Research Journal of Applied Sciences, Engineering and Technology 4(5): 510-512, 2012
ISSN:2040-7467

Submitted: September 23, 2011 Accepted: November 16, 2011 Published: March 01, 2012

Polynomial-Time Quantum Algorithms for the 0-1 Knapsack Problem

Zhong Yanhua and Nie Shuzhi
Department of Electronics and Information Technology, Jiangmen Polytechnic,
Jiangmen, China, 529090

Abstract: In this study, designed a $O(n^2\log_2 n)$ quantum mechanical algorithm, to solve the 0-1-knapsack problem on a hypothetical quantum computer. Used the special characteristics of the quantum environment, constantly divided the state of vector space, reduced the probability of state vector which don meet the conditions of magnitude, increased the probability amplitude to meet the conditions, find a larger probability of obtaining the solution. Owing to the problem of solving the time complexity by traditional exponential is too complicated, turned it into the other problem which is relatively easy, then solving polynomial time wih quantum computer, which can Reduce the difficulty of solving the problem. Analysised of the complexity of the algorithm and implementation results showed that the designed algorithm is effective and feasible. This algorithm can be extended to solve other NPC problems, such as TSP problem.

Key words: NPC problem, quantum algorithm, quantum computing, 0-1-kiapsack problem

INTRODUCTION

The 0-1 Knapsack problem belongs to the NP-complete problems. The previous quantum algorithms research about the 0-1 Knapsack problem based on Grover's search algorithm are still exponential-time (Harry *et al.*, 2001; Aaronson, 2002; Ambainis, 2002).In this paper, we considered quantum algorithms for the 0-1 Knapsack problem have a Special partition method, which can be exploited to design significantly quantum algorithms, it faster than a direct application of Grover's search algorithm would yield (Aaronson, 2002).

Give a set of n items from which we are to select a number of items to be carried in the knapsack problems. Each item has both a weight and profit. The objective is to choose the set of items which fit in the knapsack and maximize the profit (Shi, 2002; Vergis *et al.*, 1986).

Set $w_i(w_i > 0)$ to be the weight of the item and $v_i(v_i > 0)$ to be the profit, which must be accrued when the i^{th} item is carried in the knapsack, and C is the capacity of the knapsack. Set xi $\epsilon\{0,1\}$, $1 \le i \le n$ to be a variable the value of which is either zero or one. The variable x_i has one value when the i^{th} item is carried in the knapsack. Then the problem is stated as follows:

$$W = (w_1, w_2 \ldots w_n),\ V = (v_1, v_2 \ldots v_n),$$
$$g(x) = \sum_{i-1}^{n} v_i x_i,\ f(x) = \sum_{i-1}^{n} w_i x_i$$

W and V separately standed for the vectors, g(x) and f (x) separately standed for the functions. To find a binary vector x, $x = (x_1, x_2 \ldots x_n)$, then maximize the objective function profit max g (x),while satisfying the constraint $f(x) \le C$, $C > 0$.

We can solve this problem by exhaustively enumerating the feasible solutions and select the one with the highest profit (Shuzhi, 2010). However, since there are 2^n possible solutions, the running time required for the brute-force solution becomes impossible when n gets large (Yanhua and Xiaomin, 2005; Yanhua and Xiaomin, 2004). In this way, we designed a $O(n^2\log_2 n)$ quantum mechanical algorithm to solve the 0-1-knapsack problem on a hypothetical quantum computer.Used the special characteristics of the quantum environment, constantly divided the state of vector space, reduced the probability of state vector which don't meet the conditions of magnitude, increased the probability amplitude to meet the conditions, find a larger probability of obtaining the solution. Owing to the problem of solving the time complexity by traditional exponential is too complicated, turned it into the other problem which is relatively easy, then solving polynomial time wih quantum computer, which can Reduce the difficulty of solving the problem. Analysised of the complexity of the algorithm and implementation results showed that the designed algorithm is effective and feasible. This algorithm can be extended to solve other NPC problems, such as TSP problem.

QUANTUM ALGORITHMS DESIGN

The 0-1 knapsack problem require whether it is to find any combination of n binary variables that satisfies a certain set of clauses C, the crucial issue in NP-completeness is whether it is possible to solve it in time

Corresponding Author: Zhong Yanhua, Department of Electronics and Information Technology, Jiangmen Polytechnic, Jiangmen, China, 529090

polynomial in n. So there are $N= 2^n$ possible combinations which have to be searched for any that satisfy the specified property, the question is whether can do that in a time which is polynomial $O(n^k)$, Thus if it were possible to reduce the number of steps to a finite power of $O(n^k)$, it would yield a polynomial time algorithm for NP-complete problems.

Set a system have two registers $|R_1>$ and $|R_2>$, the two registers have n qubits independently, and which are labeled $x_1,x_2\ldots x_n$. These are 2^n states which is represented as n bit strings after the Walsh-Hadamard transformation for the registers $|R_1>$. The object is to find a unique state $|x>=(x_1,x_2\ldots x_n)$ which satisfied the condition of the 0-1knapsack problem. The quantum algorithms design described as follow:

Step 1: Start with the register $|R_1>$ in the states $|s>=|R_1>=100\cdots>$

Step 2: Perform the Walsh-Hadamard transformation H in $|R_1>$ on each bit independently in sequence, then:

$$H:|s> \longrightarrow |s_1 \geq \frac{1}{\sqrt{2^n}}\sum_{x=0}^{n-1}|x>|0> \qquad (1)$$

where, $|x>$ is a binary string, $|x\geq (x_1,x_2\ldots x_n)$ $x_i \in\{0,1\}$ $1,\leq i\leq n$.the register R_1 preserves the superposition of all possible answers from 0 to n. That is, R_1 save a choice of all possible items.

Step 3: Perform the unitary transformation operation into as follow:

$$|R_1>|\varphi_1 \geq P_a U_a|s_2> \qquad (2)$$

Use U_a to classify according to the capacity C of packsack, then:

$$U_a:|s' \geq \frac{(-1)^{a(x)}}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x> \qquad (3)$$

where the function a(x) = {1 C >f(x),0 C ≤ f(x), to any state $|x>$, In case C > f(x), rotate the phase by π radians; In case C > f(x), leave the system unaltered.

After apply the transformation matrix P_a, the probability amplitude of the state that unsatisfied the condition is zero, and equally magnify the probability amplitude of the state that satisfied the condition. The transformation P_a is defined as follow:

$$P_a \leq \varphi_1 \mid s' > \qquad (4)$$

Here, the first register $|R_1>$ remained the states which satisfied the capacity of the knapsack as a result of quantum entanglement. The unitary transformation operation UP will formally proved in further.

Step 4: Repeat the following unitary operations j times.Then:

$$|\varphi^{j+1} \geq P_b^j U_b^j \ldots P_b^1 U_b^1|\varphi_1 > \qquad (5)$$

Before the iteration every time, calculate the average value c1 in $|R_2>$,b(x)= $\begin{cases}1 & c1> g(x)\\ 0 & c1\leq g(x)\end{cases}$, The purpose of unitary operator U_b is to classify according to the value c1, rotate the phase of state in $|R_1>$, by π radians that the condition is; The phase of state is unaltered that the condition is c1 >g(x). So:

$$U_b:(-1)^{b(x)}|x> \qquad (6)$$

The effect of transformation matrix P_b is the same as P_a, the probability amplitude of the state that unsatisfied the condition is zero, and equally magnify the probability amplitude of the state that satisfied the condition, and is defined as follow:

$$P_b^{(i+1)} \leq \varphi_{i+1}|\varphi_i> \qquad (7)$$

Step 5: Goto step 4, repeat j times until the system is a basic state. The unitary transformation operation UP will formally proved in the next section as theorem 1.

In a practical implementation of this algorithm would involve one portion of the quantum system sensing the state and then deciding whether or not to rotate the phase. It would do it in a way so that no trace of the st ate of the system will be left after this operation,so as to ensure that paths leading to the same final state were indistinguishable and could interfere. The implementation does not involve a classical measurement.

ALGORITHM COMPLEXITY ANALYSIS AND IMPLEMENTATION

The loop of 5 steps above is the heart of the algorithm. It is clear that the search in step 4 become a binary tree, the time of the iteration is the depth of this binary tree, and is decided by the value c1. The average value c1 in $R_2>$, can be computed efficiently using only $n\log_2$noperations as well (Yanhua and Xiaomin, 2004).

The iterate equation can be constructed as follow:

$$T(2^n) = \begin{cases} C & n = 1 \\ 2T(2^{n/2}) + C_2 n \log_2 n & n > 1 \end{cases} \quad (8)$$

The answer is $T(2^n) = O(n^2 \log_2 n)$, the desired answer can be obtained in only $O(n^2 \log^2 n)$ steps.

The algorithm is likely to be implemented to compared to other quantum mechanical algorithms, because in the 0-1 knapsack problem, the operations required are: the Walsh-Hadamard transforms; the conditional phase shift operation both of which are relatively easy as compared to operations required for other quantum mechanical algorithms.

The implementation of the quantum transform UP is very similar to the diffusion transform D of the Grover's algorithm, and can be express from one representation to the other. Let A as input representation before the transformation and B as representation after the transformation, $\{|a_m>\}$ is the basic vector of the representation A and $\{|b_n>\}$ is the basic vector of, it was discussed in the previous section that the representation A satisfies the complete condition. Now we discuss the contact of the unitary state vector $|\varphi>$ in this two representation.Then:

$$<b_n|\phi> = \sum_n <b_n|a_m><a_m|\phi> \quad (9)$$

Equation (9) can be written as $\varphi(B) = S\varphi(A)$, $\varphi(B)$, and $\varphi(A)$ are respectively expressed the representation A and B. the matrix S is defined as follows:

$$S_{mn} = <b_n|a_m> \quad (10)$$

The matrix S is a transformation matrix from the representation A to the representation B. As:

$<a_m|b_n> = <b_n|a_m>^* = S^*_{mn} = S^+_{mn}$, then $S^{-1} = S^+$It

it is clear that the transformation keep unitary from one representation to the other representation B. The dimensionality of the matrix S can be identical also can be different, the effectiveness of the designed algorithm has been proved

CONCLUSION

The designed algorithm combined the principles of the Grover's quantum mechanical algorithm and the approximate quantum Fourier transforms. The implementation of the quantum transform UP used the principle of Grover's quantum mechanical algorithms and calculation of the average value c1 in $|R_2>$ used the principle of the quantum Fourier transform Alternatively, it might combine other quantum algorithms to develop a fast algorithm for other NP-search problem in general. The algorithm for the 0-1 Knapsack problem is an example of such an application.

REFERENCES

Aaronson, S., 2002. Quantum Lower Bound for the Collision Problem. Proceedings of the 34th Annual ACM Symp on Theory of Computing, pp: 635-642.

Ambainis, A., 2002. Quantum lower bounds by quantum arguments. J. Comp. Syst. Sci., 64: 750-767.

Harry, B., D.W. Ronald, D. Christoph, H. Mark, H. Peter, M. Frederic and S. Miklos, 2001. Quantum Algorithms for Element Distinctness. 16th Annual Conference on Computational Complexity, pp: 131-137.

Shi, Y., 2002. Quantum lower bounds for the collision and the element distinctness problems. In Proceedings of Symposium on the Foundations of Computer Science.

Shuzhi, N.I.E., 2010. Research on Collaborative Manufacturing Resources Optimization Deployment based on DNA genetic algorithm. Doctoral Dissertation of South China University of Technology, pp: 34-66.

Vergis, A., K. Striglitz and B. Dickinson, 1986. The complexity of analog computation. Math. Comp. Simul., 28: 91-113.

Yanhua, Z. and Y.U. Xiaomin, 2004. A quantum algorithm to resolve TSP. Comp. Eng. Design, 6: 1032-1033.