems are something like switching circuits and in the most cases summation operators or switches had been used in these systems [8]. These operators were defined before and are situated among behaviors in constant form. For complicated agents predicting such controls are very difficult even impossible and by increasing the number of behaviors, controlling and coordinating will be more complicated [9]. Therefore learning is essential here and according to Behavior-Based Agents' structures, the learning should be distributed and there should not be any central learning parts. Researches on learning show that Reinforcement Learning is a suitable way for reactive behaviors, because in dynamic environments due to high rate changes the learned data can not be put together. In other words, unlike other learning methods like neural networks which require more learning data, agents learn how to do the proper action in different conditions by getting environment status and reward of submitted action using reinforcement learning method. In other words, the agents receive reward or punishment due to its result of work. So they learn which behavior under which conditions have the most benefit by evaluating different mixtures of behaviors and gained rewards or punishments.

2. Artificial Intelligence based on Behavior

Brooks [4] has made an architecture based on behavior in order to control robots. He describes an alternative approach to handling the dynamics of the real world [2], [3] that involves decomposing an agent system to behaviors instead of decomposing it to functional modules like observing, modeling and planning. This method describes an agent by its behaviors towards goal [4]. Behaviors are independent processes each responsible for some specific aspect of the agent's interactions with the world. In behavior based approach, emphasis is on a set of interactive distributed concurrent behaviors [5].

To cooperate in multi-agent systems it requires having a control structure or learning algorithm. For agents who act in dynamic and indeterminate domains, it is difficult or impossible to build a model to define agent's work and it's dynamic exactly. Thus, traditional control based on model is not efficient for a multi-agent system's agents. Because in such controls, the agent creates model of environment and updates it by preceding time and environment's changes. Since, in the dynamic domains like rescue simulation domain, environment's changes are very extensive, so it sounds modeling the world and updating it is not well approach.

Behavior-Based approach is an adequate implement to built agents which act in dynamic environments. In this approach, regardless to world model, the goal is gained by answering to behaviors [1]. In our proposed approach, in addition to Behavior-Based Approach problems, the scalability in cooperative multi agents systems is also

considered. In other words, by increasing the number of agents in a multi agent system, this type of cooperative model can be easily extended. Our proposed approach consists of using conditions and behaviors in order to cooperate and also to minimize state space by converting it to behavior space.

3. Control structure of behaviors

According to figure 1, our control structure categorizes agents' actions in three layers. In the Bottommost layer there are actions that are applicable in the environment and influence on environment directly. In this layer we have defined some skills for each agent with regards to their types. We have designed and implemented these skills as some components for each agent. In the second layer high level behaviors are situated that from conceptual viewpoint, they are accomplished by agents. And in third layer imagery, believes and goals of agents are situated. Unlike [6], [7] in this mechanism, actions are not in a tree form and the third layer in addition to its lower layer has direct impact on physical layer (first layer). The relationship among three layers are not created as constant forms in designing phase, but, behavior is chosen by strategy layer with regards to environment current condition. Agents' duty is to learn to do which action under which condition.

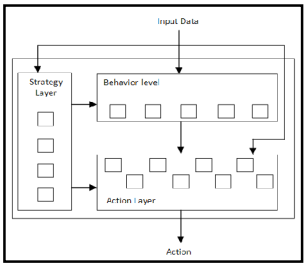

Fig. 1. Control Structure based on behavior.

In this general structure we can build agents that do assured work in different conditions by applying learning in each layer.

4. Fire Brigade Agent

Ability of overcoming environment changes is called Adaptability. Learning follows two major goals:
1. Adapting to inner and outer changes
2. Simplifying Built in Knowledge

The mechanism which we used to learn is based on Reinforcement Learning. Agents' learning from each other is for the sake of cooperative among them. The mean of learning here, is not to learn the way of doing an action by agents, but the agents learn to do which behavior under which time or condition. Agents by getting

visual and vocal information can be aware about the environment status and decide to do suitable behavior. And as a result of doing each action, they learn its reward or fine. According to reinforcement learning method, behaviors that the agents present in each status are towards to increase received reward or is towards to explore new behavior. We will show how our fire brigade agents cooperate with each other without any communication or without any central control. In this implementation a fire brigade agent with regards to environment status learns which one of the buildings has the highest priority for extinguishing in each cycle.
We have defined three behaviors for our fire brigade agents:

1. Extinguishing the most difficult fiery building
2. Extinguishing the easiest fiery building
3. Extinguishing the nearest fiery building

In order to evaluate the most difficult and the easiest fiery building we do some calculations on buildings properties. These properties are: number of floors, building's area and ignition time.

We taught our agents in two phases: Off-line Q Learning and On-line Q learning.

4.1. Off-line Q Learning

There are many problems and factors in rescue simulation domain. According to our previous experiments, agents can not extinguish fiery buildings alone; instead by assigning 5 or 6 brigades to fiery buildings they can minimize damage. And also the highest performance will be achieved when 5 fire brigade are assigned to extinguish one fiery building. Also our primary tests results show that at least 5 fire brigades are needed to achieve proper performance however it depends on the number of fiery points.

We started our tests with 5 fire brigade agents and 3 fiery points. All simulations were done in 300 cycles. And in all simulations, agents chose mentioned behaviors randomly. Q values of each behavior in each state are updated by formula (1) considering their rewards or penalties. In this formula which is called updating Q values in Q learning method, i is the environment state at the beginning of doing action, j is the environment state after action, Q_{a_i} is Q value of choosing goal a in state i, $maxQ_{a_j}$ is maximum value of Q in state j, $R[i]$ is reward or penalty in result of changing to state i, Res_a is reward or penalty in result of doing action a and lr is learning rate. $R[i]$ and Res_a are called two different ways of giving reward mechanism. If environment is in appropriate state, R[i] will be positive and if an agent does his work well Res_a will be positive value.

$$Q_{a_i} = (1 - lr)\, Q_{a_i} + lr(R[i] + Res_a + maxQ_{a_j}) \quad \textbf{(1)}$$

After 50 simulations, agents got ideal results by choosing their behaviors with considering achieved Q values.

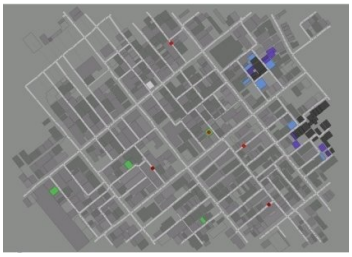

Fig. 3. *Environment state at the end of one simulation after learning*

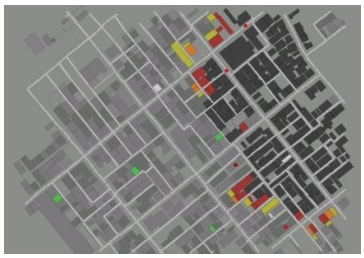

Fig. 2. *Environment state at the end of one simulation without learning*

Figure 2 is the status of city at the end of one simulation without learning. As it is seen in fig. 2 most of the city was completely destroyed. Figure 3 is the status of city after learning agents. Here fire brigade agents could restrain fires spreading.

4.2. On-line Q Learning

The aim of doing this test was applying online learning and reaching to actual conditions from viewpoint of number of agents. To do so, the number of fiery points increased up to 4. The 4th fiery point was situated far from others to be complicated. As mentioned before, the number of fiery points has direct influence in determining number of agents to do work. Primary tests with 6, 7, 8 fire brigades showed that they were unable to restrain fires of this simulation (after 50 simulation there wasn't any improvement in agents work and it was because of lack of agents, not agents' cooperation). By increasing fire brigades up to ten, it showed the improving of agents' ability in restraining fires. To apply on-line learning, initially agents had used off-line Q learning values. After 60 online simulations, agents got acceptable results and they showed intelligent behaviors by dividing to groups to extinguish fires. The result of choosing each behavior by agents shows that Q values were converged.

Fig. 4. *Environment state before on-line learning.*

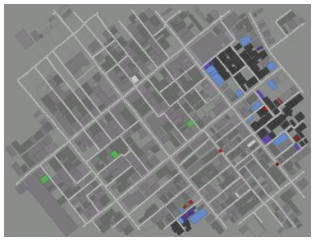

Fig. 5. *Environment state after on-line learning.*

Diagrams 6 and 7 show the Q values of agents after 50 offline and 60 online learning. Value of Q in each state shows the probability of choosing behavior. In our tests the maximum value of Q was used. This means that in different States, the

behaviors with highest Q value is selected to be done. The following points were concluded from diagrams:

1. In preliminary states – approximately from state 1 to 6 – agents prefer to do the easiest work. This means that at the first cycles of simulation, the state of city is not very fatal so the agents choose the easiest work to prevent spreading fires.
2. From state 6 to 35, they never choose the most difficult work. This means that the agents learned to choose the easiest or the nearest work because the city state not reached to the critical threshold yet.
3. After state 35 the agents seldom choose the easiest work and with little deference between the nearest and the most difficult work, they choose the nearest work. In this states agents learned that because of spreading lots of fires, the easiest work may be far from them and so they don't waste their times by going to other places and prefer to extinguish the nearest fire to minimize damage.
4. At the last states, after state 51, the fires states are terrible and choosing each one of fires to extinguish will not be more useful.

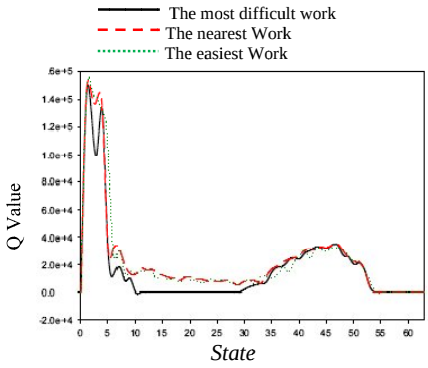

Fig. 6. *Q values based on state after 50 off-line simulations. Q value in each state show the probability of selecting a behavior.*

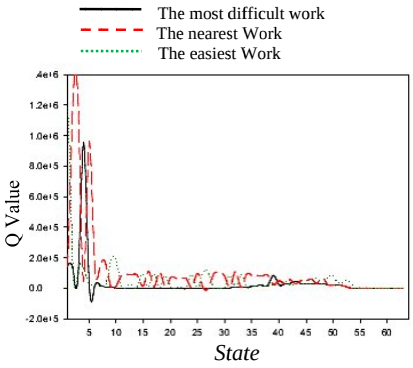

Fig. 7. *Q values based on state after 50 on-line simulations. Q value in each state show the probability of selecting a behavior.*

5. Ambulance Team

Ambulance agents play an important role to rescue civilians and collapsed agents that are buried under collapsed buildings. The way of their operation directly affects the number of civilians in the city whose lives can be saved after the occurring of the disaster.

No doubt this is the most important goal of a rescue team. Ambulance Teams require a lot of useful information to accomplish their tasks. These kinds of information include the position of the collapsed agents and civilians, some parameters showing whether they are dead or alive and the fire spreading too.

There are large numbers of state and there are the vast numbers of possible actions in such a complex environment. Whit considering these facts and the fact that we are

going to have a complex cooperation with minimum communication among agents, we have based our ambulance teams' work on behavior again.

As we mentioned in previous sections, by Behavior-Based Approaches we can handle the complexity of environment. So to extend Ambulance Teams performance near to optimal based on behavior, at first we had to define some behaviors to our ambulance agents.

By considering the fire brigades behaviors and also our experiments, we have defined three behaviors to our Ambulance Teams as follows:

1. Rescuing the nearest civilian
2. Rescuing the nearest civilian to fiery building
3. Rescuing a civilian with lowest dead time (or with highest priority)

We found out that these behaviors are more important than others. We will try to consider other factors in our future work.

Ambulance Teams should choose one of these behaviors during the simulation. Again we have used Q Learning technique to teach our agents to choose the suitable behavior in each situation. By the way, using this technique made us to reduce our inter agent communication. In our previous teams we had used some algorithms to manage our agents to work in groups, to do so, agents had to send some information about what they are doing and which civilian should be rescued. And sometimes they chose a leader to manage themselves. Therefore using of communication had increased. As we mentioned before, intelligent behavior decreases the inter agents communication and if necessary, high level information are exchanged.

Our test results showed that in this Behavior-Based Approach, our agents reached to optimal complex cooperation by doing individual actions, by applying learning to the behavior layer. From agents' viewpoint, they have done just some simple actions through simulation, but from an outer viewpoint, they have worked in some groups and they have a complex cooperation, whereas we did not implement any algorithm to group them or they do not use any communication to cooperate with each other. This is what we expected.

In the learning phase we consider time limitations to fine agents. For example if an agent could not reach to a selected target in a limited time, it was punished, this cause that in the primary cycles they choose the nearest civilian as regards to blockade roads.

6. Communication less

Communication is necessary for both information retrieval and coordination. RCRSS (RoboCup Rescue Simulation System) provides two facilities for the agents to get in touch with their environment. The first one of them is the local perception or visual sense which contains the agents' surrounding region information provided for them. And the second one is the communication capabilities of the agents accompanied by some limitations. It is very obvious that as in any other complex multi-agent domain, no single agent might have a complete knowledge of the global state of the

environment if it just wants to rely upon its local perception. Therefore communication becomes an essential part of an agent's job in order to accomplish an effective team-work. In this part we focus on our agents' communication strategy.

Communication-less maps idea was applied for the first time in China 2008 RoboCup competitions. In these maps there is not any radio communication among agents and they can only communicate with each other via channel 0 which is not a radio channel. Channel 0 is a say channel and its amplitude is limited to 30 meter. In fact in these situations we can never use centralized strategies to submit tasks or update agents' world model.

The important thing in these situations is exploring environment and it should be considered that the agents unable to communicate with each other as easy as other maps.

We have designed a strategy to handle these kinds of problems.

At the beginning of the simulation, when Police agents received map and environment information, they divide map in to the number of virtual partitions which equal to the number of Police agents using partitioning algorithm. Each partition is assigned to a Police agent. In the first cycles the Communication-less condition is checked by sending Hello message to center and other agents. If no reply is received it means that we must change to Communication-less state.

In this state each Police agent goes to their partition and explores that area for specific duration of time. This time is depending on how large the map is. After exploration, all agents go to one of the partitions and renew their world model via communication on channel 0. By this way agents will nearly be aware of environment information. After this operation they do their jobs separately. But every cycle agents broadcast their useful information via say channel to aware other agents which are located in 30 meter radius from itself.

7. References

1. Z. Coa, B. Zhang, M. Tan, "Research of Behavior-Based Real time Formation Marching for Multiple Mobile Robots".
2. R.A. Brooks, "Intelligence without Reason", A.I. Memo No.1293, Prepared for computers and Thought, IJCAI-19, April 1991.
3. R.A. Brooks, "A Robust Layered Control System for a mobile Robot", IEEE Journal of Robotics and Automation, RA-2, April 1986.
4. P. Maes and R.A. Brooks, "Learning to Coordinate Behavior", Massachusetts Institute Of Technology.
5. P. Scerri, "Engineering Strategic Behavior in a Multi-Layered Behavior Based Agent", Department of Computer Science, RMIT, May 1997.
6. S.Franklin and A. Graesser, "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", University of Memphis, 1996.
7. N. R. Jennings, "Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems using Joint Intentions", University of London, 1995.
8. P. Stone, R. S. Sutton and S. Singh, "Reinforcement Learning for 3 vs. 2 Keepaway", AT & T Labs-Research, 2000.
9. C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo and A. Sgorbissa, "Communication and Coordination among Heterogeneous Mid-size Players: ART99", 2000.