Extending the BDI Model with Q-learning in Uncertain Environment

Qian Wan†
Hubei Province Key Laboratory of Intelligent Robot
School of Computer Science and Engineering, Wuhan Institute of Technology
Wuhan, China
1007831839@qq.com

Wei Liu
Hubei Province Key Laboratory of Intelligent Robot
School of Computer Science and Engineering, Wuhan Institute of Technology
Wuhan, China
liuwei@wit.edu.cn

Longlong Xu
Hubei Province Key Laboratory of Intelligent Robot
School of Computer Science and Engineering, Wuhan Institute of Technology
Wuhan, China
1561429775@qq.com

Jingzhi Guo
Hubei Province Key Laboratory of Intelligent Robot
School of Computer Science and Engineering, Wuhan Institute of Technology
Wuhan, China
2366137345@qq.com

ABSTRACT

The BDI model has solved the problem of reasoning and decision-making of agents in a particular environment by procedure reasoning. But in uncertain environment which the context is unknown the BDI model is not applicable, because in BDI model the context must be matched in plan library. To address this issue, in this paper we propose a method extending the BDI model with Q-learning which is one algorithm of reinforcement learning, and make an improvement to the decision-making mechanism on the ASL as a implement model of BDI. Finally we completed the simulation of maze on Jason simulation platform to verify the feasibility of the method.

KEYWORDS

BDI model, Agent, Q-learning, Jason, Plan Library

ACM Reference format:

Qian Wan, Wei Liu, Longlong Xu and Jingzhi Guo. 2018. Extending the BDI Model with Q-learning in Uncertain Environment. In *Proceedings of 2018 International Conference on Algorithms, Computing and Artificial Intelligence (ACAI'18). Sanya, China, 6 pages.*

†Corresponding author: 1007831839@qq.com

ACAI '18, December 21–23, 2018, Sanya, China

 ACM ISBN 978-1-4503-6625-0/18/12…$15.00
https://doi.org/10.1145/3302425.3302432

https://doi.org/10.1145/3302425.3302432

1 Introduction

The research on agents, acting in an uncertain and dynamic environment is a challenge, BD I [1] agents is designed for agent-orient programming model and build multi-agent systems. BDI model is concerned with agent's rule description and logic reasoning in multi-agent systems. However these two factors are based on context and must be designed in advance. Reinforcement learning [2] (RL) is applied to solve this problem that BDI agent don't know environmental model. RL assumes that an agent is using observed rewards that are perceived from the environment to measure its utility following its actions in an uncertain and dynamic environment. According to reward value, the agent can determine the sequence of actions though in uncertain environment.

BDI concepts used to describe people's behavior and intention at first, then was introduced into artificial intelligence, and the earliest BDI abstract model was put forward by Georgeff. On the basis of the model, different Procedure Reasoning System (PRS) are designed for reasoning. Based on PRS mode, the researchers developed the Multi-Agent System (MAS) based on BDI model including JACK [3], DECAF [4], IRMA [5], JADEX [6],ASL [7], etc. Because the ASL which has been added with the plan library on the foundation model of BDI has a simulation system and a better extension interface, we see ASL as a starting point for studying the BDI model. The plan problem of the agent in uncertain environment, so our research is aimed at improving the planning part of the ASL which is the implementation model of the BDI Agent.

RL is learning to map situations to actions so as to maximize a numerical reward. Without knowing which actions to take, the learner must discover which actions yield the most reward by trying them. Actions may affect not only the immediate reward but also the next situation and all subsequent rewards. Agent in

RL has the ability of learning and planning, but lacks the logic and reasoning ability of BDI Agent.RL can be divided into two types by the: model-based and model-free, model-free RL is mainly used to solve the planning of agents in situations where environmental information is not known, Q-learning is one of non-model algorithm that has high learning efficiency. Therefore, this study put forward the method which can solve the problem that BDI Agent can't decision under dynamic and uncertain environment by RL.

The structure of this paper is as follows. After this introductory section, we follow in Section II with a brief discussion of related works. Section III introduces BDI Agent and AgentSpeak(L), RL and q-learning algorithm. Section IV describes our method that decision improvement algorithm based on q-learning in ASL system. Section V describes the simulation experiment and evaluates and analyzes the results of experiment. Section VI summarizes our research and point to possible future developments.

2 Related Works

The inference mechanism in BDI model is based on the preset environment, so the BDI system lacks the planning under unknown environment. For the known environment model, there have been many methods are used to solve the planning in which including decision tree, self-aware neural network, and rules learning algorithm are used to optimize the Agent's decision. Pereira apply Markov decision process to generate the optimal strategy of BDI plan, but the method in unknown environments is difficult to build Markov environment model, so these methods do not apply in unknown environment information.

For the unknown environment model, the self-adaptive research of agent in the unknown environment of BDI model has been paid more attention. Google recently proposed a method that building the BDI model of Agent by deep reinforcement learning to understand the current real intention of Agent in order to improve its planning [8], J. L. Feliu propose to have an offline training session for plan generation in an uncertain and dynamic environment, the use of Q-learning from training sessions consisting of interactions between agent and environment generates plans for agents in ASL[9]. Although this method has solved the problem that agents know how to act when considering the efficiency of achieving goals in every state under uncertain environment, but when the state set is too large, quantity of generated plan will be explosive which lead to deviation that agents focus on the internal logic in original BDI model. Joost Broekens propose to solve the problem for selection of rules of which priority is learned by RL, and to use a state to represent independent heuristic rules on active targets [10]. However, it is difficult for agents to express the state in dynamic environment. Autonomous Agent mentioned in the literature [11], incentive signal of agent's behavior is given by the person. Supervised learning is used to generate action selector according to the excitation signal which finally enables the agent to perform the task efficiently in complex and unknown environment, this method enables the Agent to learn the human's perception and decision. But the human resource is consumed by the human being as the trainer.

Compared with the above method, this study avoids the unit of plan with the state as the plan and the decision- making in the unknown environment will become a plan which added to the plan library after the Q-learning algorithm has explored the environment. The plan is the unit after learning which avoids the problem of oversize planning library, and the state of agent is easy to define.

3 Background

3.1 BDI-Agent and AgentSpeak(L)

Programming paradigm based on agent gives higher intelligent computer software module and adaptive ability, BDI is an agent oriented programming model which is composed of Belief, Desire and Intention. The belief includes the environmental information acquired by the agent from the environment, the information of its own operation and the information received from other agents. The desire represents the possible state of the agent when performing task, which is driven by intention in the actual operation. The intention represents the state that the agent decides to achieve in the actual operation of the agent. There are two types of intention one of which is the target to be executed by the agent, and another is the plan based on the context matching in the BDI model. The BDI conceptual model is described as follows: agent initializes beliefs and intentions, then perceives environmental information. By belief update function, belief is update. According to the current intentions and beliefs, some desires are selected as a candidate, one of them can be selected as the execution plan through the matching rules that have been designed in advance. Agent finish the task by the plan. BDI model is a conceptual model in which the agent has basic logical reasoning ability.

ASL (Agent Speak Language)[12] is extended on the basis of BDI model, and the specific system structure diagram is shown in figure 1.

1. The agent target and belief library are initialized, the belief library is updated by the BRF (Belief Revision Function) according to the target agent perceives information from the environment or from other agents.
2. The agent cognitive the environment changes, and updates the belief library and verify whether the change triggers the event in the plan library. When multiple events are triggered simultaneously, the sequence of event execution is determined by the SE function.
3. According to the trigger event, the predicate symbol is matched in plan library. For example, if default trigger event in the plan *library* is *+color(Object,color)*. When agent perceives the blue box from the environment, the data will be transformed into expression *+color (box1,blue)[source(percept)]*, then after SE(Select Event) is used to match trigger event,box1 Object, blue Color. The applicable plan will be matched, and the value

of Object and Color is instantiated into box1 and blue entities in the plan.

4. Select a plan through the S_0 function which context is matched in the applicable plan, the plan structure is *trigger_event:context->body*, *body* content includes the action sequence, subtarget and child trigger event.
5. According to the context, such as *box1 = 1* m^2, context expression *Object > 0.5* m^2, and there may be subgoal which is matched similarly in body. The plans that conform to context are pushed into the stack of intent. The agent performs the action in turn to pop up the stack. When the stack is empty, the system enters the next loop.

AgentSpeak(L) builds a multi-agent system based on BDI model, which facilitates the programming and research of Agent. This study mainly focuses on the improvement of ASL which is the implementation model of BDI.

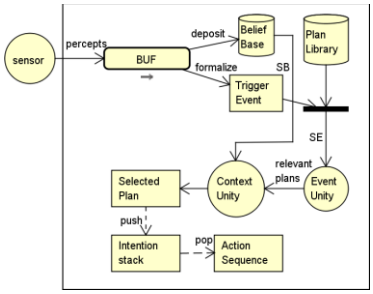

Figure 1: ASL Architecture

3.2 RL and Q-learning

Reinforcement learning is completely different from other machine learning algorithms such as supervised learning and unsupervised learning. It requires interaction with the environment and obtains "experience" from the environment. Most of the reinforcement learning adopts Markov model of which the most important feature is that the next step in the current state is not relevant to the past, RL can be described in the Markov decision process. The definition of Markov decision process is as follow $M = (S, A, P, r, \gamma, N)$, S is the state set of agent. A denotes the action set of agent. P represents the transfer probability between states, r is the immediate reward of a state transition. $\gamma \in [0,1]$ is a discount factor, N is the number of steps from the initial state to the final state. The cumulative reward denote with $R=\sum_{n=0}^{N} \gamma^t r_t$, The goal of reinforcement learning is to find the optimal strategy to maximize the cumulative reward value of the function which denote with $\max_{\pi} \int R(\tau) P_{\pi}(\tau)\, d\tau$, τ represents the behavior trajectory of the agents τ =(s_0,a_0,s_1,a_1,....).

Q-learning algorithm which is one of the RL is mainly used for agent decision learning under the model in which agent does not know the information of environment. Q-learning utilize the state-behavior function Q(s,a) to indicate the cumulative reward of the new state to the final state when the state s performs action a, and the maximum value of Q(s,a) is updated every time the state is passed. Its iterative update equation is given in equation (1):

$$Q_{k+1}(s_t,a_t) = Q_k(s_t,a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q_k(s_{t+1},a) - Q_k(s_t,a_t)\right) \quad (1)$$

$Q_{k+1}(s_t,a_t)$ represents the value of the updated cumulative reward obtained by performing the action at time t. $Q_k(s_t,a_t)$ represents the cumulative value of the last time the Agent passed the state. $\alpha \in (0,1)$, the larger α is, the later reward is more considered. r_{t+1} represents the value of the reward obtained when s_t performs action a to s_{t+1}. γ is the discount cumulative reward. $\max_a Q_k(s_{t+1},a)$ denotes the maximum value at s_{t+1} in Q table. Q-learning is a greedy strategy algorithm. The value of Q(s, a) tends to be stable after the finite iteration of formula (1), the maximum current state Q value in the Q table is selected until the final state is reached to form the optimal strategy.

4 The ASL Optimal Decision Algorithm Based On Q-learning

The ASL reasoning model is shown in figure 1. The context of an agent consists of trigger events and environmental information. But in an uncertain environment, there be no plan for matching context, another problem is that agent need to consider which plan is the best when multiple contexts are matched in plan library. In view of the above problems, this paper proposes an ASL optimal decision algorithm based on q-learning by which BDI agent improve the plan library in ASL. This study mainly improves the planning library from two aspects. (i) For multiple plan decision problems, the task is completed at each times, the cumulative reward value is recorded and compared with the latest plan. Choose the plan with the largest cumulative reward as the next execution plan. (ii) In the uncertain environment, agents explore the environment and accept the feedback of the environment. By using q-learning algorithm training, the best sequence of actions is completed and the plan is added to the plan library.

The detailed algorithm of ASL optimal decision improvement algorithm improvement details are shown in figure 2. In the initial environment model, because RL is based on markov decision model，you need to define the state s that the agent may exist, the corresponding reward value *r*, the plan library *Pu* and trigger event *E*.

ASL optimal decision algorithm based on q-learning:

Input：E /* E are initial trigger event */
P /* Pu are initial plans in library */
Su，rt /* Su is set of states，rt are reward */
output：π /* π is the agent's strategy */

1: if match(E,P)
2: N++; /* calculate number of matched plan */
3: if (N==1)

```
4:          push(plan) in stack
5:      end if
6:      if(N>1)          /* When matching multiple plans*/
7:      Random(plans)
8:      for each plan in plans
9:          if(R>Rmax)   /* The largest cumulative reward is selected */
10:         Rmax=R
11:         push(plan) in stack  /*Best plan is pushed into intention stack  */
12:         π =plan(Rmax)
13:         end if
14:   end if
15:   else
16:     for(N=0;State!=terminal_state&&N<limit(M);N++){
17:       /* M is the number of trial */
18:       Random(Actions);
19:       update(Q-table)     /* Q values in the q table are updated.*/
20:       state=next_state     /* Next state is the result of random actions*/
21:       π =plan(Rmax in Q-table) }
22:       plan library.add( π )   /*Add the policy to the plan library  */
23   end if
```

Fig. 2 The ASL Optimal Decision Algorithm based on Q-Learning.

1. When an external event is triggered as the target event E, E matches plans P which has own E as precondition in the initial plan library.
2. If the match between E and P is successful, selector function will select the applicable P, and if there is a unique P, it will be put directly into intent stack. if there are multiple P matched, agent randomly selects P after execution and calculates cumulative reward, the next execution time agent choose (usually in the form of more than 90%) current known optimal choice, or randomly select the P which has not been executed, and record cumulative reward, if the value of cumulative reward greater than Rmax compared with Rmax, Rmax and P will be recorded.
3. If E does not match P, the agent will explore the unknown environment and record the value Q(s,a) in Q table of the state-action function in each state. If within the limited time t, agent fail to reach the final state, intention is supposed to be unable to achieve. If inside limited t, agent reach the final state, Q(s,a) will be recorded and updated by Q table, the above exploration process is repeat N times with the state of Q (s, a) which the larger value update. In a limited time after exploration, the Q table can be concluded optimal strategy which will be added to the plan library.

In the improved model, the best plan is selected when the agent matches multiple plans, so the execution efficiency is improved. When there is no plan to match in unknown environment, agent can get the "experience" after a limited number of trials. Finally according to the "experience", the agent carrys out the optimal plan. By this method the problem that BDI agent is unable to perform a task in unknown environment is solved. And one plan is added to the plan library at one time when faced with a task, so the amount of plan will not too large.

5 Perimental Simulation And Evaluation

In order to verify the validity of the ASL optimal decision algorithm which based on Q-learning in an uncertain environment, Jason [13]simulation platform which has been designed for multi-Agent system is used for scenarios of the maze. The maze has two characteristics that are suitable for the experiment (i) agent has obvious state division. (ii) agent do not know what the world model and need to explore environment and get the optimal plan.

Maze simulation as shown in figure 3, agent need find a optimal strategy which is added to plan library under the condition of unknown environment information. When next time the same task triggers, the decision system in ASL model can directly call the plan in the library. R1 represents the position of the agent, and R (the reward value) represents the stimulus signal of the environment to the Agent, and the R value which usually determined by a function is set by the person. G represents an obstacle. Its R is set to -10. R_2 represents the exports of maze which the value is set to +10. Other states (each grid denote a state) because of the shortest path is the best plan, each additional state, the cumulative reward value minus one. so the white grid state of the reward value R is set to -1.

Fig. 3 Maze Simulation Scenario

The core code of the ASL optimal decision algorithm is as follows:

```
1: +!update1(St,R,A) : A\==null&&below_limit(M)&&
   non_terminal_state(St)<-
2: ?last_state(N);
3: ?state(N,pos(_,St1,R1));
4: ?utility(St1,Q1,N1);
5: M1=N1+1;
6: ?utility(St,Q,_);
7: Q2=Q1+0.8*((R1+0.9*max*Q-Q1));
8: -utility(St1,_,_);
9: +utility(St1,Q2,M1).
```

Line 1 *+!Update1* represents the trigger event of which target is to update each status. The following statement "*:*" represents

the execution condition including the action which is not empty, the number of iterations which is less than *M*, and the final state *R2* which is not reached. The line 2-6 represents the retrieval information, "*?*" denote retrieved information including state, the number of iteration of the state updates *N*, *N* is less than *M*, *N1* denote the number of each state that have been updated, *R1* is the executive reward value when agent perform action *A* and *Q1* is the cumulative reward value which has been calculated at the last time the agent passed , *St* indicates that *St1* is transferred to *St* state when performing action *A*, *Q* represents the possible set of all state – action pairs which also has been recorded in Q table at *St*. Line 7-9 indicates that the *Q* value of state *St1* is updated and stored with the q-learning algorithm. Considering the faster learning speed for agent, the learning rate α is set to 0.8 and the long-term benefit γ is set to 0.9. Then the new *Q2* which replace the *Q1* and the time *M1* the *St1* have been passed at St1 is stored in the utility. Finally the optimal strategy can be selected according to the value of Q2 in the utility.

With *R1* as the origin (0,0), the best strategy of *R1* to *R2* can be obtained after 1000 iterations. Take as an example the agent in coordinates (1,1)-specify the fifth state，the change of Q value results as shown in figure 4. (5, _), Q(5, 0), Q(5, 1), Q(5, 2), and Q(5, 3) respectively represent the action of up, down, left and right at the coordinate (1,1). After 1000 iterations, the utility (state-action pair) convergent with Q(5, _) is 5.3, -4.6, 0.8, 14.3, The maximum value of 14.3 of Q(5,3) is selected which denote agent go right at (1, 1). So on and so forth for other states, the best action of each state can be obtained as shown in figure 3 which describe the maze simulation. Each state of agent can take best action which has *Q* value compared by arrows. The number n of optimal strategy can be obtained because the Q value of the same article n strategy, randomly one of strategies is chosen. In this experiment, we choose the sequence number of the arrow mark is 1-5 as the best plan in Fig 3. The strategy of the target state as the planned goal of triggering event, the initial state is the context of the plan, the action execution sequence in plan library is formed of policy action sequences, and the strategy is added to the library. The generated plan is coded as follow. When trigger event Out_Maze[position(X,Y)] is encountered again and the context which has retrieved is position(0,0), agent will directly perform the sequences of action.

The generated plan of the ASL optimal decision algorithm is as follows：

```
1: +! Out _Maze[position(X,Y)] 2:: X=0&&Y=0
3:          <- move_right;
4:              move_down;
5:              move_down;
6:              move_right;
7:              move _down.
```

In original ASL system, the agent cannot learn "experience" from the interaction with the environment in the process of exploring target, the "experience" can be prior knowledge of the next exploration so as to find the best plan. And because there is no optimization, the time that agent arrives at the target is also uncertain. In this study, q- learning was used in ASL system has solved the decision problem of BDI Agent in unknown environment, and the agent under the system can find its optimal plan in a short time in order to better preparation for the next same task. Therefore, this experiment proves that the improvement of decision algorithm in the ASL system based on q-learning enables agents to have better decision-making power in dynamic and uncertain environments.

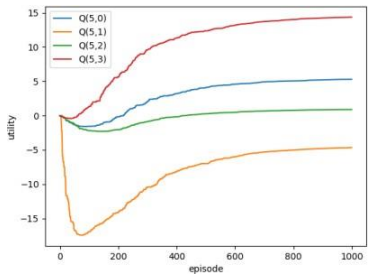

Fig. 4 Coordinate (1,1) State-Action the Cumulative Return Value after the Iteration

5 Conclusion and Future Works

This study aimed at BDI model in which the agent fail to make a decision in an uncertain environment, we put forward a new decision algorithm that mainly improves the planning mechanism of ASL, In uncertain environment, agent approach the best strategy by explore and recording the interaction evaluation value of the environment, and adding the action sequences correspond to the best strategy into the planning library of the decision-making system. The simulation results prove that the ASL system after the new algorithm is added, the Agent makes the best decision after a certain attempt in uncertain environment.

Because of the new algorithm combines the Q-learning, the value of Q in the iteration may result in a slow convergence of the final result due to the too large number of states and actions that lead to the slow operation of ASL system. Another problem is exploration of uncertain environment is likely to cause irreparable damage to the Agent such as impact damage. In the case of slow convergence, we will consider adding deep learning methods in the future to speed up the convergence rate, and consider adding more cautious learning algorithms such as Sarsa to avoid the damage of Agent.

REFERENCES

[1] Morreale, Vito, et al. "Goal-Oriented Development of BDI Agents: The PRACTIONIST Approach." Ieee/wic/acm International Conference on Intelligent Agent Technology IEEE, 2006, pp.66-72.

[2] Sutton R S, Barto A G. Reinforcement Learning: An Introduction, Adaptive Computation and Machine Learning. Cambridge, MA, :A Bradford Book. The MIT Press,1998.

[3] Ancona, Davide, and V. Mascardi. "Coo-BDI: Extending the BDI Model with Cooperativity." Lecture Notes in Computer Science, 2003, pp.109-134.

[4] Mcgeary, Foster, and K. Decker. Modeling a Virtual Food Court Using DECAF. Multi-Agent-Based Simulation. Springer Berlin Heidelberg, 2000, pp.68-81.

[5] Burgemeestre, Brigitte, J. Hulstijn, and Y. H. Tan. "Towards an Architecture for Self-regulating Agents: A Case Study in International Trade." Lecture Notes in Computer Science 6069,2009, pp.320-333.

[6] Pokahr, Alexander, L. Braubach, and W. Lamersdorf. Jadex: A BDI Reasoning Engine. Multi-Agent Programming. Springer US, 2005, pp.149-174.

[7] Bordini, Rafael H., J. F. Hübner, and M. Wooldridge. "Programming Multi-Agent Systems in AgentSpeak using Jason." 2007, pp. 39-67.

[8] Rabinowitz, Neil C, et al. "Machine Theory of Mind." (2018).

[9] J. L. Feliu. (2013) Use of reinforcement learning (rl) for plan generation in belief-desire-intention (bdi) agent systems. master thesis. paper 160. University of Rhode Island, US. [Online]. Available: http://digitalcommons.uri.edu/theses/160/

[10] Broekens, Joost, K. Hindriks, and P. Wiggers. Reinforcement Learning as Heuristic for Action-Rule Preferences. Programming Multi-Agent Systems. Springer Berlin Heidelberg, 2010 pp.25-40.

[11] Li, Guangliang, et al. "Social interaction for efficient agent learning from human reward." Autonomous Agents and Multi-Agent Systems 32.1(2018):1-25.

[12] "Jason: a java-based interpreter for an extended version of agentspeak." [Online]. Available: http://jason.sourceforge.net/

[13] Habib, Arafat, M. I. Khan, and U. Jia. "Optimal route selection in complex multi-stage supply chain networks using SARSA(λ)." International Conference on Computer and Information Technology IEEE, 2017.