Superposition-Inspired Reinforcement Learning and Quantum Reinforcement Learning

Chun-Lin Chen and Dao-Yi Dong
Nanjing University & Institute of Systems Science, CAS
China

1. Introduction

Reinforcement Learning (RL) remains an active research area for a long time (Kaelbling et al., 1996; Sutton & Barto, 1998) and is still one of the most rapidly developing machine learning methods in recent years (Barto & Mahadevan, 2003). Related algorithms and techniques have been used in different applications such as motion control, operations research, robotics and sequential decision process (He & Jagannathan, 2005; Kondo & Ito, 2004; Morimoto & Doya, 2001; Chen et al., 2006b). However how to speed up learning has always been one of the key problems for the theoretical research and applications of RL methods (Sutton & Barto, 1998).

Recently there comes up a new approach for solving this problem owning to the rapid development of quantum information and quantum computation (Preskill, 1998; Nielsen & Chuang, 2000). Some results have shown that quantum computation can efficiently speed up the solutions of some classical problems, and even can solve some difficult problems that classical algorithms can not solve. Two important quantum algorithms, Shor's factoring algorithm (Shor, 1994; Ekert & Jozsa, 1996) and Grover's searching algorithm (Grover, 1996; Grover, 1997), have been proposed in 1994 and 1996 respectively. Shor's factoring algorithm can give an exponential speedup for factoring large integers into prime numbers and its experimental demonstration has been realized using nuclear magnetic resonance (Vandersypen et al., 2001). Grover's searching algorithm can achieve a square speedup over classical algorithms in unsorted database searching and its experimental implementations have also been demonstrated using nuclear magnetic resonance (Chuang et al., 1998; Jones, 1998a; Jones et al., 1998b) and quantum optics (Kwiat et al., 2000; Scully & Zubairy, 2001). Taking advantage of quantum computation, the algorithm integration inspired by quantum characteristics will not only improve the performance of existing algorithms on traditional computers, but also promote the development of related research areas such as quantum computer and machine learning. According to our recent research results (Dong et al., 2005a; Dong et al., 2006a; Dong et al., 2006b; Chen et al., 2006a; Chen et al., 2006c; Chen & Dong, 2007; Dong et al., 2007a; Dong et al., 2007b), in this chapter the RL methods based on quantum theory are introduced following the developing roadmap from Superposition-Inspired Reinforcement Learning (SIRL) to Quantum Reinforcement Learning (QRL).

As for SIRL methods we concern mainly about the exploration policy. Inspired by the superposition principle of quantum state, in a RL system, a probabilistic exploration policy

Source: Reinforcement Learning: Theory and Applications, Book edited by Cornelius Weber, Mark Elshaw and Norbert Michael Mayer
ISBN 978-3-902613-14-1, pp.424, January 2008, I-Tech Education and Publishing, Vienna, Austria

is proposed to mimic the state collapse phenomenon according to quantum measurement postulate, which leads to a good balance between exploration and exploitation. In this way, the simulated experiments show that SIRL may accelerate the learning process and allow avoiding the locally optimal policies.

When SIRL is extended to quantum mechanical systems, QRL theory is proposed naturally (Dong et al., 2005a, Dong et al., 2007b). In a QRL system, the state value can be represented with quantum state and be obtained by randomly observing the quantum state, which will lead to state collapse according to quantum measurement postulate. The occurrence probability of eigenvalue is determined by probability amplitude, which is updated according to rewards. So this approach represents the whole state-action space with the superposition of quantum state, which leads to real parallel computing and a good tradeoff between exploration and exploitation using probability as well.

Besides the introduction of SIRL and QRL methods, in this chapter, the relationship between different theories and algorithms are briefly analyzed, and their applications are also introduced respectively. The organization of this chapter is as follows. Section 2 gives a brief introduction to the fundamentals of quantum computation, which include the superposition principle, parallel computation and quantum gates. In Section 3, the SIRL method is presented in a probabilistic version through mimicking the quantum behaviors. Section 4 gives the introduction of QRL method based on quantum superposition and quantum parallelism. Related issues and future work are discussed as a conclusion in Section 5.

2. Fundamentals of quantum computation

2.1 State superposition and quantum parallel computation

In quantum computation, information unit (also called as qubit) is represented with quantum state and a qubit is an arbitrary superposition state of two-state quantum system (Dirac's representation) (Preskill, 1998):

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1}$$

where α and β are complex coefficients and satisfy $|\alpha|^2 + |\beta|^2 = 1$. $|0\rangle$ and $|1\rangle$ are two orthogonal states (also called basis vectors of quantum state $|\psi\rangle$), and they correspond to logic states 0 and 1. $|\alpha|^2$ represents the occurrence probability of $|0\rangle$ when the qubit is measured, and $|\beta|^2$ is the probability of obtaining result $|1\rangle$. The physical carrier of a qubit is any two-state quantum system such as two-level atom, spin-1/2 particle and polarized photon. The value of classical bit is either Boolean value 0 or value 1, but a qubit can be prepared in the coherent superposition state of 0 and 1, i.e. a qubit can simultaneously store 0 and 1, which is the main difference between classical computation and quantum computation.

According to quantum computation theory, the quantum computing process can be looked upon as a unitary transformation U from input qubits to output qubits. If one applies a transformation U to a superposition state, the transformation will act on all basis vectors of this superposition state and the output will be a new superposition state by superposing the

results of all basis vectors. So when one processes function $f(x)$ by the method, the transformation U can simultaneously work out many different results for a certain input x. This is analogous with parallel process of classical computer and is called quantum parallelism. The powerful ability of quantum algorithm is just derived from the parallelism of quantum computation.
Suppose the input qubit $|z\rangle$ lies in the superposition state:

$$|z\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{2}$$

The transformation U_z describing computing process is defined as the following:

$$U_z : |z, y\rangle \to |z, y \oplus f(z)\rangle \tag{3}$$

where $|z, y\rangle$ represents the input joint state and $|z, y \oplus f(z)\rangle$ is the output joint state. Let $y = 0$ and we can easily obtain (Nielsen & Chuang, 2000):

$$U_z |z\rangle = \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle) \tag{4}$$

The result contains information about both $f(0)$ and $f(1)$, and we seem to evaluate $f(z)$ for two values of z simultaneously.
Now consider an n-qubit cluster and it lies in the following superposition state:

$$|\psi\rangle = \sum_{x=00\cdots0}^{\overbrace{11\cdots1}^{n}} C_x |x\rangle \qquad (\text{where } \sum_{x=00\cdots0}^{\overbrace{11\cdots1}^{n}} |C_x|^2 = 1) \tag{5}$$

where C_x is complex coefficients and $|C_x|^2$ represents occurrence probability of $|x\rangle$ when state $|\psi\rangle$ is measured. $|x\rangle$ can take on 2^n values, so the superposition state can be looked upon as the superposition state of all integers from 0 to $2^n - 1$. Since U is a unitary transformation, computing function $f(x)$ can give (Preskill, 1998):

$$U \sum_{x=00\cdots0}^{\overbrace{11\cdots1}^{n}} C_x |x, 0\rangle = \sum_{x=00\cdots0}^{\overbrace{11\cdots1}^{n}} C_x U |x, 0\rangle = \sum_{x=00\cdots0}^{\overbrace{11\cdots1}^{n}} C_x |x, f(x)\rangle \tag{6}$$

Based on the above analysis, it is easy to find that an n-qubit cluster can simultaneously process 2^n states. However, this is different from the classical parallel computation, where multiple circuits built to compute $f(x)$ are executed simultaneously, since quantum parallel computation doesn't necessarily make a tradeoff between computation time and needed physical space. In fact, quantum parallelism employs a single circuit to evaluate the function for multiple values of x simultaneously by exploiting the quantum state superposition principle and provides an exponential-scale computation space in the n-qubit linear physical space. Therefore quantum computation can effectively increase the computing speed of some important classical functions. So it is possible to obtain significant result through fusing quantum computation into reinforcement learning theory.

2.2 Quantum gates

Analogous to classical computer, quantum computer accomplishes some quantum computation tasks through quantum gates. A quantum gate or quantum logic gate is a basic quantum circuit operating on a small number of qubits. They can be represented by unitary matrices. Here we will introduce several simple quantum gates including quantum NOT gate, Hadamard gate, phase gate and quantum CNOT gate. The detailed description of quantum gates can refer to (Nielsen & Chuang, 2000).

A quantum NOT gate maps $|0\rangle \to |1\rangle$ and $|1\rangle \to |0\rangle$ respectively and that can be described by the following matrix:

$$\mathrm{U}_{\mathrm{NOT}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{7}$$

When a quantum NOT gate is applied on a single qubit with state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, then the output will become $|\psi\rangle = \alpha|1\rangle + \beta|0\rangle$. The symbol for the NOT gate is drawn in Fig.1 (a).

The Hadamard gate is one of the most useful quantum gates and can be represented as:

$$\mathrm{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{8}$$

Through the Hadamard gate, a qubit in the state $|0\rangle$ is transformed into a superposition state in the two states, i.e.

$$\mathrm{H}|0\rangle \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{9}$$

Another important gate is phase gate which can be expressed as

$$\mathrm{U}_{\mathrm{p}} = \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{bmatrix} \tag{10}$$

U_{p} generates a relative phase π between the two basis states of the input state, i.e.

$$\mathrm{U}_{\mathrm{p}}|\psi\rangle = \alpha|0\rangle + \mathrm{i}\beta|1\rangle \tag{11}$$

The CNOT gate acts on two qubits simultaneously and can be represented by the following matrix:

$$U_{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{12}$$

The symbol for the CNOT gate is shown as in Fig.1 (b). If the first control qubit is equal to $|1\rangle$, then CNOT gate flips the target (second) qubit. Otherwise the target remains unaffected. This can be described as follows:

$$\begin{cases} U_{CNOT}|00\rangle = |00\rangle \\ U_{CNOT}|01\rangle = |01\rangle \\ U_{CNOT}|10\rangle = |11\rangle \\ U_{CNOT}|11\rangle = |10\rangle \end{cases} \tag{13}$$

Just like AND and NOT form a universal set for classical boolen circuits, the CNOT gate combined with one qubit rotation gate can implement any kind of quantum calculation.

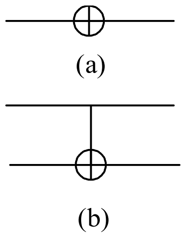

Fig. 1. Symbols for NOT and CNOT gate

3. Superposition-inspired reinforcement learning

Similar the standard RL, SIRL is also a RL method that is designed for the traditional computer, instead of a quantum algorithm. However, it borrows the ideas from quantum characteristics and provides an alternative exploration strategy, i.e., action selection method. In this section, the SIRL will be presented after a brief introduction of the standard RL theory and the existing exploration strategies.

3.1 Reinforcement learning and exploration strategy

Standard framework of RL is based on discrete-time, finite Markov decision processes (MDPs) (Sutton & Barto, 1998). RL algorithms assume that state S and action $A_{(s_n)}$ can be divided into discrete values. At a certain step, the agent observes the state of the

environment (inside and outside of the agent) s_t, and then choose an action a_t. After executing the action, the agent receives a reward r_{t+1}, which reflects how good that action is (in a short-term sense).
The goal of reinforcement learning is to learn a mapping from states to actions, that is to say, the agent is to learn a policy $\pi : S \times \cup_{i\in S} A_{(i)} \to [0,1]$, so that expected sum of discounted reward of each state will be maximized:

$$\begin{aligned} V^{\pi}_{(s)} &= E\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots \mid s_t = s, \pi\} \\ &= E[r_{t+1} + \gamma V^{\pi}_{(s_{t+1})} \mid s_t = s, \pi] \\ &= \sum_{a\in A_s} \pi(s,a)[r_s^a + \gamma \sum_{s'} p_{ss'}^a V^{\pi}_{(s')}] \end{aligned} \tag{14}$$

where $\gamma \in [0,1)$ is discounted factor, $\pi(s,a)$ is the probability of selecting action a according to state s under policy π, $p_{ss'}^a = \Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\}$ is probability for state transition and $r_s^a = E\{r_{t+1} \mid s_t = s, a_t = a\}$ is expected one-step reward. Then we have the optimal state-value function

$$V^*_{(s)} = \max_{a\in A_s}[r_s^a + \gamma \sum_{s'} p_{ss'}^a V^*_{(s')}] \tag{15}$$

$$\pi^* = \arg\max_{\pi} V^{\pi}_{(s)}, \quad \forall s \in S \tag{16}$$

In dynamic programming, (15) is also called Bellman equation of V^*.
As for state-action pairs, there are similar value functions and Bellman equations, where $Q^{\pi}(s,a)$ stands for the value of taking action a in state s under policy π:

$$\begin{aligned} Q^{\pi}_{(s,a)} &= E\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots \mid s_t = s, a_t = a, \pi\} \\ &= r_s^a + \gamma \sum_{s'} p_{ss'}^a V^{\pi}(s') \\ &= r_s^a + \gamma \sum_{s'} p_{ss'}^a \sum_{a'} \pi(s',a') Q^{\pi}_{(s',a')} \end{aligned} \tag{17}$$

$$Q^*_{(s,a)} = \max_{\pi} Q_{(s,a)} = r_s^a + \gamma \sum_{s'} p_{ss'}^a \max_{a'} Q^*_{(s',a')} \tag{18}$$

Let α be the learning rate, the one-step update rule of Q-learning (a widely used reinforcement learning algorithm) (Watkins & Dayan, 1992) is:

$$Q(s_t,a_t) \leftarrow (1-\alpha)Q(s_t,a_t) + \alpha(r_{t+1} + \gamma \max_{a'} Q(s_{t+1},a')) \tag{19}$$

Besides Q-learning, there are also many other RL algorithms such as temporal difference (TD), SARSA and multi-step version of these algorithms. For more detail, please refer to (Sutton & Barto, 1998).
To approach the optimal policy effectively and efficiently, the RL algorithms always need a certain exploration strategy. One widely used exploration strategy is ε-greedy $(\varepsilon \in [0,1))$, where the optimal action is selected with probability $1-\varepsilon$ and a random action is selected with probability ε. Sutton and Barto (Sutton & Barto, 1998) have compared the performance of RL for different ε, which shows that a nonzero ε is usually better than $\varepsilon = 0$ (i.e., blind greedy strategy). Moreover, the exploration probability ε can be reduced over time, which moves the agent from exploration to exploitation. The ε-greedy method is simple and effective, but it has one drawback that when it explores it chooses equally among all actions. This means that it makes no difference to choose the worst action or the next-to-best action. Another problem is that it is difficult to choose a proper parameter ε which can offer the optimal balancing between exploration and exploitation.
Another kind of action selection methods are randomized strategies, such as Boltzmann exploration (i.e., Softmax method) (Sutton & Barto, 1998) and Simulated Annealing (SA) method (Guo et al., 2004). It uses a positive parameter τ called the temperature and chooses action with the probability proportional to $\exp(Q_{(s,a)}/\tau)$. Compared with ε-greedy method, the greedy action is still given the highest selection probability, but all the others are ranked and weighted according to their value estimates. It can also move from exploration to exploitation by adjusting the "temperature" parameter τ. It is natural to sample actions according to this distribution, but it is very difficult to set and adjust a good parameter τ and may converge unnecessarily slowly unless the parameter τ is manually tuned with great care. It also has another potential shortcoming that it may works badly when the values of the actions are close and the best action can not be separated from the others. A third problem is that when the parameter τ is reduced over time to acquire more exploitation, there is no effective mechanism to guarantee re-exploration when necessary.
Therefore, the existing exploration strategies usually suffer from the difficulties to hold the good balancing between exploration and exploitation and to provide an easy method of parameter setting. Hence new ideas are necessary to explore more effective exploration strategies to achieve better performance. Inspired by the main characteristics of quantum computation, we present the SIRL algorithm with a probabilistic exploration policy.

3.2 Superposition-inspired RL

The exploration strategy for SIRL is inspired by the state superposition principle of a quantum system and collapse postulate, where a combined action form is adopted to provide a probabilistic mechanism for each state in the SIRL system. At state s, the action to be selected is represented as:

$$a_s = f(s) = \frac{c_1}{a_1} + \frac{c_2}{a_2} + \ldots + \frac{c_m}{a_m} = \sum_{i=1}^{m} \frac{c_i}{a_i} \qquad (20)$$

Where $\sum_{i=1}^{m} c_i = 1$, $0 \le c_i \le 1$, $i = 1,2,...m$. a_s is the action to be selected at state s and the action selection set is $\{a_1, a_2, ..., a_m\}$. Equation (20) is not for numerical computation and it just means that at the state s, the agent will choose the action a_i with the occurrence probability c_i, which leads to a natural exploration strategy for SIRL.

After the execution of action a_i from state s, the corresponding probability c_i is updated according to the immediate reward r and the estimated value of the next state $V(s')$.

$$c_i \leftarrow c_i + k(r + V(s')) \tag{21}$$

where k is the updating step and the probability distribution $(c_1, c_2, ..., c_m)$ is normalized after each updating process. The procedural algorithm of standard SIRL is shown as in Fig. 2.

```
Procedural SIRL:
Initialize $V(s)$ arbitrarily, $\pi$ to the policy to be evaluated
        $\pi : a_s = f(s) = \frac{c_1}{a_1} + \frac{c_2}{a_2} + ... + \frac{c_m}{a_m} = \sum_{i=1}^{m} \frac{c_i}{a_i}$
  Repeat (for each episode):
    Initialize $s$
    Repeat (for each step of episode):
        $a \leftarrow$ action given by $\pi$ for $s$
        Take action $a$ : observe reward, $r$, and next state, $s'$
          $V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$
          $c_i \leftarrow c_i + k(r + V(s'))$
          $s \leftarrow s'$
    until $s$ is terminal
  until the learning process ends
```

Fig. 2. A standard SIRL algorithm

In the SIRL algorithm, the exploration policy is accomplished through a probability distribution over the action set. When the agent is going to choose an action at a certain state, the action a_i will be selected with probability c_i, which is also updated along with the value funcion updating. Comparing the SIRL algorithm with basic RL algorithms, the main difference is that with the probabilistic exploration policy, the SIRL algorithm makes better tradeoff between exporation and exploitation without bothering to tune it by the designers.

3.3 Simulated experiments

The performance of the SIRL algorithm is tested with two examples, which are a puzzle problem and a mobile robot navigation problem.

1. The puzzle problem

First, let's consider a puzzle problem as shown in Fig. 3, which is in a $13\times13(0\sim12)$ gridworld environment. From any state the agent can perform one of four primary actions: up, down, left and right, and actions that would lead into a blocked cell are not executed. The task is to find an optimal policy which will let the agent move from *S*(11,1) to *G*(1,11) with minimized cost (number of moving steps).

The experiment setting is as follows. Once the agent finds the goal state it receives a reward of 100 and then ends this episode. All steps are punished by a reward of -1. The discount factor γ is set to 0.99 for all the algorithms that we have carried out in this example. In this experiment, we compare the proposed method with TD algorithm. For the action selection policy of TD algorithm, we use ε-greedy policy (ε = 0.01). As for SIRL method, the action selecting policy uses the values of c_i to denote the probability of an action, which is defined as $a_s = f(s) = \frac{c_1}{a_1} + \frac{c_2}{a_2} + \ldots + \frac{c_m}{a_m} = \sum_{i=1}^{m} \frac{c_i}{a_i}$. For the four cell-to-cell actions c_i is initialized uniformly.

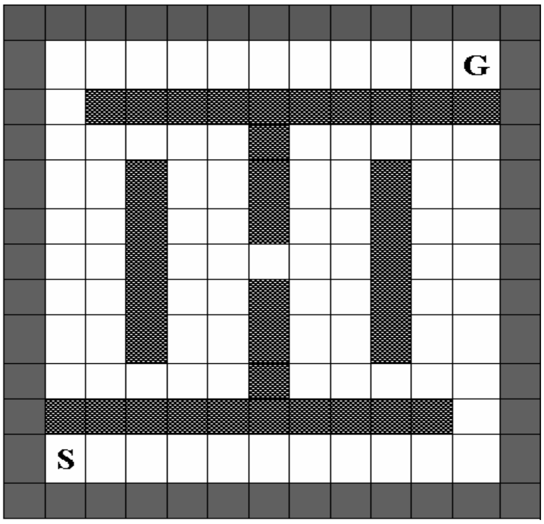

Fig. 3. A puzzle problem. The task is to move from start (S) to goal (G) with minimum number of steps

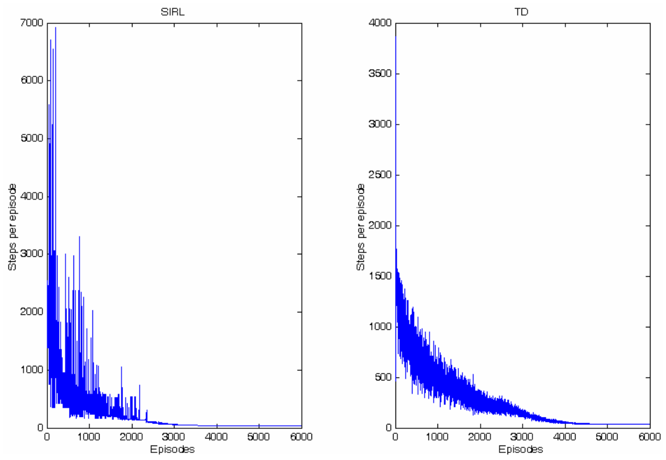

Fig. 4. Performance of SIRL (the left figure) compared with TD algorithm (the right figure)

The experimental results of the SIRL method compared with TD method are plotted in Fig. 4. It is obvious that at the beginning phase SIRL with this superposition-inspired exploration strategy learns extraordinarily fast, and then steadily converges to the optimal policy that costs 40 steps to the goal *G*. The results show that the SIRL method makes a good tradeoff between exploration and exploitation.

2. Mobile robot navigation

A simulation environment has also been set up with a larger grid-map of 400×600. And the configuration of main parameters is as follows: learning rate $\alpha = 0.5$, discount factor $\gamma = 0.9$. Fig. 5. shows the result in complex indoor environment, which verifies the effectiveness of robot learning using SIRL for navigation in large unknown environments.

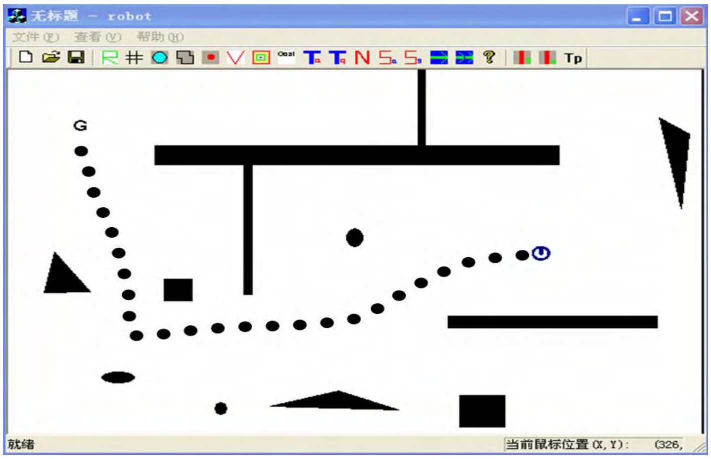

Fig. 5. Simulation result of robot navigation in indoor environment

4. Quantum reinforcement learning

When the SIRL is applied to a real quantum system, for example, to run the algorithm on a quantum computer, the representation and the computation mode will be dramatically different, which will lead to quantum reinforcement learning (QRL). Then we can take the most advantages of this quantum algorithm, such as the speeding up due to quantum parallel computation.

4.1 Representation

One of the most fundamental principles of quantum mechanics is the state superposition principle. As we represent a QRL system with quantum concepts, similarly, we have the following definitions and propositions for QRL.

Definition 1: (Eigenvalue of states or actions) States s or actions a in a RL system are denoted as corresponding orthogonal quantum states $|\mathrm{s}_\mathrm{n}\rangle$ (or $|\mathrm{a}_\mathrm{n}\rangle$) and are called the eigenvalue of states or actions in QRL.

Then we get the set of eigenvalues of states: $\mathrm{S} = \{|\mathrm{s}_\mathrm{n}\rangle\}$ and that of actions for state i: $\mathrm{A}_{(\mathrm{i})} = \{|\mathrm{a}_\mathrm{n}\rangle\}$.

Corollary 1: Every possible state $|\mathrm{s}\rangle$ or action $|\mathrm{a}\rangle$ can be expanded in terms of an orthogonal complete set of functions, respectively. We have

$$|s\rangle = \sum_n \beta_n |s_n\rangle \tag{22}$$

$$|a\rangle = \sum_n \beta_n |a_n\rangle \tag{23}$$

where β_n is probability amplitude, which can be a complex number, $|s_n\rangle$ and $|a_n\rangle$ are eigenvalues of states and actions, respectively. And the β_n in equation (22) is not necessarily the same as the ones in equation (23), which just mean this corollary holds for both of $|s\rangle$ and $|a\rangle$. $|\beta_n|^2$ means the probability of corresponding eigenvalues and satisfies

$$\sum_n |\beta_n|^2 = 1 \tag{24}$$

Proof: (sketch)

(1) State space $\{|s\rangle\}$ in QRL system is a N-dimension Hilbert space,

(2) States $\{|s_n\rangle\}$ in traditional RL system are the eigenvalue of states $|s\rangle$ in QRL system, (Definition 1)

Then $\{|s_n\rangle\}$ are N linear independent vectors for this N-dimension Hilbert space, according to the definition of Hilbert space, any possible state $|s\rangle$ can be expanded in terms of the complete set of $|s_n\rangle$. And it is the same for action space $\{|a\rangle\}$.

So the states and actions in QRL are different from those in traditional RL.

1. The sum of several states (or actions) does not have a definite meaning in traditional RL, but the sum of states (or actions) in QRL is still a possible state (or action) of the same quantum system, and it will simultaneously take on the superposition state of some eigenvalues.
2. The measurement value of $|s\rangle$ relates to its probability density. When $|s\rangle$ takes on an eigenstate $|s_i\rangle$, its value is exclusive. Otherwise, its value has the probability of $|\beta_i|^2$ to be one of the eigenstate $|s_i\rangle$.

Like what has been described in Section 2, quantum computation is built upon the concept of qubit. Now we consider the systems of multiple qubits and propose a formal representation of them for QRL system.

Let N_s and N_a be the numbers of states and actions respectively, then choose numbers *m* and *n*, which are characterized by the following inequalities:

$$N_s \le 2^m \le 2N_s,\ N_a \le 2^n \le 2N_a \tag{25}$$

And use *m* and *n* qubits to represent eigenstate set $S = \{s\}$ and eigenaction set $A = \{a\}$ respectively:

$$s:\begin{bmatrix} a_1 \\ b_1 \end{bmatrix}\!\!\begin{matrix} a_2 \\ b_2 \end{matrix}\Bigg|\cdots\Bigg|\begin{matrix} a_m \\ b_m \end{matrix}\Bigg],\ \text{where } |a_i|^2+|b_i|^2=1,\ i=1,2,...m$$

$$a:\begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix}\!\!\begin{matrix} \alpha_2 \\ \beta_2 \end{matrix}\Bigg|\cdots\Bigg|\begin{matrix} \alpha_n \\ \beta_n \end{matrix}\Bigg],\ \text{where } |\alpha_i|^2+|\beta_i|^2=1,\ i=1,2,...n$$

Thus the states and actions of a QRL system may lie in superposition states:

$$|s^{(m)}\rangle = \sum_{s=00\cdots0}^{\overbrace{11...1}^{m}} C_s\,|s\rangle \tag{26}$$

$$|a^{(n)}\rangle = \sum_{a=00\cdots0}^{\overbrace{11...1}^{n}} C_a |a\rangle \tag{27}$$

where C_s and C_a can be complex numbers and satisfy

$$\sum_{s=00\cdots0}^{\overbrace{11...1}^{m}} |C_s|^2 = 1 \tag{28}$$

$$\sum_{a=00\cdots0}^{\overbrace{11...1}^{n}} |C_a|^2 = 1 \tag{29}$$

4.2 Action selection policy

In QRL, the agent is also to learn a policy $\pi\ : S \times \cup_{i\in S} A_{(i)} \to [0,1]$, which will maximize the expected sum of discounted reward of each state. That is to say, the mapping from states to actions is $f(s) = \pi : S \to A$, and we have

$$f(s) = |a_s^{(n)}\rangle = \sum_{a=00\cdots0}^{\overbrace{11...1}^{n}} C_a |a\rangle \tag{30}$$

where C_a is probability amplitude of action $|a\rangle$ and satisfies (29).

Definition 2: (Collapse) When a quantum state $|\psi\rangle = \sum_n \beta_n |\psi_n\rangle$ is measured, it will be changed and collapse randomly into one $|\psi_n\rangle$ of its eigenstates with corresponding probability $|\langle\psi_n|\psi\rangle|^2$:

$$|\langle\psi_n|\psi\rangle|^2 = |(|\psi_n\rangle)^* |\psi\rangle|^2 = |\beta_n|^2 \tag{31}$$

Then when an action $|a_s^{(n)}\rangle$ is measured, we will get $|a\rangle$ with the occurrence probability of $|C_a|^2$. In QRL algorithm, we will amplify the probability of "good" action according to corresponding rewards. It is obvious that the *collapse* action selection method is not a real action selection method theoretically. It is just a fundamental phenomenon when a quantum state is measured, which results in a good balancing between exploration and exploitation and a natural "action selection" without setting parameters.

4.3 Value function updating and reinforcement strategy

In Corollary 1 we pointed out that every possible state of QRL $|s\rangle$ can be expanded in terms of an orthogonal complete set of eigenstate $|s_n\rangle$: $|s\rangle = \sum_n \beta_n |s_n\rangle$. If we use an *m*-qubit register, it will be $|s^{(m)}\rangle = \sum_{s=00\cdots0}^{\overbrace{11...1}^{m}} C_s |s\rangle$.

According to quantum parallel computation theory, a certain unitary transformation U from input qubit to output qubit can be implemented. Suppose we have such a "quantum black box" which can simultaneously process these 2^m states with the value updating rule

$$V(s) \leftarrow V(s) + \alpha(r + V(s') - V(s)) \tag{32}$$

where α is learning rate, and r is the immediate reward. It is like parallel value updating of traditional RL over all states, however, it provides an exponential-scale computation space in the *m*-qubit linear physical space and can speed up the solutions of related functions.

The reinforcement strategy is accomplished by changing the probability amplitudes of the actions according to the updated value function. As we know that action selection is executed by measuring action $|a_s^{(n)}\rangle$ related to certain state $|s^{(m)}\rangle$, which will collapse to $|a\rangle$ with the occurrence probability of $|C_a|^2$. So it is no doubt that probability amplitude updating is the key of recording the "trial-and-error" experience and learning to be more intelligent. When an action $|a\rangle$ is executed, it should be able to memorize whether it is "good" or "bad" by changing its probability amplitude C_a . For more details, please refer to (Chen et al., 2006a; Dong et al., 2006b; Dong et al., 2007b).

As action $|a_s^{(n)}\rangle$ is the superposition of *n* possible eigenactions, to find out $|a\rangle$ and to change its probability amplitudes are usually interactional for a quantum system. So we simply update the probability amplitude of $|a_s^{(n)}\rangle$ without searching $|a\rangle$, which is inspired by Grover's searching algorithm (Grover, 1996).

The updating of probability amplitude is based on Grover iteration. First, prepare the equally weighted superposition of all eigenactions

$$|a_0^{(n)}\rangle = \frac{1}{\sqrt{2^n}}\left(\sum_{a=00\cdots0}^{\overbrace{11...1}^{n}} |a\rangle\right) \tag{33}$$

This process can be done easily by applying the Hadamard transformation to each qubit of an initial state $|a = 0\rangle$. We know that $|a\rangle$ is an eigenaction and can get

$$\langle a \mid a_0^{(n)} \rangle = \frac{1}{\sqrt{2^n}} \tag{34}$$

Now assume the eigenaction to be reinforced is $| a_j \rangle$, and we can construct Grover iteration through combining two reflections U_{a_j} and $U_{a_0^{(n)}}$ (Preskill, 1998; Nielsen & Chuang, 2000)

$$U_{a_j} = I - 2 \mid a_j \rangle \langle a_j \mid \tag{35}$$

$$U_{a_0^{(n)}} = 2 \mid a_0^{(n)} \rangle \langle a_0^{(n)} \mid - I \tag{36}$$

where I is unitary matrix. U_{a_j} flips the sign of the action $| a_j \rangle$, but acts trivially on any action orthogonal to $| a_j \rangle$. This transformation has a simple geometrical interpretation. Acting on any vector in the 2^n-dimensional Hilbert space, U_{a_j} reflects the vector about the hyperplane orthogonal to $| a_j \rangle$. On the other hand, $U_{a_0^{(n)}}$ preserves $| a_0^{(n)} \rangle$, but flips the sign of any vector orthogonal to $| a_0^{(n)} \rangle$. Grover iteration is the unitary transformation

$$U_{Grov} = U_{a_0^{(n)}} U_{a_j} \tag{37}$$

By repeatedly applying the transformation U_{Grov} on $| a_0^{(n)} \rangle$, we can enhance the probability amplitude of the basis action $| a_j \rangle$ while suppressing the amplitude of all other actions. This can also be looked upon as a kind of rotation in two-dimensional space. Applying Grover iteration U_{Grov} for K times on $| a_0^{(n)} \rangle$ can be represented as

$$U_{Grov}^K \mid a_0^{(n)} \rangle = \sin((2K+1)\theta) \mid a_j \rangle + \cos((2K+1)\theta) \mid \phi \rangle \tag{38}$$

where $| \phi \rangle = \sqrt{\frac{1}{2^n - 1}} \sum_{a \neq a_j} | a \rangle$, θ satisfying $\sin\theta = 1/\sqrt{2^n}$. Through repeating Grover iteration, we can reinforce the probability amplitude of corresponding action according to the reward value.

Thus when an action $| a_0^{(n)} \rangle$ is executed, the probability amplitude of $| a_j \rangle$ is updated by carrying out $[k(r + V(s'))]$ (an integer) times of Grover iteration. k is a parameter and the probability amplitudes will be normalized with $\sum_a | C_a |^2 = 1$ after each updating.

4.4 Quantum reinforcement learning algorithm

The procedural form of a standard QRL algorithm is described as Fig. 6 (Dong et al., 2007b). QRL is inspired by the superposition principle of quantum state and quantum parallel computation. The state value can be represented with quantum state and be obtained by randomly observing the simulated quantum state, which will lead to state collapse according to quantum measurement postulate. And the occurrence probability of eigenvalue is determined by probability amplitude, which is updated according to rewards. So this approach represents the whole state-action space with the superposition of quantum state and makes a good tradeoff between exploration and exploitation using probability. The merit of QRL is twofold. First, as for simulation algorithm on traditional computer it is an effective algorithm with novel representation and computation methods. Second, the representation and computation mode are consistent with quantum parallel computation system and can speed up learning in exponential scale with quantum computer or quantum logic gates.

In this QRL algorithm we use temporal difference (TD) prediction for the state value updating, and TD algorithm has been proved to converge for absorbing Markov chain when the stepsize is nonnegative and digressive (Sutton & Barto, 1998; Watkins & Dayan, 1992). Since QRL is a stochastic iterative algorithm and Bertsekas and Tsitsiklis have verified the convergence of stochastic iterative algorithms (Bertsekas & Tsitsiklis, 1996), we give the convergence result about the QRL algorithm as Theorem 1. The proof and related discussions can be found in (Dong et al., 2006a; Chen et al., 2006c; Dong et al., 2007b):

Theorem 1: For any Markov chain, quantum reinforcement learning algorithm converges at the optimal state value function $V(s)^*$ with probability 1 under proper exploration policy when the following conditions hold (where α_k is stepsize and nonnegative):

$$\lim_{T\to\infty}\sum_{k=1}^{T}\alpha_k = \infty, \qquad \lim_{T\to\infty}\sum_{k=1}^{T}\alpha_k^2 < \infty \tag{39}$$

From the procedure of QRL in Fig. 6, we can see that the learning process of QRL is carried out through parallel computation, which also provides a mechanism of parallel updating. Sutton and Barto (Sutton & Barto, 1998) have pointed out that for the basic RL algorithms the parallel updating does not affect such performances of RL as learning speed and convergence in general. But we find that the parallel updating will speed up the learning process for the RL algorithms with a hierarchical setting (Sutton et al., 1999; Barto & Mahadevan, 2003; Chen et al., 2005), because the parallel updating rules give more chance to the updating of the upper level learning process and this experience for the agent can work as the “sub-goals” intrinsically that will speed up the lower learning process.

```
Procedure QRL:
Initialize $|s^{(m)}\rangle = \sum_{s=00\cdots0}^{11\cdots1} C_s |s\rangle$, $f(s) = |a_s^{(n)}\rangle = \sum_{a=00\cdots0}^{11\cdots1} C_a |a\rangle$ and $V(s)$ arbitrarily
    Repeat (for each episode)
        For all states $|s^{(m)}\rangle = \sum_{s=00\cdots0}^{11\cdots1} C_s |s\rangle$:
            1. Observe $f(s) = |a_s^{(n)}\rangle$ and get $|a\rangle$;
            2. Take action $|a\rangle$, observe next state $|s'\rangle$, reward $r$, then
                (a) Update state value: $V(s) \leftarrow V(s) + \alpha(r + \gamma V(s') - V(s))$
                (b) Update probability amplitudes:
                    repeat for $[k(r + V(s'))]$ times
                        $\mathrm{U}_{Grov} |a_s^{(n)}\rangle = U_{a_0^{(n)}} U_a |a_s^{(n)}\rangle$
    Until for all states $|\Delta V(s)| \le \varepsilon$.
```

Fig. 6. The algorithm of a standard QRL (Dong et al., 2007b)

4.5 Physical implementation

Now let's simply consider the physical realization of QRL and detailed discussion can be found in (Dong et al., 2006b). In QRL algorithm, the three main operations occur in preparing the equally weighted superposition state for calculating the times of Grover iteration, initializing the quantum system for representing states or actions, and carrying out a certain times of Grover iteration for updating probability amplitude according to reward value. In fact, we can initialize the quantum system by equally weighted superposition for representing states or actions. So the main operations required are preparing the equally weighted superposition state and carrying out Grover iteration. These can be implemented using the Hadamard transform and the conditional phase shift operation, both of which are relatively easy in quantum computation.

Consider a quantum system described by *n* qubits, it has 2^{n} possible states. To prepare an equally weighted superposition state, initially let each qubit lie in the state $|0\rangle$, then we can perform the transformation H on each qubit independently in sequence and thus change the state of the system. The state transition matrix representing this operation will be of dimension $2^{\mathrm{n}} \times 2^{\mathrm{n}}$ and it can be implemented by *n* shunt-wound Hadamard gates. This process can be represented into:

$$H^{\otimes n} | \overbrace{00\cdots0}^{n} \rangle = \frac{1}{\sqrt{2^n}} \sum_{a=00\cdots0}^{\overbrace{11\ldots1}^{n}} | a \rangle \qquad (40)$$

The other operation is the conditional phase shift operation which is an important element to carry out the Grover iteration. According to quantum information theory, this transformation may be efficiently implemented using phase gates on a quantum computer. The conditional phase shift operation does not change the probability of each state since the square of the absolute value of the amplitude in each state stays the same.

4.6 Simulated experiments

The presented QRL algorithm is also tested using two examples: Prisoner's Diploma and the control of a five-qubit system.

1. Prisoner's Diploma

The first example is derived from typical Prisoners' Dilemma. In the Prisoners' Dilemma, each of the two prisoners, prisoner I and prisoner II, must independently make the action selection to agree to give evidence against the other guy or to refuse to do so. The situation is as described in Table 1 with the entries giving the length of the prison sentence (years in prison) for each prisoner, in every possible situation. In this case, each of the prisoners is assumed to minimize his sentence. As we know, this play may lead to Nash equilibrium by giving the action selection *(agree to give evidence, agree to give evidence)* with the outcome of (3, 3) years in prison.

prisoner II / **Prisoner I**	**Agree to give evidence**	**Refuse to give evidence**
Agree	(3, 3)	(0, 5)
Refuse	(5, 0)	(1, 1)

Table 1. The Prisoners' Dilemma

Now, we assume that this Prisoners game can be played repeatedly. Each of them can choose to agree or refuse to give evidence against the other guy and the probabilities of the action selection *(agree to give evidence, agree to give evidence)* are initially equal. To find a better outcome, the two prisoners try to improve their action selection using learning. By applying the QRL method proposed in this chapter, we get the results as shown in Fig. 6 and Fig. 7 (Chen et al., 2006a; Chen et al., 2006c). From the results, it is obvious that the two prisoners get smarter when they try to cooperate indeliberately and both of them select the action of *"Refuse to give evidence"* after about 40 episodes of play. Then they steadily get the outcome of (1, 1) instead of (3, 3) (Nash equilibrium).

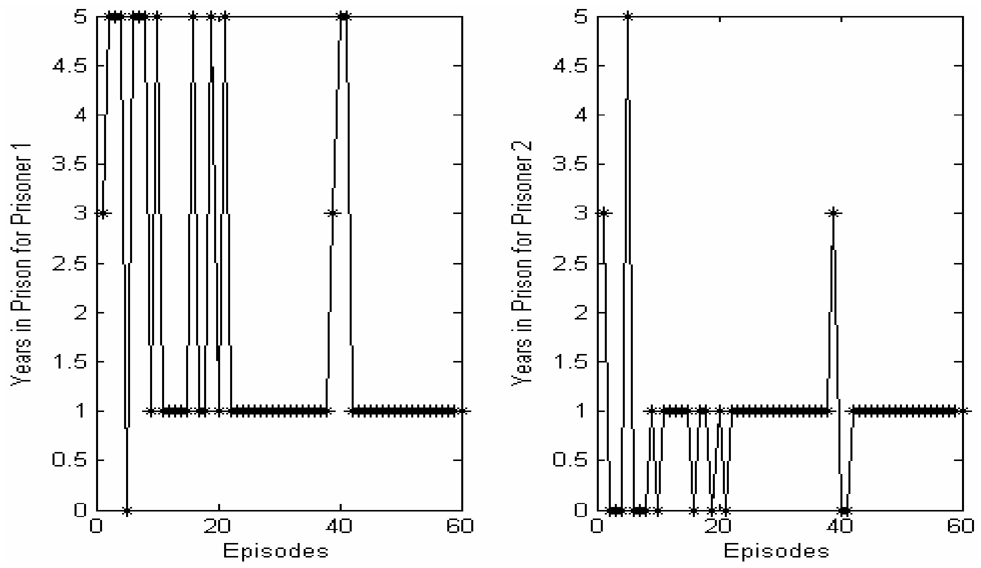

Fig. 6. The outcome (years in prison) of the Prisoners problem for each prisoner

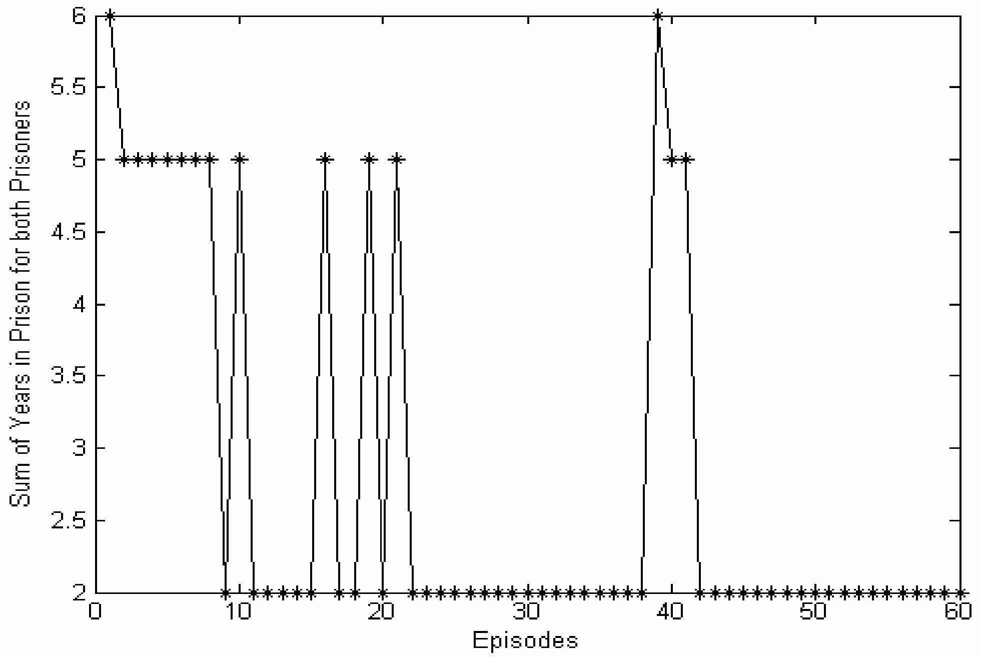

Fig. 7. The whole outcome of the Prisoners problem (Sum of years in prison for both prisoners)

2. Control of a five-qubit system

The second axample is about the control of a five-qubit system (Dong et al., 2006c). With the development of quantum information technology, quantum control theory has drawn the attention of many scientists (Chen et al., 2005). The objective of quantum control is to determine how to drive quantum systems from an initial given quantum state to a pre-determined target quantum state with some given time. According to quantum mechanics, the state $|\psi(\mathrm{t})\rangle$ of arbitary time *t* can be reached through an evolution on the initial state $|\psi(0)\rangle$. It can be expressed as

$$|\psi(\mathrm{t})\rangle = \hat{\mathrm{U}}\,|\psi(0)\rangle \tag{41}$$

where $\hat{\mathrm{U}}$ is a unitary operator and satisfies:

$$\hat{\mathrm{U}}\hat{\mathrm{U}}^{+} = \hat{\mathrm{U}}^{+}\hat{\mathrm{U}} = \mathrm{I} \tag{42}$$

where $\hat{\mathrm{U}}^{+}$ is the Hermitian conjugate operator of $\hat{\mathrm{U}}$. So the control problem of quantum state can be converted into finding appropriate unitary operator $\hat{\mathrm{U}}$.

In this example, we consider the five-qubit system, it has 32 eigenstates. In practical quantum information technology, some state transitions can easily be completed through appropriate unitary transformations but the other ones are not easy to be accomplished. Assume we know its state transitions satisfy the following equations through some experiments:

$|00001\rangle = \hat{\mathrm{U}}_{00}\,|00000\rangle$; $|00010\rangle = \hat{\mathrm{U}}_{01}\,|00001\rangle$; $|00011\rangle = \hat{\mathrm{U}}_{02}\,|00010\rangle$;

$|00100\rangle = \hat{\mathrm{U}}_{03}\,|00011\rangle$; $|00101\rangle = \hat{\mathrm{U}}_{04}\,|00100\rangle$; $|00111\rangle = \hat{\mathrm{U}}_{11}\,|00001\rangle$;

$|01000\rangle = \hat{\mathrm{U}}_{12}\,|00010\rangle$; $|01010\rangle = \hat{\mathrm{U}}_{14}\,|00100\rangle$; $|01011\rangle = \hat{\mathrm{U}}_{15}\,|00101\rangle$;

$|01000\rangle = \hat{\mathrm{U}}_{21}\,|00111\rangle$; $|01011\rangle = \hat{\mathrm{U}}_{24}\,|01010\rangle$; $|01101\rangle = \hat{\mathrm{U}}_{31}\,|00111\rangle$;

$|10000\rangle = \hat{\mathrm{U}}_{34}\,|01010\rangle$; $|10001\rangle = \hat{\mathrm{U}}_{35}\,|01011\rangle$; $|01101\rangle = \hat{\mathrm{U}}_{40}\,|01100\rangle$;

$|10001\rangle = \hat{\mathrm{U}}_{44}\,|10000\rangle$; $|10010\rangle = \hat{\mathrm{U}}_{50}\,|01100\rangle$; $|10110\rangle = \hat{\mathrm{U}}_{54}\,|10000\rangle$;

$|10111\rangle = \hat{\mathrm{U}}_{55}\,|10001\rangle$; $|10101\rangle = \hat{\mathrm{U}}_{62}\,|10100\rangle$; $|10110\rangle = \hat{\mathrm{U}}_{63}\,|10101\rangle$;

$|10111\rangle = \hat{\mathrm{U}}_{64}\,|10110\rangle$; $|11000\rangle = \hat{\mathrm{U}}_{70}\,|10010\rangle$; $|11100\rangle = \hat{\mathrm{U}}_{74}\,|10110\rangle$;

$|11101\rangle = \hat{\mathrm{U}}_{75}\,|10111\rangle$; $|11001\rangle = \hat{\mathrm{U}}_{80}\,|11001\rangle$; $|11101\rangle = \hat{\mathrm{U}}_{84}\,|11100\rangle$;

$|11111\rangle = \hat{\mathrm{U}}_{91}\,|11001\rangle$

In the above equations, $\hat{\mathrm{U}}$ is reversible operator. For example, we can easily get

$$|\,00000\rangle = \hat{\mathrm{U}}_{00}^{-1}\,|\,00001\rangle \tag{43}$$

Assume the other transitions are impossible except the above transitions and corresponding inverse transitions. If the initial state and the target state are $|\,11100\rangle$ and $|\,11111\rangle$ respectively, the following task is to find optimal control sequence through QRL.

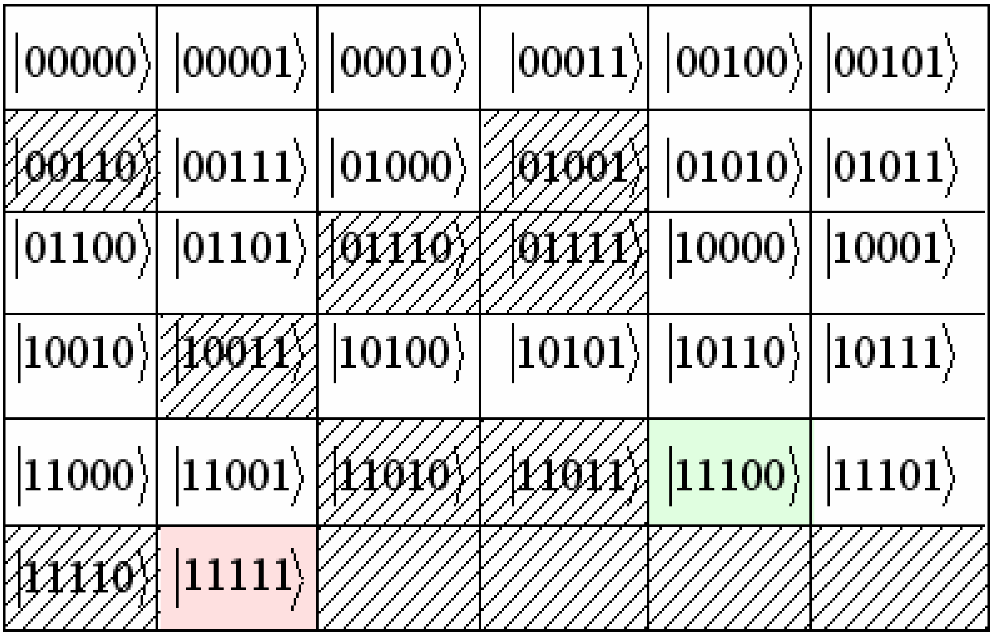

Fig. 8. The grid representation for the quantum control problem of a five-qubit system

Therefor we first fill the eigenstates of five-qubit system in a grid room and they can be described as shown in Fig. 8. Every eigenstate is arranged in a corresponding grid and the hatched grid indicates that the corresponding state can not be attained. The two states with a common side are mutually reachable through one-step control and other states can not directly reach each other through one-step control. Now the task of the quantum learning system is to find an optimal control sequence which will let the five-qubit system transform from $|\,11100\rangle$ to $|\,11111\rangle$. Using the QRL method proposed previously, we get the results as shown in Fig. 9. And more experimental results are shown in Fig. 10 to demonstrate its performance with different learning rates. From the results, it is obvious that the control system can robustly find the optimal control sequence for the five-qubit system through learning and the optimal control sequences are shown in Fig. 11. We can easily obtain two optimal control sequences from Fig. 11:

$$\text{Sequence\_1} = \{\hat{U}^{-1}_{74}, \hat{U}^{-1}_{54}, \hat{U}^{-1}_{34}, \hat{U}^{-1}_{14}, \hat{U}^{-1}_{03}, \hat{U}^{-1}_{02}, \hat{U}_{12}, \hat{U}^{-1}_{21}, \hat{U}_{31}, \hat{U}^{-1}_{40}, \hat{U}_{50}, \hat{U}_{70}, \hat{U}_{80}, \hat{U}_{91}\} \qquad (44)$$

$$\text{Sequence\_2} = \{\hat{U}^{-1}_{74}, \hat{U}^{-1}_{54}, \hat{U}^{-1}_{34}, \hat{U}^{-1}_{14}, \hat{U}^{-1}_{03}, \hat{U}^{-1}_{02}, \hat{U}^{-1}_{01}, \hat{U}_{11}, \hat{U}_{31}, \hat{U}^{-1}_{40}, \hat{U}_{50}, \hat{U}_{70}, \hat{U}_{80}, \hat{U}_{91}\} \qquad (45)$$

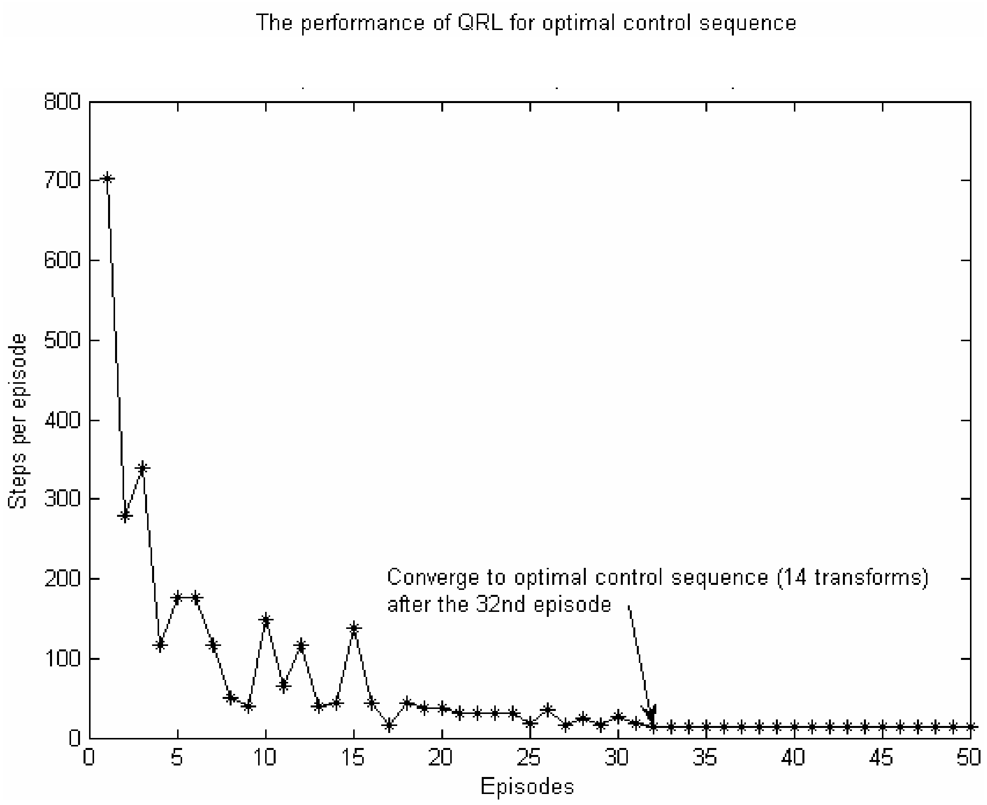

Fig. 9. The performance of QRL for optimal control sequence

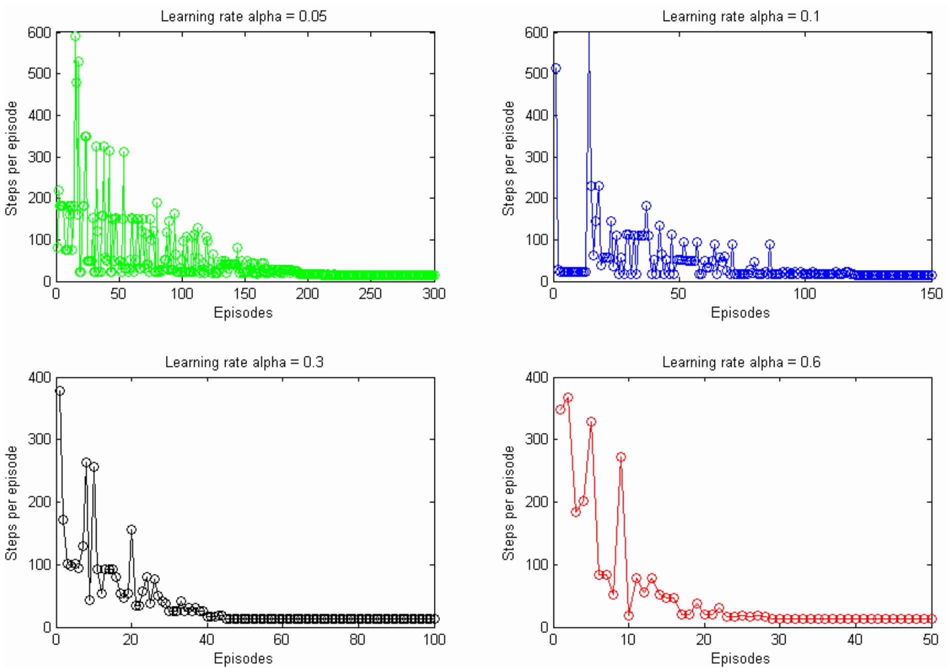

Fig. 10. The performance of QRL with different learning rates

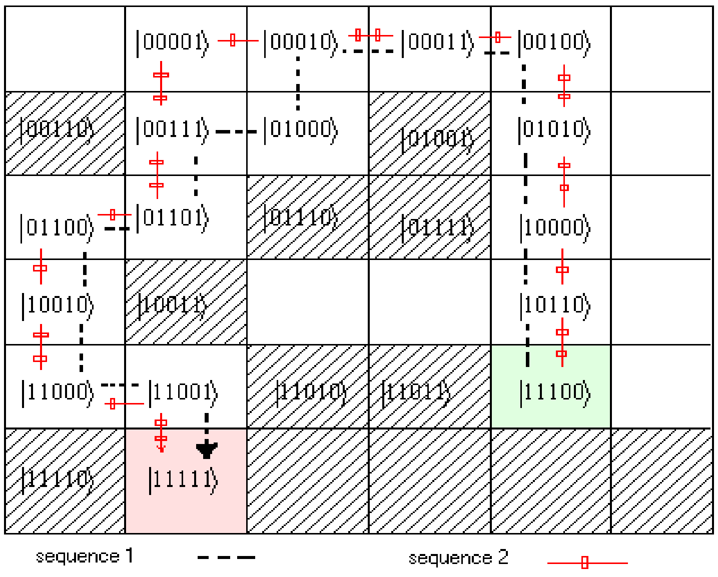

Fig. 11. The control paths for the control of a five-qubit system

5. Conclusion

According to the existing problems in RL area, such as low learning speed and tradeoff between exploration and exploitation, SIRL and QRL methods are introduced based on the theory of RL and quantum computation in this chapter, which follows the developing roadmap from the superposition-inspired methods to the RL methods in quantum systems. Just as simulated annealing algorithm comes from mimicking the physical annealing process, quantum characteristics also broaden our mind and provide alternative approaches to novel RL methods.

In this chapter, SIRL method emphasizes the exploration policy and uses a probabilistic action selection method that is inspired by the state superposition principle and collapse postulate. The experiments, which include a puzzle problem and a mobile robot navigation problem, demanstrate the effectiveness of SIRL algorithm and show that it is superior to basic TD algorithm with ε-greedy policy. As for QRL, the state/action value is represented with quantum superposition state and the action selection is carried out by observing quantum state according to quantum collapse postulate, which means a QRL system is designed for the real quantum system although it can also be simulated on a traditional computer. The results of simulated experiments verified its feasibility and effectiveness with two examples: Prisoner's Dilemma and the control of a five-qubit system. The contents presented in this chapter are mainly the basic ideas and methods related to the combination of RL theory and quantum computation. More theoretic research and applictions are to be investigated in the future.

6. References

Barto, A.G. & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning, *Discrete Event Dynamic Systems: Theory and applications,* Vol. 13, pp. 41-77

Bertsekas, D.P. & Tsitsiklis, J.N. (1996). *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA

Chen, C.L. & Chen, Z.H. (2005). Reinforcement learning for mobile robot: from reaction to deliberation. *Journal of Systems Engineering and Electronics*, Vol. 16, No. 3, pp. 611-617

Chen, C.L.; Dong, D.Y. & Chen, Z.H. (2006a). Quantum computation for action selection using reinforcement learning. *International Journal of Quantum Information*, Vol. 4, No. 6, pp. 1071-1083

Chen, C.L.; Dong, D.Y. & Chen, Z.H. (2006b). Grey reinforcement learning for incomplete information processing. *Lecture Notes in Computer Science*, Vol. 3959, pp. 399-407

Chen, C.L.; Dong, D.Y.; Dong, Y. & Shi, Q. (2006c). A quantum reinforcement learning method for repeated game theory. *Proceedings of the 2006 International Conference on Computational Intelligence and Security*, Part I, pp. 68-72, Guangzhou, China, Nov. 2006, IEEE Press

Chen, C.L. & Dong D.Y. (2007). Quantum mobile intelligent system, In: *Quantum-Inspired Evolutionary Computation*, N. Nedjah, L. S. Coelho & L. M. Mourelle (Eds.), Springer, in press

Chen, Z.H.; Dong, D.Y. & Zhang, C.B. (2005). *Quantum Control Theory: An Introduction*, University of Science and Technology of China Press, ISBN 7-312-01863-7/TP. 363, Hefei (In Chinese)

Chuang I.L.; Gershenfeld N. & Kubinec M. (1998). Experimental implementation of fast quantum searching, *Physical Review Letters*, Vol. 80, pp. 3408-3411

Dong, D.Y.; Chen, C.L. & Chen, Z.H. (2005a). Quantum reinforcement learning. *Lecture Notes in Computer Science*, Vol. 3611, pp. 686-689

Dong, D.Y.; Chen, C.L.; Zhang, C.B. & Chen, Z.H. (2005b). An autonomous mobile robot based on quantum algorithm. *Lecture Notes in Artificial Intelligence*, Vol. 3801, pp. 394-399

Dong, D.Y.; Chen, C.L.; Zhang, C.B. & Chen, Z.H. (2006a). Quantum robot: structure, algorithms and applications. *Robotica*, Vol. 24, No.4, July 2006, pp. 513-521

Dong, D.Y.; Chen, C.L.; Chen, Z.H. & Zhang, C.B. (2006b). Quantum mechanics helps in learning for more intelligent robots. *Chinese Physics Letters*, Vol. 23, No. 7, pp. 1691-1694

Dong, D.Y.; Chen, C.L.; Chen, Z.H. & Zhang, C.B. (2006c). Control of five-qubit system based on quantum reinforcement learning. *Proceedings of the 2006 International Conference on Computational Intelligence and Security*, Part I, pp. 164-167, Guangzhou, China, Nov. 2006, IEEE Press

Dong, D.Y.; Chen, C.L. & Li, H.X. (2007a). Reinforcement strategy using quantum amplitude amplification for robot learning. *Proceedings of the 26th Chinese Control Conference*, Part VI, pp. 571-575, Zhangjiajie, China, Jul. 2007, IEEE Press

Dong, D.Y.; Chen, C.L.; Li, H.X. & Tarn, T.J. (2007b). Quantum reinforcement learning. IEEE *Transaction on System, Man, and Cybernetics B,* under review

Ekert, A. & Jozsa, R. (1996). Quantum computation and Shor's factoring algorithm, *Reviews of Modern Physics*, Vol. 68, pp. 733-753

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search, *Proceedings of the 28th Annual ACM Symposium on the Theory of Computation*, pp. 212-219, New York, 1996, ACM Press

Grover, L. K. (1997). Quantum mechanics helps in searching for a needle in a haystack, *Physical Review Letters*, Vol. 79, pp. 325-327, 1997

Guo, M.Z.; Liu, Y. & Malec, J. (2004). A new Q-learning algorithm based on the metropolis criterion, *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics,* Vol. 34, No. 5, pp. 2140-2143

He, P. & Jagannathan, S. (2005). Reinforcement learning-based output feedback control of nonlinear systems with input constraints, *IEEE Transactions on Systems Man and Cybernetics, Part B-Cybernetics,* Vol. 35, No. 1, pp. 150-154

Jones, J.A. (1998a). Fast searches with nuclear magnetic resonance computers, *Science*, Vol. 280, pp. 229

Jones, J.A.; Mosca, M. & Hansen R.H. (1998b). Implementation of a quantum Search algorithm on a quantum computer, *Nature*, Vol. 393, pp. 344-346

Kaelbling, L.P.; Littman, M.L. & Moore, A.W. (1996). Reinforcement learning: a survey, *Journal of Artificial Intelligence Research,* Vol. 4, pp. 237-287

Kondo, T. & Ito, K. (2004). A reinforcement learning with evolutionary state recruitment strategy for autonomous mobile robots control, *Robotics and Autonomous Systems,* Vol. 46, pp. 111-124

Kwiat, P.G.; Mitchell, J.R.; Schwindt, P.D.D. et al. (2000). Grover's search algorithm: an optical approach, *Journal of Modern Optics*, Vol. 47, pp. 257-266

Morimoto, J. & Doya, K. (2001). Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning, *Robotics and Autonomous Systems,* Vol. 36, pp. 37-51

Nielsen, M.A. & Chuang, I.L. (2000). *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, England

Preskill, J. (1998). Physics 229: *Advanced Mathematical Methods of Physics--Quantum Information and Computation*. California Institute of Technology, 1998. Available electronically via http://www.theory.caltech.edu/people/preskill/ph229/

Scully, M.O. & Zubairy, M.S. (2001). Quantum optical implementation of Grover's algorithm, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 98, pp. 9490-9493

Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring, *Proceedings of the 35th Annual Symposium on Foundations of Computer Science,* pp. 124-134, Los Alamitos, CA, IEEE Press

Sutton, R. & Barto A.G. (1998). *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA

Sutton, R.; Precup, D. and Singh, S. (1999). Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning, *Artificial Intelligence,* Vol. 112, pp. 181-211

Vandersypen, L.M.K.; Steffen, M.; Breyta, G. et al. (2001). Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, *Nature*, Vol. 414, pp. 883-887

Watkins, J.C.H. and Dayan, P. (1992). Q-learning, *Machine Learning,* Vol. 8, pp. 279-292