Automated Planning Using Quantum Computation

S. Naguleswaran and L. B. White
School of Electrical and Electronic Engineering,
University of Adelaide,
SA 5005,
Australia.

I. Fuss
Defence Science and Technology Organisation(DSTO),
Edinburgh,
SA 5111,
Australia.

Abstract

This paper presents an adaptation of the standard quantum search technique to enable application within Dynamic Programming, in order to optimise a Markov Decision Process. This is applicable to problems arising from typical planning domains that are intractable due to computational complexity when using classical computation. The proposed method is able to balance state-space exploration with greedy selection of the local minima by setting appropriate thresholds via quantum counting. A quantum walk is used to propogate through a graphical representation of the state space.

Introduction

The recent advances in the field of quantum computation opens up the possibility of exploiting the computational power of quantum computers to solve previously intractable problems (Feynman 1982). There is considerable research being undertaken on the physical realisation of quantum computers and with this consideration it is imperative that algorithmic development is also undertaken in order to fully utilise the computational power as it becomes available. Typical planning problems, when represented as Markov Decision Processes (MDP) suffer from the problem of "State Space Explosion" . Therefore, despite the fact that Dynamic Programming is a polynomial-time algorithm with respect to the number of states, due to the fact that the number of states increases exponentially with the number of state variables, realistic problems quickly become intractable. This work seeks to apply the concept of quantum search and quantum walks to optimise MDP representations of planning problems. Specifically, Deterministic Shortest Path problems with a well defined initial and goal state are considered in this paper.

Quantum computation provides a means of achieving dramatic increases in computing speed as well as secure data transfer due to the quantum mechanical principles of superposition and entanglement. Quantum computation is accomplished by the manipulation of qubits which are much like classical bits in that they can be in one of two states (0, 1) which are known as basis states. However, a qubit posseses the added property of being able to be in a number of intermediate states.

This paper is organised as follows. Firstly the formulation of the problem is given. Then the principles involved in quantum search are described, followed by a description of the quantum walk. The development of a Dynamic Programming method leveraging on these quantum methods is then explored. The conclusions are given in the final Section.

Problem Statement

The planning problem can be represented as an MDP on a directed graph corresponding to either a deterministic or stochastic shortest path problem (Bertsekas & Tsitsiklis 1995).

Assuming a discrete-finite-state system, the Deterministic Shortest Path problem consists of finding the path formed by a minimum-cost sequence of successor states starting at the initial state i and terminating at a special cost-free goal state g. The system can be represented on a directed graph consisting of nodes 1,2,...,N, where node g is a goal state and non-negative costs are assigned to each edge. At each node $s \neq s'$, a successor node s' such that (s,s') is an edge, is selected.

The Stochastic Shortest Path problem is a generalisation of the Deterministic Shortest path problem, where a probability distribution over all possible succeeding states, selected from a pre-determined set of probability distributions, is chosen. The real and non-negative transition probabilities between states s and s' are given by $p_a(s'|s)$, where a represents the action.

Finding a Plan with Dynamic Programming

Dynamic Programming techniques can be used to find the optimal path through a directed graph representing the planning domain, in order to find the goal state. Bellman's Equation for the action-value-function is given by

$$Q(s,a) = C(s,a) + \gamma \sum_{s' \in S} P_a(s'|s)V(s'), \tag{1}$$

where C is the total for performing action a at state s, V is the expected cost for s', P is the probability of reaching s' and γ is a discount factor, which is set to unity for Shortest Path problems. Bellman's optimality condition is given by

$$V(s) = min_{a \in A} Q(s,a). \tag{2}$$

Equation (2) can be solved using value iteration, which is a well known Dynamic Programming technique.

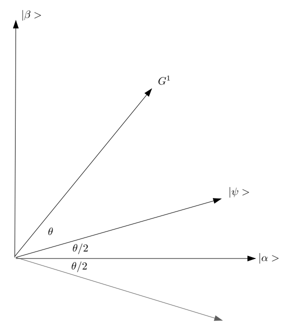

Figure 1: Geometric picture of GSA. Each iteration consists of two reflections in a plane that result in a rotation.

Quantum Search

A Quantum Search based on the Grover Search Algorithm (GSA) is proposed for the minimisation operation in (2). The term "state" in this section refers to the states of the quantum computation and should not be confused with the state-space of the planning problem. The search is performed over the actions and therefore each action corresponds to a state in the context of the quantum computation.

The search procedures considered are based on the original Grover Search Algorithm (Grover 1997) which is based on amplifying the amplitude of the desired state. If n is the number of qubits, then the number of states are $N = 2^n$. The system is initialised to the superposition: $\psi = \frac{\sum_x |x>}{\sqrt{N}}$, ensuring all states have the same amplitude. The following operations are performed O($\sqrt{N}$) times:

Let the system be in any state S: For $C(s) = 1$, the phase is rotated by π radians. For $C(S) = 0$, the system is unaltered. The function C is either based on the contents of the memory location corresponding to state S or evaluated by a computer (in polynomial time). This is also refered to as the oracle operation O. The diffusion transform matrix D defined as $D_{ij} = \frac{2}{N}$ if $i \neq j$ and $D_{ij} = -1 + \frac{2}{N}$ if $i = j$, is then applied.

The measured final state will be S_n (i.e. the desired state that satisfies the condition $C(S) = 1$) with minimum probability of 0.5.

Geometric view of GSA

Eigenvector Basis of G If $M = |\beta|$, the number of solutions to the search problem with N items, then the number of non-solutions $|\alpha| = N - M$. If $M = 0$ or N, $H|0\rangle$ is an eigenvector of G with eigenvalue $e^{\pi i M/N}$. For $0 < M < N$, $|\alpha\rangle = \frac{1}{\sqrt{(N-M)}} \sum_{x\in\alpha} |x\rangle$ and $|\beta\rangle = \frac{1}{\sqrt{(M)}} \sum_{x\in\beta} |x\rangle$.

The eigenvectors of G can then be written as $|a\rangle = \frac{1}{\sqrt{(2)}}|\beta\rangle + \frac{i}{\sqrt{(2)}}|\alpha\rangle$ and $|b\rangle = \frac{1}{\sqrt{(2)}}|\beta\rangle - \frac{i}{\sqrt{(2)}}|\alpha\rangle$. The corresponding eigenvalues are $e^{2\pi\varphi}$and $e^{-2\pi\varphi}$, where $0 < \varphi < \frac{1}{2}$, with $cos(2\pi\varphi) = 1 - 2M/N$ and $sin(2\pi\varphi) = 2\sqrt{(M(N-M)}/N$. It should also be noted that $\frac{1}{\sqrt{(2)}}|a\rangle + \frac{1}{\sqrt{(2)}}|b\rangle = |\beta\rangle$, which is the state we want to observe.

Visualisation The effect of G, which can also be written as $(2|\psi><\psi| - I)O$, can be visualised by considering Figure 1 where the oracle O performs a reflection about $|\alpha\rangle$ in the plane defined by $|\alpha\rangle$ and $|\beta\rangle$. Similarly, the diffusion operator also performs a reflection about $|\psi\rangle$ in this plane. The two reflections couple to produce a rotation. Letting $cos(\frac{\theta}{2}) = \sqrt{N - M/N}$ so that $|\psi\rangle = cos(\frac{\theta}{2})|\alpha\rangle + sin(\frac{\theta}{2})|\beta\rangle$ we have

$$G^k|\psi\rangle = cos(\frac{2k+1}{2}\theta)|\alpha\rangle + sin(\frac{2k+1}{2}\theta)|\beta\rangle. \quad (3)$$

In the $|\alpha\rangle$, $|\beta\rangle$ basis, G can be written as

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (4)$$

with eigenvalues $e^{i\theta}$and $e^{i(2\pi-\theta)}$.

Phase Flip using a Threshold set by Quantum Counting

The algorithm proposed for application within Dynamic Programming uses amplitude amplification in a manner similar to the GSA, where the phase of the desired states is flipped by π radians. However, the construction of the oracle would require setting a threshold T, where if the condition $cost < T$ is satisfied, the phase is flipped for that particular state (action). Therefore, the algorithm considers all states with costs below T as solutions.

The method of quantum counting can be used to rapidly calculate the number of solutions for a given threshold. Thus, it is possible to balance exploration of the state-space with greediness by selection of a threshold with the appropriate number of solutions. Quantum counting is based on the method of quantum phase estimation, which can estimate the angle θ in (4) . Using the fact that $sin^2(\theta/2) = M/N$, an estimate for the number of solutions can be found (Nielsen & Chuang 2000).

Discrete Quantum Walks

Previous studies of discrete time quantum walks are described in (Aharanov, Davidovich, & Zagury 1992; Aharanov *et al.* 2000). In the classical world, Markov Chains or random walks are a fundamental tool in studying the properties of a structured set with the repeated application of certain transition rules. In the case of a Quantum walk on a graph $G(V,E)$, an auxiliary Hilbert space spanned by the coin-flip outcomes that control the walk is introduced in addition to the Hilbert space spanned by the vertices of G. The transition rule for the Quantum walk is unitary and is a transformation on the tensor product $H_c \otimes H_s$ of the two Hilbert spaces. The probability amplitude is non-zero on an edge connecting two vertices.

The quantum walk on a line can be realised using the following two operations a coin-flip operation C,where,

$$C|n,0\rangle = a|n,0\rangle + b|n,1\rangle,$$
$$C|n,1\rangle = c|n,0\rangle + d|n,1\rangle. \quad (5)$$

and a translational operation S, where,

$$S|n,0\rangle = |n-1,0\rangle, S|n,1\rangle = |n+1,1\rangle. \quad (6)$$

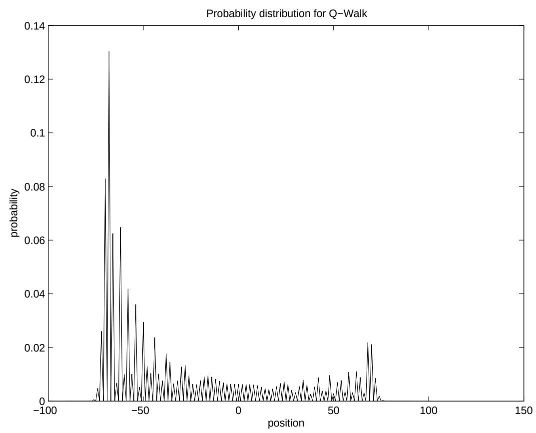

Figure 2: A quantum walk using a Hadamard coin

A step of the quantum walk is then given by the operation, SC. The Hadamard transform is a common unitary operator chosen for C.

The quantum walk begins to differ from the classical walk after the first two steps, where instead of choosing left and right directions with equal amplitude the quantum walk goes to the left with all the amplitude due to the effect of quantum interference. A classical walk starting at location 0 and run for t steps approaches a normal distribution. A quantum walk starting at location $|0,0\rangle$ and run for t steps results in the distribution shown in Figure 2.

The drift to the left is due to the non-symmetric coin-flip operation. A symmetric coin or a starting state that is an equal superposition of the coin-states can be used to produce a symmetric distribution. The classical random walk on a line has a variance $\sigma^2 = t$ after t steps of the walk, and thus the expected distance from the origin is of the order $\sigma = \sqrt{t}$. However, the variance of the quantum walk is $\sigma_q^2 \approx t^2$ and hence the expected distance from the origin is of order t, implying that the quantum walk propagates quadratically faster.

Quantum Walks in two dimensions

A continuous walk on a line is easily generalised to a walk on a general graph by appropriately defining the Hamiltonian (Ambainis 2004). There have been recent advances in describing a discrete quantum walk on an arbitrary graph (Kendon 2003).

Consider a graph $G(V,E)$ and use the basis states $|v,e\rangle$ for $v \in V$ and $e \in E$ such that e connects to the vertex v. One step of the quantum walk consists of performing a coin-flip C_v on the states $|v,e\rangle$ for each v and then performing a shift S such that $S|v,e\rangle = |v',e\rangle$ and $S|v',e\rangle = |v,e\rangle$. A common choice for C_v is Grover's Diffusion operator defined in section 3, where there are N edges adjacent to a vertex v.

Quantum Walk Search

Previous work on searching through a graph using quantum walks was described in (Ambainis 2005). The algorithms basically work by alternatively querying the vertices for a marked state or moving to another vertex in each step of the walk. The Grover Search Algorithm (GSA) (Grover 1997) can be used to query the vertices in order to find the marked state. In a shortest path planning problem the goal state would be the marked state.

	Classical	Quantum
Calculate Q	$O(\|A\|\|S\|\|S_b\|)$	$O(\|A\|\|S'\|\|S_b\|)$
Update V	$O(\|A\|)$	$O(\sqrt{\|A\|})$
Quantum Walk/Search	-	$O(\sqrt{\|S'\|log(\|S'\|)})$

Table 1: Computational complexity per iteration for Value Iteration

Proposed Dynamic Programming Method and Complexity Analysis

The algorithm that was investigated in this study utilises the quantum computational concepts introduced above and accomplishes the task of extracting a valid plan with reduction in computational complexity when compared with classical Dynamic Programming.

The Algorithm

The proposed algorithm consists of the following steps:

$V_0(S) \leftarrow 0$;
$k \leftarrow 1$;
numsol $\leftarrow$*set number of solutions; This controls the threshold* T.
while $V_k - V_{k-1} > \varepsilon$

1. *At initial state calculate* $Q(s_i, a)$ *and find all actions* a_t *below a threshold* T.
2. *Update* $V(s_i)$ *with first* $Q(s_i, a)$ *below threshold.*
3. *quantum walk a step where* a_t *constitute edges of the graph.*

while not goal state
for all states reached by walk repeat steps 1, 2, 3
return policy π.

If V does not converge the threshold can be lowered by picking a lower number of possible actions from a given state.

Table 1 shows the comparison of computational complexity between the classical and quantum cases. $|A|$ is the number of actions, $|S|$ the number of states, $|S_b|$the average resultant states of an action (branching) and $|S'|$ the number of states visited by the quantum walk when constrained by the threshold used in the search.

It should be noted that it is the number of states $|S|$ that typically governs the complexity in a planning problem. The judicious use of the proposed heuristic in the quantum case could make $|S'|$ small enough to ensure that a previously intractable problem becomes computable in a realistic period of time.

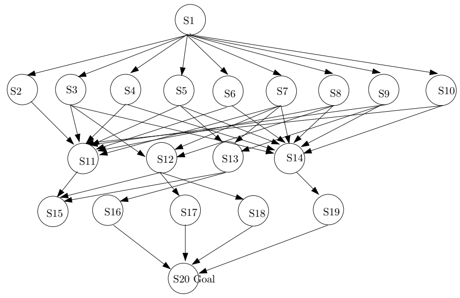

Figure 3: The state diagram representing MDP for the logistical planning problem

A Simple Planning Scenario

It is illustrative to consider an example problem depicted by the MDP shown in Figure 3. An MDP of this form was derived for a two-truck, two-city, two-box planning problem in a logistical domain using the concept of Markov-nets in contemporaneous work (Naguleswaran, Hickmott, & White 2005) . However, for the purpose of this paper it is sufficient to analyse the MDP without regard to its derivation. The cost of reaching each of the states using any action leading to it is given by

$$c(S1) = c(S11) = c(S15) = \infty,$$
$$c(S3) = x,\, c(S4) = x,\, c(S5) = 3x,\, c(S6) = 3x,$$
$$c(S7) = 4x,\, c(S8) = 4x,\, c(S9) = 4x,\, c(S10) = 4x,$$
$$c(S12) = 4x,\, c(S13) = 6x,\, c(S14) = 0,$$
$$c(S16) = 4x,\, c(S17) = 6x,\, c(S18) = 5x,\, c(S19) = 5x.$$

The goal state $S20$ is self absorbing with zero cost to reach it.

Using the aforementioned algorithm, the values for V shown in Table 2 are output during each of four iterations. Each visited state was updated with the lowest of the two action-values Q that were below the threshold, although states resulting from both these actions were visited in the subsequent move of the quantum walk.

The resultant optimal policies are given as $\pi_1^* = [(S1,S3),(S3,S14),(S14,S19),(S19,S20)]$ and $\pi_2^* = [(S1,S4),(S4,S14),(S14,S19),(S19,S20)]$.

Conclusions

A method for adapting standard quantum search techniques to be used in the context of solving planning problems using Dynamic Programming was proposed in this paper. The method is also able to provide a useful heuristic for controlling a quantum walk and thus reducing the number of states visited during an iteration of Dynamic Programming.

A simple deterministic example MDP was optimised where the effective state-space was reduced by half on each iteration. However, the proposed method becomes more powerful when stochastic problems are considered. The ability of the quantum walk to visit all subsequent states of non-zero probability on selected edges in a single move, would result in significant complexity savings when compared to classical forms of propagation.

The problems that can be solved efficiently on a quantum computer with a bounded probability of error are classified to be in the BQP complexity class. However, the relationship of BQP to the complexity classes P, NP and PSPACE is yet unknown although it is thought to be between P and PSPACE.

References

Aharanov, D.; Ambainis, A.; Kempe, J.; and Vazirani, U. 2000. Quantum walks on graphs. *arXiv:quant-ph/0012090*.

Aharanov, Y.; Davidovich, L.; and Zagury, N. 1992. *Phys. Rev. A* 48:1687.

Ambainis, A. 2004. Quantum walks and their algorithmic applications. *arXiv:quant-ph/0403120v3*.

Ambainis, A. 2005. Quantum search algorithms. *arXiv:quant-ph/05040120v1*.

Bertsekas, D., and Tsitsiklis, J. 1995. *Neuro-Dynamic Programming*. Belmont, Mass., USA: Athena Scientific.

Feynman, R. P. 1982. Simulating physics with computers. *Int. J. Theor. Phys.* 21:467–488.

Grover, L. K. 1997. Quantum mechanics helps in searching for a needle in a haystack. *PRL* 78:325–328.

Kendon, V. 2003. Quantum walks on general graphs. *arXiv:quant-ph/0306140 v3*.

Naguleswaran, S.; Hickmott, S. L.; and White, L. B. 2005. Reduced mdp representation of a logistical planning problem using petri-nets. *LNAI* 3809:830–833.

Nielsen, M. A., and Chuang, I. L. 2000. *Quantum Computation and Quantum Information*. Cambridge, UK: Cambridge University Press.

State/iter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	x	-	0	0	-	-	-	-	-	-	∞	$5x$	-	$5x$	-	-	0	0	0	0
2	x	-	$5x$	$5x$	-	-	-	-	-	-	∞	$5x$	-	$5x$	-	-	0	0	0	0
3	$3x$	-	-	-	$5x$	$5x$	-	-	-	-	∞	-	$4x$	$5x$	∞	0	-	-	0	0
4	$6x$	-	$5x$	$5x$	-	-	-	-	-	-	∞	$5x$	-	$5x$	-	-	0	0	0	0

Table 2: The values of $V(S)$ are shown for each of the states that are updated on a particular iteration.