Mutiple query search in a classical database using quantum algorithm

First Author:
L. Monisha
Bachelor Of Technology(Third Year)
Department Of CSE
PONDICHERRY ENGINEERING COLLEGE

monishals@yahoo.com

Second Author:
S.Nanthini
Bachelor Of Technology(Third Year)
Department Of CSE
PONDICHERRY ENGINEERING COLLEGE
nanthini1992@yahoo.co.in

Abstract-- A quantum search for a classical object can be performed with no coherent evolution on the quantum computer being used for the search. It is done so by using interaction free measurement as a subroutine in a quantum search algorithm. In addition to providing a simple example of how non-unitary processes which approximate unitary ones can be useful in a quantum algorithm, this procedure requires only one photon regardless of the size of the database, thereby establishing an upper bound on the amount of energy required to search an arbitrarily large database.

This paper applies quantum computing to a problem in information processing and presents an algorithm that is significantly faster than any classical algorithm can be. The problem is this: there is an unsorted database containing *N* items out of which just one item satisfies a given condition - that one item has to be retrieved. Once an item is examined, it is possible to tell whether or not it satisfies the condition in one step. However, there does not exist any sorting on the database that would aid its selection. The most efficient classical algorithm for this is to examine the items in the database one by one. If an item satisfies the **required condition stop; if it does not, keep track of this item so that it is** not examined again. It is easily seen that this algorithm will need to look at an average of N/2 items before

finding the desired item and also makes a faster approach compared to the classical algorithm.

Index Terms-- *Quantum algorithm,Grover's algorithm,Walsh Hadamard transform,selective rotation.*

I.INTRODUCTION

The quantum search algorithm was originally phrased in terms of searching an unsorted database for a marked item. But this did not meet the practical needs of retrieving information from the database. As with so many things in quantum information theory, the ultimate definition should conveniently coincide with a problem which can be solved by using the algorithm. In theoretical computer science, the typical problem can be looked at as that of examining a number of different possibilities to see which, if any, of them satisfy a given condition. This is analogous to the search problem stated in the abstract above, except that usually there exists some structure to the problem, i.e some sorting does exist on the database. Most interesting problems are concerned with the effect of this structure on the speed of the algorithm. For example the **SAT problem [3]** asks whether it is possible to find any combination of *n* binary variables that satisfies a certain set of clauses C, the crucial

MAIN AUTHOR AFFILIATION INFORMATION GOES HERE.

issue in NP-completeness is whether it is possible to solve it in time polynomial in *n*. In this case there are $N = 2n$ possible combinations which have to be searched for any that satisfy the specified property and the question is whether we can do that in a time which is polynomial in , i.e. Thus if it were possible to reduce the number of steps to a finite power , it would yield a polynomial time algorithm for NP-complete problems.

*A. Search Problems in Computer Scienc**e:***

In view of the fundamental nature of the search problem in both theoretical and applied computer science, it is natural to ask - how fast can the basic identification problem be solved without assuming anything about the structure of the problem? It is generally assumed that this limit is since there are *N* items to be examined and a classical algorithm will clearly take steps. However, quantum systems can simultaneously be in multiple Schrodinger cat states and carry out multiple tasks at the same time. This paper presents an step algorithm [2] for the search problem.

B. Evolution Of the Quantum Computer:

Quantum computers will consist of quantum states instead of classical ones. So, the electric potential can be replaced by some quantum state: the quantum bit (qubit for short). Just as a bit has a state 0 or 1, a qubit also has a state **|0›or |1›.** This is called the **Dirac notation** and it is the standard notation for states in Quantum Mechanics. The difference between bits and qubits is that a qubit **|Ψ›** can also be in a linear combination
of states |0› and |1›

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

The state |0› is not the zero vector, but simply the first vector of the basis.
The matrix representations of the vectors |0› and |1› are given by

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \qquad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

II.UNSORTED DATABASE SEARCH:

Let the database contain N distinct objects arranged in a random order. A certain object has to be located in the database by asking a set of questions. Each query is a yes/no question based on a property of the desired object (e.g. is this the object that I want or not?). In the search process, the same query is repeated using different input states until the desired object is found. Let Q be the number of queries required to locate the desired object in the database.

Using classical probability analysis, it can be easily seen that

Option (a) :

‹Q› = N when all objects are available with equal probability for each query (i.e. each query has a success probability of 1/N)

Option (b):

‹Q› = (N + 1)/2 when the objects which have been rejected earlier in the search process are not picked up again for a query. Here the angular brackets represent the average expectation values.

Option (b) is available only when the system possesses memory to recognise what has already been tried before. In the random cellular environment, the rejected object is thrown back into the database, and only option (a) is available to a classical **ASSEMBLY** operation.

Lov Grover discovered a quantum database search algorithm that locates the desired object using fewer queries. Quantum algorithms work with amplitudes, which evolve in time by unitary transformations. At any stage, the observation probability of a state is the absolute value square of the corresponding amplitude. The quantum database is represented as an N−dimensional Hilbert space, with the N distinct objects as its orthonormal basis vectors. The quantum query can be applied not only to the basis vectors, but also to their all possible superpositions (i.e. to any state in the Hilbert space). Let |bi be the desired state and |si be the symmetric superposition state .

$$|b\rangle = [\ (0\ .\ .\ .\ 010\ .\ .\ .\ 0)]\hat{}T$$

$$|s\rangle = [(1/\sqrt{N})\ \ (1.\ .\ .\ 1)]\hat{}T \qquad -\ (1)$$

Let $Ub = 1 - 2|b\rangle\langle b|$ and

$$Us = 1 - 2|s\rangle\langle s|$$

be the reflection operators corresponding to these states .

The operator Ub distinguishes between the desired state and the rest. It flips the sign of the amplitude in the desired state, and is the query or the quantum oracle. The operator Us treats all objects on an equal footing. It implements the reflection about the average operation. Grover’s algorithm starts with the input state |si, and at each step applies the combination –UsUb to it. Each step just rotates the state vector by a fixed angle (determined by |b›|s›| = 1/√N) in the plane formed by |b› and |s›|. Q applications of −UsUb rotate the state vector all the way to |b›|, at which stage the desired state is located and the algorithm is terminated.

$$(-UsUb)Q|s\rangle = |b\rangle\ \ -\ (2)$$

This relation is readily solved, since the state vector rotates at a constant rate, giving

$$(2Q + 1)\ \mathrm{sin}{-}1(1/\sqrt{N}) = \prod/2\ \ -\ (3)$$

A. Grover’s Algorithm:

i. For a given N, the solution for Q satisfying Eq.(3) may not be an integer. This means that the algorithm will have to stop without the final state being exactly |bi on the r.h.s. of Eq.(2). There will remain a small admixture of other states in the output, implying an error in the search process. The size of this admixture is determined by how close one can gets to ∏/2 on the r.h.s. of Eq.(3). Apart from this, the algorithm is fully deterministic.

ii. The algorithm is known to be optimal going from |s›| to |b›| along a geodesic. No other algorithm, classical or quantum, can locate the desired object in an unsorted database with a fewer number of queries.

iii. The iterative steps of the algorithm can be viewed as the discretised evolution of the state vector in the Hilbert space, governed by a Hamiltonian containing two terms, |b›|‹b| and |s›|‹s|. The former represents a potential energy attracting the state towards |bi, while the latter represents a kinetic energy diffusing the state throughout the Hilbert space. The alteration between Ub and Us in the discretised steps is reminiscent of **Trotter’s formula[8]** used in construction of the transfer matrix from a **discretised Feynman’s path[11]** integral.

iv. Asymptotically, **Q = ∏√N/4**. The best that the classical algorithms can do is to random walk through all the possibilities, and that produces **Q = O(N)** as mentioned above. With the use of superposition of all possibilities at the start, the quantum algorithm performs a directed walk to the final result and achieves the square-root speed-up.

v. The result in Eq.(3) depends only on |h›|s›|; the phases of various components of |s› can be arbitrary, i.e. they can have the symmetry of bosons, fermions or even anyons.

B. Walsh – Hadamard Transformation:

If x be the *n*-bit binary string describing

If y the starting state and the *n*-bit binary string. describing the resulting string, the sign of the amplitude of y is determined by the parity of the bitwise dot product of x and y, i.e. . $(-1)^{x.y}$. This transformation is referred to as the Walsh-Hadamard transformation This operation (or a closely related operation called the Fourier Transformation) is one of the things that makes quantum mechanical algorithms more powerful than classical algorithms and forms the basis for most significant quantum algorithms.

When *N=2n,* the discrete Walsh Hadamard Tranform is:

$$W(u) = \sum_{x=0}^{N-1} f(x) \frac{1}{\sqrt{N}} \prod (x+a)^n \quad = (-1)^{b_1(x) b_{n-i-1}(u)}$$

Where *bk(z)* is the *k*th bit in the binary representation of *z*.

III. THE ABSTRACTED PROBLEM:

Let a system have ***N* = *2n*** states which are labelled *S1,S2,...SN.* These *2n* states are represented as *n* bit strings. Let there be a unique state, say *S*n, that satisfies the condition ***C(Sn)* = *1*,** whereas for all other states *S, C(S)* = (assume that for any state *S*, the condition *C(S)* can be evaluated in unit time). The problem is to identify the state *S*n.

IV. QUANTUM ALGORITHM:

The above mentioned Grover's algorithm is not typically projected to solve multiple query states. So this quantum algorithm might pave the way to solve complex query searches with the help of multiple unitary states.

(i) Initialize the system to the distribution: $\left(^1/_{\sqrt{N}}, ^1/_{\sqrt{N}}, ^1/_{\sqrt{N}}, ^1/_{\sqrt{N}}, \right)$ i.e. there is the same amplitude to be in each of the *N* states. This distribution can be obtained by simulating annealing[5].

ii)Repeat the following unitary operations O(log N) times (the precise number of repetitions is important as discussed in

(a) Let the system be in any state S:

Corollary 1: In case **C(S)= 1**, rotate the phase by radians;

Corollary 2: In case **C(S)= 0**, leave the system unaltered.

(b) Apply the diffusion transform *D* which is defined by the matrix *D* as follows:

i) $\mathbf{D_{ij} = \frac{2}{N}}$ **if i≠j**

ii) $\mathbf{D_{ii} = -1 + \frac{2}{N}}$

This diffusion transform, *D*, can be implemented as , where *R* the Rotation matrix & *W* the Walsh-Hadamard Transform Matrix are defined as follows:

i) *Rij=0 if i≠j;*

ii) *Rii=1 if i=0 Rii=-1 if i≠0;*

As discussed in **section II.B**:

$$Wij = 2^{-n/2} - n/2(-1)^{i1.j1}$$

where is the
i1 is binary representation of i, and
i1.j1 denotes the bitwise dot product of the two *n* bit i and j strings.

(iii) Sample the resulting state. In case there is a unique state *S*n such that the final state is *S*n with a probability of at least ½.

Note that step (ii) (a) is a phase rotation transformation by using **selective rotation.**
of the phase of the amplitude in certain states. The transformation describing this for a 4 state system is of the form ,

$$\begin{bmatrix} e^{j\phi 1} & 0 & 0 & 0 \\ 0 & e^{j\phi 2} & 0 & 0 \\ 0 & 0 & e^{j\phi 3} & 0 \\ 0 & 0 & 0 & e^{j\phi 4} \end{bmatrix}$$

where **j=√-1** and **ϕ1, ϕ2, ϕ3, ϕ4** are **arbitrary real numbers.** Note that, unlike the Walsh-Hadamard transformation and other state transition matrices, the probability in each state stays the same since the square of the absolute value of the amplitude in each state stays the same.

In a practical implementation this 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 state of the system 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.

V.INVERSION ABOUT AVERAGE OPERATION:

The loop in step (ii) above, is the heart of the algorithm. Each iteration of this loop increases the amplitude in the desired state by O($\frac{1}{\sqrt{N}}$), as a result in repetitions of the loop, the amplitude and hence the probability in the desired state reach O(1). In order to see that the amplitude increases by O($\frac{1}{\sqrt{N}}$), in each repetition, we first show that the diffusion transform, *D*, can be interpreted as an ***inversion about average*** operation. A simple inversion is a phase rotation operation and by **SELECTIVE ROTATION** ,which it is unitary.

The *inversion about average* operation also a unitary operation and is equivalent to the diffusion transform *D* as used in step (ii)(a) of the algorithm and at the same time The *inversion about average* operation is applied to a distribution in which all but one of the components is initially negative. This operation is precisely explained below:

CASE I:

Let α denote the average amplitude over all states, i.e. if α_i be the amplitude in the i^{th}state, then the average is $\frac{1}{\sqrt{N}}\sum_{i=1}^{N-1}\alpha_i$

As a result of the operation *D*, the amplitude in each state increases (decreases) so that after this operation it is as much below (above) a as it was above (below) a before the operation.

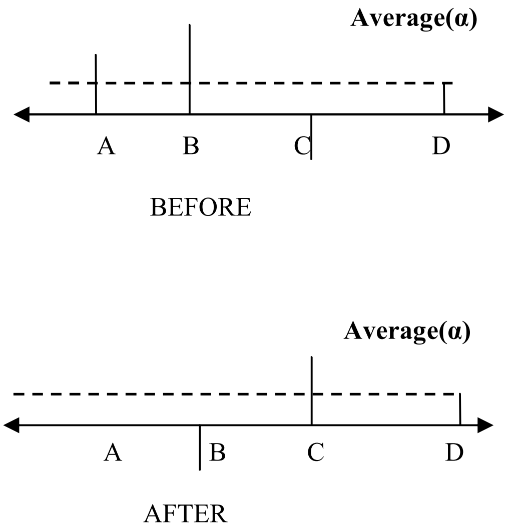


Figure 1:Inversion about average Operation

The diffusion transform, , is defined as follows:

$$\text{i) } \mathbf{D_{ij}} = \frac{2}{N} \text{ if } i \neq j$$

$$\text{ii) } \mathbf{D_{ii}} = -1 + \frac{2}{N}$$

Next it is proved that is indeed the ***inversion about average*** as shown in figure 1 above. Observe that *D* can be represented in the form where is the identity matrix and is a projection matrix with for all .

The following two properties of *P* are easily verified:

First: It is $\mathbf{P^2 = P}$

Second: that P acting on any vector gives a vector each of whose components is equal to the average of all components.

Using the fact that $\mathbf{P^2 = P}$, it follows immediately from the representation

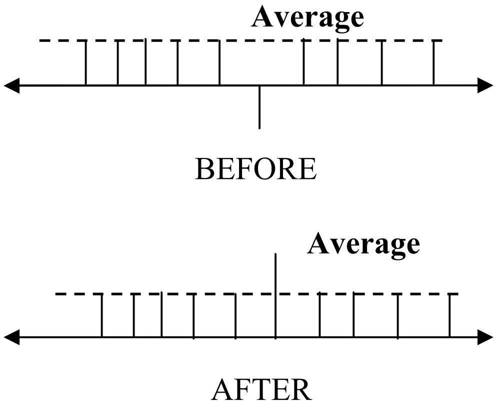
D= - I + 2P that $\mathbf{D^2 = I}$

and hence *D* is unitary.

CASE II:

Next consider what happens when the *inversion about average* operation is applied to a vector where each of the components, except one, are equal to avalue, say *C*, which is approximately $\frac{1}{\sqrt{N}}$; the one component that is different is negative

The average *A* is approximately equal to *C*. Since each of the **(N-1)** components is approximately equal to the average, it does not change significantly as a result of the inversion about average. The one component that was negative to start out, now becomes positive and its magnitude increases by approximately 2C , which is approximately $\frac{2}{\sqrt{N}}$



.

Figure 2 : The *inversion about average* operation is applied to a distribution

VI.EVOLUTION OF THIS ALGORITHM:

It is very much important that the above algorithm must be implemented in way that how accurately is it possible to find the desired element from the database There is a matching lower bound that suggests that it is not possible to identify the desired element in fewer than Ω(√N) steps.[4] This result states that any quantum algorithm running for *T* steps is only sensitive to $\mathbf{O(T^2)}$ queries (i.e. if there are more possible queries, then the answer to at least one can be flipped without affecting the behaviour of the algorithm).

So in order to correctly decide the answer which is sensitive to multiple queries will take a running time of Ω(√N). To see this assume that C(S)=0 for all states and the algorithm returns the right result, i.e. that no state satisfies the desired condition. Then, by T< Ω(√N),the answer to at least one of the queries about C(S) for some *S* can be flipped without affecting the result, thus giving an incorrect result for the case in which the answer to the query was flipped. gives a direct proof of this result along with tight bounds showing the algorithm of this paper is within a few percent of the fastest possible quantum algorithm.

A. Implementation Considerations:

This algorithm is likely to be simpler to implement as compared to other quantum mechanical algorithms for the following reasons:

i)The only operations required are, first, the Walsh -Hadamard transform, and second, the conditional phase shift operation both of which are relatively easy as compared to operations required for other quantum mechanical algorithms.

(ii) Quantum algorithms based on the Walsh-Hadamard transform are likely to be much simpler to implement than those based on the large scale Fourier transform.

(iii) The conditional phase shift would be much easier to implement if the algorithm was used in the mode where the function at each point was computed rather than retrieved from memory. This would eliminate the storage requirements in quantum memory.

B. A Real time Example – Quantum Search:

Imagine a phone directory containing *N* names arranged in completely random order. In order to find someone's phone number with a probability of $\frac{1}{2}$, any classical algorithm (whether deterministic or probabilistic) will need to look at a minimum of $\frac{N}{2}$ names.

Quantum systems can be in a superposition of states and simultaneously examine multiple names. By properly adjusting the phases of various operations, successful computations reinforce each other while others interfere randomly. As a result, the desired phone number can be obtained in only **O(√N)**, steps. The algorithm is within a small constant factor of the fastest possible quantum algorithm.

C.Efficiency And Query Complexity:

By Reichardt' span-program formalism [7], it is now known that the quantum query complexity of any formula of ***O*(1)** fan in on *N* variables is **O(√N)** This result culminates that a query search on 2n variables can be evaluated on quantum computers in time **O(20.5n)** using a continuous-time quantum walk, whereas classical computers require **Ω(20.753 n)** queries.

So the query Complexity is less when compared to classical algorithms. Discrete Quantum Walks[10] are specialized in this algorithm.

VII. OTHER OBSERVATIONS:

The algorithm as discussed here assumes a unique state that satisfies the desired condition. It can be easily modified to take care of the case when there are multiple
states satisfying the condition and it is
required to find one of these.

Two ways of achieving this are:

(i) The first possibility would be to repeat the experiment so that it checks for a range of degeneracy, i.e.
redesign the experiment so that it checks for the degeneracy of the solution being in the range (k,k+1,k+2,… 2k) for various *k.* Then within log *N* repetitions of this procedure, one can ascertain whether or not there exists at least one out of the *N* states that satisfies
the condition. [4]

(ii) The other possibility is to slightly perturb the problem in a random fashion as discussed in so that with a high probability the degeneracy is removed. There is also a scheme discussed in [3] by which it is possible to modify any algorithm that solves an NP search problem with a unique solution and use it to solve an NP-search problem in general.

VIII.CONCLUSION:

It is possible for quantum system to make ***interaction-free measurements*** by using the duality properties of photons]. In these the presence (or absence) of an object can be deduced by allowing for a very small probability of a photon interacting with the object. Therefore most probably the photon will not interact, however, just allowing a small probability of interaction is enough to make the measurement.

This suggests that in the multiple search problem also, it might be possible to find the object without examining all the objects but just by allowing a certain probability of examining the desired object which is something like what happens in the algorithm in this paper.

IX.REFERENCES:

[1]A. Berthiaume and G. Brassard, *Oracle quantum computing,* Journal of Modern Optics, Vol.41, no. 12, December 1994, pp. 2521-2535

[2]Quantum searching a classical database (or how we learned to stop worrying and love the bomb).pdf

[3]http://www.proofwiki.org/wiki/CNF_SAT_is_NP-complete

[4] C.H. Bennett, E. Bernstein, G. Brassard & U.Vazirani, *Strengths and weaknesses of quantum computing*, to be published in the SIAM Journal on Computing.

[5]. M. Boyer, G. Brassard, P. Hoyer & A. Tapp, *Tight bounds on quantum searching,* Proceedings, PhysComp 1996 (lanl e-print quant-ph/9605034).

[6] P. W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring,* Proceedings, 35th Annual Symposium on Fundamentals of Comp. Science (FOCS), 1994, pp. 124-134.

[7]www.stp.dias.ie/~dorlas/Papers/Feynman4.pdf

[8] Hazewinkel, Michiel, ed. (2001), "Trotter product formula",*Encyclopedia of Mathematics*, Springer, ISBN 978-1-55608-010-4

[9] Terry Rudolph and Dr.(Strange)Lov Grover Bell Labs, 600-700 Mountain Ave., Murray Hill, NJ 07974, U.S.A.(Dated: February 1, 2008).

[10]http://pra.aps.org/abstract/PRA/v48/i2/p1687_1