Is partial quantum search of a database any easier?

Lov K. Grover[⋆1] and Jaikumar Radhakrishnan[⋆⋆2]

[1] Bell Laboratories, Lucent Technologies,
600–700 Mountain Avenue, Murray Hill, NJ 07974
e-mail: lkgrover@bell-labs.com
[2] School of Technology and Computer Science,
Tata Institute of Fundamental Research, Mumbai 400005, India
e-mail:jaikumar@tifr.res.in

Abstract. We consider the partial database search problem where given a quantum database $\{0,1\}^n \xrightarrow{f} \{0,1\}$ with the condition that $f(x) = 1$ for a unique $x \in \{0,1\}^n$, we are required to determine *only* the first k bits of the address x. We derive an algorithm and a lower bound for this problem. Let $q(k,n)$ be the minimum number of queries needed to find the first k bits of the required address x with certainty (or with very high probability, say $1 - O(N^{-\frac{1}{4}})$). We show that there exist constants c_k (corresponding to the algorithm) and d_k (corresponding to the lower bound) such that

$$\frac{\pi}{4}\left(1 - \frac{d_k}{\sqrt{K}}\right)\sqrt{N} \leq q(k,n) \leq \frac{\pi}{4}\left(1 - \frac{c_k}{\sqrt{K}}\right)\sqrt{N},$$

where $K = 2^k$ and $N = 2^n$. Our algorithm returns the correct answer with probability $1 - O(1/\sqrt{N})$, and can be easily modified to give the correct answer with certainty. The lower bound for algorithms that return the correct answer with certainty is proved by reducing the usual database search problem to this partial search problem, and invoking Zalka's lower bound showing that Grovers algorithm is optimal for the usual database search problem. We then derive a lower bound that is applicable for database search algorithms that err with small probability, and use it to show that our lower bound also applies to partial search algorithms that return the correct answer with probability at least $1 - O(N^{-\frac{1}{4}})$. Thus, it is easier to determine some k bits of the target address than to find the entire address, but as k becomes large this advantage reduces rapidly.

⋆ Research was partly supported by NSA & ARO under contract no. DAAG55-98-C-0040.
⋆⋆ This work was done during the author's visit to Bell Labs which was supported by the above contract; current affiliation: Toyota Technological Institute at Chicago, 1427 E. 60th Street, Chicago, IL 60637.

1 Introduction

In the database search problem, we are required to determine the address where a given item is stored. In this paper, we consider a variant of the problem, where we do not want the entire address but only its first k bits. The motivation for considering this problem is that quite often in real databases, just a small part of the target address contains all the information we need. For example: the items in a database may be listed according to the order of preference (say a merit-list which consists of a ranking of students in a class sorted by the rank). We want to know roughly where a particular student stands—we might want to know whether he/she ranks in the top 25%, the next 25%, the next 25%, or the bottom 25%. In other words, we want to know the first two bits of the rank and since the list was sorted by rank, we need just the first two bits of the address.

1.1 Partial classical search is easier

First consider classical algorithms. We will assume that our algorithms make no errors. Using a simple randomized classical search algorithm one can find an element in such a database using, on an average, $\frac{N}{2}$ queries (N is the number of elements in the database). This bound is tight if we wish to to determine the location of the element exactly.

Now, consider partial search. That is, the range $\{1, 2, \ldots, N\}$ into K equal blocks (in the example above, these blocks are intervals) and we are asked to determine in which of these blocks the target item lies. A natural algorithm for this problem is to randomly choose $K-1$ of the K blocks and probe the locations in these blocks in some random order. If the target lies in the block that is being searched, the algorithm will find it; however, if it does not find the target, we come to know that it is one of the $\frac{N}{K}$ items that the algorithm has not probed. A simple analysis shows that the expected number of queries made by this algorithm is $\frac{N}{2}\left(1-\frac{1}{K^2}\right)$. So, this algorithm saves a little for the seemingly easier problem. It can be verified that no classical randomized algorithm can do better (see Appendix A for a proof). So, the savings necessarily reduce very fast as K grows. If we restrict ourselves to deterministic algorithms, then using the same idea we can derive an algorithm that makes only $N\left(1-\frac{1}{K}\right)$ queries, a saving of $\frac{N}{K}$ queries over any algorithm that solves the original database problem with certainty.

1.2 Why partial quantum search may be easier

In this paper we study this problem in the quantum setting. Again, we restrict ourselves to algorithms that make no error. It is well known that if the database is supplied in the form of a suitable quantum oracle, then it is possible to use *quantum parallelism* and determine the rank of any element using $\left(\frac{\pi}{4}\right)\sqrt{N}$ queries [4]. Furthermore, this algorithm is optimal [8] (also [1,2]). The ideas used to speed up partial search classically can be used to reduce the number of queries by a factor of $\sqrt{\frac{K-1}{K}} \approx 1-\frac{1}{2K}$ over the standard quantum search algorithm. That is, we randomly pick $K-1$ of the blocks and run the quantum search algorithm on the $N\left(1-\frac{1}{K}\right)$ locations in the chosen blocks. This would require

$$\frac{\pi}{4}\sqrt{\frac{(K-1)N}{K}} \;\approx\; \frac{\pi}{4}\left(1-\frac{1}{2K}\right)\sqrt{N}$$

queries. As in the deterministic classical search algorithm, the savings in the number of queries are $O(\frac{1}{K})$ times the number of queries needed for exact search. In the classical case, that is the best that we could do. In this paper, we show a better quantum algorithm that saves $\theta(\frac{1}{\sqrt{K}})$ times the number number of queries needed for exact search, and argue that up to constant factors this the best one can hope for. Our main result is the following.

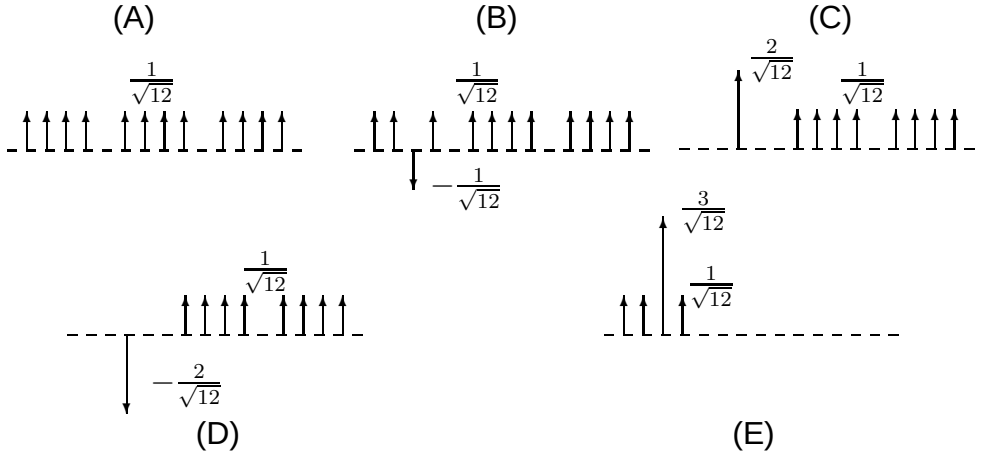
Theorem 1. *Assume that N is much larger than K.*

Upper bound: *For all K there is a constant $c_k > 0$, such that there is a quantum algorithm for partial search that makes at most $(\frac{\pi}{4})(1 - c_K)\sqrt{N}$ queries to the database and returns the answer with probability at least $1 - O(N^{-\frac{1}{2}})$; this algorithm can be modified to return the correct answer with certainty while increasing the number of queries by at most a constant. For large K, we have $c_K \geq \frac{0.42}{\sqrt{K}}$.*

Lower bound: *Fix K. Suppose there is a quantum algorithm for partial search that makes at most $\frac{\pi}{4}(1 - d_k)\sqrt{N}$ queries (for all large N) and returns the correct answer with probability at least $1 - O(N^{-\frac{1}{4}})$, then $d_K \leq \frac{1}{\sqrt{K}}$.*

The upper bound is obtained by judicious combinations of amplitude amplification steps used for the standard quantum database search problem. The lower bound is obtained by observing that an algorithm for the regular database search problem can be obtained by using the partial search algorithm as a subroutine and using lower bounds for the regular database search (for the zero-error case we can use Zalka [8]'s result, which we refine in order to derive our lower bounds for partial search algorithms with small error).

1.3 An example



(A) Start with the uniform superposition of the twelve states.
(B) Invert the amplitude of the target state.
(C) Invert about the average in each of the three blocks.
(D) Invert the amplitude of the target state again.
(E) Invert about the global average.

Fig. 1. Partial quantum search in a database of twelve items

Consider a list with twelve items. It is easily checked that to find the target with certainty, we would need at least three (quantum) queries. Suppose we just want to know whether the given element appears

in the first one-third, the second one-third or the last one-third. Can we now manage with fewer queries? Consider the algorithm shown in Figure 1. At the end, all the amplitude is concentrated on the target block. So, by performing a measurement, we can determine the target block with probability one, and, in fact, recover the target state itself with probability $\frac{3}{4}$. Note that this algorithm makes just two queries, for the transitions from (A) to (B) and (C) to (D).

2 Background and notation

Before we describe our algorithm, we review the framework for quantum search. We assume that the reader is familiar with the basics of quantum circuits, especially the quantum database search algorithm of Grover [4] (see, for example, Nielsen and Chuang [7, Chapter 6]).

2.1 Database search

The database is modeled as a function $f : [N] \to S$, where $[N] \stackrel{\Delta}{=} \{0, 1, 2, \ldots, N-1\}$ is the set of addresses and S is the set of possible items to be stored in the database. For our purposes, we can take S to be $\{0, 1\}$. When thinking of bits we identify $[N]$ with $\{0, 1\}^n$. We will assume that there is a unique location x such that $f(x) = 1$. This assumption is natural and widely used when modeling the search problem for unsorted databases with unique keys. We will call this the target location and denote it by t. In the quantum model, this database is provided to us by means of an oracle unitary transformation T_f which acts on an $(n+1)$-qubit space by sending the basis vector $|x\rangle|b\rangle$ to $|x\rangle|b \oplus f(x)\rangle$.

In the version of the quantum search algorithm useful for us (see Nielsen and Chuang [7, Chapter 6]), one starts with n-qubits in the uniform superposition

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle.$$

Then, one repeatedly applies the following operation in order to amplify the probability of the target state:

$$\mathsf{A} \stackrel{\Delta}{=} I_0 I_t.$$

Here, the unitary transformation $I_t = I - 2|t\rangle\langle t|$ selectively inverts the amplitude of the target state, and $I_0 = 2|\psi_0\rangle\langle\psi_0| - I$ selectively inverts the amplitude of the component orthogonal to $|\psi_0\rangle$. One can implement I_t using a quantum circuit that makes one query to the oracle. The state vector always stays in the two dimensional space spanned by $|\psi_0\rangle$ and $|t\rangle$. Each application of the transformation A moves the state vector towards the target state by an angle of about $\frac{2}{\sqrt{N}}$. Thus, after $(\frac{\pi}{4})\sqrt{N}$ iterations the state vector reaches (very close to) the target state. One curious feature of this algorithm is that further applications of the transformation move the state vector away from $|t\rangle$ and $|\psi_0\rangle$ so that very soon its component along the $|\psi_0\rangle$ becomes negative. Thus, one has to choose the number of iteration carefully in order to optimize the probability of success. *Interestingly, this drift away from the target state, which is usually considered a nuisance, is crucial for our general partial search algorithm. In fact, we made use of it in the example above.*

Quantum algorithms that make no error: Grover's database search algorithm [4] finds the location where an element is stored using approximately $\left(\frac{\pi}{4}\right)\sqrt{N}$ queries to the oracle. The original version of the algorithm could return a wrong address with probability $O(1/N)$. (Recall that we assume that the element is stored at a unique location in the database.) However, it is possible to modify the algorithm so

that the correct answer is returned with certainty (for example, one can modify the last iteration slightly so that the state vector does not overshoot its target see also [3,5,6,9]). In the next section, we present an algorithm for partial search that returns the correct answer with high probability. It is easy to see that this algorithm can be modified to ensure that the correct answer is returned with probability one. It is easy to reduce the usual database problem to the partial search problem considered in this paper and conclude from Zalka's [8] lower bound of $\left(\frac{\pi}{4}\right)\sqrt{N}$ that the bound we obtain are essentially optimal for partial search algorithms that return the correct answer with certainty. A similar lower bound applies to partial search algorithm whose probability of error is very small say $O(N^{-\frac{1}{4}})$. For this we need to observe that a version of Zalka's [8] lower bound holds even when the quantum database search algorithm errs with some very small probability. A referee of an earlier version of this paper pointed out that this fact is not explicitly stated in Zalka's paper; so, we provide a detailed derivation in Appendix B. In Section 4, we show that the savings achieved by our algorithm are essentially optimal for algorithms that err with very small probability.

2.2 The partial search problem

In the partial search problem, the set of address, R, is partitioned into K equal blocks of size N/K each. We will then think of the addresses in R as pairs (y, z) where $y \in [K] \stackrel{\Delta}{=} \{1, 2, \ldots, K\}$ and $z \in [N/K]$. For example, when $R = \{0,1\}^n$ and $K = 2^k$, each such block could correspond to those addresses that have the same first k bits; then, y gives the first k bits of the address and z the remaining $n-k$ bits. We refer to the block containing the target $t = y_t z_t$ as the *target* block and the other blocks as *non-target* blocks. In our algorithm we will apply the database search iteration to the different blocks in parallel. This is implemented by means of an operator $\mathsf{A}_{[N/K]}$ which acts as follows. First, it applies the the operator I_t defined above, to invert the amplitude of target state. Then, it performs an inversion about the average in each block in parallel. Formally, let $|\psi_{0,[N/K]}\rangle \stackrel{\Delta}{=} \frac{1}{\sqrt{N/K}} \sum_{z\in[N/K]} |z\rangle$ and $I_{0,[N/K]} = 2|\psi_{0,[N/K]}\rangle\langle\psi_{0,[N/K]}| - I_{[N/K]}$ (where $I_{[N/K]}$ is the identity operator on the space of dimension N/K). Then,

$$\mathsf{A}_{[N/K]} = (I_{[K]} \otimes I_{0,[N/K]}) I_t.$$

Note this operator acts on the space of dimension N. With this notation, we are ready to describe our algorithm and the lower bound in detail.

3 Partial search is easier ...

In this section, we present an algorithm that uses fewer queries to the database than the algorithm presented in the introduction. The goal is to use a suitable combination of operators A and $\mathsf{A}_{[N/K]}$ so that, in the end, only the basis states in the target block have positive amplitude. Clearly, after this a measurement of the state vector with respect to the standard basis will give us the information we want.

The algorithm has three steps, of which the first two involve amplitude amplification. In the first step, we use the operator A on the entire address space (of dimension N). However, we stop short of finding the target exactly, making $\theta\left(\sqrt{\frac{N}{K}}\right)$ fewer queries than we would for a complete search. In the second step, we use the operator $\mathsf{A}_{[N/K]}$ which does amplification in parallel in each block. The amplitudes of the states in the non-target blocks are not affected in this step; the states in the target block, however, acquire negative amplitudes. The number of iterations is chosen so that the overall average amplitude is exactly half of the amplitude of each state in non-target blocks. In particular, if we perform an inversion

Step 1: Prepare the address register in the state

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x \in [N]} |x\rangle.$$

Perform $\ell_1(\epsilon) = (\frac{\pi}{4})(1-\epsilon)\sqrt{N}$ iterations of the standard amplification step. The resulting state (see Figure fig:step1) is

$$|\psi_1\rangle \stackrel{\Delta}{=} \mathsf{A}^{\ell_1}|\psi_0\rangle.$$

Step 2: We perform $\ell_2(\epsilon)$ iterations of the $\mathsf{A}_{[N/K]}$, so that the average amplitude of all the non-target states is exactly half the amplitude of every state in every non-target block.

Step 3: There are two operations performed in this step. First, we move the target state out. That is, we take an ancilla qubit b (initially in state $|0\rangle$) and perform the operation M:

$$\begin{aligned} \text{for the target state } t\text{:} \quad & |b\rangle|t\rangle \mapsto |b \oplus 1\rangle|t\rangle; \\ \text{for other basis states } |x\rangle\text{:} \quad & |b\rangle|x\rangle \mapsto |b\rangle|x\rangle. \end{aligned}$$

Controlled on $b = 0$, we perform an inversion about the average. All states in the non-target blocks now have amplitude zero!

Fig. 2. The algorithm for partial search

about the average for all non-target states, the amplitude of the states in the non-target blocks will become zero. This inversion about the average for non-target states is implemented in the third step; it requires just one query to the database. It will turn out that the savings in the first step are significantly more than the number of queries needed to implement the second step, so that in the end we are still left with a savings of $\theta\left(\sqrt{\frac{N}{K}}\right)$ queries.

The algorithm in Figure 3 uses a parameter ϵ that controls the number of iterations. Later, in order to minimize the number of queries we will need to choose ϵ optimally. On first reading, it might be helpful to assume that K is a large constant and $\epsilon = \frac{1}{\sqrt{K}}$.

3.1 Estimating the number of queries

Remark about approximations: In the following we will assume that N is large. To keep the presentation simple, we will often make approximate calculations. When we say LHS $\sim$ RHS, we mean that the two sides differ by a quantity that goes to 0 as N becomes large, say like $O(\frac{1}{\sqrt{N}})$. For example, we sometimes pretend that the target state is orthogonal to the state $|\psi_0\rangle$. Our algorithm will produce the correct answer with probability very close to 1; the probability of error is $O\left(\frac{1}{\sqrt{N}}\right)$. However, as remarked above, one can modify the algorithm so that it returns the correct answer with certainty.

Analysis of Step 1: We will write $x = yz$ with $y \in [K]$ and $z \in [N/K]$, as described above, and express $|\psi_1\rangle$ as

$$\sum_{y \in [K]} \alpha_y |y\rangle|\psi_y\rangle,$$

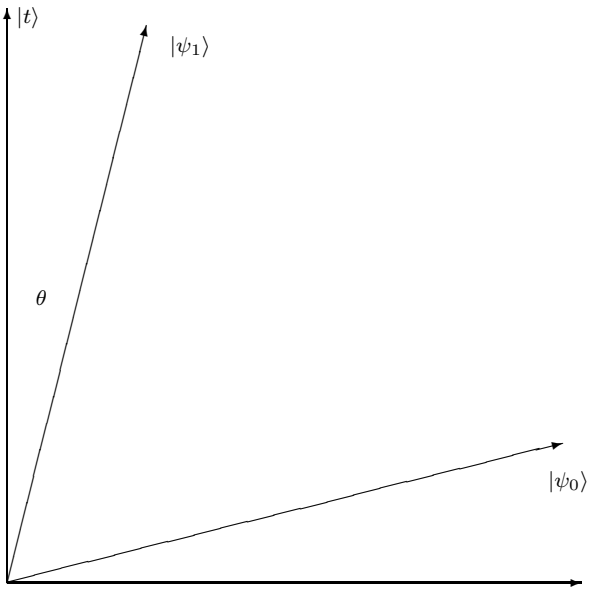


Fig. 3. Step 1 moves the state vector close to the target state by performing $\frac{\pi}{4}(1-\epsilon)\sqrt{N}$ iterations of the standard quantum search

where the α_y's are complex amplitudes whose squares sum to 1, and $|\psi_y\rangle$ is a state of the form

$$\sum_{z\in[N/K]} \beta_{yz}|z\rangle \text{ where } \sum_{z\in[N/K]} |\beta_{yz}|^2 = 1.$$

Since,

$$|\psi_1\rangle = \cos(\theta)|t\rangle + \frac{\sin(\theta)}{\sqrt{N}} \sum_{x:x\neq t} |x\rangle,$$

we get the following information about α_y and $|\psi_y\rangle$. (See also the first histogram of Figure 5). Recall that we write $t = y_t z_t$, where y_t is the address of the target block and z_t is the address inside the target block.

In a non-target block: Suppose $y \neq y_t$. Then, the projection of the current state vector $|\psi_1\rangle$ on the subspace corresponding the block of y, is a uniform superposition of the basis vectors but with length $\frac{1}{\sqrt{K}}\sin(\theta)$. That is,

$$\begin{aligned} \alpha_y &\sim \frac{1}{\sqrt{K}}\sin(\theta); \\ \text{and } |\psi_y\rangle &= \frac{1}{\sqrt{N/K}} \sum_{z\in[N/K]} |z\rangle. \end{aligned} \tag{1}$$

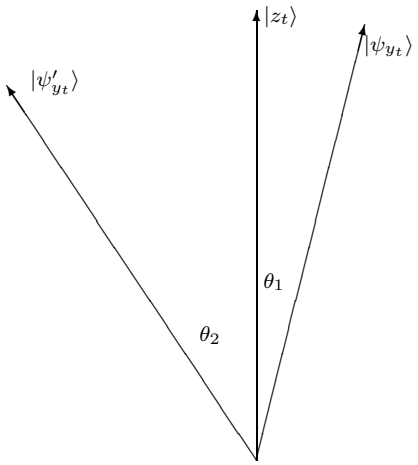


Fig. 4. Step 2 consists of independent quantum searches for each block; in the target block the state vector moves past the target

In the target block: For the target block, we have

$$\alpha_{y_t} \sim \sqrt{1 - \left(\frac{K-1}{K}\right)\sin^2(\theta)}\,;$$
$$\text{and } |\psi_{y_t}\rangle \sim \frac{\cos(\theta)}{\alpha_{y_t}}|z_t\rangle + \frac{\sin(\theta)}{\alpha_{y_t}\sqrt{K}}\frac{1}{\sqrt{N/K}}\sum_{z\in[N/K]}|z\rangle. \tag{2}$$

The first term corresponds to the component along the target state; the second corresponds to the uniform superposition on all states in the target block.

Analysis of Step 2: In this step, we work on each block separately but in parallel by applying the operator $A_{[N/K]}$ described above. This operator has the following effect. The amplitudes of basis states in non-target blocks remain the same (i.e. $\frac{\sin(\theta)}{\sqrt{N}}$). However, the non-target basis states in the target block transfer their amplitude to the target state and then acquire negative amplitudes. For $y \neq y_t$, α_y and $|\psi_y\rangle$ do not change. However, the projection on the subspace corresponding to the target block moves from $|\psi_{y_t}\rangle$ to ψ'_{y_t} (see Figure 4). Thus, at the end of Step 2, the overall state is

$$|\psi_2\rangle = \alpha_{y_t}|y_t\rangle|\psi'_{y_t}\rangle + \sum_{y:y\neq y_t}\alpha_y|y\rangle|\psi_y\rangle.$$

We choose ℓ_2 so that the component of the vector $|\psi'_{y_t}\rangle$ along $\frac{1}{\sqrt{N/K}}\sum_{z\in[N/K]}|z\rangle$ is an appropriate negative quantity. The histogram in Figure 5 shows the amplitudes of all the basis states before and after Step 2. The dotted line in the histogram is the average of all the non-target states and is arranged to be half the amplitude of the states in the non-target blocks.

We are now ready to determine ℓ_2. We work in the N/K dimensional space corresponding to the target block. Let θ_1 be the initial angle between $|\psi_{y_t}\rangle$ and $|z_t\rangle$. Then, by (2), $\sin(\theta_1) = \frac{1}{\alpha_{y_t}\sqrt{K}}\sin(\theta)$,

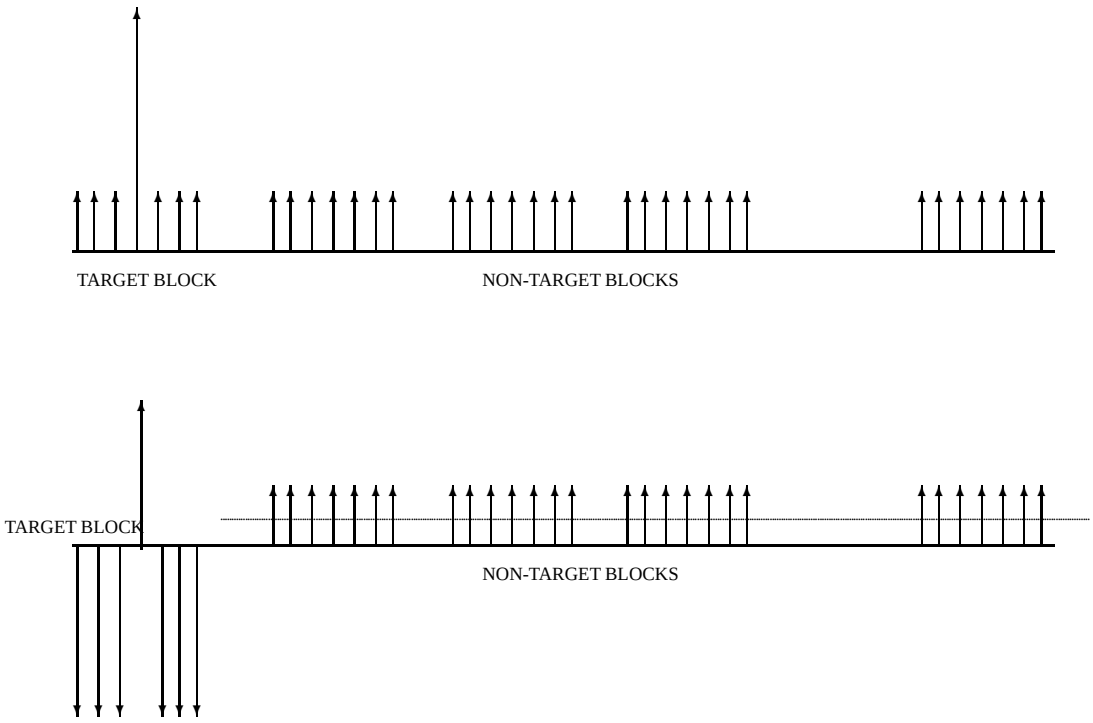


Fig. 5. After Step 2, the non-target states in the target block acquire negative amplitudes

that is,

$$\theta_1 = \arcsin\left(\frac{1}{\alpha_{y_t}\sqrt{K}}\sin\theta\right). \tag{3}$$

Let θ_2 be the angle between the final state $|\psi'_{y_t}\rangle$ and $|z_t\rangle$ that we want to achieve in Step 2. To determine θ_2 we need to do a small calculation. Let X be the sum of the amplitudes of all the states in non-target blocks. Then, we have from (1) that

$$X \sim \frac{(K-1)\sqrt{N}}{K}\sin\theta.$$

Let Y denote the sum of the amplitudes of all non-target states in the target block at the end of Step 2. We want the overall average amplitude per block (that is, $(X+Y)/K$) to be half the average in the non-target blocks (that is, $X/(2(K-1))$); therefore,

$$Y \sim -\frac{K-2}{2(K-1)}X \sim -\frac{(K-2)\sqrt{N}}{2K}\sin\theta.$$

On the other hand, $Y = -\alpha_{y_t}\sin(\theta_2)\sqrt{N/K}$. Thus,

$$\theta_2 \sim \arcsin\left(\frac{1}{2\alpha_{y_t}}\frac{K-2}{\sqrt{K}}\sin\theta\right). \tag{4}$$

The vector $|\psi_{y_t}\rangle$ needs to traverse a total angle of $\theta_1+\theta_2$. In one iteration of $A_{[N/K]}$ it covers $\frac{2}{\sqrt{N/K}}$. We thus have an estimate for ℓ_2:

$$\ell_2(\epsilon) = \frac{\sqrt{N/K}}{2}(\theta_1+\theta_2),$$

where (3) and (4) give us the values of θ_1 and θ_2 in terms of ϵ. We choose ϵ so that the total number of queries, namely,

$$(\ell_1(\epsilon)+\ell_2(\epsilon))\sqrt{N}+1 \sim \left(\frac{\pi}{4}\right)(1-\epsilon)\sqrt{N} + \frac{1}{2\sqrt{K}}\left[\arcsin\left(\frac{1}{\alpha_{y_t}\sqrt{K}}\sin\theta\right) + \arcsin\left(\frac{1}{2\alpha_{y_t}}\frac{K-2}{\sqrt{K}}\sin\theta\right)\right]\sqrt{N}+1$$

is as small as possible.

Note that with $\epsilon = 0$, the algorithm reduces to the usual database search algorithm and takes $(\frac{\pi}{4})\sqrt{N})$ queries. Also, for $K \geq 2$, the derivative (w.r.t. ϵ) of

$$\ell_1(\epsilon)+\ell_2(\epsilon),$$

is negative at $\epsilon = 0$. It follows that with the optimum choice of ϵ, the running time is bounded by $\frac{\pi}{4}(1-c_k)\sqrt{N}$ for some constant $c_k > 0$.

We have not been able to obtain general expression for the optimum number of queries by an optimum choice of ϵ. For some small values of K, the following table lists the optimum values obtained by using a computer program.

	Upper bound	Lower bound
Database search	$0.785\sqrt{N}$	$0.785\sqrt{N}$
K=2 (first bit)	$0.555\sqrt{N}$	$0.23\sqrt{N}$
K=3 (first trit)	$0.592\sqrt{N}$	$0.332\sqrt{N}$
K=4 (first two bits)	$0.615\sqrt{N}$	$0.393\sqrt{N}$
K=5	$0.633\sqrt{N}$	$0.434\sqrt{N}$
K=8 (first three bits)	$0.664\sqrt{N}$	$0.508\sqrt{N}$
K=32 (first five bits)	$0.725\sqrt{N}$	$0.647\sqrt{N}$

For large values K, one can obtain some estimates on the behavior of c_k. For example, assume that K is large (and N even larger). Take $\epsilon = \frac{1}{\sqrt{K}}$. Then, we approximate $\sin\theta$ by $\theta = \frac{\pi}{2\sqrt{K}}$ and bound the number of queries by

$$\frac{\pi}{4}\left[1-\left(1-\frac{2}{\pi}\arcsin\frac{\pi}{4}\right)\frac{1}{\sqrt{K}}+O\left(\frac{1}{K}\right)\right]\sqrt{N}+1 \leq \frac{\pi}{4}\left(1-\frac{0.42}{\sqrt{K}}\right)\sqrt{N}.$$

4 ...but not much easier

Theorem 2. *Fix K. Suppose for all large N, there is an algorithm for the partial search problem that uses at most $\alpha_K\sqrt{N}$ queries and return the correct answer with probability of error $O(N^{-\frac{1}{4}})$. Then,*

$$\alpha_K \geq \left(\frac{\pi}{4}\right)\left(1-\frac{1}{\sqrt{K}}\right).$$

Proof. We reduce the database search problem to the partial search problem. First, we derive the lower bound assuming that the algorithms for the partial search problem return the correct answer with certainty. We start by applying the algorithm for partial search for databases of size N. This yields the first $\log K$ bits of the target state. Next, we restrict ourselves to those addresses x that have the correct first k bits and determine the next k bits of the universe by using the partial search algorithm for databases of size N/K. Continuing in this way, we converge on the target state after making a total of at most

$$\alpha\left(1+\frac{1}{\sqrt{K}}+\frac{1}{K}+\frac{1}{K\sqrt{K}}+\cdots\right) \leq \alpha_k\left(\frac{\sqrt{K}}{\sqrt{K}-1}\right)\sqrt{N}$$

queries to the database. Thus, using Zalka's lower bound [8], we have

$$\alpha_k\left(\frac{\sqrt{K}}{\sqrt{K}-1}\right)\sqrt{N} \geq \left(\frac{\pi}{4}\right)\sqrt{N},$$

implying that

$$\alpha_k \geq \left(\frac{\pi}{4}\right)\left(1-\frac{1}{\sqrt{K}}\right).$$

The reduction for algorithms with errors is similar. We apply the above method until the problem size becomes very small (say less than $N^{\frac{1}{3}}$). At that point we solve the database search problem by brute force. Since, we invoked the partial search problem only $O(\log N)$ times, and each such invocation had probability of error at most $N^{-\frac{1}{12}}$, the probability that all these invocations gave the correct answer is $1-O(N^{-\frac{1}{12}}\log N)$. This gives a database search algorithm that gives the correct answer with probability $1-O(N^{-\frac{1}{12}}\log N)$. Now, Theorem 3 in the appendix implies that such an algorithm must make $\frac{\pi}{4}\sqrt{N}(1-o(1))$ queries to the oracle. A calculation very similar to the one above gives the claimed lower bound for α_K. □

5 Summary

In yet another variant of the quantum search problem we ask whether it is any easier to partially search a database than it is to completely search it. This question is significant, because the search algorithm of [4] has been shown in [8] to be precisely optimal for the complete search problem. The short answer is that partial search is slightly easier. We derive a lower bound indicating how much easier it is and an algorithm that closely matches this lower bound. In the appendix, we present a lower bound on the expected number of queries made by zero-error randomized algorithms for partial search; we also derive an explicit form of Zalka's lower bound for quantum search algorithms that make very small error.

Acknowledgments

We thank the referees of on an earlier version of this paper for their detailed comments.

References

1. C.H. Bennett, E. Bernstein, G. Brassard and U. Vazirani: Strengths and Weaknesses of Quantum Computing, SIAM J. Computing, 26:5 (1997) , 1510–1523. Also quant-ph/9701001.
2. M. Boyer, G. Brassard, P. Hoyer, and A. Tapp: Tight bounds on quantum searching, Fortsch.Phys. 46 (1998) 493-506. Also quant-ph/9605034.

3. G. Brassard, P. Hoyer, M. Mosca, A. Tapp: Quantum Amplitude Amplification and Estimation. quant-ph/0005055.
4. L.K. Grover: A fast quantum mechanical algorithm for database search. STOC (1996): 212-219.
5. C-R. Hu: A family of sure-success quantum algorithms for solving a generalized Grover search problem. quant-ph/0201049.
6. G.L. Long: Grover Algorithm with zero theoretical failure rate. Phys. Rev. A 64 (2001) 022307.
7. M.A. Nielsen and I.L. Chuang: Quantum Computation and Quantum Information. Cambridge University Press (2000).
8. C. Zalka: Grover's quantum searching algorithm is optimal. Phys. Rev. A 60 (1999) 2746-2751.
9. C. Zalka: A Grover-based quantum search of optimal order for an unknown number of marked elements. quant-ph/9902049.

A Randomized complexity of partial search

We are considering algorithms that make no error. To show a lower bound of t queries on the expected number of probes made by a randomized algorithm on a worst-case input, it is enough to describe a distribution on inputs for which every deterministic algorithm makes at least t queries on the average.

Let $\ell_1, \ell_2, \ldots,$ be the sequence of location this deterministic algorithm probes if $f(x) = 0$, for all x. Now, consider the uniform distribution on inputs obtained by choosing a random location $t \in [N]$ and setting $f(x) = 1$ if and only $x = t$. Let $\mathcal{E}$ denote the event that t is one of $\ell_1, \ell_2, \ldots, \ell_{N-N/K}$. Then, the probability of $\mathcal{E}$ is exactly $1 - 1/K$ and conditioned on the event $\mathcal{E}$, the expected number of probes made by the deterministic algorithms is exactly $\frac{N}{2}\left(1 - \frac{1}{K}\right)$. If $\mathcal{E}$ does not hold then, the algorithm makes at least $N\left(1 - \frac{1}{K}\right)$ probes (because it is not allowed any errors). Thus, the overall average number of probes made by this algorithm is at least

$$\left(1 - \frac{1}{K}\right) \times \frac{N}{2}\left(1 - \frac{1}{K}\right) + \frac{1}{K} \times N\left(1 - \frac{1}{K}\right) = \frac{N}{2}\left(1 - \frac{1}{K^2}\right).$$

B Zalka's bound revisited

In this section we give a detailed proof of the following theorem.

Theorem 3. *Suppose a quantum database search algorithm makes T queries to a database of size $N \geq 100$ and returns the correct answer with probability at least $1 - \epsilon \geq 0.9$. Then,*

$$T \geq \frac{\pi}{4}\left(1 - O\left(\sqrt{\epsilon} + N^{-\frac{1}{4}}\right)\right).$$

To justify this, we will make the arguments in Zalka's paper [8] more explicit and rigorous. Let O_y be the oracle corresponding to the database where $y \in [N]$ is the only marked element. For $t = 0, 1, \ldots, T$, let

$|\phi_t\rangle$ be the state of the registers just before the $(t+1)$-st query is made assuming that all instances of the oracle are replaced by the identity transformation;

$|\phi_t^y\rangle$ be the state of the registers just before the $(t+1)$-st query is made to the oracle, assuming that the oracle is O_y.

Thus, $|\phi_0\rangle = |\phi_0^y\rangle$ is the initial state and $|\phi_T\rangle$ and $|\phi_T^y\rangle$ are the states reached at the end. Like several other lower bound proofs, our proof will also consider the variation in the state caused by each query to the oracle. To state this, we will denote by $|\phi_t^{y,i}\rangle$ the state of the registers just before the $(t+1)$-st query is made assuming that the first $T - i$ queries are made to the identity oracle and the remaining i queries are made to the oracle O_y. Thus, $|\phi_t^{y,0}\rangle = |\phi_t\rangle$ and $|\phi_t^{y,T}\rangle = |\phi_t^y\rangle$.

It will be convenient to have only real valued amplitudes appearing in our state vectors when they are expressed in the computational basis. The standard way to arrange this is by adding an extra qubit

register and associating the component $(a + ib)|x\rangle$ appearing in the original space with the vector $a|x\rangle|0\rangle + b|x\rangle|1\rangle$. We will assume that the first register (whose state is a superposition of values in $[N]$, when we identify $[N]$ with $\{0,1\}^n$, this register will consist of n qubits) contains the query made to the oracle and also that this *address* register is measured at the end to recover the answer. Let V be the Hilbert space associated with the remaining registers, and let P_y be the orthogonal projection operator for the subspace $|y\rangle \otimes V$. Note that $\|P_y|\phi\rangle\|^2$ is the probability that we observe y when we measure the first register in the state $|\phi\rangle$.

As in Zalka's argument, we will use angles to measure the distance between (real) unit vectors. Let $\theta(\phi, \phi') = \arccos |\langle\phi|\phi'\rangle|$. Note that this distance is always in the range $[0, \frac{\pi}{2}]$, and that the triangle inequality holds for it. Our proof rests on three lemmas.

Lemma 1. $\displaystyle\sum_{y\in[N]} \theta(\phi_T, \phi_T^y) \geq \frac{\pi}{2}\left(1 - O\left(\sqrt{\epsilon} + N^{-\frac{1}{4}}\right)\right).$

Lemma 2. *For $i = 1, 2, \ldots, T$, $\theta(\phi_T^{y,i-1}, \phi_T^{y,i}) \leq 2 \arcsin \sqrt{p_{T-i,y}}$, where $p_{T-i,y} = \|P_y|\phi_{T-i}\rangle\|^2$ is the probability that the first address register of $|\phi_{T-i}\rangle$ contain the value y.*

Let us assume that these lemmas hold and proceed with the proof of our theorem. Then,

$$\begin{aligned}
\sum_{y\in[N]}\sum_{i=0}^{T-1} 2\arcsin\sqrt{p_{i,y}} &\geq \sum_{y\in[N]}\sum_{i=0}^{T-1} \theta(\phi_T^{y,i}, \phi_T^{y,i+1}) \qquad \text{(by Lemma 2)} \\
&\geq \sum_{y\in[N]} \theta(\phi_T^{y,0}, \phi_T^{y,T}) \qquad \text{(triangle inequality)} \\
&\geq \sum_{y\in[N]} \theta(\phi_T, \phi_T^y) \\
&\geq N \cdot \frac{\pi}{2}\left(1 - O\left(\sqrt{\epsilon} + N^{-\frac{1}{4}}\right)\right). \qquad \text{(by Lemma 1)}
\end{aligned}$$

On changing the order of summation, we obtain

$$\sum_{i=0}^{T-1}\sum_{y\in[N]} 2\arcsin\sqrt{p_{i,y}} \geq N \cdot \frac{\pi}{2}\left(1 - O\left(\sqrt{\epsilon} + N^{-\frac{1}{4}}\right)\right). \tag{5}$$

Let us consider the inner sum for each value of i.

Lemma 3. *For $i = 0, 1, \ldots, T-1$,* $\displaystyle\sum_{y\in[N]} \arcsin\sqrt{p_{i,y}} = \sqrt{N}\left(1 + O\left(\frac{1}{N}\right)\right).$

Now, our theorem follows immediately by using this bound in (5). It remains to prove the lemmas.

B.1 Proofs of the lemmas

Proof of Lemma 1: Recall that $\theta(\phi_T, \phi_T^y) = \arccos|\langle\phi_T|\phi_T^y\rangle|$, and P_y is the orthogonal projection operator onto the subspace $|y\rangle \otimes V$. Since $|\phi_T\rangle$ is a unit vector, we have $\sum_y \|P_y|\phi_T\rangle\|^2 \leq 1$, implying

$$\frac{1}{N}\sum \|P_y|\phi_T\rangle\| \leq \left(\frac{1}{N}\sum_y P_y|\phi_T\rangle\|^2\right)^{1/2} \leq \frac{1}{\sqrt{N}}.$$

By Markov's inequality, for all but a fraction $N^{-\frac{1}{4}}$ of the y's in $[N]$,

$$\|P_y|\phi_T\rangle\| \leq N^{-\frac{1}{4}}. \tag{6}$$

On the other hand, because the algorithm errs with probability at most ϵ, we have for all y, $\|P_y|\phi_T^y\rangle\|^2 \geq 1-\epsilon$,, and since $\|P_y|\phi_T^y\rangle\|^2 + \|(I-P_y)|\phi_T^y\rangle\|^2 = 1$, we conclude that

$$\|(I-P_y)|\phi_T^y\rangle\| \leq \sqrt{\epsilon}. \tag{7}$$

By combining (6) and (7) we conclude that for all but an $N^{-\frac{1}{4}}$ fraction of the y's in $[N]$,

$$|\langle\phi_T|\phi_T^y\rangle| \leq |\langle\phi_T^y|P_y|\phi_T\rangle| + |\langle\phi_T|(I-P_y)|\phi_T^y\rangle| \leq \sqrt{\epsilon} + N^{-\frac{1}{4}}.$$

Let $\theta_y \stackrel{\Delta}{=} \arccos\langle\phi_T|\phi_T^y\rangle$. We claim that for all but an $N^{-\frac{1}{4}}$ fraction of the y's in $[N]$,

$$\theta_y \geq \frac{\pi}{2}\left(1 - O\left(\epsilon^{\frac{1}{2}} + N^{-\frac{1}{4}}\right)\right). \tag{8}$$

To justify this, note that since $\epsilon \leq \frac{1}{10}$ and $N \geq 100$, we have $\sqrt{\epsilon} + N^{-\frac{1}{4}} \leq \frac{1}{\sqrt{2}}$ (say), and $\theta_y \geq \frac{\pi}{4}$. For $\theta_y \leq \theta \leq \frac{\pi}{2}$, $\frac{\mathrm{d}\cos\theta}{\mathrm{d}\theta} = -\sin\theta \leq -\frac{1}{\sqrt{2}}$. So,

$$\left(\frac{\pi}{2} - \theta_y\right) \times -\frac{1}{\sqrt{2}} \geq \cos\frac{\pi}{2} - \cos\theta_y \geq -\left(\sqrt{\epsilon} + N^{-\frac{1}{4}}\right).$$

Our claim (8) follows from this.

Finally, since $\theta_y \geq 0$ for all y, we have

$$\sum_y \theta_y = \sqrt{N}\left(1 - N^{-\frac{1}{4}}\right)\frac{\pi}{2}\left(1 - O\left(\sqrt{\epsilon} + N^{-\frac{1}{4}}\right)\right) = \sqrt{N}\frac{\pi}{2}\left(1 - O\left(\sqrt{\epsilon} + N^{-\frac{1}{4}}\right)\right).$$

□

Proof of Lemma 2: It follows from elementary trigonometry that for unit vectors v and w,

$$\theta(v,w) = 2\arcsin(\frac{1}{2}\min\{\|v-w\|, \|v+w\|) \leq 2\arcsin(\frac{1}{2}\|v-w\|).$$

On the other hand, since the oracle O_y differs from the identity transformation only for basis vectors in the subspace $|y\rangle \otimes V$, one can show that (see, e.g., [8] for a similar derivation)

$$\|\phi_T^{y,i-1} - \phi_T^{y,i}\| \leq 2\|P_y|\phi_{T-i}\rangle\|.$$

Our lemma follows from these two observations. □

Proof of Lemma 3: We will show that $\sum_y \arcsin\sqrt{p_{i,y}}$ achieves its maximum when each $p_{i,y}$ is $\frac{1}{N}$. For, let $\sum_y \arcsin\sqrt{p_{i,y}}$ take its maximum value at $(p_{i,y} : y \in [N])$. Now, the function $\arcsin\sqrt{x}$ is concave in the interval $[0, \frac{1}{2}]$ (for its second derivative is $-\frac{1}{4}(1-2x)(x-x^2)^{\frac{3}{2}}$). Thus, our claim will follow from Jensen's inequality if we can show that each $p_{i,y} \in [0, \frac{1}{2}]$. To show this, suppose for contradiction that for some (necessarily unique) $\hat{y}$, we have $p_{i,\hat{y}} > \frac{1}{2}$. Then,

$$\sum_y \arcsin\sqrt{p_{i,y}} \leq (N-1)\arcsin\sqrt{\frac{1-p_{i,\hat{y}}}{N-1}} + \arcsin\sqrt{p_{i,\hat{y}}} \leq (1+o(1))\sqrt{\frac{N-1}{2}} + \frac{\pi}{3}.$$

But for large N, we have $N \arcsin \frac{1}{\sqrt{N}} \geq \sqrt{N}$ is larger than this quantity. This contradiction implies that our assumption about the existence of $\hat{y}$ was incorrect. It follows that

$$\sum_y \arcsin \sqrt{p_{i,y}} \leq N \arcsin \frac{1}{\sqrt{N}} \leq \sqrt{N} \left(1 + O\left(\frac{1}{N}\right)\right),$$

where for the first inequality we used the concavity of the function $\arcsin \sqrt{x}$ for $x \in [0, \frac{1}{2}]$ to apply Jensen's inequality. □