2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)
Dec 20-22, 2013, Shenyang, China

# The Method Based on Q-Learning Path Planning in Migrating Workflow

Song Xiao
School of Computer Science and Technology
Shandong University
Jinan, China
xiaosong2008@yeah.net

Xiao-lin Wang *
School of Computer Science and Technology
Shandong University
Jinan, China
xlwang@sdu.edu.cn

***Abstract*—In a goal-oriented migrating workflow management system, each migrating instance is regarded as a mobile agent, the path planning for migrating instance is the path planning for mobile agent. The migrating workflow path is an ordered set of working positions that can achieve the sequence of goals carried by mobile agent. How to plan out a most efficient and most rational migrating path is one of the problems needs to be solved in the research of migrating workflow. This article puts forward a method that in a based-on social acquaintance network environment, mobile agent dynamically plan out a migrating work path by reinforcement learning. This method is suitable for the target-oriented migrating workflow management system, which can well solve the problem of the migrating path planning in the uncertain or partially observable environment of mobile agent.**

***Keywords—reinforcement learning; migrating path; mobile agent; social acquaintance network***

## I. INTRODUCTION

With the development of computer and network, the study of the migrating workflow management system draw people's much attention. During the implementation process of the mobile agent-based workflow, mobile agent is an agent who continuously migrates between different working positions and exploits local services to achieve the business goals. When a subgoal is completed, mobile agent determines the next working position on the basis of the current state and the next sub-goal. Working position provides services for mobile agent, including the runtime environment, runtime services and workflow services etc. Therefore, mobile agent finds out a migrating path through path-planning, and orderly achieves the sequence of the business goals.

Reference [1] introduced a agent-based migrating workflow management system, but its implementation depends on the workflow model, which has some deficiencies such as immobility structure, poor scalability and the lack of good adaptability in the complicated and fickle network environment etc. Literature [2] built up a migrating path planning model based on the navigation tree for mobile agent, which agent only finds these migrating locations according to the navigation station's advice, and meanwhile highly improve the system efficiency. However, in the large and complex network environment, the costs of knowledge organization, sharing and maintenance between the navigation stations are larger and lack of flexibility. Literature [3] proposed a goal-oriented path planning migrating workflow optimization algorithm, through optimizing the AND-OR graph, that significantly improve the execution efficiency of mobile agent, however, with the changing of environmental conditions, mobile agent cann't adaptively change the migrating path. Literature [4] proposed a greedy particle swarm optimization algorithm for workplace planning in migrating workflow. Whereas the entire environmental status information is opaque to mobile agent. Consequently, the needs to the interaction, updating in real time between the two workplace cause many difficulties, such as the high cost to sharing and maintaining the entire environment, the larger network overhead and the poorer scalability etc. For these reasons, this algorithm can't solve the problem of the path-planning for mobile agent in an uncertain or partially observable environment.

In the practical applications, because of the complexity, flexibility and uncertainty of network environment, it is very difficult to plan out a most effective and most reasonable migrating path for mobile agent. Reinforcement learning(RL) is a computational approach to understanding and automating goal-directed learning, which is distinguished from other computational approaches by its emphasis on learning by the individual from direct interaction with its environment, without relying on exemplary supervision or complete models of the environment[5]. Reinforcement learning uses a formal framework defining the interaction between a learning agent and its environment in terms of states, actions, and rewards. The working path planning of mobile agent combined with reinforcement learning , can satisfy the needs that mobile agent solve the path-planning problem in a uncertain or partially observable environment. Therefore, this article puts forward a method of dynamical based-on reinforcement learning workplace path programming for mobile agent in a environment of the social acquaintance network, which can well solve the path-planning problem in an uncertain or partially observable environment.

## II. SOCIAL ACQUAINTANCE NETWORK

In the realistic society, every social member has directly built relationships with the acquaintances by their peers or partnership. By the way of the acquaintance's relationships, all of social members formed a social acquaintance network[6].

*Corresponding Author

This research is partially supported by National Science Foundation of China (Grant No. 61173068), Natural Science Foundation of Shandong Province of China (Grant No. ZR2009GM021) and DNSLAB, China Internet Network Information Center (K201206007)

978-1-4799-2565-0/13/$31.00 ©2013 IEEE

*Definition1* Social Acquaintance Network(SAN) is a directed graph, and constitutes a social network by all of the acquaintance's relationships, which we denoted by $SoANet = (V, E)$, where, $V$ is the set of vertices, each vertex $v \in V$ represents a social member(SM); $E$ is the set of directed edges, each directed edge $e_{i,j} \in E$ connects the vertice $v_i$ to another vertice $v_j$, that represents the social member $v_i$ is a acquaintance of $v_j$.

*Definition2* Any one of the social members $v_i$ has fall within the scope of its social acquaintances, which is referred to as Social Acquaintance Group(SAG), we write $SAG_{v_i}$. For instance, Fig. 1 shows the example of the structure of the social acquaintance network, for social member $v_1$, its social acquaintance group is $SAG_{v_1} = \{v_2, v_4, v_7\}$.

In the social acquaintance network, any one of the social members provides a set of services for mobile agent, and in which mobile agent achieves a subgoal by executing a service.

*Definition3* For one of social members $v_i$, the set of its available services is recorded as $S_{v_i}$, then $S_{v_i} = \{s_1, s_2, \ldots, s_n\}$, in which $s_i$ is a service, n is the total number of services for a social member $v_i$. So the set of services in the entire social acquaintance network is $S_{SAN} = S_{v_1} \cup S_{v_2} \cup S_{v_3} \cup \ldots \cup S_{v_m}$, where m is the total number of social members.

*Definition4* Suppose $g_i$ is the goal by completing a service $s_i$ for a social member $v_i$. Then, to one of social members $v_i$, the corresponding set of its goals to available services $S_{v_i} = \{s_1, s_2, s_3, \ldots, s_n\} \subset S_{SAN}$ is $GS_{v_i} = \{g_1, g_2, g_3, \ldots, g_n\}$. The goal set of all members in the entire network is $GS_{SAN} = GS_{v_1} \cup GS_{v_2} \cup GS_{v_3} \cup \ldots \cup GS_{v_m}$, where m is the total number of social members.

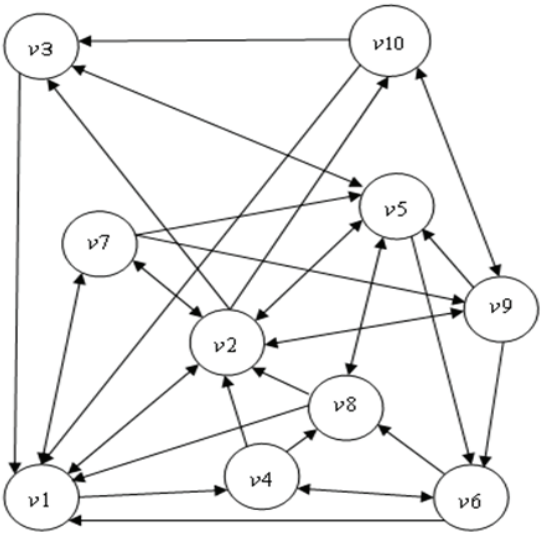


Figure 1. Show the structure of the social acquaintance network

## III. PROBLEM OF PATH PLANNING

In the environment of SAN, every social member can offer service for mobile agent, including the runtime environment, runtime services and workflow services etc. Due to these influences, such as the service ability of social members and the service efficiency, mobile agent needs the migrating operations between social members, in order to achieve more efficient, higher-quality services.

*Definition5* Suppose $G$ is a sequence of the workflow business goals achieved by mobile agent, and consists of s different goals orderly, i.e. $G = (g_1, g_2, \ldots, g_s \mid g_i \in GS_{SAN}, 1 \le i \le s)$.

*Definition6* Assume that in the process of migration of mobile agent, for each social member, the costs of the service completing a goal are different. Let $C_{serv}(v_i, g)$ be the service cost of completing the goal $g$ for the social member $v_i$. Meanwhile, assume that the transmission costs among all of social members are nonidentical. Let $C_{tran}(v_j, v_i)$ be the transmission cost of moble agent who migrate from the social member $v_j$ to $v_i$.

The path planning problem for mobile agent is that, in a given environment of SAN, according to the business goals G,mobile agent find out a path $P: v_{r_0} \to v_{r_1} \to v_{r_2} \to \cdots \to v_{r_p}$. Mobile agent migrates along this path to sequentially realize the goal sequence $G$ with the minimum of the sum of $C_{serv}$ and $C_{tran}$, that is said to be an optimal path $\pi_P$, where $v_{r_0}$ is the initial position.

Therefore, path planning for mobile agent is to find out a optimal path $\pi_P$. The objective function of solving the problem of path planning is

$$Min \sum_{i=0}^{|G|} \left( C_{tran}(v_{r_i}, v_{r_{i+1}}) + C_{serv}(v_{r_{i+1}}, g_{r_{i+1}}) \right) \tag{1}$$

having all of the following characteristics:

a) $v_{r_{i+1}} \in SAG_{v_{r_i}} + v_{r_i}$;

b) Mobile agent migrating along the path P sequentially realizes the goal sequence G.

## IV. METHOD OF PATH PLANNING BASED-ON Q-LEARNING

### A. Related concepts

Reinforcement learning[5] is an unsupervised learning method, the environment of feedback should be considered as a reinforcement signal, to choose the optimal action by learning, commonly, describing it using Markov Decision Process(MDP) model[7]. Traditional MDP model is defined as a 4-tuple $< S, A, T, R >$, in which $S$ is a finite set of states; $A$ is a finite set of actions; $T$ is a state-transition probability function; $R$ is the reward function.

An solution of MDP is known as an strategy $\pi$, maping from the states set to the actions set, namely, $\pi: S \to A$. For any a viable sequence of the strategies, we hope that mobile agent maximizes the long-term profits according to certain criteria to selecting action, i.e. $\max E[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t))]$, where $R(s_t, \pi(s_t))$ is the reward in step t; $\gamma \in (0,1)$ is the discount factor, ensure that the total expected revenue for mobile agent is convergent.

Q-learning is a form of model-free reinforcement learning[5]. The expected return performing an action $a$ in a state $s$ is recorded as $Q(s, a)$. The solving function for optimal strategy is

$$\pi(s) = \max_a Q(s, a) \tag{2}$$

In Q-learning, the transition function isunknown, the expected profit is acquired by finite loop iteration. Iteratively, the Q values are updated by ( 3).

$$Q_{t+1}(s,a) \leftarrow (1-\rho)Q_t(s,a) + \rho[R(s,a) + \gamma\, max_{a'}\, Q_t(s',a')] \quad (3)$$

in which $\rho$ is the learning rate; $\gamma$ is the discount rate; $t$ stands for the number of iterations.

### B. Model of Path Planning Based-on RL

Reinforcement learning is a kind of target-oriented learning, and learn by directly interacting with the environment to find out an optimal action strategy. In a SAN, mobile agent migrate between social members who provide different services, combined with reinforcement learning, and form a work path. So the basic thought of path-planning based-on RL method is that, due to the lack of ability to the current work position, mobile agent chooses a migration work position from the SAN. When accepted the migrating action, state change at the same time produce a strengthening feedback signal to the mobile agent.Mobile agent choose the next action according to the reinforcement signal and the condition of the current environment, until achieve all goals.Repeating the above processes, getting a strengthening signal pathis an optimal path.

*Definition7* Let $S = \{s_1, s_2, s_3, \dots, s_{|V|\times|G|}\}$ be set of states, in which V is the social member set, G is the goal sequence. Each state $s \in S$ is composed of a two-dimensional vector, for example, $s = (v_i, g_j)$, means that mobile agent will achieve the goal $g_j \in G$ in the social member $v_i \in V$.

*Definition8* In social acquaintances in the network, mobile agent migrates along the adjacent members of the current member, namely,the social acquaintance group of the current member. Mobile agent maybe migrates next work position, that constitutes a set of next actions $Action(s) = \{a_{v_s} | v_s \in SAG_{v_i}\} + a_{v_i}$ in the state $s = (v_i, g_j) \in S$. For example, in the fig. 1, the set of next actions for mobile agent may be $Action(s) = \{a_{v_1}, a_{v_2}, a_{v_3}, a_{v_5}, a_{v_7}, a_{v_9}, a_{v_{10}}\}$ in a state $s = (v_2, g)$.

In the paper, we will consider the design of the reward function from two aspects: the computational cost of social members and the transmission cost of mobile agent.

According to the definition of state and action, mobile agent takes the next step $a = a_v \in SAG_u$ in the state $s = (u, g)$, whose reward is that mobile agent reckon in the cost of completing goal from the social member u to the next social member v, namely,

$$R(s,a) = R(s = (u,g), a = a_v \in SAG_u) = \begin{cases} \vartheta \frac{1}{C_{serv}} + \beta \frac{1}{C_{tran}} & u \neq v, g \in Gw_v \\ \vartheta \frac{1}{C_{serv}} & u = v, g \in Gw_v \\ \beta \frac{1}{C_{tran}} & u \neq v, g \notin Gw_v \end{cases} \quad (4)$$

where $\vartheta$,$\beta$ are the coefficients.

### C. State Updating

With the changing of environmental conditions, mobile agent makes constantly decision for migrating work path to complete different services among social members. Algorithm 1 below gives the process of the state-updating. And if the goal and the position of mobile agent are both changed, update the current state according to step1; If the goal of mobile agent is changed and the position isn't, update the current state according to step2; If the position of mobile agent is changed and the goal isn't, update the current state according to step3. It should be emphasized, however, that it does not exist the migrating process that the goal and the position of mobile agent aren't both changed, which goes against the principle of migrating workflow.

*Algorithm1*: Updating State

Input: current state $s = (u, g)$, the next goal $g'$and next action $a_v \in Action(s)$

Output: the updated state $s'$

1. if $u \neq v$ and $g \in GS_v$ then $s' \leftarrow (v, g')$;
2. if $u = v$ and $g \in GS_v$ then $s' \leftarrow (u, g')$;
3. if $u \neq v$ and $g \notin GS_v$ then $s' \leftarrow (v, g)$;

Figure 2. State updating algorithm

### D. Action Selecting

Reinforcement learning is a kind of trial and error learning, the local best action may not be the global best. So, in the paper, we use a soft strategy method[8] to select actions.

*Definition9* $P(a|s)$ is the probability of taking an action a in the state $s$, according to Boltzman formula:

$$P(a|s) = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_{b\in Action(s)} e^{\frac{Q(s,b)}{\tau}}} \quad (5)$$

where $\tau$ is the temperature parameter.

*Algorithm2*: Selecting Action

Input: *current states* $= (v, g)$, all $Q(s, a|a \in Action(s))$and next probable action set $Action(s) = \{a_v, a_{v_{s1}}, a_{v_{s2}}, \dots, a_{v_{sk}} | v_{si} \in SAG_v,\ 1 \le i \le k\}$

Output: the selected action $a \in Action(s)$

1. initiate $\tau$;
2. if $g \notin GS_v$
3. calculate all $P(a|s)$ using (5) where $a \in Action(s) - a_v$;
4. $a \leftarrow$ randomly select an action according to all of $P(a|s)$ from $Action(s) - a_v$;
5. if $g \in GS_v$
6. calculate all $P(a|s)$ using (5) where $a \in Action(s)$;
7. $a \leftarrow$ randomly select an action according to all of $P(a|s)$ from $Action(s)$;

Figure 3. Action selection algorithm

Algorithm 2 gives the action selecting algorithm. In the algorithm, firstly initialize the temperature parameter ; And then determining whether social member can achieve goal, if not, select next action on step 3 and 4; Otherwise, according to step 6 and 7 choose the next action.

### E. Path-planning Algorithm

The remainder of this article focuses on providing details about the based on Q-learning path-planning algorithm.

In Q-learning, $Q(S, Action)$ is a function maping from state-action pairs to the value by learning, mobile agent chooses next action according to the Q values. Therefore, based-on Q-learning path-planning method is divided into two stages: 1) The learning phase, constantly updating Q value by finite iterations; 2) The searching optimal-path phase, mobile agent find out an optimal migrating path in the light of Q values obtained by the learning phase.

In the learning phase, mobile agentconstantly update Q values by finite iteration. The process of learning Q values is divided into two steps: the first step is one-episode learning. For instance, at the time t, mobile agent migrates from the current state $s_t$ by decision-making mechanism (Algorithm 2) to select the next action. And then updating Q values according to (3), mobile agent moves to the next workplace in the state $s_{t+1}$ at the time $t+1$. Repeating circularly above the process, finally agent get a work path from the initial state to the final state. Fig. 4 shows the process of the one-episode learning. In the figure, the migrating path along the direction of the black arrow is acquired by one-episode learning. The second step is the iteration of episodes, mobile agent chooses interactively next action and updates Q values through the multiple episodes learning.

The next stage is the searching optimal-path phase. In this phase, mobile agent find out an optimal migrating path in the light of Q values obtained by the learning phase. At the stage, in comparison with the process of the learning phase, The difference is that the next action is selected not by the algorithm 2, but by (2). By the greedy algorithm, for each step, the next action select from the set of possible actions who has the greatest Q value.

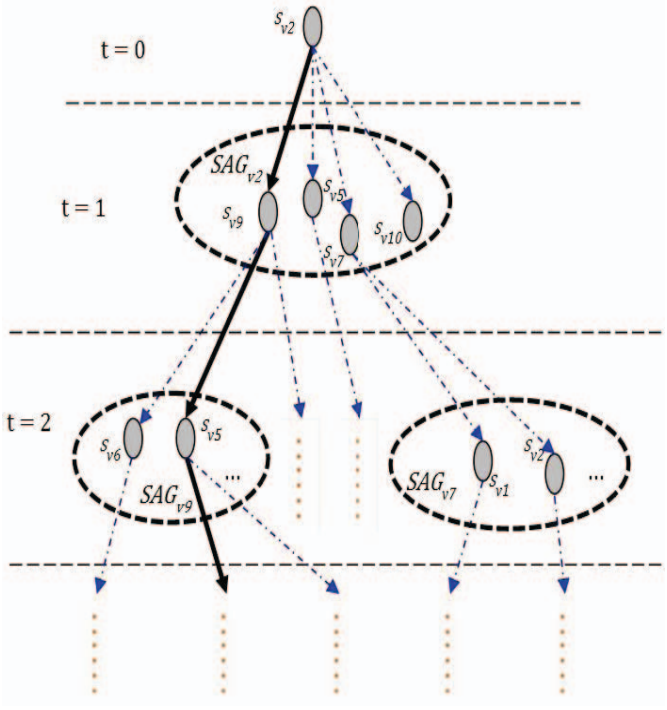


Figure 4. The process of one-episode learning

Fig. 5 shows the path planning algorithm based on Q-learning for mobile agent.

*Algorithm3*: path-planning (PP)

Input: $SoANet$ // the structure of SAN;
$GS_{SAN} = \{g_1, g_2, g_3, \dots, g_n\}$ // The goals set of SAN;
$G = \{g'_0, g'_1, g'_2, \dots, g'_m\}$ //the goalSequence;
$s_0 = (v, g_0)$ // the original state $s_0$;
Output: the path policy $\pi_P$

//learning Q value;
1. The state space $S$ defined by $SoANet$ and $G$ and initiate arbitrarily all $Q(s, a)$ values, $s \in S, a \in Action$;
2. Repeat (for each episode):
   current state $s \leftarrow s_0$;
3. Repeat (for each step in the episode)
4. i. Select an action $a$ according to Algorithm2;
5. ii. Taking the action $a$ in state $s$, calculate immediate reward $R(s,a)$ using(4), and then get the next state $s'$ according to Algorithm1;
6. iii. Update $Q$ value using (3);
7. iv. $s \leftarrow s'$;
8. Until all of the goal in the set of $G$ realized;
9. Until the desired number of episodes have been investigated;

//Finding the Path
10. current state $s \leftarrow s_0$;
11. Repeat
12. finding next action $a$ according to $Q$-value using (2) and then get the next state $s'$ according to Algorithm1;
13. $s \leftarrow s'$;
14. Until all of the goal in the set of $G$ realize;
15. Path policy $\pi_P \leftarrow$ the Sequence of $a$;

Figure 5. Based on Q-learning path planning algorithm for mobile agent

## V. SIMULATION EXPERIMENT

We have already realized the based on Q-learning path planning algorithm for mobile agent on matlab programming platform. During implementation, firstly, constructing three of the SAN respectively consist of 500, 1000 and 2000 social members. Mobile agent carries out the dynamic work path planning algorithm to find out a optimal migrating path. In the process, the following parameters are set, the learning rate $\rho = 0.9$, the discount rate $\gamma = 0.9$, the goal sequence $G = \{g_1, g_2, g_3, g_4, g_7, g_6, g_8, g_5, g_9\}$ and the initial Q value=0.

At the beginning, mobile agent is located in the social member $v_5$, i.e., the initial state $s_0 = (v_5, g_1)$. Mobile agent orderly complete the goal sequence G to obtain the migrating path strategies in the three of the SAN respectively. In the SAN with 500 social members, the optimal migrating path is $\pi_{500} = v_5 \rightarrow (v_2|g_1) \rightarrow (v_5|g_2) \rightarrow v_2 \rightarrow (v_{10}|g_3) \rightarrow (v_{10}|g_4) \rightarrow (v_{20}|g_7) \rightarrow v_{50} \rightarrow v_{124} \rightarrow (v_{310}|g_6) \rightarrow v_8 \rightarrow v_{52} \rightarrow v_{59} \rightarrow v_{12} \rightarrow (v_{11}|g_8) \rightarrow (v_{419}|g_5) \rightarrow v_2 \rightarrow (v_5|g_9)$, that is, mobile agent migrates from member $v_5$ to $v_2$ and achieves $g_1$ at the work member $v_2$, along the migrating strategy $\pi_{500}$, then ultimately achieve all of the goals.

In order to validate the efficiency of the algorithm 3, mobile agent plans respectively the migrating path in the three of the SAN with 500, 1000 and 2000 social members. And the variation of Q values as shown in fig. 6.

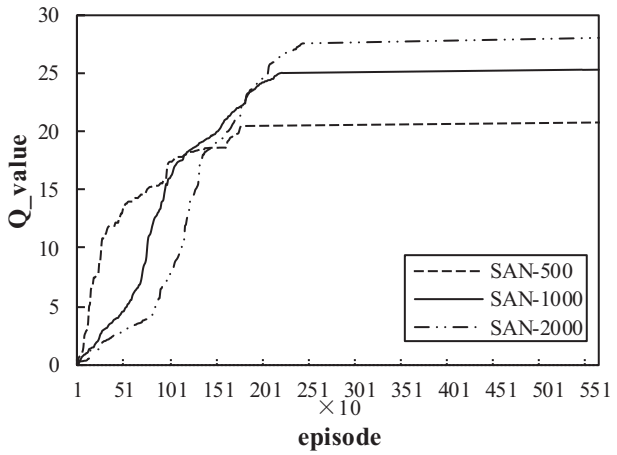


Figure 6. The variation of the Q_value in the three of the SAN with 500, 1000 and 2000 social members

From fig. 6, we can find that the Q_value is convergent after about 1700 iterations for mobile agent in the SAN with 500 social members(SAN-500), about 2200 iterations in SAN-1000, about 2400 iterations in the SAN-2000. So this algorithm of solving the migrating path planning problem in a large scale, complex network has higher efficiency.

## VI. CONCLUSIONE

In a goal-oriented migrating workflow management system, each migrating instance is regarded as a mobile agent, the path planning for migrating instance is the path planning for mobile agent. In this paper, we propose a method of dynamical work path programming based-on reinforcement learning for mobile agent in a environment of the social acquaintance network. This method makes mobile agent easy to find out a optimal migrating path by autonomous learning in a uncertain or partially observable environment. However, this method has some problems: 1) One is that in the learning phase, how to choose a next action for mobile agent is difficult. Designing a good action-selecting algorithm can improve the convergence speed of this algorithm, and thus find out a optimal migrating path as soon as possible; 2) The other is that in the process of path planning, if there maybe has a ring, mobile agent will repeatedly choose the next work position within the ring, which slows down the convergence speed of learning algorithm, thereby influences on the solving speed. Our next jobs are to study and solve these problems and explore to apply reinforcement learning to the dynamic migrating path programming for mobile multiagent.

## REFERENCES

[1] Zeng G, Dang Y. The Study of Migrating Workflow Based on the Mobile Computing Paradiam[J]. Chinese Journal of Computers, 2003, 26(10): 1343-1349.

[2] Cheng J, Zeng G,He H. Research on Migrating Instance Path Planning in Migrating Workflow System[J]. Journal of Frontiers of Computer Science and Technology, 2008, 2(6): 658-665.

[3] Zhang J,Zeng G,Han F. Path Search and Optimization in Goal-oriented Migrating Workflow[J]. Application Research of Computers, 2008, 25(9): 2623-2624.

[4] Cheng J, Zeng G. A Greedy Particle Swarm Optimization Algorithm for Workplace Planning in Migrating Workflow[C]. Natural Servputation,2008. ICNC '08. 2008, 6: 407-411.

[5] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge: MIT press, 1998.

[6] Boguñá M, Pastor-Satorras R, Díaz-Guilera A, et al. Models of social networks based on social distance attachment[J]. Physical Review E, 2004, 70(5): 056122.

[7] Puterman M L. Markov decision processes: discrete stochastic dynamic programming[M]. Wiley-Interscience, 2009.

[8] Tokic M. Adaptive ε-greedy exploration in reinforcement learning based on value differences[M].KI 2010: Advances in Artificial Intelligence. Springer Berlin Heidelberg, 2010: 203-210.