International Journal of Automation and Computing 8(1), February 2011, 100-106
DOI: 10.1007/s11633-010-0560-2

Path Selection in Disaster Response Management Based on Q-learning

Zhao-Pin Su[1, 2, 3] Jian-Guo Jiang[1, 2, 4] Chang-Yong Liang[2, 3] Guo-Fu Zhang[1, 2, 4]

[1]Key Laboratory of Special Display Technology (Hefei University of Technology), Ministry of Education, Hefei 230009, PRC

[2]School of Computer and Information, Hefei University of Technology, Hefei 230009, PRC

[3]Postdoctoral Research Station for Management Science and Engineering, Hefei University of Technology, Hefei 230009, PRC

[4]Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei 230009, PRC

Abstract: Suitable rescue path selection is very important to rescue lives and reduce the loss of disasters, and has been a key issue in the field of disaster response management. In this paper, we present a path selection algorithm based on Q-learning for disaster response applications. We assume that a rescue team is an agent, which is operating in a dynamic and dangerous environment and needs to find a safe and short path in the least time. We first propose a path selection model for disaster response management, and deduce that path selection based on our model is a Markov decision process. Then, we introduce Q-learning and design strategies for action selection and to avoid cyclic path. Finally, experimental results show that our algorithm can find a safe and short path in the dynamic and dangerous environment, which can provide a specific and significant reference for practical management in disaster response applications.

Keywords: Disaster response management, path selection, agent, self-organizing, Markov decision process, Q-learning.

1 Introduction

In the light of recent events throughout the world, ranging from natural disasters, such as the Asian Tsunami, hurricane Katrina in New Orleans, the serious floods of Yangtze River Basin, and the 5/12 large earthquake in Wenchuan (happened on May 12, 2008), to the man-made disasters such as the 7/7 terrorist attacks (happened on July 7, 2005) in London, and 9/11 attacks in New York (happened on September 11, 2001), the topic of disaster management[1−3], also known as disaster response, has become a key social and political concern. For example, in 2003, the Chinese government developed emergency management plans for natural disasters, accidents, public health events, social security, and so on. In November 1, 2007, the national emergency response law was effective in China.

However, it can be seen from these and many other similar disasters that there is also an overwhelming need of better information technology to support efficient and effective decisions for disaster response management.

Generally speaking, most of the researches within this area mainly focus on the following three aspects.

1) Communications

Efficient communications are crucial for disaster response management, and many researchers have done much work on this[4−7]. Tsai et al.[4] presented a building blackbox system to bridge the gap between the first responders and the building systems and to provide reliable and accurate building information over a mobile ad hoc network. Bello et al.[5] designed ubiquitous mobile infrastructures to match the response environment′s communication flows and telecommunications, and produced a ubiquitous mobile communication system. Yu et al.[6] studied and enhanced the interoperability of land mobile radio (LMR) with commercial wireless cellular networks, by which a wide variety of benefits could be offered to disaster responders, including new multimedia services, increased data rates, and low cost devices. Télécoms Sans Frontières (TSF)[7], a non-government organization specialized in emergency telecommunications, provided broadband internet connections, as well as phone and fax lines and technical assistance to quake effected regions across India and Pakistan.

2) Decision-making support system

After disasters happen, efficient decision-making support systems can be used to reduce the time needed to make crucial decisions regarding task assignment and resource allocation, and to guide longer term decisions involving resource acquisition as well as training and the evaluation of command and control[8−11]. Thompson et al.[8] analyzed a number of factors contributing to current lacklustre response efforts, such as the complex, rapidly changing decision-making environments, the slow, ineffective strategies for gathering, processing, analyzing data, and so on. Liu et al.[9] employed control system technology to develop a general framework for the disaster response management system, which also incorporates an adaptive decision system, and presented a model with networked critical infrastructure (CI) systems. Hu et al.[10] developed an integrative application platform for flood disaster emergency response, fast loss evaluation, and salvation decision-making support based on existing data, software and methodolo-

Manuscript received September 14, 2009; revised May 20, 2010
This work was supported by National Basic Research Program of China (973 Program) (No. 2009CB326203), National Natural Science Foundation of China (No. 61004103), the National Research Foundation for the Doctoral Program of Higher Education of China (No. 20100111110005), China Postdoctoral Science Foundation (No. 20090460742), National Engineering Research Center of Special Display Technology (No. 2008HGXJ0350), Natural Science Foundation of Anhui Province (No. 090412058, No. 070412035), Natural Science Foundation of Anhui Province of China (No. 11040606Q44, No. 090412058), Specialized Research Fund for Doctoral Scholars of Hefei University of Technology (No. GDBJ2009-003, No. GDBJ2009-067)

gies such as the object-oriented, reuse, geodatabase, component object model, remote sensing (RS), and geographic information system (GIS) software technologies. Fiedrich and Burghardt[11] used agent-based simulation systems to model human and system behaviors during or after disaster events, and proposed a disaster response agent-based system to envision to support emergency managers by helping maintain common situational awareness and aiding the planning and coordination of response activities. Yang et al.[12] explored the design specification of on-site emergency response information systems for emergency first responders, and formulated the basic design principles for on-site dynamic information collection information sharing. Turoff et al.[13] developed a set of general and supporting design principles and specifications for a "dynamic emergency response management information system" (DERMIS), and presented a framework for the system design and development that addressed the communication and information needs of first responders as well as the decision making needs of command and control personnel.

3) Optimal rescue/evacuation path planning

Selection of suitable rescue/evacuation path is very important to reduce the loss of disasters[14−17]. Chandio et al.[14] presented a GIS based guiding system for route decision making to supply the relief work and relief aid on the affected area. Ozdamar et al.[15] identified a feasible, acceptable solution to the emergency logistic problem based on a greedy l-neighborhood search technique, and developed a simple and fast constructive heuristic path-builder to construct all vehicle itineraries in parallel and iterative, and to exploit foreseeable opportunities within the vehicle′s limited neighborhood. Chiou and Lai[16] proposed an integrated multi-objective model to determine the optimal rescue path, which consists of three sub-models: rescue shortest path model, post-disaster traffic assignment model, and traffic controlled arcs selection model, and to minimize four objectives: travel time of rescue path, total detour travel time, number of unconnected trips of non-victims, and number of police officers required; they used genetic algorithms and K-shortest path methods to determine optimal rescue path and controlled arcs, and used fuzzy system reliability theory (weakest t-norm method) to measure the access reliability of rescue path. Yuan and Wang[17] presented a single-objective path selection model to minimize total travel time along a path and designed a modified Dijkstra algorithm to solve the model, and further presented a multi-objective path selection model to minimize the total travel time along the path and the path complexity, and used ant colony optimization algorithm to solve the model.

However, the most important activity after a disaster happens is how to select a path to arrive at the affected area safely, rapidly know the disaster condition and supply the relief work and relief aid on the affected area. The problem is called disaster response path selection (DRPS). Although most researches[14−17] on optimal rescue/evacuation path planning can solve the problem of DRPS to a certain extent, there are some shortcomings as follows:

1) Most of the existing researches use a directed graph[18] $G(V, A)$ to model the disaster environment, assume that nodes in V and the arcs $(v_i, v_j) \in A$ between nodes v_i and v_j are safe, and can be obtained safely. However, after the disaster happens, the path between nodes v_i and v_j is not always safe in nature. Therefore, the proposed model cannot commendably describe the actual dynamic and complex environment.

2) In order to solve the model $G(V, A)$, researchers have set overmany parameters and brought much trouble in actual applications.

3) The model $G(V, A)$ makes the proposed algorithms unsuitable for large scale problems, and the complexity will increase exponentially with the number of nodes in V.

Thus, against this background, this paper is absorbed in a novel model for DRPS. To achieve the goal, we advance the state of the art in the following ways:

1) We address the problem of DRPS in a two-dimensional geographic grids, and assume that a rescue team is an agent.

2) We develop a novel DRPS algorithm based on Q-learning to search the safest and shortest path, or approximate one.

The remainder of this paper is organized as follows. Section 2 gives the model of DRPS and testifies that the problem of DRPS is a Markov decision process. In Section 3, we show our algorithm based on Q-learning. Section 4 gives experiments to evaluate our algorithm performance. Finally, Section 5 discusses the conclusions and presents the future work.

2 Model of DRPS

2.1 Environment

In the immediate aftermath of a disaster, the environment around the affected location B is usually very complex, and previously safe areas might become dangerous or unaccessible. Therefore, the dynamic and complex environment considered in this paper assumes a two-dimensional geographic grids V: $(m+1)\cdot(n+1)$ (See Fig. 1), where we identify as the following:

1) Safe areas. Any agent can traverse this kind of areas safely and quickly (white areas in Fig. 1).

2) Danger areas. The agent can traverse this kind of areas, but may waste more time than in safe areas (slash areas in Fig. 1).

3) Unaccessible areas. The agent cannot traverse this kind of areas (gray areas in Fig. 1).

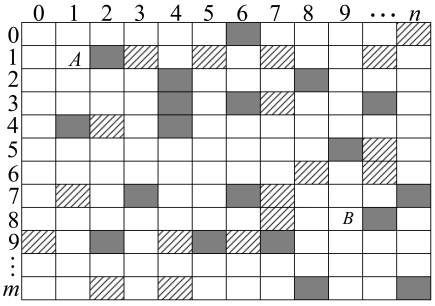


Fig. 1 The model of DRPS

Assume that a rescue team is in location A. The problem is how the rescue team arrives at B safely and rapidly.

The model can be extended in a straightforward way to

involve more than three area types which affect the movement of the rescue team in a variety of ways.

2.2 Agent

In this paper, we assume that a rescue team is an agent, and each agent has a well-defined physical position and movement capability. In practice, these agents can be teams of robots, persons, wrecking cars, rescue materials, and so on. The goal of the agent is to reach the affected location B from the rescue location A.

Assume that the current position of an agent is (x, y), and the agent can move up, down, left, and right to its neighbor areas, and will receive different signals γ according to the neighbor area types. The agent can arrive at B by its continuous movement according to a certain strategy related with the signal γ.

Our objective is to find the strategy, and obtain the safest and shortest path from the rescue location A to the affected location B.

In the proposed model, an agent selects a path as short as possible, and may traverse the danger areas unavoidably. There are two choices for the agent: first, make a detour, select more safe areas, and increase the path; second, select danger areas and consume more time.

Definition 1. When an agent is in a grid (x, y) (see Fig. 1), we denote its state as $s = (x, y)$.

Definition 2. An agent can move to its any neighbor area by selecting an action from a set $Action = \{up, down, left, right\}$.

The agent can transfer its state from s_i to s_{i+1} by executing an action $a_j \in Action$, and the relation between s_i and s_{i+1} can be shown as

$$s_i = (x_i, y_i) \tag{1}$$

$$s_{i+1} = (x_{i+1}, y_{i+1}) \tag{2}$$

$$s_{i+1} = \begin{cases} (x_i - 1, y_i), & j = up \\ (x_i + 1, y_i), & j = down \\ (x_i, y_i - 1), & j = left \\ (x_i, y_i + 1), & j = right. \end{cases} \tag{3}$$

The agent should select an action in every area (x, y), so the agent disaster response path selection is a sequential decision making process.

2.3 Markovity of disaster response path selection

In order to testify the process of path selection, which is a Markov decision process, we first give the definition.

Definition 3. A Markov decision process (MDP)[19, 20] model contains: a set of possible states S, a set of possible actions A, a real valued reward function $R : S \times A \to R$, and a state transition function $T : S \times A \to P(S)$. Let $R(s, a, s')$ denote the immediate reward after transition from state s to s' by executing action a, and $P(s, a, s')$ denote the state transition probability from state s to s' by executing action a.

The essence of MDP is that the effects of an action taken in a state depend only on that state but not on the prior history.

Assumption 1. The process of disaster response path selection is a Markov decision process.

Proof. According to the properties of MDP, here, we only need to prove that the process constructed by $S = (s_1, s_2, \cdots, s_i, \cdots)$ of agent disaster response path selection is a Markov decision process.

According to Definition 3, we only need to prove that state s_{i+1} only depends on s_i. Note that at time $i + 1$, according to (1)–(3), state s_{i+1} only depends on s_i. Therefore, this process is a Markov decision process. □

3 Our path selection algorithm

3.1 Agent selecting optimal DRPS based on Q-learning

Q-learning is an effective model-free reinforcement learning algorithm proposed by Watkins[21−25] in the environment which supposed to be a discrete state Markov decision process. In the theory of Q-learning, an agent is provided the ability to act optimally by evaluating the Q value which represents the total consequences of a series of actions. In each step of interaction, the agent receives an immediate reward for the selected action. Then, the current state and Q value are updated, the agent continues to select the next action with a certain strategy. By comparing the effects of learning, the agent can find out the optimal strategy. As the process of DRPS is a Markov decision process, the agent aims at the maximum reward by selecting the optimal strategy in every discrete state.

In the model in Section 2, the objective is to obtain a short path and a minimum time, while Q-learning is to obtain a large reward. To solve the conflict, we define the signal γ as follows.

Definition 4. There are four different signals according to area types (See in Fig. 1) and location B:

$$\gamma = \begin{cases} \gamma_{\rm ua}, & \text{if the area type is unaccessible area} \\ \gamma_{\rm da}, & \text{if the area type is danger area} \\ \gamma_{\rm sa}, & \text{if the area type is safe area} \\ \gamma_B, & \text{if the area is location } B \end{cases} \tag{4}$$

and $\gamma_B \gg \gamma_{\rm sa} > \gamma_{\rm da} > \gamma_{\rm ua} = 0$.

Now, we will give the definition of "state value" and "immediate reward" according to the nature of the DRPS problem.

Definition 5. The immediate reward r of the agent is defined as the reward obtained after action $a \in Action$ is taken and the agent's state is driven from s to s'. In the problem of DRPS, $r = \gamma_{s'}$.

Definition 6. The state value $\nu(s)$ is defined as the sum of immediate rewards on the path before the agent reaches its state s.

Given this, the task of the agent is to select an optimal behavior strategy to maximize its long-term reward. Considering a behavior strategy π, the state value s is as follows, where λ is a discount factor:

$$v^{\pi}(s) = r(\pi(s)) + \lambda \sum_{s \in S} P(s, a, s') v^{\pi}(s'). \tag{5}$$

The theory of dynamic programming can guarantee at least an optimal strategy π^* for the agent to obtain the

maximum reward as follows:

$$v^{\pi^*}(s) = \max_a \{r + \lambda \sum_{s' \in S} P(s, a, s') v^{\pi^*}(s')\}. \quad (6)$$

We note that Q-learning is to optimize directly the iterative Q function, but not estimate the environment model. When an action yields a state transition from s_i to s_{i+1} and the learning agent receives immediate reward r, the value of Q function is updated in the following way, where α is the learning rate:

$$Q_{i+1}(s, a) \leftarrow (1 - \alpha) Q_i(s, a) + \alpha \left[r + \lambda \max_{a' \in Action} Q_i(s', a') \right]. \quad (7)$$

Therefore, the algorithm can obtain an optimal strategy only by optimizing directly the iterative Q function. Waltkins and Dayan[21] proved that Q-learning can converge to a global optimal behavior strategy through infinitely searching the state space by the agent.

The basic learning steps are as follows:

Step 1. Initialization: for each $s \in S$, $a \in Action$, set $Q(s, a) = 0$; set the iteration number $t = 0$.

Step 2. Set the initial state $s = A$.

Step 3. If $s \neq B$, select an action a according to a certain policy (see Section 3.2), and go to Step 4; otherwise, go to Step 7.

Step 4. Execute a, drive agent to state s' from state s, obtain an immediate reward r according to the type of s', and update the value of Q function according to (7).

Step 5. If the type of s' is unaccessible area, go to Step 2.

Step 6. Set $s = s'$, and go to Step 3.

Step 7. Set $t = t + 1$.

Step 8. If $t = t_{\max}$, where $t_{\max}$ is the maximum number of iterations, end the learning; otherwise, go to Step 2.

3.2 Strategy for action selection

In Q-learning algorithm, selecting actions should follow a certain strategy. For example, one can select actions stochastically according to Boltzmann distribution[26] or to the roulette selection referring to the probabilities of actions in each state[27].

In this paper, at time t, we select actions according to the state-action pair value in time $t + 1$[24], that is,

$$a = \arg \max_{a' \in Action} Q(s', a'). \quad (8)$$

3.3 Strategy for avoiding cyclic path

In practical applications, we find that the action selection policy in Section 3.2 is easy to drive the agent to drop into a cyclic path, which decreases the performance of the algorithm. Therefore, we design a new function $check(s, a)$ to check whether the agent has dropped into a loop or not.

Assume that $s = (x, y)$ is the current state of the agent, and L is the set of traversed states. Then, $check(s, a)$ can be expressed as follows:

Step 1. Select action a according to the proposed policy in Section 3.2.

Step 2. Execute action a, and drive the agent to state s' from s.

Step 3. Obtain $a_{\text{next}} = \arg \max_{a' \in Action} Q(s', a')$.

Step 4. Execute action a_{next}, and drive the agent to state s'' from s'.

Step 5. If $s'' \in L$, there is a cyclic path. Then, reselect another different action a stochastically from $Action$ except the last selected action.

3.4 Strategy for unaccessible areas

The agent enters an unaccessible area through its moving means that its current path is invalid. If we discard the invalid solutions and re-start the exploration from A, it will waste much time and decrease the performance of the algorithm. To tackle this shortcoming, we present a "step back" strategy to prevent the agent from entering an unaccessible area.

Definition 7. The "step back" strategy is stated as follows: when the agent selects an action a in the current state s, if its next state s' is an unaccessible area after executing action a, it will stochastically re-select another action a' from $Action$ which may drive it into a safe area or a danger area.

3.5 Our path selection algorithm

Our path selection algorithm is described as follows:

Step 1. Initialization: for each $s \in S$, $a \in Action$, set $Q(s, a) = 0$; set the iteration number $t = 0$.

Step 2. Set the initial state $s = A$.

Step 3. If $s \neq B$, select an action a according to the policy in Section 4.2, and go to Step 4; otherwise, go to Step 8.

Step 4. Use function $check(s, a)$ to avoid cyclic path.

Step 5. Execute a, drive the agent to state s' from state s, obtain an immediate reward r according to the type of s', and update the value of Q function according to (7).

Step 6. If the type of s' is unaccessible area, the agent adopts "step back" strategy, and selects another different action a, go to Step 5.

Step 7. Set $s = s'$, and go to Step 3.

Step 8. Set $t = t + 1$.

Step 9. If $t = t_{\max}$ ($t_{\max}$ is the maximum number of iterations), end the algorithm; otherwise, go to Step 2.

3.6 Computational complexity

If $p = |Action| = 4$, then the efficiency of our algorithm can be judged from computations as follows:

1) In Step 1, we need to initialize $Q(s, a)$ for every $s \in S$ and $a \in Action$, while the number of states and actions is $m \cdot n \cdot p$. Therefore, the complexity of Step 1 is $O(m \cdot n \cdot p)$;

2) The complexity of selecting action is $O(p \cdot \log_2 p)$, and the complexity of $check(s, a)$ is $O(p \cdot p \cdot \log_2 p)$;

3) The complexity of calculating Q value is $O(p \cdot \log_2 p)$, and the complexity of "step back" strategy is p.

Therefore, the complexity of our algorithm for disaster response path selection is $O(t_{\max} \cdot (m \cdot n \cdot p + p \cdot \log_2 p + p \cdot p \cdot \log_2 p + p \cdot \log_2 p + p))$, which is close to $O(n^4)$.

4 Experimental results and discussion

In order to evaluate the performance of our algorithm, we consider the environment shown in Fig. 2, where A and B

are randomly generated, and the environment is dynamic. The signals of γ of different types are $\gamma_{\rm sa} = 10$, $\gamma_{\rm da} = 2$, $\gamma_{\rm ua} = 0$, and $\gamma_B = 100$.

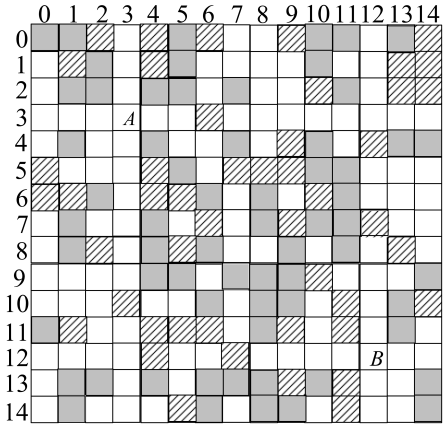


Fig. 2 Environment of DRPS

The shortest and safest path is determined purely by optimizing the objective shown in (7). It is because the idea of the proposed algorithm is that if at a given state s an agent has to choose among different actions, those with a high value of $Q(s, a)$ (see (8)) are chosen with higher probability, and the process is thus characterized by a positive feedback based on reinforcement learning mechanism, which ensures that an agent may at least find an approximately optimal solution within a finite number of loops.

We made six different independent experiments based on different parameters in the environment, and the partial parameters are shown in Table 1.

Table 1 Parameters of DRPS

Experiment	1	2	3	4	5	6
$t_{\max}$	–	–	100	–	–	–
m	–	–	15	–	–	–
n	–	–	15	–	–	–
α	0.1	0.1	0.2	0.2	0.3	0.3
λ	0.8	0.9	0.8	0.9	0.8	0.9

Fig. 3 shows the resultant paths which have the same state value of B. The path (1) of $(3, 3)$ $(3, 4)$ $(3, 5)$ $(3, 6)$ $(3, 7)$ $(3, 8)$ $(3, 9)$ $(3, 10)$ $(3, 11)$ $(3, 12)$ $(4, 12)$ $(5, 12)$ $(6, 12)$ $(7, 12)$ $(8, 12)$ $(9, 12)$ $(10, 12)$ $(11, 12)$ $(12, 12)$ was generated by the parameters $\alpha = 0.1$, $\lambda = 0.8$, and $\alpha = 0.3$, $\lambda = 0.8$; path (2) of $(3, 3)$ $(4, 3)$ $(5, 3)$ $(6, 3)$ $(7, 3)$ $(8, 3)$ $(9, 3)$ $(9, 2)$ $(10, 2)$ $(11, 2)$ $(12, 2)$ $(12, 3)$ $(12, 4)$ $(12, 5)$ $(12, 6)$ $(12, 7)$ $(12, 8)$ $(12, 9)$ $(12, 10)$ $(12, 11)$ $(12, 12)$ was generated by the parameters $\alpha = 0.2$ and $\lambda = 0.9$.

Fig. 4 shows the iteration curves of steps. We can see that the number of steps is decreasing through continuous learning, and the algorithm can find the shortest path in a limited iteration number. Fig. 5 gives the details of iteration curves of steps. When $\alpha = 0.3$ and $\lambda = 0.8$, the algorithm can converge to an optimal or approximately optimal solution. Table 2 lists the partial Q value after 100 iterations when $\alpha = 0.3$ and $\lambda = 0.8$.

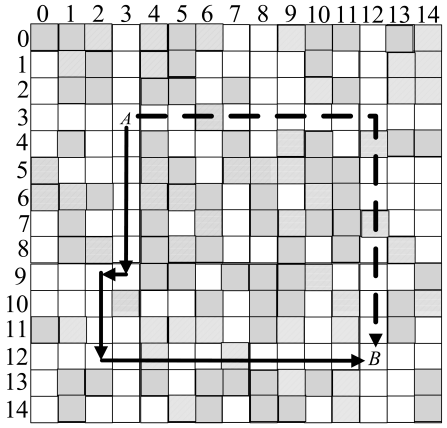


Fig. 3 Resultant paths of the proposed algorithm

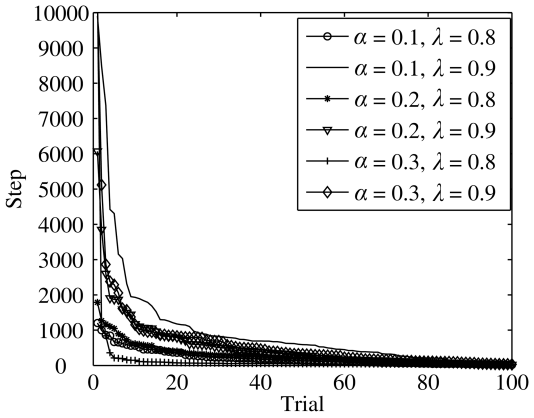


Fig. 4 The iteration curves of steps

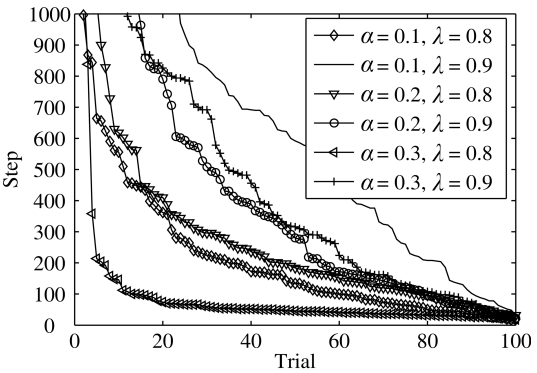


Fig. 5 The details of iteration curves of steps

Table 2 Partial Q value after 100 iterations

Q value	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)	(3,8)	(3,9)	(3,10)	(3,11)
up	0	0	0	10	0	35	50	42	0
$down$	19	0	13	16	0	27	41	0	46
$left$	27	17	15	21	26	34	51	52	53
$right$	51	51	51	52	53	53	54	55	57
Q value	(3,12)	(4,12)	(5,12)	(6,12)	(7,12)	(8,12)	(9,12)	(10,12)	(11,12)
up	55	53	55	59	62	62	70	76	82
$down$	58	61	63	66	70	76	82	90	100
$left$	55	48	0	0	0	0	70	68	74
$right$	55	0	58	60	55	58	70	0	0

The experiment results show that our algorithm can find the shortest and safest path in a limited iteration number without any unaccessible areas and cyclic path.

In general, most of the existing researches use optimization algorithms, such as genetic algorithms[16] and ant colony optimization algorithm[17], to solve the path selection problem. However, under the model proposed in Section 2, it is unsuitable to solve our model in Section 2 because of the following reasons:

1) Optimization algorithms do not identify any environmental information to search a solution. But in the DRPS problem, environment is an important factor, because it is dynamic and complex. Q-learning can obtain an optimal solution through constant trial and error interactions with the environment.

2) Because of the complexity of the environment, optimization algorithms may be unable to identify different types of grids, and easily run into an unfeasible solution, which contains some unaccessible areas. Q-learning can avoid the problem by setting a different signal γ according to area types.

3) The environment of DRPS cannot be easily predicted, that is, there may be no any priori knowledge. In this case, optimization algorithm cannot be used, while Q-learning can work without a priori knowledge due to its interaction with the environment.

Therefore, in this paper, we used Q-learning to solve the DRPS problem, rather than any optimization algorithm.

5 Conclusions and future work

In this paper, we developed a novel path selection algorithm based on Q-learning for disaster response management, and evaluated the performance of our algorithm by experiments. This algorithm can find a safer and shorter path in dynamic and dangerous environment, and avoid cyclic path dropping into unaccessible areas, thus providing a specific and significant reference for practical management tasks in disaster response applications. Therefore, our algorithm can be seen to represent a significant advance in the state of the art.

For future work, we will concentrate on solving path selection in large scale areas with a mass of agents.

References

[1] A. Mansourian, A. Rajabifard, M. J. V. Zoej, I. Williamson. Using SDI and web-based system to facilitate disaster management. *Computers and Geosciences*, vol. 32, no. 3, pp. 303–315, 2006.

[2] K. Friesen, D. Bell. Regional health authorities, disaster management, and geomatics: Opportunities and barriers. *International Journal of Emergency Management*, vol. 4, no. 2, pp. 141–165, 2007.

[3] W. E. Roper. Waste management policy revisions: Lessons learned from the Katrina disaster. *International Journal of Environmental Technology and Management*, vol. 8, no. 2/3, pp. 275–309, 2008.

[4] M. H. Tsai, L. Y. Liu, F. P. Mora, C. Arboleda. A preliminary design of disaster-survivable building blackbox system for Urban disaster response. *Electronic Journal of Information Technology in Construction*, vol. 13, no. 2, pp. 179–192, 2008.

[5] P. G. Bello, I. Aedo, F. Sainz, P. Diaz, J. Munnelly, S. Clarke. Improving communication for mobile devices in disaster response. In *Proceedings of the 1st International Conference on Mobile Information Technology for Emergency Response, Lecture Notes in Computer Science*, Springer, vol. 4458, pp. 126–134, 2007.

[6] F. R. Yu, J. Zhang, H. Tang, H. C. B. Chan, V. C. M. Leung. Enhancing interoperability in heterogeneous mobile wireless networks for disaster response. In *Proceedings of IEEE Military Communications Conference*, Orlando, Florida, USA, pp. 1–7, 2007.

[7] Télécoms Sans Frontières. *Emergency Communications Aid Disaster Response*, Geneva: International Telecommunications Union, pp. 34, 2005.

[8] S. Thompson, N. Altay, W. G. Green III, J. Lepetina. Improving disaster response efforts with decision support systems. *International Journal of Emergency Management*, vol. 3, no. 4, pp. 250–263, 2006.

[9] Y. L. Tu, W. J. Zhang, X. Liu, W. Li, C. L. Chai, R. Deters. A disaster response management system based on the control systems technology. *International Journal of Critical Infrastructures*, vol. 4, no. 3, pp. 274–295, 2008.

[10] Z. W. Hu, X. J. Li, Y. H. Sun, L. Zhu. Flood disaster response and decision-making support system based on remote sensing and GIS. In *Proceedings of International Geoscience and Remote Sensing Symposium*, Barcelona, Spain, pp. 2435–2438, 2008.

[11] F. Fiedrich, P. Burghardt. Agent-based systems for disaster management. *Communications of the ACM*, vol. 50, no. 3, pp. 41–42,2007.

[12] L. L. Yang, R. Prasanna, M. King. On-site information systems design for emergency first responders. *Journal of Information Technology Theory and Application*, vol. 10, no. 1, pp. 5–27, 2009.

[13] M. Turoff, M. Chumer, B. Van de Walle, X. Yao. The design of a dynamic emergency response management information system. *Journal of Information Technology Theory and Application*, vol. 5, no. 4, pp. 1–36, 2004.

[14] A. F. Chandio, L. Y. Shu, N. M. Memon, A. Khawaja. GIS based route guiding system for optimal path planning in disaster/crisis management. In *Proceedings of the 10th IEEE International Multitopic Conference*, Islamabad, Pakistan, pp. 207–210, 2006.

[15] L. Ozdamar, W. Yi. Greedy neighborhood search for disaster relief and evacuation logistics. *IEEE Intelligent Systems*, vol. 13, no. 1, pp. 14–23, 2008.

[16] Y. C. Chiou, Y. H. Lai. An integrated multi-objective model to determine the optimal rescue path and traffic controlled arcs for disaster relief operations under uncertainty environments. *Journal of Advanced Transportation*, vol. 42, no. 4, pp. 493–519, 2008.

[17] Y. Yuan, D. W. Wang. Path selection model and algorithm for emergency logistics management. *Computers and Industrial Engineering*, vol. 56, no. 3, pp. 1081–1094, 2009.

[18] E. G. Lopez. Efficient graph-based genetic programming representation with multiple outputs. *International Journal of Automation and Computing*, vol. 5, no. 1, pp. 81–89, 2008.

[19] L. P. Kaelbling, M. L. Littman, A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, vol. 4, no. 1, pp. 237–285, 1996.

[20] M. Kumar, A. K. Verma, A. Srividya. Analyzing effect of demand rate on safety of systems with periodic proof-tests. *International Journal of Automation and Computing*, vol. 4, no. 4, pp. 335–341, 2007.

[21] C. J. C. H. Watkins, P. Dayan. Technical note: Q-learning. *Machine Learning*, vol. 8, no. 3–4, pp. 279–292, 1992.

[22] V. S. Borkar. Q-learning for risk-sensitive control, *Mathematics of Operations Research*, vol. 27, no. 2, pp. 294–311, 2002.

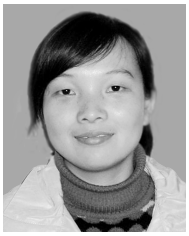
[23] M. Andrecut, M. K. Ali. Q learning in the minority game. *Physical Review E*, vol. 64, no. 6, pp. 1–4, 2001.

[24] J. G. Jiang, Z. P. Su, M. B. Qi, G. F. Zhang. Multi-task coalition parallel formation strategy based on reinforcement learning. *Acta Automatica Sinica*, vol. 34, no. 3, pp. 349–352, 2008.

[25] S. M. Lucas. Computational intelligence and games: Challenges and opportunities. *International Journal of Automation and Computing*, vol. 5, no. 1, pp. 45–57, 2008.

[26] S. Q. Yu, H. Q. Wang, F. M. Ye, S. Mabu, K. Shimada, K. Hirasawa. A Q value-based dynamic programming algorithm with Boltzmann distribution for optimizing the global traffic routing strategy. In *Proceedings of SICE Annual Conference*, Tokyo, Japan, pp. 619–622, 2008.

[27] K. Takadama, T. Kawai, Y. Koyama. Can agents acquire human-like behaviors in a sequential bargaining game? — Comparison of Roth′s and Q-learning agents. In *Proceedings of the 7th International Workshop on Multi-agent-based Simulation*, Hakodate, Japan, pp. 156–171, 2006.



Zhao-Pin Su received the B. Sc. and Ph. D. degrees in computer science from Hefei University of Technology, Hefei, PRC in 2004 and 2008, respectively. Currently, she is a lecturer in the School of Computer and Information, Hefei University of Technology. Also, she is now working together with Guo-Fu Zhang to model and solve disaster response coalition formation for unconventional emergency at Postdoctoral Research Station for Management Science and Engineering in Hefei University of Technology.

Her research interests include autonomous agent, reinforcement learning, and immune algorithm.

E-mail: szp@hfut.edu.cn (Corresponding author)

Jian-Guo Jiang received the M. Sc. degree in computer science from Hefei University of Technology, Hefei, PRC in 1989. He is currently a professor in the School of Computer and Information, Hefei University of Technology. He is head of the Texas Instruments-Hefei University of Technology DSPS Laboratory in Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education.

His research interests include automatic control, image processing, and software engineering.

E-mail: jgjiang@hfut.edu.cn

Chang-Yong Liang received the Ph. D. degree from Harbin Institute of Technology, PRC in 2001. He is currently a professor in the School of Management, Hefei University of Technology, PRC.

His research interests include collaborative filtering and intelligent decision support system.

E-mail: cyliang@163.com

Guo-Fu Zhang received the B. Sc. and Ph. D. degrees in computer science from Hefei University of Technology, Hefei, PRC in 2002 and 2008, respectively. He is currently a lecturer in the School of Computer and Information, Hefei University of Technology.

His research interests include evolutionary computation, intelligent agent, and multi-agent systems, especially in coalition formation.

E-mail: zgf@hfut.edu.cn