Fixed-Point Quantum Search with an Optimal Number of Queries

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1103/PhysRevLett.113.210501">http://dx.doi.org/10.1103/PhysRevLett.113.210501</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>American Physical Society</td>
</tr>
<tr>
<td>Version</td>
<td>Final published version</td>
</tr>
<tr>
<td>Accessed</td>
<td>Sun Apr 24 20:08:03 EDT 2016</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/91683">http://hdl.handle.net/1721.1/91683</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Article is made available in accordance with the publisher’s policy and may be subject to US copyright law. Please refer to the publisher’s site for terms of use.</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td></td>
</tr>
</tbody>
</table>
Fixed-Point Quantum Search with an Optimal Number of Queries

Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA
(Received 10 September 2014; published 18 November 2014)

Grover’s quantum search and its generalization, quantum amplitude amplification, provide a quadratic speedup over classical algorithms for solving a broad class of problems. Included are the many important, yet computationally prohibitive nondeterministic polynomial time (NP) problems [2], which can always be solved, albeit inefficiently, by searching the space of possible solutions. Because the problem Grover’s algorithm solves is so simple to understand—given an oracle function that recognizes marked items, locate one of \( M \) such marked items among \( N \) unsorted items—its classical time complexity \( \mathcal{O}(N/M) \) is obvious, making the quantum speedup that much more conclusive.

Conceptually also, Grover’s algorithm is compelling—the iterative application of the oracle and initial state preparation rotates from a superposition of mostly unmarked states to a superposition of mostly marked states in just \( \mathcal{O}(\sqrt{N/M}) \) steps [3]. This interpretation of Grover’s algorithm as a rotation is very natural because the Grover iterate is a unitary operator. However, this same unitarity is also a weakness. Without knowing exactly how many marked items there are, there is no knowing when to stop the iteration. This leads to the soufflé problem [4], in which iterating too little “undercooks” the state, leaving mostly unmarked states, and iterating too much “overcooks” the state, passing by the marked states and leaving us again with mostly unmarked states.

The most direct solution of the soufflé problem is to estimate \( M \) by either using full-blown quantum counting [5,6] or a trial-and-error scheme where iterates are applied an exponentially increasing number of times [5,7]. Although scaling quantum mechanically, these strategies are unappealing for search as they work best not by monotonically amplifying marked states, but rather by getting “close enough” before resorting to classical random sampling.

An alternative approach, in line with what we advocate here, is to construct, either recursively or dissipatively, operators that avoid overcooking by always amplifying marked states. Such algorithms are known as fixed-point searches. For example, running Grover’s \( \pi/3 \) algorithm [8] or the comparable ancilla algorithm [9] longer can only ever improve its success probability. Yet, a steep price is paid for this monotonicity—in both cases, the quadratic speedup of the original quantum search is lost.

This disappointing fact means that current fixed-point algorithms take time \( \mathcal{O}(N/M) \) for small \( M/N \), and their usefulness is relegated to large \( M/N \), where they conveniently avoid overcooking, but where classical algorithms are also already successful. Several results [10,11] improve the performance of fixed-point algorithms on wide ranges of \( M/N \); however, these algorithms are numerical, and as such, their time scaling cannot be assessed. Indeed, the \( \pi/3 \) algorithm was shown to be optimal in time [12], ostensibly proving it impossible to find a search algorithm that both avoids the soufflé problem and provides a quantum advantage.

Nevertheless, here we present a fixed-point search algorithm, which, amazingly achieves both goals—our search procedure cannot be overcooked and also achieves optimal time scaling, a quadratic advantage over classical unordered search. We sidestep the conditions of the impossibility proof by requiring not that the error monotonically improve as in the \( \pi/3 \) algorithm, but that the error become bounded by a tunable parameter \( \delta \) over an ever widening range of \( M/N \) as our algorithm is run longer. The polynomial method [13] is typically used to prove lower bounds on quantum query complexities; however, we instead use the fact that the success probability is a polynomial to adjust the phases of Grover’s reflection operators [14,15] and effect an optimal output polynomial with bounded error \( \delta \). In fact, our algorithm becomes the \( \pi/3 \) algorithm and Grover’s original search algorithm in the special cases of \( \delta = 0 \) and \( \delta = 1 \), respectively.

Our results apply just as cleanly, and more generally, to amplitude amplification [7], so we proceed in that
framework. We are given a unitary operator $A$ that prepares
the initial state $|s\rangle = A|0\rangle^{\otimes n}$. From $|s\rangle$, we would like to
extract the target state $|T\rangle$ with success probability $P_L \geq 1 - \delta^2$, where the overlap $\langle T | s \rangle = \sqrt{\lambda e^{i\delta}}$ is not zero and
$\delta \in [0, 1]$ is given. To do so, we are provided with the oracle $U$ which flips an ancilla qubit when fed the target state. That is, $U|T\rangle = |T\rangle|b_1\rangle$ and $U\langle T | b_\rangle = \langle T | b \rangle$ for $\langle T | T \rangle = 0$. Below, we show how to solve this problem and
extract $|T\rangle$ by performing on $|s\rangle$ a quantum circuit $S_{\lambda}$
consisting of $A, A^\dagger, U$, and efficiently implementable $n$-qubit gates, such that

$$P_L = |\langle T | S_{\lambda} | s \rangle|^2 = 1 - \delta^2 T_{L}(1/\delta) \sqrt{1 - \lambda}. \quad (1)$$

Here $T_{L}(x) = \cos[L \cos^{-1}(x)]$ is the $L^{\text{th}}$ Chebyshev
polynomial of the first kind [16], and $L - 1$ is the query complexity: the number of times $U$ is applied in the circuit $S_{\lambda}$. Furthermore, we construct $S_{\lambda}$ for any odd integer $L \geq 1$ and any $\delta$. Some examples of $P_L$ and a comparison to the
$\pi/3$ algorithm are shown in Fig. 1.

Assuming for now the existence of $S_{\lambda}$—its construction
is given later—we can already see that the success probability $P_L$ possesses both the fixed point property and optimal query complexity. First, note that as long as $|T_{1/L}(1/\delta)| \sqrt{1 - \lambda} \leq 1$, the fact that $|T_{L}(x)| \leq 1$ for $|x| \leq 1$ implies $P_L \geq 1 - \delta^2$. Therefore, for all $\lambda \geq w = 1 - T_{1/L}(1/\delta)^{-2}$, the probability $P_L$ meets our error tolerance. For large $L$ and small $\delta$, this width $w$ can be approximated as

$$w \approx \left(\frac{\log(2/\delta)}{L}\right)^2. \quad (2)$$

This equation demonstrates the fixed-point property—as $L$
increases, $w$ decreases, and we achieve success probability $P_L \geq 1 - \delta^2$ over an ever-increasing range of $\lambda$. Equivalently, this means we cannot overcook the state because if a sequence $S_{\lambda}$ achieves bounded error at $\lambda$, then so does $S_{\lambda'}$ for any $L' > L$. Second, note that to ensure the probability is bounded we must choose $L$ such that $w \leq \lambda$.

That is, for $\delta > 0$,

$$L \geq \frac{\log(2/\delta)}{\sqrt{\lambda}}. \quad (3)$$

Thus, query complexity goes as $L = \mathcal{O}(\log(2/\delta)(1/\sqrt{\lambda}))$
for our algorithm, achieving, for amplitude amplification, the best possible scaling in $\lambda$ [7]. See also Fig. 1 (inset).

Having seen two defining attributes, the fixed-point property and optimality, of the success probability from Eq. (1), let us now create it using the operators provided: the state preparation $A$ and oracle $U$. This problem simplifies when interpreted in the two-dimensional subspace $T$ spanned by $|s\rangle$ and $|T\rangle$ rather than in the full $2^n$-dimensional Hilbert space of all $n$ qubits. First, define $|t\rangle = e^{-i \delta/2}|T\rangle$ and $\bar{t} = (|s\rangle - \langle t | s \rangle |t\rangle) / \sqrt{1 - \lambda}$ so that

$$|s\rangle = \sqrt{1 - \lambda} \bar{t} + \sqrt{\lambda} |t\rangle = \left(\frac{1 - \lambda}{\sqrt{\lambda}}\right). \quad (4)$$

The matrix notation comes from the definitions $|t\rangle = (0_1^n)$ and $\bar{t} = (1_1^n)$. The location of $|s\rangle$ on the Bloch sphere is in the $X Z$ plane at an angle $\phi$ from the north pole, where $\phi \in [0, \pi]$ is defined by $\sin(\phi/2) = \sqrt{\lambda}$. Our goal of achieving the $P_L$ of Eq. (1) is equivalently expressed as
constructing, up to a global phase, a Chebyshev state

$$|C_L\rangle = \sqrt{1 - P_L} \bar{t} + \sqrt{P_L} e^{i \chi |t\rangle} = \left(\frac{1 - P_L}{\sqrt{P_L}}\right)^{1/\sqrt{\lambda}}. \quad (5)$$

for some relative phase $\chi$. For large enough $\lambda$ or large enough $L$, the Chebyshev states lie near the south pole of the Bloch sphere.

Similarly, Grover’s reflection operators can be interpreted as $\text{SU}(2)$ unitaries acting on $T$. As in previous work [14,15], we add arbitrary phases to the reflections to define generalized reflections. In Fig. 2 we show explicitly how to implement these generalized reflections using $A$, $U$, and

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{(color online). A comparison of search algorithms, plotting the overlap $P_L$ of the target state with the output state versus the overlap $\lambda$ of the target state with the initial state. We weigh our fixed-point (FP) algorithm (thick solid) against the $\pi/3$ algorithm (dashed) for the task of achieving output success
probability $P_L$ greater than $1 - \delta^2 = 0.9$ for all $\lambda > \lambda_0$. The query complexities of the algorithms vary based on $\lambda_0$ (dotted vertical lines). For $\lambda_0 = 0.25$ (blue), our algorithm makes 4 queries while the $\pi/3$ algorithm makes 8. For $\lambda_0 = 0.03$ (red), our algorithm makes 12 queries while the $\pi/3$ algorithm makes 80. For comparison, also shown is Grover’s non-fixed-point (NFP) search with 8 queries (thin black). The width and error for our 4-query algorithm are labeled $w$ and $\delta$, respectively. Inset: we plot the query complexity against $\lambda$ for our algorithm with $\delta^2 = 0.1$ (solid), the $\pi/3$ algorithm (dashed), and non-fixed-point Grover’s (dotted). While our FP algorithm and Grover’s NFP algorithm scale as $L \sim 1/\sqrt{\lambda}$, the $\pi/3$ algorithm scales as $L \sim 1/\lambda$.}
\end{figure}

210501-2
efficiently implementable n-qubit operations. With \( I \) the identity operator, their SU(2) representations are

\[
S_x(\alpha) = I - (1 - e^{-i\alpha})|s\rangle\langle s| = \\
\begin{pmatrix}
1 - (1 - e^{-i\alpha})\lambda & -\sqrt{1 - \lambda} \\
(1 - e^{-i\alpha})\sqrt{\lambda} & 1 - (1 - e^{-i\alpha})\lambda
\end{pmatrix},
\]

where \( \lambda = 1 - \lambda \). The product of the reflection operators is often called the Grover iterate \( G(\alpha, \beta) = -S_x(\alpha)S_z(\beta) \). The original Grover iterate [1] used \( \alpha = \pm \pi \) and \( \beta = \pm \pi \).

The generalized reflection operators are also expressible as rotations on the Bloch sphere. Defining \( R_{\phi}(\theta) = \exp \{-i\frac{\phi}{2}\cos(\theta)Z + \sin(\theta)X\} \) for Pauli operators \( X \) and \( Z \), we find

\[
S_x(\alpha) = e^{-i\alpha/2}R_{\phi}(\alpha),
\]

\[
S_z(\beta) = e^{i\beta/2}R_{\phi}(\beta).
\]

When \( \alpha = \pm \pi \) and \( \beta = \pm \pi \), these rotations map the \( XZ \) plane to the \( XZ \) plane, reproducing the O(1) rotation picture of Grover’s original non-fixed-point algorithm [3].

Yet, why limit ourselves to O(1) when, by using general phases \( \alpha \) and \( \beta \), we can access the whole of SU(2)? To that end, we consider a sequence of \( l \) generalized Grover iterates. Since each generalized Grover iterate contains two queries to \( U \), such a sequence would have query complexity \( L-1=2l \). We thus set out to find, for any \( \lambda > 0 \), phases \( \alpha_j \) and \( \beta_j \) such that the sequence

\[
S_L = G(\alpha_1, \beta_1) \ldots G(\alpha_L, \beta_L) = \prod_{j=1}^{l} G(\alpha_j, \beta_j)
\]

attains success probability \( P_L \) by preparing, up to a global phase, a Chebyshev state: \( |\langle C_L|S_L|s\rangle| = 1 \).

Indeed, such phases exist for all \( I \) and all \( \delta \in [0, 1] \), and moreover, they may be given in very simple analytical forms. For all \( j = 1, 2, \ldots, l \), we have

\[
\alpha_j = -\beta_{l-j+1} = 2\cot^{-1}\left(\frac{\tan(\pi j/L)}{\sqrt{1 - \gamma^2}}\right),
\]

where \( L = 2l+1 \) as before and \( \gamma^-1 = T_{1/L}(1/\delta) \). Notice Grover’s non-fixed-point search is subsumed by this solution—if \( \delta = 1 \), then \( \alpha_j = \pm \pi \) and \( \beta_j = \pm \pi \) for all \( j \), values that we saw above give Grover’s original non-fixed-point algorithm [1]. Thus, when \( \delta = 1 \), our algorithm is exactly Grover’s search.

The proof that Eq. (11) implies Eq. (1) begins by rearranging \( S_L \). Let \( A_z = \exp\{-i\frac{\phi}{2}\cos(\zeta)X + \sin(\zeta)Y\} \). With this definition, the state preparation operator is \( A = A_{\pi/2} \). Also note the identities \( R_{\phi}(\alpha) = A_{\pi/2}R_{\phi}(\alpha)A_{-\pi/2} \) and \( A_{\pi/2}R_{\phi}(\alpha)A_{-\pi/2} = R_{\phi}(\alpha)A_{\pi/2}R_{\phi}(\alpha) \). Then, using Eqs. (8) and (9), we find, up to a global phase, that

\[
S_L|s\rangle \sim R_0(\zeta_1)(A_{\zeta_1} \ldots A_{\zeta_L} A_{\zeta_1})R_0(-\zeta_1)|0\rangle.
\]

Here the phases \( \zeta_k = \zeta_{L-k+1} \) are palindromic, a consequence of the phase matching \( \alpha_j = -\beta_{l-j+1} \). With \( \alpha_j \) defined by Eq. (11), all \( \zeta_k \) can be found recursively using \( \zeta_{l+1} = (-1)^{l+1}\pi/2 \) and

\[
\zeta_{k+1} - \zeta_k = (-1)^{k}\pi - 2\cot^{-1}\left(\frac{\tan(k\pi/L)}{\sqrt{1 - \gamma^2}}\right)
\]

for all \( k = 1, \ldots, L-1 \).

From Eq. (12), we set up a recurrence relation to study the amplitude in states \( |t\rangle \) and \( |\bar{t}\rangle \) after each application of \( A_z \). That is, we let \( (a_0, b_0) = (1, 0) \), and for \( h = 1, \ldots, L \) define \( a_h \) and \( b_h \) by the matrix equation
\[
\left( \begin{array}{c} a_h \\ b_h \sin(\phi/2) \end{array} \right) = A_{\zeta_i} \left( \begin{array}{c} a_{h-1} \\ b_{h-1} \sin(\phi/2) \end{array} \right). \tag{14}
\]

Letting \( x = \cos(\phi/2) \), we can decouple this recurrence by defining
\[
b'_h = -x a_h - i \sqrt{1 - x^2} e^{-i \zeta_i} b_h.
\]
Rearranging Eq. (14), we find
\[
a_h = x(1 + e^{-i(\zeta_i - \zeta_{i-1})}) a_{h-1} - e^{-i(\zeta_i - \zeta_{i-1})} a_{h-2}. \tag{15}
\]

for \( h = 2, \ldots, L \) with initial values \( a_0 = 1 \) and \( a_1 = x \). This recurrence is strikingly similar to that defining the Chebyshev polynomials: \( T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \). Indeed, using Eq. (13), the Chebyshev recurrence is exactly recovered when \( \gamma = \delta = 1 \). For other values of \( \gamma \), the complex, degree-\( h \) polynomials \( a^{(\gamma)}_L(x) \) generalize the Chebyshev polynomials. In fact, it can be shown using combinatorial arguments analogous to those in [19] that
\[
a^{(\gamma)}_L(x) = T_L(x/\gamma)/T_L(1/\gamma). \quad \text{Since } T_L(1/\gamma) = 1/\delta \quad \text{and} \quad P_L = 1 - |a^{(\gamma)}_L(x)|^2, \quad \text{this completes the proof of Eq. (1)}.
\]

While the solutions in Eq. (11) are extremely simple to express, there are other solutions. Indeed, solutions of small-length \( l \) and large-width \( w \) can be combined to create solutions of larger length and smaller width through a process we call nesting. The general idea of nesting is that, within a sequence \( S_{L_1} \), the state preparation \( A \) can be replaced by another sequence \( S_{L_2} \) to recursively narrow the region of high failure probability. An intuition for this recursion can be noted in the similarity of Eqs. (4) and (5). Nesting is similar to concatenation in composite pulse sequence literature [20] and has already been employed in special cases of fixed-point search [8].

Although nesting would work to widen any fixed-point sequence (those found in [10,11], for instance), with our sequences using phases from Eq. (11), nesting neatly preserves the form of the success probability \( P_L \). For notational convenience, let us denote by \( S_{L_1}(B) \) a sequence of generalized Grover iterates as in Eq. (10) that uses \( BA \) in place of the state preparation operator \( A \). For instance, we have already shown that
\[
S_{L_1}(I)|s\rangle = \sqrt{1 - P_{L_1}(\lambda)|t\rangle} + \sqrt{P_{L_1}(\lambda)} e^{i\zeta_1}|t\rangle, \tag{16}
\]
where we have made explicit the dependence of \( P_L \) from Eq. (1) on \( \lambda \). By the same logic,
\[
S_{L_2}(S_{L_1}(I))S_{L_1}(I)|s\rangle = \sqrt{1 - P_{L_2}(P_{L_1}(\lambda))|t\rangle} + \sqrt{P_{L_2}(P_{L_1}(\lambda))} e^{i(x_1 + x_2)}|t\rangle. \tag{17}
\]

Consider \( P_{L_2}(P_{L_1}(\lambda)) \) and say that we choose the error bound for sequence 1 to be \( \delta_1 = (T_1/L_1)/|\lambda| \) and that for sequence 2 to be \( \delta_2 = \delta \). Using the semigroup property of the Chebyshev polynomials, \( T_p(T_q(x)) = T_{pq}(x) \), simple algebra yields
\[
P_{L_2}(P_{L_1}(\lambda, \delta_1, \delta_2)) = P_{L_1L_2}(\lambda, \delta), \tag{18}
\]
where we have further explicated the dependence of \( P_L \) from Eq. (1) on its error bound \( \delta \).

Therefore, as a result of nesting we can combine sequences of complexities \( L_1 \) and \( L_2 \) to obtain a sequence of complexity \( L_1L_2 \). In terms of Grover iterations, sequences with \( l_1 \) and \( l_2 \) iterations can be combined into one with \( l = l_1 + 2l_1l_2 + l_2 \) iterations. If the phase angles of the component sequences are denoted \( \alpha_j^{(1)} \) and \( \alpha_j^{(2)} \), then the nested sequence has phase angles
\[
\alpha_j^{(1,2)} = \begin{cases} 
\alpha_h^{(1)} & j \equiv h \ (\text{mod} \ L_1) \\
-a_k^{(1)} & j \equiv -h \ (\text{mod} \ L_1) \\
\alpha_k^{(2)} & j = kL_1,
\end{cases} \tag{19}
\]
for \( h \in \{1, 2, \ldots, l_1\} \) and \( k \in \{1, 2, \ldots, l_2\} \). The accompanying phase angles \( \beta_j^{(1,2)} \) can be taken to be phase matched, \( \beta_j^{(1,2)} = -\alpha_j^{(1,2)} \).

With nesting, we can see that the \( \pi/3 \) algorithm [8] is a special case of ours. From Eq. (11), note that our \( l = 1 \) sequence with \( \delta = 0 \) has phases \( -\alpha_1 = \beta_1 = \pi/3 \), and nesting it with itself gives exactly the \( \pi/3 \) algorithm. The query complexity argument represented by Eq. (3) breaks down when \( \delta = 0 \). In fact, the complexity of the \( \pi/3 \) algorithm scales classically as \( O(\frac{3}{\delta}) \) [8,9].

A strong argument for using nesting, even though explicit solutions at all lengths are available in Eq. (11), is that it lends our algorithm a nice property: adaptability. At the end of any sequence \( S_{L_1} \), we can choose to keep the result, a Chebyshev state \( |C_{L_1}\rangle \), or enhance it further to a Chebyshev state \( |C_{L_1L_2}\rangle \) for any odd \( L_2 \). So, conveniently, sequences can be extended without restarting the algorithm from the initial state \( |s\rangle \). This works because \( S_{L_1} \) is a prefix of the nested sequence in Eq. (17). This is not something the phases with the form in Eq. (11) allow as written, since they are prefix free.

Our fixed-point algorithm can be used as a subroutine in any scenario where amplitude amplification or Grover’s search is used [21], including quantum rejection sampling [22], optimum finding [23,24], and collision problems [25]. The obvious advantage of our approach over Grover’s original algorithm is that there is no need to hunt for the correct number of iterations as in [5], and this consequently eliminates the need to ever remake the initial state and restart the algorithm. Ideally, no measurements at all are required if \( \delta \) and \( L \) are chosen so the error of any amplitude amplification step will not significantly affect the error of the larger algorithm of which it is a part. Thus, our
fixed-point amplitude amplification could make such algorithms completely coherent.

An interesting direction for future work is relating quantum search to filters. In fact, the Dolph-Chebyshev function in Eq. (1) is one of many frequency filters studied in electronics [26]. For our purposes, the Dolph-Chebyshev function guarantees the maximum range of \( \lambda \) over which the bound \( P_L \geq 1 - \delta^2 \) can be satisfied by a polynomial of degree \( L \) [27]. Moreover, since the probability of success is guaranteed to be polynomial in \( \lambda \) and its degree is proportional to the number of queries made [13], we can also see this range is the maximum achievable with \( L - 1 \) queries.

Our algorithm is also easily modified to avoid the target state—simply using \( \alpha_j \) from Eq. (11), but with \( \beta_{l-j+1} = \alpha_j \) instead, will amplify the component of \( |s\rangle \) that lies perpendicular to \( |T\rangle \) so that \( |\langle T|S_L |s\rangle|^2 = P_L \). Using this insight, it is tempting for instance to consider “trapping” magic states [28] by repelling a slightly nonstabilizer state from all the stabilizer states nearby.

Similar to the \( \pi/3 \) algorithm [29], our sequences also have application to the correction of single qubit errors, as suggested by Eq. (12). For instance, if a perfect bit-flip \( X \) is desired, but only another nonidentity operation \( A \in \text{SU}(2) \), its inverse \( A^\dagger \), and perfect \( Z \) rotations are available, then still the operator \( X \) can be implemented with high fidelity. Such a situation is reality for some experiments—for example, those with amplitude errors [30].

We gratefully acknowledge funding from NSF RQCC Project No. 1111337 and the ARO Quantum Algorithms Program. T. J. Y. acknowledges the support of the NSF iQuISE IGERT program.