Optimization of Grover's Search Algorithm

Varun Garg * Anupama Pande †

Abstract—**Quantum mechanical computers unlike classical computers can be in a superposition of states and carry out multiple operations at the same time. Grover has shown that a quantum computer can do a random database search in O($\sqrt{N}$) steps. He has further proved that using the superposition of the states in quantum computing, a quantum computer should be able to do a search in a single step. He has stated that the query generated to do this search will require Ω(NlogN) steps.**

In this paper we aim at optimizing the Grover's search algorithm. In our algorithm, we have repeated the inversion step a number of times instead of stopping after a single step. Measurement after a single step required a larger number of subsystems which impaired the effectiveness of the algorithm leading to Ω(NlogN) steps in the preparation and processing of the query. Repetition of the inversion step brought the the number of subsystems down drastically by increasing the amplitude of the desired state.We have proved that by doing so the overall effectiveness of Grover's algorithm can be brought down to Ω($\sqrt[3]{N}$).

Keywords: Quantum mechanical computers, Grover's search algorithm, inversion step, probability and amplitude.

1 Introduction

The idea of a computational device based on quantum mechanics was first explored in the 1970's and early 1980's by physicists and computer scientists such as Charles H. Bennett of the IBM Thomas J. Watson Research Center, Paul A. Benioff of Argonne National Laboratory in Illinois, David Deutsch of the University of Oxford, and the late Richard P. Feynman of the California Institute of Technology (Caltech). The idea emerged when scientists were pondering the fundamental limits of computation. They understood that if technology continued to abide by Moore's Law, then the continually shrinking size of circuitry packed onto silicon chips would eventually reach a point where individual elements would be no larger than a few atoms. Here a problem arose because at the atomic scale the physical laws that govern the behavior and properties of the circuit are inherently quantum mechanical in nature, not classical. This then raised the question of whether a new kind of computer could be devised based on the principles of quantum physics.

Feynman was among the first to attempt to provide an answer to this question by producing an abstract model in 1982 that showed how a quantum system could be used to do computations. He also explained how such a machine would be able to act as a simulator for quantum physics. In other words, a physicist would have the ability to carry out experiments in quantum physics inside a quantum mechanical computer [1].

This dramatic advantage of quantum computers has only been discovered for these problems so far: factoring, discrete logarithm. However, there is no proof that the advantage is real: an equally fast classical algorithm may still be discovered. There is one other problem where quantum computers have a smaller, though significant (quadratic) advantage. It is quantum database search, and can be solved by Grover's algorithm. In this case the advantage is provable. This establishes beyond doubt that (ideal) quantum computers are superior to classical computers for at least one problem [6]. In this paper we aim at optimizing the Grover's search algorithm. Grover in his algorithm stopped after one single inversion step. Measurement after a single step required a larger number of subsystems which impaired the effectiveness of the algorithm. This led to Ω(NlogN) steps in the preparation and processing of the query. In our modified algorithm, we have repeated the inversion step a number of times and because of this the number of subsystems required is brought down drastically. The increase in the amplitude of the desired state by the repetition of the inversion step improved the overall effectiveness of the algorithm. We established that the lower bound of database search using Grover's search algorithm is Ω($\sqrt[3]{N}$).

2 Preliminaries

A quantum bit or qubit is a unit of quantum information. That information is described by a state vector in a two-level quantum mechanical system which is formally equivalent to a two-dimensional vector space over the complex numbers. Benjamin Schumacher discovered a way of interpreting quantum states as information. He came up with a way of compressing the information in a state, and storing the information on a smaller number

*Varun Garg is a B.Tech from IIT Kanpur. Currently he is working as a reservoir engineer with Shell EP solutions. Tel: +91-9900540020 Email: varun.garg@shell.com

†Anupama Pande is a B.Tech from HBTI Kanpur. Currently she is working as software engineer with IBM India software Labs. Tel: +91-990025972 Email: anupamapande@in.ibm.com

of states. This is now known as Schumacher compression. A qubit has some similarities to a classical bit, but is overall very different. Like a bit, a qubit can have two possible valuesnormally a 0 or a 1. The difference is that whereas a bit must be either 0 or 1, a qubit can be 0, 1, or a superposition of both. The states a qubit may be measured in are known as basis states (or vectors). As is the tradition with any sort of quantum states, Dirac, or bra-ket notation is used to represent them.

A pure qubit state is a linear superposition of those two states. This means that the qubit can be represented as a linear combination of (0) and (1):

$$\omega = \alpha(0) + \beta(1), \tag{1}$$

where α and β are probability amplitudes and can in general both be complex numbers.

When we measure this qubit in the standard basis, the probability of outcome of (0) is α^2 and the probability of outcome of (1) is β^2. Because the absolute squares of the amplitudes equate to probabilities, it follows that α and β must be constrained by the equation

$$\alpha^2 + \beta^2 = 1, \tag{2}$$

simply because this ensures you must measure either one state or the other.

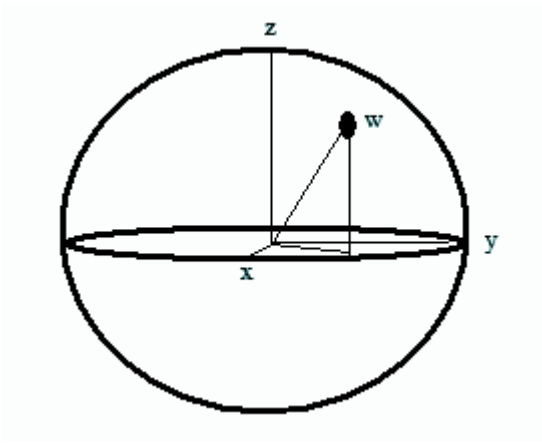

Figure 1: A qubit representation by a Bloch sphere

An important distinguishing feature between a qubit and a classical bit is that multiple qubits can exhibit quantum entanglement. Quantum entanglement allows qubits that are separated by incredible distances to interact with each other instantaneously (not limited to the speed of light). No matter how great the distance between the correlated particles, they will remain entangled as long as they are isolated [1].

Taken together, quantum superposition and entanglement create an enormously enhanced computing power. Where a 2-bit register in an ordinary computer can store only one of four binary configurations (00, 01, 10, or 11) at any given time, a 2-qubit register in a quantum computer can store all four numbers simultaneously, because each qubit represents two values. If more qubits are added, the increased capacity is expanded exponentially [6].

The class of problems that can be efficiently solved by quantum computers is called BQP which stands for bounded error, quantum, polynomial time. Quantum computers only run probabilistic algorithms, so BQP on quantum computers is the counterpart of BPP on classical computers. It is defined as the set of problems solvable with a polynomial-time algorithm, whose probability of error is bounded away from one quarter. A quantum computer is said to solve a problem if, for every instance, its answer will be right with high probability. If that solution runs in polynomial time, then that problem is in BQP [10].

BQP is suspected to be disjoint from NP-complete and a strict superset of P, but that is not known. Both integer factorization and discrete log are in BQP. Both of these problems are NP problems suspected to be outside BPP, and hence outside P. Both are suspected to not be NP-complete. There is a common misconception that quantum computers can solve NP-complete problems in polynomial time. That is not known to be true, and is generally suspected to be false.

An operator for a quantum computer can be thought of as changing a vector by multiplying it with a particular matrix. Multiplication by a matrix is a linear operation. Daniel S. Abrams and Seth Lloyd have shown that if a quantum computer could be designed with nonlinear operators, then it could solve NP-complete problems in polynomial time. It could even do so for #P-complete problems. They do not believe that such a machine is possible [6].

A quantum database search can be solved by Grover's algorithm. In this case the advantage is provable. Grover proved that if a problem has following four properties:

i. The only way to solve it is to guess answers repeatedly and check them,

ii. There are n possible answers to check,

iii. Every possible answer takes the same amount of time to check, and

iv. There are no clues about which answers might be better: generating possibilities randomly is just as good as checking them in some special order.

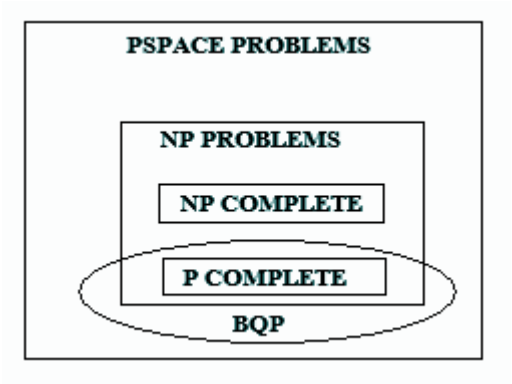

Figure 2: The suspected relationship of BQP to other problem spaces

then the time for a quantum computer to solve this will be proportional to the square root of n (it would take an average of $\frac{n+1}{2}$ guesses to find the answer using a classical computer.) [6]

In short, Grover's algorithm is a quantum algorithm for searching an unsorted database with N entries in $O(\sqrt{N})$ time and using O(logN) storage space [7]. Like many quantum computer algorithms, Grover's algorithm is probabilistic in the sense that it gives the correct answer with high probability. Grover proved that the number of steps taken for a quantum computer to search for an item in a database will be of $O(\sqrt{N})$ in comparison to its classical counterpart which will take O(N) steps [7].

Grover further showed that since the quantum computer can search all N items simultaneously, it is possible to search an arbitrarily large database in a single query [8]. But in the process of preparing and processing the query, the over all effectiveness of the algorithm goes down to Ω(NlogN).

3 Optimization of Grover's Search Algorithm

Quantum mechanical computers can be in a superposition of states and carry out multiple operations at the same time. An algorithm that uses this parallelism is which searches an N item database for a single marked item in $O(\sqrt{N})$ quantum queries where each query pertains to only one of the N items [7]. Grover has also justified the quantum approach of being able to access all the items in the database simultaneously and hence being able to search an element in a single query [8]. In the process, the over all effectiveness of the algorithm dipped down to Ω(NlogN). This is primarily due to the complexity of the nature of the query generated and the post processing of the data.

Let a system have $N = 2^n$ states which are labeled S_1 , $S_2 ... S_N$. These 2n states are represented as n bit strings. Let there be a unique state, say S_v, that satisfies the condition $C(S_v) = 1$, whereas for all other states S, $C(S_v) = 0$. The problem is to identify the state S_v. Initialize the system to the superposition: $(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}} ... \frac{1}{\sqrt{N}})$i.e., there is the same amplitude to be in each of the N states. This superposition can be obtained in O(logN) steps [7].

Let the system be in any state S:

In case

$$C(S_v) = 1, \tag{3}$$

rotate the phase by π radians ;

In case

$$C(S_v) = 0, \tag{4}$$

leave the system unaltered.

By using such a selective inversion followed by an inversion about average operation, Grover showed that the magnitude of the amplitude in marked state(s) can be increased by a certain amount [8]. The inversion about average operation is defined by the following unitary operation

$$D_{ij} = \frac{2}{N} \tag{5}$$

if i ≠ j and as;

$$D_{ij} = -1 + \frac{2}{N} \tag{6}$$

if i = j.

This can be physically implemented as a product of three local unitary matrices [7].

Assume that D is applied to a superposition with each component of the superposition, except one, having amplitude equal to $\frac{1}{\sqrt{N}}$; the one component that is different has amplitude of $-\frac{1}{\sqrt{N}}$. The one that was negative, now becomes positive and its magnitude increases to approximately $\frac{3}{\sqrt{N}}$, the rest stay virtually unchanged as shown in the figure 3.

Consider a quantum system composed of multiple subsystems. Each subsystem has an N dimensional state space like the one used in the Grover's search algorithm [7]. Each basis state of a subsystem corresponds to an item in the database. It is shown that with a single quantum query, pertaining to information regarding all N items, the amplitude (and thus probability) in the state corresponding to the marked item(s) of each subsystem can be amplified by a small amount. By choosing the number of

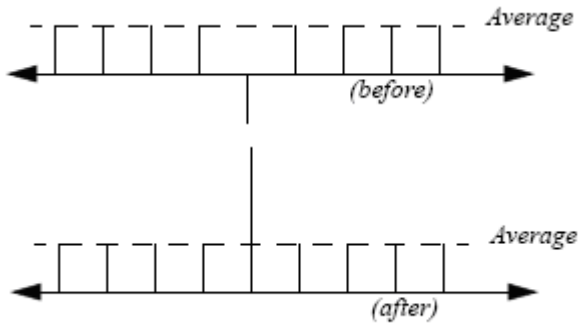

Figure 3: The inversion about average operation is applied to a superposition in which all but one of the components amplitude is initially $\frac{1}{\sqrt{N}}$; one of the components is initially at amplitude of -$\frac{1}{\sqrt{N}}$.

subsystems to be appropriately large, and by repeating such queries, this small difference in probabilities can be estimated by making a measurement to determine which item of the database each subsystem corresponds to - the item pointed to by the most subsystems is the marked item.

Consider a tensor product of η identical quantum mechanical subsystems - all subsystems have an N dimensional state space. Each of the N basis states corresponds to an item in the database. All subsystems are placed in a superposition with equal amplitude in all N states.

Assuming N to be a power of 2, the state of each subsystem is initialized by taking a set of $\log_2$N qubits which gives N states. The system consists of η such subsystems. Each qubit is placed in the superposition $(\frac{1}{\sqrt{2}})((0) + (1))$, thus obtaining equal amplitudes in all N states. Denoting the N states by S_1, S_2...S_N, the state vector is proportional to $((S_1)+(S_2) +...+ (S_N))^\eta$ which may be written as $((S_1\ S_1...S_1) + (S_1\ S_1..S_2) +...+ N^\eta$ such terms).

Query the database as to whether the number of subsystems (out of the η subsystems) in the state corresponding to the marked item, is odd or even. In case it is odd, invert the phase; if it is even, do nothing. This is achieved by using the technique described in section 3.2.

Let S_1 be the state corresponding to the marked item. The state vector after this operation becomes as $(\pm(S_1\ S_1...S_1) \pm (S_1\ S_1...S_2)$ ...N^η such terms). The sign of each term is determined by whether the state corresponding to the marked item (-S_1) is present an odd or even number of times in the respective term. This state vector can be factored and written as $((S_1)+(S_2)+...+ (S_N))^\eta$. The system is now in a tensor product of η identical quantum mechanical subsystems, each of which has an N dimensional state space. In each of the subsystems, the phase of the amplitude in the basis state corresponding to the marked item is inverted.

Note that by a single operation on the multi-system wave function, the wave function of each subsystem has been altered in a suitable way. Using a single query, the phase of the amplitude in the state corresponding to the marked item in each of these η subsystems is inverted - the reason it needs only a single query is that the new phase can have only two possible values ($\pm$1). The only statistic needed from the oracle is: Is the number of subsystems in the state corresponding to the marked item is odd or even?

After this a single inversion about average operation is carried out on each of the subsystems separately.Since the system is in a tensor product of η identical quantum mechanical subsystems, each subsystem can be independently operated on.This increases the amplitude of the marked state, which was negative, by a factor of 3 by an inversion about average operation.The state vector after carrying out this operation becomes approximately: $(3(S_1)+(S_2) +...+ (S_N))^\eta$.

This is where the Grover's algorithm stops and makes a measure of the number of subsystems in the marked state [8]. In our algorithm we apply the inversion step and inversion about the average step a number of times. The state vector after single inversion is $(3(S_1)+(S_2) +...+ (S_N))^\eta$ which may be written as $(3^\eta(S_1\ S_1...S_1) + 3^{\eta-1}(S_1\ S_1...S_2) +...+ N^\eta$ such terms).

The next inversion step is carried in a similar way as the first inversion step. The state vector after this operation becomes $(\pm\ 3^\eta(S_1\ S_1...S_1) \pm 3^{\eta-1}(S_1\ S_1...S_2) \pm...+ N^\eta$ such terms). The sign of each term is determined by whether the state corresponding to the marked item (S1) is present an odd or even number of times in the respective term. This state vector can be factored and written as $(\ -3(S_1) + (S_2) +...+ (S_N))^\eta$.

After this a single inversion about average operation is carried out on η the subsystems separately. This increases the amplitude of the marked state, which was negative, by an inversion about average operation. The state vector after carrying out this operation becomes approximately: $(5(S_1) + (S_2) +...+ (S_N))^\eta$.

Next the inversion step is followed by the inversion about the average step n-1 times. After n-1 steps, the state vector can be written as $((2n-1)(S_1) + (S_2) +... + (S_N))^\eta$. After the nth step, the state vector becomes $(\pm(2n-1)^\eta(S_1\ S_1...S_1) \pm (2n-1)^{\eta-1}(S_1\ S_1...S_2) \pm ... + N^\eta$ such terms. This state vector can be factored and written as $(-(2n-1)(S_1) + (S_2) +...+ (S_N))^\eta$. Now, we do the inversion about the average step for the nth time. The state vector after carrying out this operation becomes approximately: $((2n+1)(S_1) + (S_2) +...+ (S_N))^\eta$.

Now, in the final step the algorithm makes a measurement

that projects each subsystem into one of its basis states that points to an item in the database. The item that the most subsystems point to is the marked item.

Assume the number of subsystems (i.e. η) and the number of steps iteration carried out (i.e. n) to be sufficiently large. The probability of obtaining the basis state corresponding to the marked item (S_N) in each of the η subsystems is approximately $\frac{(2n+1)^2}{N}$ and the probability of obtaining another basis state is approximately $\frac{1}{N}$. We assume number of subsystems (i.e. η) and the number of steps iteration has been carried out (i.e. n) to be sufficiently large. The algorithm assumes 2n+1 $\approx$ 2n, since n $\gg$ 1. Hence the probability of finding a subsystem in the marked state can be written as $4n^2/N$.

It follows by the law of large numbers, that out of η subsystems, $\frac{4n^2\eta}{N} \pm O(\sqrt{\frac{n^2\eta}{N}})$ lie in state S_1 while $\frac{\eta}{N} \pm O(\sqrt{\frac{\eta}{N}})$ lie in each of the other basis state [3]. To test the lower limit of the algorithm, we now put $\eta = n = \sqrt[3]{N}$. This will lead to the probability of obtaining the basis state corresponding to the marked state item (S_1) in each of the η subsystems to be $4 \pm O(1)$. Hence, the item that the most subsystems point to is the marked item.

4 Conclusion and Comparison with Grover's Algorithm

The Grover's single step search algorithm made the measurement after a single inversion step [8]. This made the probability of obtaining the basis state corresponding to the marked item in each of η subsystems approximately $\frac{9}{N}$. This implied that out of η subsystems, $\frac{9\eta}{N} \pm O(\sqrt{\frac{\eta}{N}})$ lie in state S_1 while $\frac{\eta}{N} \pm O(\sqrt{\frac{\eta}{N}})$ lie in each of the other basis state. This made the requirement to make the number of subsystems (i.e. η) large enough to neglect the uncertainty caused by $O(\sqrt{\frac{\eta}{N}})$ term. As shown by Grover this number comes out to be Ω(NlogN) [8]. This hampers the overall effectiveness of the algorithm. While in Grover's original algorithm the number of subsystems has been kept to 1 while iterating the inversion steps to $O(\sqrt{N})$. This way the amplitude and hence the probability of obtaining the desired state reaches O(1).

In our optimised algorithm,the advantages of both the algorithms have been merged. The algorithm uses the advantage of going through multiple inversion steps and of having multiple subsystems to get a more efficient form. This takes over all $\Omega(\sqrt[3]{N})$ quantum steps in comparison to $O(\sqrt{N})$ steps and Ω(NlogN) steps used inGrover's Algorithm.

References

[1] A. Elitzur and L. Vaidman in Foundations of Physics 23, 1993, pp. 987-997.

[2] C. Durr and P. Hoyer, A quantum algorithm for finding the minimum, lanl preprint, quant-ph/9602016.

[3] D. Beckman, A.N. Chari, S. Devabhaktuni and J.Preskill in Phys. Rev. A 54(1996), 1034-1063.

[4] D. Deutsch and R. Jozsa in Proc. Royal Society of London, A400, 1992, pp. 73-90.

[5] http://whatis.techtarget.com

[6] http://www.wikipedia.com

[7] L.K.Grover, Quantum Mechanics Help in Searching for a Needle in a Haystack, Phys. Rev. Letter 79, 325-328, 1997.

[8] L.K.Grover, Quantum Computers can Search Arbitrarily Large Databases by a Single Query, Phys. Rev. Letter 79, 4709-4712, 1997.

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

[10] W. Feller, An Introduction to Probability Theory and its Applications, Vol. I and II, John Wiley Publishers,1971.