RoboCupRescue 2010 – Rescue Simulation League Team Description <BonabRescue (Iran)>

Farshid Faraji, Mohammad Reza Khojasteh, Hekmat Hajizadeh, Reza Moheb Alizadeh, Naser Irani

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

Abstract. In this paper with regards to existing problems in cooperation among agents, especially in the dynamic environments, a method based on behavior and reinforcement learning is presented. Here, we investigate and analyze the treat of Learning Automata as agents decision strategy in multi agents systems in order to reach coordinated behaviors. 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. Our experiments show the successes of this method in achieving grouped intelligent behavior. Obtained Results in these experiments show that agents by doing individual behaviors and using reinforcement learning, learned to present suitable behaviors in different conditions.

1. Introduction

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 simulated team, 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.

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 cannot 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. So they learn which behavior under which conditions have the most benefit by evaluating different mixtures of behaviors and gained rewards or penalties.

2. Artificial Intelligence based on Behavior

Reactive robot became popular at a later date. This approach involves programming the robot to react quickly to its environment. The robot must react to the problems it encounters. The robots do not build a model of their world, they simply act in response to the things they encounter whilst existing there. Achieving this paradigm requires developing behaviors for the robot to execute or exhibit. These behaviors enable the robot to operate in environment.

Brooks [1] 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 [1]. 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 [4].

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 [5]. Our approach consists of using conditions and behaviors in order to cooperate and also to minimize state space by converting it to behavior space.

3. Getting cooperation via behaviors

Testing a behavior based systems is quite different from deliberative systems. We can test individual parts of the system, we are able to build up the behaviors the agent will have and then test each for correctness. It is possible to develop and check a behavior before developing another behavior. This type of modular approach gives another added benefit. If one behavior breaks down it does not mean that the whole system collapses. This added benefit gives the system extra robustness something required to operate in an uncertain would. A set of behaviors for a reactive (behavior based) fire brigade rescue agent might be:

1. Extinguish The Nearest Fiery Building
2. Extinguish The Most Difficult Fiery Building
3. Extinguish The Easiest Fiery Building
4. Extinguish The Biggest Fiery Building
5. Extinguish The Smallest Fiery Building
6. Extinguish The Most Dangerous Fiery Building
7. Extinguish The Nearest Fiery Building To Civilian
8. Extinguish The Fiery Building Which Is Extinguishing by Neighbor Agent

We have used these behaviors in our team. 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 kind of building (wooden, steel frame or reinforced concrete). Also to evaluate the most dangerous fiery building we use a formal which we used in our robocup 2006 rescue team (Persia 2006) [9].

In order to investigate each one of these behaviors and analyze the effect of them in the agents' prosperity, we did some tests by the following way:

We design a map with three initial fiery points, 740 building and 72 civilian then the scenario was conducted. The tests were started with 5 fire brigade agents. All simulations were done in 300 cycles. At first we test with fire brigade agents who have no goal and they select their targets randomly.

As the result shows in fig.3, most of the buildings and civilians have burned. We continued our test and this time, in each test, fire brigade agents handled one of the 8 mentioned behaviors.

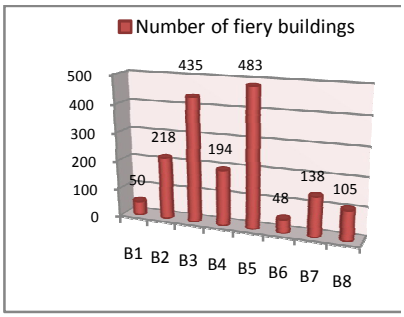

Fig. 1. The number of fiery buildings at the end of tests by selecting each one of the behaviors.

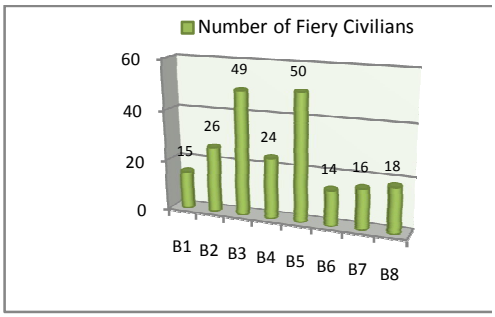

Fig. 2. The number of burned civilians at the end of tests by selecting each one of the behaviors.

Results in fig.1 and fig.2 show the effect of mentioned behaviors in the rescue simulation domain. Choosing the most dangerous fiery building has the highest performance and choosing the smallest fiery building has the worst performance. Also it shows, by defining suitable behaviors we can improve the performance of agents.

To complete our tests in the next step we forced our agents to do one of the 8 mentioned behaviors randomly. Fig.4 shows obtained results.

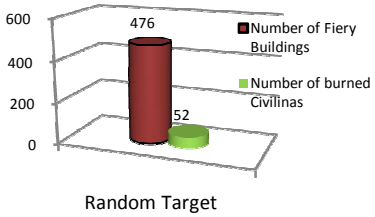

Fig. 3. Number of fiery buildings and burned civilians by selecting random targets.

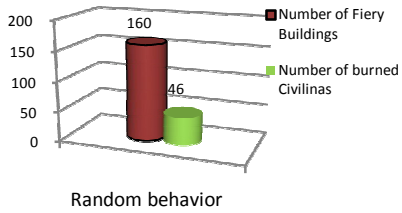

Fig. 4. Number of fiery buildings and burned civilians by doing random behaviors.

Results show that without cooperation among agents they cannot improve their performance.

4. Evaluating reinforcement learning and cooperation among agents

The goal of agent, who acts in dynamic environment, is perform an optimal decision. If the agents be not aware about reward of various actions in the environment, selecting an action will be difficult. In reinforcement learning, agents do not need to model the world explicitly, because their actions can be established via rewards. So

this type of learning will be useful, especially when agents have less information about the environment.

It should be considered that learning problem in multi agent systems has more complexity and agent in addition to learning its reactions' effects, should be learned how to coordinate its reactions with others. Current works show that reinforcement learning lead to achieve a coordinate treatment.

The mechanism which we have used to learn is based on Learning Automata and Q 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 in which time or condition. 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.

To do so, at first we taught the agents with Q Learning method. In this phase we considered behaviors number 1 to 3 which mentioned above. We started our tests with 5 fire brigade agents and 3 fiery points. Q values of each behavior in each state were updated by formula (1) considering their rewards or penalties.

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

Here 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.

Learning had done in two phases: off-line and on-line. In off-line phase agents chose mentioned 3 behaviors randomly. After 50 simulations, agents got ideal results by choosing their behaviors with considering achieved Q vales. They could restrain fires spreading.

In on-line phase the number of fiery points increased up to 4 and the number of fire brigades increased up to 10. 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.

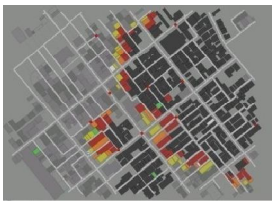

Fig. 5. Environment state before on-line learning.

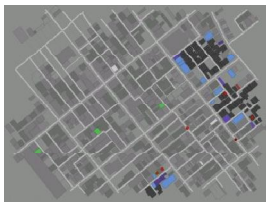

Fig. 6. Environment state after on-line learning

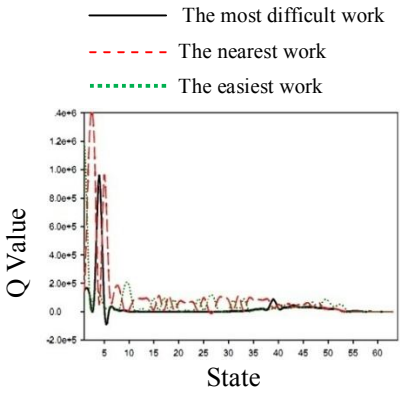

Fig. 7. Q values based on state after 50 off-line simulations. Q value in each state show the probability of selecting a behavior.

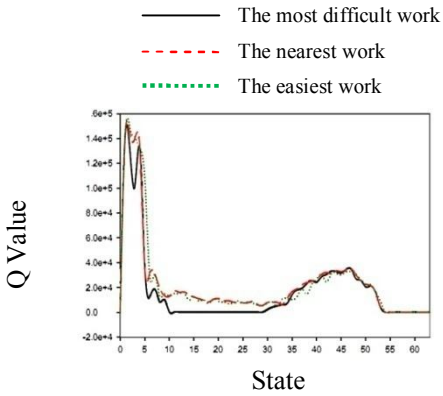

Fig. 8. Q values based on state after 50 on-line simulations. Q value in each state show the probability of selecting a behavior.

Diagrams 7 and 8 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. The following points were concluded from the 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 the simulation, the state of city is not very fatal so the agents choose the easiest work to prevent spreading fires.
2. From states 6 to 35, they never choose the most difficult work. This means that the agents learn 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 these 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 time 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.

In the next phase we evaluate Learning Automata (L_{rp}) and all defined behaviors. As a model for learning, Learning Automata act in a stochastic environment and are able to update their action probabilities considering the inputs from their environment, so optimizing their functionality as a result[6].

In RoboCup Rescue Simulation Environment, our first use of Learning Automata goes back to Persia 2005 [7].

A variable structure automaton is represented by a sextuple $\langle \alpha, \beta, \Phi, P, G, T \rangle$. In this sextuple, β is a set of inputs, Φ is a set of internal states, α is a set of outputs, P denotes the state probability vector governing the action chosen in each state at each stage k, G is the output mapping, and T is the learning algorithm. An Example of the variable structure type is *Lrp* automata that we summarize in the following paragraphs.

Let αi be the action (with index i) chosen at stage $k-1$ as a sample realization from distribution $P(k-1)$. So, the automata should update all of its actions (with index j) probabilities depending on its environment's response received at stage k. In *linear reward penalty algorithm* (L_{rp}) scheme, the recurrence equation for updating P is defined as (r is the number of actions):

$$P_j(k+1) = \begin{cases} p_j(k) + a\left(1 - p_j(k)\right) & if\ j = 1 \\ p_j(k)(1-a) & if\ j \neq i \end{cases} \quad (2)$$

If $\beta(k) = 0$ (i.e. reward received from environment) and

$$P_j(k+1) = \begin{cases} p_j(k)(1-b) & if\ j \neq i \\ \dfrac{b}{1-r} + (1-b)p_j(k) & if\ j \neq i \end{cases} \quad (3)$$

if $\beta(k) = 1$ (i.e. penalty received from environment).

The parameters a and b represent reward and penalty parameters, respectively. The parameter a (b) determines the amount of increase (decreases) of the action probabilities.

The number of states in a simulated robotic rescue domain is very large and therefore, for an agent to consider all of them is impossible. In fact, the most important job in this regard, is to design a proper generalization of the environmental state space for the agent. If we call the set of agent's actions A, each agent will have |A| possible actions in each of |V| states and therefore, the set to be learned for the agent will have at most V × A elements. If we choose the sets V and A wisely, our agents can learn effectively in a complex and real-time environment using limited samples. In fact, the sets V and A should have the property that cover all states and actions as much as possible and they should be good mappings of the sets of all possible states and ac-

tions that exist in the domain of agents[8]. For generalization of the environment we map the state of fire brigade agent in to 6 states as fallow:

State 1: 0% ~ 2% of buildings are burned
State 2: 2% ~ 5% of buildings are burned
State 3: 5% ~ 10% of buildings are burned
State 4: 10% ~ 20% of buildings are burned
State 5: 20% ~ 50% of buildings are burned
State 6: 50% ~ 100% of buildings are burned

By doing so, at any moment of the simulation, each agent is in one of these states. For each of these 6 states we use an Lrp learning automata. Each automata, has 8 actions. These actions are the mentioned 8 behaviors. We increased the agents' vision to able them to see the entire environment and their actions' results.
By the following pseudo code our fire brigade agents change their automata values:

```
if a civilian has burnt after handle a behavior
          give itself penalty
else if the number of fiery buildings after handle a
behavior were increased
          give itself penalty
else if the number of fiery buildings after handle a
behavior were decreased and no civilian has burnt
          give itself reward
else if civilians have not burnt after handle a beha-
vior
          give itself reward
```

In fact, our agents simply give itself reward if select a proper behavior in each state to prevent fire spreading and burning civilians. Similarly the agent gives itself penalty if a civilian was burnt or number of fiery buildings was increased.
After 50 off-line and on-line simulations, obtained results in fig.9 show that agents select proper behavior in each cycle.

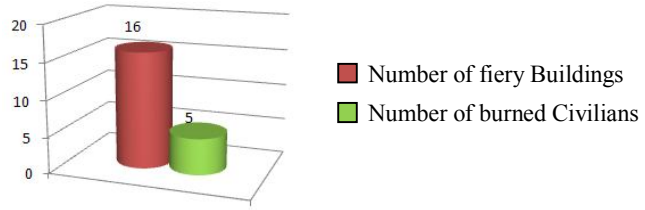

Fig. 9. Number of Fiery Buildings and burned civilians after learning agents

5. Conclusion and Future work

In this article we focused on cooperation in multi agent systems in dynamic and complex environments. By implementing intelligent behaviors, communication among agents decreased and if necessary, high level information be exchanged. So by decreasing communication, effective cooperation among agents is achieved. Obtained Results in these experiments show that agents by doing individual behaviors and using reinforcement learning, learned to present suitable behaviors in different conditions to reach team cooperation. We will use effective behaviors for our police force agents and ambulance agents to make cooperation among them.

6. References

1. P. Maes and R.A. Brooks, "Learning to Coordinate Behavior", Massachusetts Institute Of Technology.
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. Scerri, "Engineering Strategic Behavior in a Multi-Layered Behavior Based Agent", Department of Computer Science, RMIT, May 1997.
5. Z. Coa, B. Zhang, M. Tan, "Research of Behavior-Based Real time Formation Marching for Multiple Mobile Robots".
6. Narendra, K.S., Thathachar, M.A.L.: Learning Automata: An Introduction. Prentice Hall, Inc. (1989)
7. Khojasteh, M.R., Heidari H., Faraji, F.: Persia 2005 Team Description. Team DescriptionPaper (2005)
8. Khojasteh M.R.: Cooperation in multi-agent systems using Learning Automata. M.Sc. thesis. Department of Computer Engineering and Information Technology, Amirkabir University of Technology (2002)
9. Khojasteh, M.R., Faraji, F., Kazimi, A., Ghaseminik, Z.: Persia 2006, Towards a Full Learning Automata-Based Cooperative Team. Team Description Paper (2006)