Chinese Journal of Electronics
Vol.19, No.3, July 2010

A Bayesian Reinforcement Learning Algorithm Based on Abstract States for Elevator Group Scheduling Systems*

CHENG Yuhu, WANG Xuesong and ZHANG Yiyang

(*School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China*)

Abstract — In order to solve the curse of dimensionality problem encountered by reinforcement learning algorithms for Elevator group scheduling (EGS) systems with large-scale state space, a kind of Bayesian reinforcement learning algorithm based on Abstract states (BRL-AS) was proposed. On one hand, an abstract state space whose size is much smaller than that of the original state space was constructed by analyzing the motion situations of EGS systems. On the other hand, a Bayesian network was used to carry out an inference operation on the abstract states and to obtain discrete real-valued variables, which is not only suitable for numerical computation of neural networks, but also can further reduce the size of the state space. The neural network model used for value-function approximating based on the inference output of the Bayesian network not only can solve the problem of continuous space expression of reinforcement learning system, but also can improve the system learning speed due to its simple topology structure. Simulation results of an EGS system for typical traffic profiles verify the feasibility and validity of the proposed reinforcement learning scheduling algorithm.

Key words — Reinforcement learning, Abstract state, Bayesian network, Elevator group scheduling, Neural network.

I. Introduction

Elevators play an important role in today's urban life. Elevator group scheduling (EGS) problem is simply stated[1]. New passengers arrive at a bank of elevators at random times and floors, making hall calls to signal for rides up or down. A ride destination is unknown until the passenger enters the car and makes a car call to request a stop. The scheduler must assign a car to serve each hall call in a way that optimizes overall system performance. The execution of the schedule is performed by alternating the direction of movement of each car and servicing all hall calls assigned to it in its current direction of motion. The EGS problem has been studied for a long time due to its high practical significance: first approaches were mainly based on analytical approaches derived from queuing theory[2], in the last decades artificial intelligence techniques such as fuzzy logic[3], neural networks[4], and evolutionary algorithms[5] were introduced.

EGS is a typical Discrete event dynamic system (DEDS) having two characteristics. First, its state space is huge. Second, its dynamics is accompanied by a large amount of uncertainty. Because Reinforcement learning (RL) technique has properties of learning, optimization and decision, it is very suitable for EGS systems. Crites and Barto firstly applied reinforcement learning to an EGS problem and proposed two kinds of RL algorithms called RLp and RLd respectively[6]. RLp and RLd denote the RL controllers, parallel and decentralized. Simulation results showed that compared with conventional elevator scheduling algorithms, the RLp and RLd can obtain smaller average wait time of all passengers in the system. Most conventional reinforcement learning algorithms assume that the task is a Markov decision process (MDP), in which a transition probability is defined for discrete states given a discrete action. Reinforcement learning will encounter the curse of dimensionality problem when it is applied to real-world EGS systems. Therefore, in order to solve the curse of dimensionality problem, reinforcement learning should have generalization ability. The essence of generalization is to use a function to approximate unlearnt mapping from a state space to an action space. In recent years, fuzzy logic[7], neural network[8] and kernel technique[9]were commonly used to solve the curse of dimensionality problem in RL. In most RL-based EGS systems, BP[6], RBF[10] and CMAC[11] Neural networks (NNs) were adopted to approximate a state-action value-function. The inputs of these neural netorks are states and actions, while the output is corresponding value-function. Because the state and action spaces of an EGS system are huge, the number of input units of a NN are large. For example a 10-story building with 4 elevator cars, both the BP network in RLp and RLd algorithms and the CMAC network in RL-CMAC algorithm has 47 input units[6,11]. The complex topology structure of NNs inevitably results in slow learning speed which influences the real-time performance of EGS systems. From the view

*Manuscript Received Sept. 2009; Accepted Oct. 2009. This work is supported by the National Natural Science Foundation of China (No.60804022, No.60974050), Program for New Century Excellent Talents in University (No.NCET-08-0836), Natural Science Foundation of Jiangsu Province (No.BK2008126), Special Grade of the Financial Support from China Postdoctoral Science Foundation (No.200902533).

of dimensionality reduction, a kind of Bayesian reinforcement learning based on abstract states (BRL-AS) algorithm for EGS systems was proposed in the paper.

II. Reinforcement Learning Model of EGS

EGS problem can be regarded as a stochastic optimal decision problem. Therefore, we can use a MDP or a Semi-MDP (SMDP) model as the reinforcement learning model of EGS[10]. EGS systems have characteristics of DEDS, *i.e.*, the arrival events of passengers and elevators mutually occur and make states change. There are two patterns to observe states of EGS systems: time-driven and event-driven. Time-driven means that EGS systems actively observe states with a periodic activation pattern. The decision behavior of assigning a car will occur after the state observation. Therefore, MDP model is suitable for EGS problems with time-driven pattern. If an EGS system belongs to event-driven, SMDP model can be used. A SMDP model generally takes into account the time span between the generation of two successive decision behavior. The time span changes randomly which accords with the independent and random arrival event of passengers and the corresponding decision. Decision signal can be produced by EGS scheduler after a hall call is created. We used a SMDP model to describe EGS problems in our study, *i.e.*, the scheduler observes system states and makes decision using event-driven pattern.

Because the objective of EGS systems is to minimize the wait time and travl time of passengers, the state value-function and state-action value-function should be minimized. Bellman equations of EGS systems are

$$V^*(s) = \min_{\pi} \left(R(s,a) + \sum_{s' \in S, \tau} \gamma_\tau P(s', \tau | s, a) V^\pi(s') \right) \tag{1}$$

$$Q^*(s,a) = \min_{\pi} \Big(R(s,a) + \sum_{s' \in S, \tau} \gamma_\tau P(s', \tau | s, a) Q^\pi(s', \pi(s')) \Big) \tag{2}$$

where, $V^*(s)$ and $Q^*(s,a)$ are optimal state value-function and state-action value-function respectively, $R(s,a)$ the expected reward of state-action pair, τ the duration of system state after executing an elevator-assignment behavior a according to policy π, $P(s', \tau | s, a)$ a transition probability from state s to state s' under the executio of action a after time τ, γ_τ a discount factor that is used to determine the proportion of the delay to the future rewards.

In the field of reinforcement learning algorithms for discrete-time, the discount factor is generally a constant. But EGS systems belong to DEDS, *i.e.*, the time span between the generation of two successive decision behavior is a variable. Therefore, we use a variable relevant to decision time steps t_k and t_{k+1} to denote the discount factor as follows[6].

$$\gamma_\tau = \int_{t_k}^{t_{k+1}} e^{-\beta t} dt \tag{3}$$

where β is a predifined constant.

III. EGS Algorithm Based on BRL-AS

In order to solve the curse of dimensionality problem and to improve the learning speed of conventional RL algorithms applied to EGS problems, a kind of Bayesian reinforcement learning algorithm based on abstract states was proposed. The architecture of the EGS system based on BRL-AS is shown in Fig.1, which has 6 main modules. The state abstract module is used to exact relevant features from original state space of an EGS system to obtain a low-dimension abstract state space $(s_1, s_2, s_3, s_4, s_5, s_6)$ where s_1 denotes the motion situation of elevators, s_2 the relative position between s_1 and the current hall call, s_3 the number of spacing floors between an elevator and the current hall call, s_4 the direction of the current hall call, s_5 the number of hall calls and s_6 the number of passengers in a car. The Bayesian network module is uesd to carry out a probability inference on the abstract states so as to obtain three discrete variables denoted as Nfloor, Npeople and Nstop which respectively mean the number of floors passed by a car, the number of times of passengers enter and exit a car, the stop times of a car. The neural network receives the outputs of the Bayesian network and an elevator-assignment signal and gives an estimate of corresponging Q value. The weights of the neural network can be tuned based on TD error. Based on the estimated Q value, the action-selection module generates a suitable elevator-assignment signal a. Under the execution of a, the EGS system will receives a immediate reward r and sends it to the TD error computation module where both r and Q are used to compute TD error.

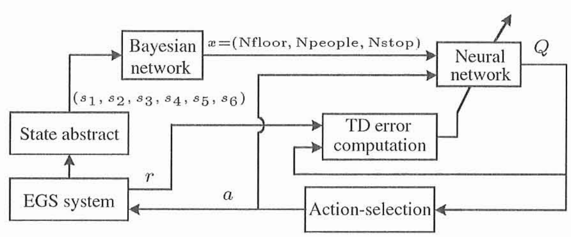


Fig. 1. EGS system based on BRL-AS

1. State space abstract

The state space of an EGS system is continuous because it includes the elapsed times since any hall calls were registered, which are real-valued. Even if these real values are approximated as binary values, the size of the state space is still immense[6]. For example a 10-story building with 4 elevator cars, its components include 2^{18} possible combinations of the 18 hall call buttons (up and down at each landing except the top and bottom), 2^{40} possible combinations of the 40 car buttons, and 18^4 possible combinations of the positions and directons of the cars (rounding off to the nearest floor). Other parts of the state are not fully observable, for example, the exact number of passengers waiting at each floor, their exact arrival times, and their desired destinations. Ignoring everything except the configuration of the hall and car hall buttons and the approximate position and direction of the cars, we obtain an extremely conservation estimate of the size of a discrete approximation to the continuous state space: $2^{18} \times 2^{40} \times 18^4 \approx 10^{22}$ states[6]. Therefore, to exact relevant feature from original states so as to reduce the scale of the original state space is a key of applicing RL algorithms to EGS problems.

The abstract states of an EGS system can be concluded as $(s_1, s_2, s_3, s_4, s_5, s_6)$. Fig.2 shows a sketch map of states s_1 and s_2. s_1 denotes not only the current but also the future situations of elevators which has five possible situations. It can be seen from Fig.2 that the five possible situations of s_1 are 'stop' (I), 'up' (II), 'down' (III), 'first up then down' (IV) and 'first down then up' (V). When s_1 is I, s_2 is ②. For other situations of s_1, s_2 has three possible situations. When s_1 is II, s_2 will be ② if a new hall call occurs between the current position of an elevator and the highest response floor; s_2 will be ① if the position of a new hall call is higher than the highest response floor; s_2 will be ③ if the position of a new hall call is lower than that of the current elevator. When s_1 is III, s_2 will be ② if a new hall call occurs between the current position of an elevator and the lowest response floor; s_2 will be ① if the position of a new hall call is higher than that of the current elevator; s_2 will be ③ if the position of a new hall call is lower than the lowest destination floor. When s_1 is IV and V, the situations of s_2 are the same as s_1 is II and III respectively.

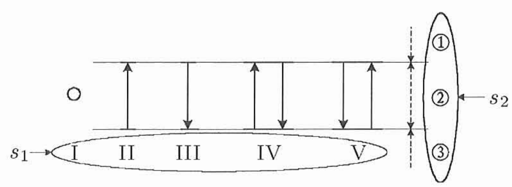


Fig. 2. Sketch map of states s_1 and s_2

If a building is a 10-story building, the value of state s_3 should be an integer ranged from 0 to 9. In order to further reduce the size of the state space, s_3 can be set as follows: s_3 is 1 if the spacing floor between elevator and the new hall call is from 0 to 3; s_3 is 2 if the spacing floor between elevator and the new hall call is from 4 to 6; s_3 is 3 other situations. State s_4 has two situations, *i.e.*, '+1' and '−1' which denote 'up' and 'down' directions respectively. If the car capacity of an EGS system is 20 passengers, *i.e.*, the sum of the hall call numbers and the passenger numbers in a car should be less than 20, the maximum hall call numbers and the maximum passenger numbers are set as 10 in our study. Based on the above analysis of abstract states, we can infer that the size of a discrete approximation to the continuous state space of the EGS system is $5 \times 3 \times 3 \times 2 \times 10 \times 10 = 9000$ states, which is much smaller than the original 10^{22} states.

2. Reward function of RL

For EGS systems, the main objective is to improve the running efficiency of elevators, to reduce the wait time of passengers and to reduce the times of start and stop of elevators, while the minor objective is to improve the service quality and to reduce the crowding degree. Therefore, we used the wait time and travl time of passengers as the performance index in the study. The reward function of RL is defined as a function relevant to the wait time and travl time[10].

$$r_{t_k} = \sqrt{\Big(\sum_p Tw(p)\Big)^2 + \Big(\sum_{p'} Tr(p')\Big)^2} \tag{4}$$

where Tw and Tr are wait time and travl time respectively. The wait time of passenger p is defined as

$$Tw(p) = t_k - t_p \tag{5}$$

where t_k is the current time and t_p the arrival time of passenger p.

Suppose passenger p' already stayed in a car at the previous decision point and the passenger entered the car at time $t_{p'}$, the travl time of passenger p' is defined as[10]

$$Tr(p') = t_k - t_{p'} \tag{6}$$

3. Bayesian network inference

In order to solve the curse of dimensionality problem, a neural network can be used to approximate $Q(s,a)$ as follows.

$$Q(s,a) = f_{NN}(s_1, s_2, s_3, s_4, s_5, s_6, a) \tag{7}$$

Neural network is a kind of powerful numerical computation tool, but it cannot deal with symbolic data such as states s_1 and s_2 of EGS systems. Therefore, in order to utilize neural network and to further reduce the size of the state space of EGS systems, a Bayesian network was adopted to obtain the discrete real-valued variables Nfloor, Npeople and Nstop. The order of variables is $\langle s_1, s_2, s_3, s_4, s_5, s_6$, Nfloor, Nstop, Npeople $\rangle$ and the Bayesian network for EGS systems is shown in Fig.3.

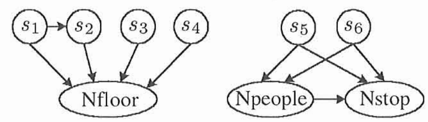


Fig. 3. Bayesian network for EGS system

There are two independent sub-graphs in Fig.3 where $\langle s_1, s_2, s_3, s_4, s_5, s_6 \rangle$ are evidence variables and $\langle$ Nfloor, Nstop, Npeople $\rangle$ are query variables. Variables s_1 and s_2 are dependent, Npeople and Nstop are dependent. It can be seen from Fig.3 that we can easily infer the predicted value of Nfloor from evidence variables $\langle s_1, s_2, s_3, s_4 \rangle$ using the variable elimination method[12], so do Npeople and Nstop from $\langle s_5, s_6 \rangle$.

4. Neural network approximator of state-action value-function

The outputs of the Bayesian network were transferred to the neural network approximator, then the state-action value-function can be expressed as

$$\begin{aligned} Q(x,a) &= \phi_{NN}(x,a) \\ &= \phi_{NN}(\text{Nfloor, Npeople, Nstop}, a) \end{aligned} \tag{8}$$

ϕ_{NN} (Nfloor, Nstop, Npeople, a) denotes a neural network model whose inputs are $x =$(Nfloor, Nstop, Npeople) and a and output is the corresponding Q value. The number of input units of the neural network is merely 4 which is far less than that of used in the RLp, RLd and RL-CMAC algorithms.

A 4-8-1 BP neural network was used as the approximator of Q value-function in our study. We adopted the direct gradient descent algorithm to tune the weights of BP network.

The learning process of RL is to descrease the temporal difference of the value-function between successive states in the state transition. TD error is calculated as

$$\delta_{TD}(t_k) = r_{t_k} + \gamma_r \min_{a_{t_{k+1}}} Q(x_{t_{k+1}}, a_{t_{k+1}}) - Q(x_{t_k}, a_{t_k}) \quad (9)$$

We used $E(t_k) = (\delta_{TD}(t_k))^2/2$ as the criterion of updating weight. The BP network can learn its weights according to the following equation

$$\begin{aligned}\Delta w(t_k) &= -\eta \frac{\partial E(t_k)}{\partial w(t_k)} \\ &= \eta \delta_{TD}(t_k) \frac{\partial Q(x_{t_k}, a_{t_k})}{\partial w(t_k)} \\ &= \eta \delta_{TD}(t_k) \frac{\partial Q(x_{t_k}, a_{t_k})}{\partial w(t_k)} \quad (10)\end{aligned}$$

where, η is a learning rate, the gradient information $\partial Q(x_{t_k}, a_{t_k})/\partial w(t_k)$ can be easily computed by referring to Ref.[7].

5. Action-selection strategy

A reinforcement learning agent is commonly confronted with a problem of selecing a suitable action. There are two factors needed to be considered. First, the agent should explore fully through the whole state and action spaces to find an optimal or sub-optimal policy. Second, the agent should take advantage of the obtained experience to select an action so as to reduce the learning cost. Generally speaking, these two factors are conflicting. In order to solve the dilemma of 'exploration' and 'exploitation', a Boltzmann-Gibbs distribution was used as the action-selection strategy. The action a_i is selected with probability

$$\begin{aligned}&\text{prob}(a_{t_k} = a_i | x_{t_k}) \\ &= \frac{\exp(-Q(x_{t_k}, a_i)/T)}{\sum_{j \le N} \exp(-Q(x_{t_k}, a_j)/T)} \quad (11)\end{aligned}$$

The temperature parameter $T > 0$ controls the stochastic degree of action-selection. i and j denote the serial number of elevators which satisfy $i, j = 1, 2, \cdots, N$. N is the number of elevators. It can be seen from Eq.(11) that the selection result depends on Q value. Bigger Q value means that the wait and travl time are longer. Therefore, the corresponding elevator-assignment strategy is unreasonable and the probability of the action is small.

IV. Simulation Research

A 10-story building with 4 elevators is our simulated object which was also studied in Refs.[6] and [11]. The system dynamics is described by the following parameters[6,13]:

- Floor time (the time to move one floor at maximum speed): 1.45 secs.
- Stop time (the time needed to decelerate, open and close the doors, and accelerate again): 7.19 secs.
- Turn time (the time needed for a stopped car to change direction): 1 sec.
- Load time (the time for one passenger to enter or exit a car): random variable from a 20th order truncated Erlang distribution with a range from 0.6 to 6 secs and a mean of 1 sec.
- Car capacity: 20 passengers.

We use a traffic profile which dictates arrival rates for every 5-minute interval during a typical down-peak rush hour. Table 1 shows the mean number of passengers arriving at each of floors 2 through 10 during each 5-minute interval who are headed for the lobby.

Table 1. The down-peak traffic profile

Time (m)	0	5	10	15	20	25	30	35	40	45	50	55
Rate	1	2	4	4	18	12	8	7	18	5	3	2

The EGS system based on BRL-AS algorithm was trained on 10 hours of simulated elevator time using the down-peak traffic profile. The simulation program was realized by using MATLAB 7.0 software on a P4/1.5G/256M computer. During the simulation process, temperature parameter $T = 10$, learning rate $\eta = 0.25$ and the constant $\beta = 0.01$. The trained BRL-AS algorithm was applied to two typical traffic profiles on 30 hours of simulated elevator time to ensure its statistical performance: a down-peak profile with down-only traffic and a down-peak profile with up and down traffic. Table 2 shows the results for the down-peak profile with down-only traffic. Table 3 shows the results for the down-peak profile with up and down traffic, including an average of 2 up passengers per minute at the lobby. There are three performance indexes used in Tables 2 and 3. The first term AvgWait means the average wait time of all passengers. Another term SquaredWait is the average squared wait time. The last index is the percentage of passengers that wait longer than some dissatisfaction threshold (usually 60 seconds).

Table 2. Results for down-peak profile with down-only traffic

Algorithms	AvgWait (s)	SquaredWait (s)	SystemTime (s)	Percent > 60 secs
SECTOR[6]	21.4	674	47.7	1.12
DLB[6]	19.4	658	53.2	2.74
BASIC HUFF[6]	19.9	580	47.2	0.76
LQF[6]	19.1	534	46.6	0.89
RLp[6]	14.8	320	41.8	0.09
RLd[6]	14.7	313	41.7	0.07
RL-BP[11]	21.2	569	/	0.09
RL-CMAC[11]	19.7	529	/	0.07
BRL-AS	15.9	389	43.1	0.059

Table 3. Results for down-peak profile with up and down traffic

Algorithms	AvgWait (s)	SquaredWait (s)	SystemTime (s)	Percent > 60 secs
SECTOR[6]	27.3	1252	54.8	9.24
DLB[6]	21.7	826	54.4	4.74
BASIC HUFF[6]	22.0	756	51.1	3.46
LQF[6]	21.9	732	50.7	2.87
RLp[6]	16.9	476	42.7	1.53
RLd[6]	16.9	468	42.7	1.40
RL-BP[11]	24.3	1140	/	9.90
RL-CMAC[11]	21.8	1048	/	9.14
BRL-AS	21.3	599	49.9	3.30

The results of SECTOR, DLB, BASIC HUFF, LQF, RLp and RLd algorithms in Tables 2 and 3 were cited from Ref.[6], and the results of RL-BP and RL-CMAC algorighms were cited from Ref.[11]. The RL-BP was the reproduced algorithm of RLp by Gao *et al.* But they did not obtain the same simulation results as RLp in Ref.[6], which may be the reason of parameters setting problem. It is easily seen from Tables 2 and 3 that even though BRL-AS is little worse than RLd and RLp, it is better than the reproduced RLp and other algorithms. It should be noted that the RLp and RLd algorithms are time-consuming even though they can obtain better performance. Crites and Barto pointed out that four days on a 100 MIPS workstation are needed to train the RLp and RLd algorithms on 60000 hours of simulated elevator time. It is only 3 seconds was needed to train the BRL-AS algorithm on one hour of simulated elevator time, which means that about less than 50 hours were taken to simulate 60000 hours of simulated elevator time. Therefore, high computing efficiency of the proposed BRL-AS algorithm ensures it is much suitable for the real-time scheduling of EGS systems.

V. Conclusion

Elevator group scheduling system is a very large scale stochastic dynamic optimization problem. Due to its vast state space, significant uncertainty, and numerous resource constraints such as finite car capacities and registered hall/car calls, it is hard to manage EGS using conventional control methods. Reinforcement learning technique is very suitable for elevator group scheduling problems due that it contains learning, optimization and decision ideas. But because EGS systems have high-dimension state spaces, conventional reinforcement learning algorithms are inevitably encountered with the curse of dimmensionality problem. In order to solve the continuous space representation problem of RL and to improve the learning speed of RL, a kind of BRL-AS algorithm was proposed. At first, we constructed a low-dimension abstract state space based on the analysis of motion situations of EGS system. Secondly, a Bayesian network was used to infer the abstract states so as to obtain discrete real-valued variables and to further reduce the size of the abstract state space. In the third step, a neural network was used to estimate Q value based on the output of the Bayesian network. At last, a suitable elevator-assignment signal can be given according to an action-selection strategy. It is verified that the proposed BRL-AS algorithm is a very effective and high-efficiency scheduling method with satisfactory performance for EGS systems. The BRL-AS algorithm is also capable to adopt for different process parameters such as the number of elevator cars, capacities of the cars, floor numbers, different traffic profiles, and it is a distinctive advantages of the algorithm.

References

[1] S.Y. Yang, J.Z. Tai, C. Shao, "Dynamic partition of elevator group control system with destination floor guidance in up-peak traffic", *Journal of Computers*, Vol.4, No.1, pp.45–52, 2009.

[2] Y. Sakai, K. Kurosawa, "Development of elevator supervisory group control system with artificial intelligence", *Hitachi Review*, Vol.33, No.1, pp.25–30, 1984.

[3] J.R. Wan, J.S. Zhang, Z.Q. Wei, "Study of elevator group-control expert system based on traffic-flow mode recognition", *Elevator World*, Vol.54, No.11, pp.130–133, 2006.

[4] C.E. Imrak, "Artificial neural networks application in duplex/triplex elevator group control system", *Strojniski Vestnik*, Vol.54, No.2, pp.103–114, 2008.

[5] L. Yu, S. Mabu, T.T. Zhang, S. Eto, K. Hirasawa, "Multi-car elevator group supervisory control system using genetic network programming", *Proceedings of IEEE Congress on Evolutionary Computation*, Trondheim, Norway, pp.2188–2193, 2009.

[6] R.H. Crites, A.G. Barto, "Elevator group using multiple reinforcement learning agents", *Machine Learning*, Vol.33, No.2-3, pp.235–262, 1998.

[7] X.S. Wang, Y.H. Cheng, J.Q. Yi, "A fuzzy Actor-Critic reinforcement learning network", *Information Sciences*, Vol.177, No.18, pp.3764–3781, 2007.

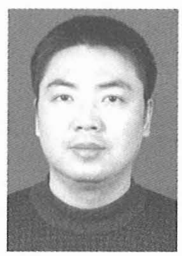
[8] A.H. Tan, N. Lu, D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback", *IEEE Transactions on Neural Networks*, Vol.19, No.2, pp.230–234, 2008.

[9] X.S. Wang, X.L. Tian, Y.H. Cheng, "Value approximation with least squares support vector machine in reinforcement learning system", *Journal of Computational and Theoretical Nanoscience*, Vol.4, No.7/8, pp.1290–1294, 2007.

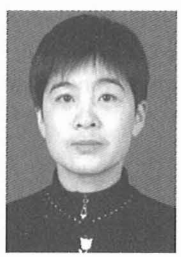
[10] Q. Zong, C.F. Song, G.S. Xing, "A study of elevator dynamic scheduling policy based on reinforcement learning", *Elevator World*, Vol.54, No.1, pp.58–58, 2006.

[11] Y. Gao, J.K. Hui, B.N. Wang, D.L. Wang, "Elevator group control using reinforcement learning with CMAC", *Acta Electronica Sinica*, Vol.35, No.2, pp.362–265, 2007. (in Chinese)

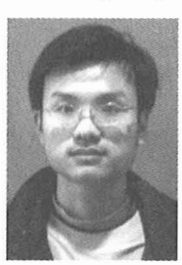
[12] A. Cano, M. Gomez-Olmedo, S. Moral, "Binary probability trees for Bayesian networks inference", *Lecture Notes in Computer Science*, Vol.5590, pp.180–191, 2009.



CHENG Yuhu received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences in 2004. He is currently an associate professor in the School of Information and Electrical Engineering, China University of Mining and Technology. His main research interests include machine learning and intelligent system.



WANG Xuesong received the Ph.D. degree from China University of Mining and Technology in 2002. From July 2002 to June 2004, she worked as a postdoctoral researcher at Beijing Institute of Technology. She is currently a professor in the School of Information and Electrical Engineering, China University of Mining and Technology. Her main research interests include machine learning, bioinformatics and intelligent system. (Email: wangxuesongcumt@163.com)



ZHANG Yiyang was born in Henan Province, China, in 1984. He received the B.S. and M.S. degrees from China University of Mining and Technology in 2006 and 2009 respectively. His main research interests include reinforcement learning and intelligent system.