How hard is to search an unsorted database containing N items? A quantum mechanical algorithm

Lov K. Grover
3C-404A, AT&T Bell Labs
600 Mountain Avenue
Murray Hill NJ 07974

Summary

An unsorted database contains N records out of which just one satisfies a particular property. The problem is to identify that one record. Any classical algorithm, deterministic or probabilistic, will clearly take $O(N)$ steps since on the average it will have to examine a large fraction of the N records. Quantum mechanical systems can do several operations simultaneously due to their wave like properties. This paper gives an $O(\sqrt{N})$ step quantum mechanical algorithm for identifying that record.

1. Introduction

1.1 Search Problems The typical problem in theoretical computer science is 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 above in the summary, 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 the structure on the speed of the algorithm. For example the SAT problem of NP-completeness asks whether it is possible to find any combination of n binary variables that satisfies a certain set of clauses C 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 $O(\log N) = O(n^k)$ time. Thus if it were possible to reduce the number of steps to $O(\log N)$ (instead of $O(\sqrt{N})$ as we do in this paper), it would give us a polynomial time algorithm for NP-complete problems.

In view of the fundamental nature of the search problem, it is natural to ask - how fast can the basic identification problem be solved without assuming anything about the structure? It has always been assumed that this limit is $O(N)$ since there are N items to be gone through. However, quantum mechanical systems can sometimes be counter-intuitive [1][3][4].

1.2 Quantum mechanical algorithms A good starting point to think of quantum mechanical algorithms is probabilistic algorithms [1] (e.g. simulated annealing). In these instead of having the system in a specified state, it is in a distribution over various states with a certain probability of being in each state. At each step, there is a certain probability of making a transition from one state to another. The evolution of the system is obtained by premultiplying this probability vector (that describes the distribution of amplitudes over various states) by a state transition matrix. Knowing the initial distribution and the transition matrix, it is possible in principle to calculate the distribution at any point in time.

Just like classical probabilistic algorithms, quantum mechanical algorithms have a probability distribution over various states. However, the probability itself does not completely describe the system. In order to completely describe the system we need the *amplitude* in each state which is a complex number. The evolution of the system is obtained by premultiplying this amplitude vector (that describes the distribution of amplitudes over various states) by a transition matrix, the entries of which are complex in general. The probabilities in any state are given by the squared of the absolute values of the amplitudes. It can be shown that in order to conserve probabilities, the state transition matrix has to be unitary [1].

2. The Abstracted Problem Let a system contain $N = 2n$ states which are labelled $S_1 S_2 \dots S_N$. 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) = 0. The problem is to identify the state S_v.

3. Algorithm

(i) Initialize the system to the state $\left(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}} \dots \frac{1}{\sqrt{N}}\right)$, i.e. there is the same amplitude to be in all N states. It is possible to obtain this state in $O(\log N)$ steps as discussed in [3][4].

(ii) Repeat the following unitary operations $O(\sqrt{N})$ times:

Let the system be in the state S:

(a) In case C(S) = 1, rotate the phase by π radians, in case C(S) = 0, leave the system unaltered.

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

$$D_{ij} = \frac{2}{N} \text{ if } i \neq j \ \& \ D_{ii} = -1 + \frac{2}{N}$$

This diffusion transform, D, can be implemented as $D = FRF$, where F is the Fourier Transform Matrix [4] and R is defined as follows:

$R_{ij} = 0$ if $i \neq j$ & $R_{ii} = 1$ if $i = 0$ & $R_{ii} = -1$ if $i \neq 0$.

(iii) Sample the resulting state.

The algorithm identifies the state that satisfies C(S) = 1 with a probability of O(1).

4. Outline of rest of paper

The following sections contains theorems which prove that the system discussed in section 3 is indeed a valid quantum mechanical system and it converges to the desired state with a probability O(1). The proofs are divided into two parts. First, it is proved that the system is a valid quantum mechanical system and, second, that it converges to the desired state. In order to prove that it is a valid quantum mechanical system we need to prove that it is unitary and that it can be implemented as a product of local transition matrices. The diffusion transform D is defined by the matrix D as follows:

(4.0) $$D_{ij} = \frac{2}{N}, \text{ if } i \neq j \ \& \ D_{ii} = -1 + \frac{2}{N}.$$

Theorem 1- D is unitary.

The way D is presented above, it is not a local transition matrix since there are transitions from each state to all N states. Using the Fourier transformation matrix [4] it can be implemented as a product of 3 unitary transformations, each of which is a local transition matrix. As discussed in

[4], the Fourier transformation matrix F is defined as $F_{ij} = 2^{-n/2}(-1)^{\bar{i} \cdot \bar{j}}$, where $\bar{i}$ is the binary representation of i, and $i \cdot j$ denotes the bitwise dot product of the two n bit strings $\bar{i}$ and $\bar{j}$.

Theorem 2 - D can be expressed as $D = FRF$, where F is the Fourier Transform Matrix [4] and R is defined as follows: $R_{ij} = 0$ if $i \neq j$ & $R_{ii} = 1$ if $i = 0$ & $R_{ii} = -1$ if $i \neq 0$.

Next it is proved that the algorithm indeed converges to the desired state. In order to follow the evolution of the state when the unitary transformations are successively applied, we first prove:

Theorem 3 - Let the state vector be as follows - for any one state the amplitude is k, for each of the remaining *(N-1)* states the amplitude is l. Then after applying the diffusion transform D, the amplitude in the one state is $k_1 = \left(\frac{2}{N} - 1\right)k + 2\frac{(N-1)}{N}l$ and the amplitude in each of the remaining *(N-1)* states is $l_1 = \left(\frac{2}{N}k + \frac{(N-2)}{N}l\right)$.

Using this we prove a simple relation among the amplitudes k & l.

Corollary 3.1 - Let the state vector be as follows - for any one state the amplitude is k, for each of the remaining *(N-1)* states the amplitude is l. Let k and l be real numbers (in general the amplitudes can be complex). Let k be negative and l be positive such that $\left|\frac{k}{l}\right| < \sqrt{N}$. Then after applying the diffusion transform both k and l are positive numbers.

As is well known, in a unitary transformation the total probability is conserved - this is proved for the particular case of the diffusion transformation.

Corollary 3.2 - Let the state vector be as follows - for the state that satisfies C(S) = 1, the amplitude is k, for each of the remaining *(N-1)* states the amplitude is l. Then if after applying the diffusion transformation D, the new amplitudes are respectively k_1 and l_1 as derived in theorem 3, then $k_1^2 + (N-1)l_1^2 = k^2 + (N-1)l^2$.

k & l vary with the number of iterations of steps (ii) of the algorithm in section 3. Using this conservation result we can immediately obtain l if we are given k at any point in the algorithm - it is thus enough to keep track of one variable k or l, not both. Using this we prove -

Theorem 4 - In case $k < O(\sqrt{N})$, the change in k (Δk) in one repetition of the loop in step (ii) of the algorithm in section 3 is lower bounded by $\frac{\Delta k}{k} > O\left(\frac{1}{\sqrt{N}}\right)$.

Using this it immediately follows that in $O(\sqrt{N})$ steps, k will approach $O(\sqrt{N})$. Since the probability of the system being found in any particular state is proportional to the square of the amplitude, it follows that the probability of the particle being in the desired state after $O(\sqrt{N})$ steps is $\frac{k^2}{N} = O(1)$. Therefore if the system is now sampled, it will be in the desired state with a probability $O(1)$. Finally it is proved that once we can design an experiment with a finite success probability ε, in $O\left(\frac{1}{\varepsilon}\right)$ repetitions of the experiment it is possible to have a success probability very close to unity.

Theorem 5 - Let the probability of a particular outcome in an experiment be ε. Then if the experiment be repeated $\frac{\kappa}{\varepsilon}$ times, the probability that the outcome A does not happen even once is less than or equal to $e^{-\kappa}$.

5. Proofs

As mentioned before (4.0), the diffusion transform D is defined by the matrix D as follows:

(5.0) $$D_{ij} = \frac{2}{N}, \text{ if } i \neq j \ \& \ D_{ii} = -1 + \frac{2}{N}.$$

Theorem 1- D is unitary.
Proof - For a matrix to be unitary it is necessary and sufficient to prove that the columns are orthonormal. The magnitude of each column vector is $\sum_{i=1}^{n} D_{ij}^2 + D_{jj}^2$, $j = 1,2...n$. Substituting from (5.0), it follows that this is $\left(1 - \frac{2}{N}\right)^2 + (N-1)\frac{4}{N^2}$ which is equal to 1. The dot product of two column vectors is $\sum_{i=1}^{n} D_{ik} D_{ij}$. This is $N\frac{4}{N^2} - \frac{4}{N} = 0$.

Theorem 2 - D can be expressed as $D = FRF$, where F is the Fourier Transform Matrix and R is defined as follows: $R_{ij} = 0$ if $i \neq j$ & $R_{ii} = 1$ if $i = 0$ & $R_{ii} = -1$ if $i \neq 0$.
Proof - We evaluate FRF and show that it is equal to D. As discussed in section 4, $F_{ij} = 2^{-n/2}(-1)^{\bar{i} \cdot \bar{j}}$, where $\bar{i}$ is the binary representation of i, and $\bar{i} \cdot \bar{j}$ denotes the bitwise dot product of the two n bit strings $\bar{i}$ and $\bar{j}$. R can be written as $R = R_1 + R_2$ where $R_1 = -I$ where I is the identity matrix (with ones on the diagonal) and $R_{2,00} = 2$, $R_{2,ij} = 0$ if $i \neq 0, j \neq 0$. It is easily proved (and well known [4]) that $FF=I$ and hence $D_1 = FR_1F = -I$. We next evaluate $D_2 = FR_2F$. By standard matrix multiplication $D_{2,ad} = \sum_{bc} F_{ab} R_{2,bc} F_{cd}$. Using the definition of R_2,

it follows that $D_{2,ad} = 2F_{a0}F_{0d} = \frac{2}{2^n}(-1)^{a\cdot\bar{0}+\bar{0}\cdot d} = \frac{2}{N}$. Thus all elements of the matrix D_2 equal $\frac{2}{N}$.

Theorem 3 - Let the state vector be as follows - for any one state the amplitude is *k*, for each of the remaining *(N-1)* states the amplitude is *l*. Then after applying the diffusion transform D, the amplitude in the one state is $k_1 = \left(\frac{2}{N}-1\right)k + 2\frac{(N-1)}{N}l$ and the amplitude in each of the remaining *(N-1)* states is $l_1 = \frac{2}{N}k + \frac{(N-2)}{N}l$.

Proof -Using the definition of the diffusion transform (5.0), it follows that

$$k_1 = \left(\frac{2}{N}-1\right)k + 2\frac{(N-1)}{N}l$$

$$l_1 = \left(\frac{2}{N}-1\right)l + \frac{2}{N}k + \frac{2(N-2)}{N}l = \frac{2}{N}k + \frac{(N-2)}{N}l$$

Corollary 3.1 - Let the state vector be as follows - for any one state the amplitude is *k*, for each of the remaining *(N-1)* states the amplitude is *l*. Let *k* and *l* be real numbers (in general the amplitudes can be complex). Let k be negative and *l* be positive and $\left|\frac{k}{l}\right| < \sqrt{N}$. Then after applying the diffusion transform both k_1 and l_1 are positive numbers.

Proof - From theorem 3 we have that $k_1 = \left(\frac{2}{N}-1\right)k + 2\frac{(N-1)}{N}l$. Assuming $N > 2$, it follows that $\left(\frac{2}{N}-1\right)$ is negative; by assumption *k* is negative and $2\frac{(N-1)}{N}l$ is positive and hence $k_1 > 0$. Similarly it follows that if $l > -\frac{2}{N-2}k$ then $l_1 > 0$. Thus if $\left|\frac{k}{l}\right| < \frac{(N-2)}{2}$, then $l_1 > 0$. If $\left|\frac{k}{l}\right| < \sqrt{N}$, then for non-trivial *N* the condition is satisfied and $l_1 > 0$.

Corollary 3.2 - Let the state vector be as follows - for the state that satisfies C(S) = 1, the amplitude is *k*, for each of the remaining *(N-1)* states the amplitude is *l*. Then if after applying the diffusion transformation D, the new amplitudes are respectively k_1 and l_1 as derived in theorem 3, then $k_1^2 + (N-1)l_1^2 = k^2 + (N-1)l^2$.

Proof - Using theorem 3 it follows that

$$k_1^2 = \frac{(N-2)^2}{N^2}k^2 + 4\frac{(N-1)^2}{N^2}l^2 - \frac{4(N-2)(N-1)}{N^2}kl$$

$$(N-1)l_1^2 = \frac{4(N-1)^2}{N^2}k^2 + \frac{(N-2)^2}{N^2}(N-1)l^2 + \frac{4(N-2)(N-1)}{N^2}kl$$

Adding the previous two equations the corollary follows.

Theorem 4 - In case $k < O(\sqrt{N})$, the change in k (Δk) in one repetition of the loop is lower bounded as: $\frac{\Delta k}{k} > O\left(\frac{1}{\sqrt{N}}\right)$.

Proof - Using theorem 3 it follows that $\Delta k = -\frac{2}{N}k + 2\left(1 - \frac{1}{N}\right)l$. Since $k < O(\sqrt{N})$, it follows from corollary 3.2 that $l = O(1)$ and by corollary 3.1 that l is positive. Therefore $\Delta k = O(1)$, thus proving the result.

In order to prove theorem 5, we first prove a simple lemma

Lemma 5 - $(1 - \gamma)^{\kappa/\gamma} < e^{-\kappa}$ provided $\kappa, \gamma > 0$.

Proof - Observe that $(1 - \gamma) < e^{-\gamma}$. Now raise both sides to the $\frac{\kappa}{\gamma}$ power.

Theorem 5 - Let the probability of a particular outcome of an experiment be ε. Then if the experiment be repeated $\frac{\kappa}{\varepsilon}$ times, the probability that the outcome A does not happen even once is less than or equal to $e^{-\kappa}$.

Proof - The probability that the outcome A does not happen even once in $\frac{\kappa}{\varepsilon}$ experiments is

$(1 - \varepsilon)^{\kappa/\varepsilon}$. By lemma 5 this is less than or equal to $e^{-\kappa}$.

6. Observations

1. As mentioned in the introduction, the algorithm discussed in this paper does not use any knowledge about the problem except for the fact that there is a unique state that satisfies C(S) = 1. There would exist simpler and faster schemes that make use of more structure of the problem at hand. For example Shor's primality testing algorithm [3] for a number N can be viewed as a search of the (N-1) numbers smaller than N to see if any of them is a factor of N. This algorithm, by making use of some features of the structure of the problem, solves the problem in $O(\log N)$ steps.

2. The algorithm as discussed here assumes a unique state that satisfies the desired condition. It can be easily modified to take care of the possibility that there are multiple states that satisfy the condition by one of the following two ways.
(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 $(k, k+1, \ldots 2k)$ for various k. Then within $\log N$ repetitions of this procedure, one can ascertain whether or not there exists at least one state that satisfies the desired property.
(ii) The other possibility is to slightly perturb the problem in a random fashion as discussed in [2] so that with a high probability the degeneracy is removed.

7. References

[1] E. Bernstein and U. Vazirani, *Quantum Complexity Theory*, Proceedings 25th ACM Symposium on Theory of Computing, pp. 11-20, 1993.

[2] K. Mulmuley, U. Vazirani and V. Vazirani, *Matching is as easy as matrix inversion,* Combinatorica, 7 (1987), pp. 105-131.
[3] P. Shor, *Algorithms for quantum computation: discrete logarithms and factoring,* Proceedings, 35th Annual Symposium on Fundamentals of Computer Science (FOCS),1994, pp. 124-134.
[4] D. Simon, *On the power of quantum computation,* Proceedings, 35th Annual Symposium on Fundamentals of Computer Science (FOCS),1994, pp. 116-123.