Synthesis of Quantum Superpositions by Quantum Computation

Lov K. Grover*

Bell Labs, Lucent Technologies, 600–700 Mountain Avenue, Murray Hill, New Jersey 07974

(Received 18 February 2000)

The quantum search algorithm can be looked at as a technique for synthesizing a particular kind of superposition—one whose amplitude is concentrated in a single basis state. This basis state is defined by a binary function $f(\overline{x})$ that is nonzero in this desired basis state and zero everywhere else. This paper extends the quantum search algorithm to an algorithm that can create an arbitrarily specified superposition on a space of size N in $O(\sqrt{N})$ steps. The superposition is specified by a complex valued function $f(\overline{x})$ that specifies the desired amplitude of the system in basis state $\overline{x}$.

PACS numbers: 03.67.Lx

The synthesis of quantum superpositions has previously been considered from the point of view of quantum control (i.e., controlling the time evolution of quantum systems). To synthesize a superposition on N states, the known algorithms for this, such as [1], take $O(N)$ steps. This paper presents an $O(\sqrt{N})$ step algorithm for this problem.

(I) Introduction.—Just as classical digital systems can be constructed out of two state systems called bits, quantum mechanical systems can be constructed out of basic two state quantum mechanical systems called *qubits.* Quantum mechanical operations that can be carried out in a controlled way are unitary operations that act on a small number of qubits in each step. In a quantum mechanical algorithm, the system is started in a state that is easy to prepare, say, one in which all qubits are in the 0 state; after this a sequence of simple operations is applied due to which the various qubits get entangled in some complicated way. When the system is observed after applying these operations, it can be analytically shown to yield the answer to a difficult computational problem with a high probability. For example, in the quantum search algorithm a function $f(\overline{x})$ is given. This function is defined over N basis states (denoted by $\overline{x}$). $f(\overline{x})$ is known to be nonzero at a single value of $\overline{x}$, say, t (t for target)—the goal is to find t. Given a particular basis state $\overline{x}$, the function $f(\overline{x})$ is easy to evaluate; however, there is no known information about the structure of $f(\overline{x})$ from which we can deduce which is the target state t. If one were using a classical computer, then on the average it would take $N/2$ function evaluations to solve this problem successfully. Quantum mechanical systems can simultaneously be in multiple basis states. By making use of this parallelism, it is possible to search for t in only $O(\sqrt{N})$ steps.

In order to design quantum computing systems, such as that for searching, we need a basic set of building blocks analogous to the NAND and NOR gates that are used to build classical digital systems. Unfortunately, by writing out the transformation matrices for NAND and NOR, it is easily seen that they are not unitary and hence cannot be implemented quantum mechanically. Fortunately, there exists an equivalent set of transformations that is unitary and can hence be implemented quantum mechanically: (i) NOT—a one-input one-output gate. The output is the NOT of the input. (ii) CNTRL-NOT—a two-input two-output gate. The first output is the same as the first input. If the first input is 1, the second output is the NOT of the second input; if the first input is 0, the second output is equal to the second input. (iii) CNTRL-CNTRL-NOT—a three-input three-output gate. The first two outputs are equal to the first two inputs, respectively. If the first two inputs are both 1's, the third output is the NOT of the third input; if either of the first two inputs is 0, the third output is equal to the third input. Note that this verbal description of the gates holds only for the 0 or 1 basis states; for superpositions, the transformations have to be obtained by the superposition principle.

The unitarity of these three is easily verified by writing out the transformation matrices and noting that the column vectors are orthonormal. Using these three gates it is possible to synthesize any Boolean function $f(\overline{x})$ that can be synthesized classically with approximately the same number of gates. Note that we need three basic gates, whereas in the classical case we needed just two (NAND and NOR). In order to develop more powerful quantum mechanical algorithms, in addition to these three gates, we need some operations that are essentially quantum mechanical, i.e., the entries of the state transition matrix are not all 0's and 1's. Two such operations that we need in the quantum search algorithm are the Walsh-Hadamard (WH) transformation operation and the selective inversion operation. These are discussed in the following two paragraphs.

A basic operation in quantum computing is the operation M performed on a single qubit—this is represented by the following matrix:

$$M \equiv \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

i.e., the state 0 is transformed into a superposition where the two states 0 and 1 have the same amplitude of $1/\sqrt{2}$, this superposition is denoted by $(1/\sqrt{2})(|0\rangle + |1\rangle)$; similarly, state 1 is transformed into the superposition $(1/\sqrt{2})(|0\rangle - |1\rangle)$. If we consider an n qubit system, we can perform the transformation M on each qubit independently in sequence, thus transforming the state of the system. A system consisting of n qubits has $N \equiv 2^n$ basis states, so the state transition matrix representing

 0031-9007/00/85(6)/1334(4)$15.00

this operation is of dimension $2^n \times 2^n$. Consider the case when the starting state is one of the 2^n basis states, i.e., a state described by a general string of n binary digits composed of some 0's and some 1's. The result of performing the transformation M on each qubit will be a superposition of states consisting of all possible n bit binary strings with amplitude of each state being $\pm 2^{-n/2}$. This transformation is referred to as the WH transformation [2]. A generalization of this is the quantum Fourier transformation that leads to applications such as factorization [3].

The other transformation that we need is the selective phase inversion of the amplitude in certain states. The transformation matrix describing this for a four state system with selective phase inversion of the second state is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Unlike the WH transformation, here the probability in each state stays the same. A quantum mechanical circuit to invert the amplitude in a certain set of states where the function $f(\overline{x})$ is nonzero can be designed if we are given a quantum mechanical black box that will evaluate the function $f(\overline{x})$ in any specified basis state $\overline{x}$—note that we do not need to know in advance which basis states the function is nonzero in. A realization of this kind of transformation can be achieved using the gates discussed so far as shown in Fig. 1 [4].

To analyze quantum circuits, such as this one, one examines the transformation of the input basis states; then by linearity, the effect on any superposition can be obtained. It is easily seen in the above circuit that if for some input basis vector $\overline{x}$, the output of the $f(\overline{x})$ gate is 1, the ancilla bit superposition is transformed from $(1/\sqrt{2}, -1/\sqrt{2})$ into $(-1/\sqrt{2}, 1/\sqrt{2})$; hence the amplitude of the entire system is inverted. If the output of the $f(\overline{x})$ gate is 0, the ancilla bit superposition stays unaltered, and hence the amplitude of the state stays the same. Thus in any superposition, the amplitude of those basis states are selectively inverted for which the function $f(\overline{x})$ is 1.

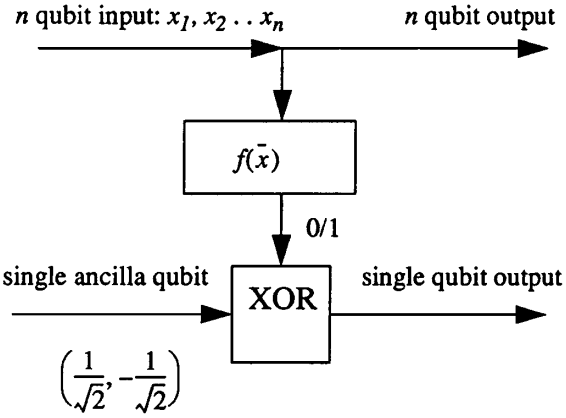

FIG. 1. The above quantum mechanical circuit inverts the amplitudes of precisely those states for which the function $f(x)$ is 1.

(II) Quantum Search.—As mentioned in the Introduction, in the quantum search algorithm a function $f(\overline{x})$ is given which is defined over N states (denoted by $\overline{x}$). $f(\overline{x})$ is known to be nonzero at a single value of $\overline{x}$, say, t (t for target), and the goal is to find t.

The quantum search algorithm consisted of $\sqrt{N}$ repetitions of the operator $-I_t W I_{\overline{0}} W$ starting with the state $\overline{0}$ (here W denotes the WH transformation, I_t denotes the selective phase inversion of the target state t, and $I_{\overline{0}}$ denotes the selective phase inversion of the $\overline{0}$ state) [5]. This creates a superposition all of whose amplitude is in the t state. A measurement will then immediately identify the t state. In case there are multiple t states, the quantum search algorithm creates a superposition with equal amplitude in all the t states. Reference [4] showed that if there are η t states, then after approximately $\sqrt{N/\eta}$ repetitions of the operator $-I_t W I_{\overline{0}} W$, a superposition is obtained that has equal amplitude in all t states and zero amplitude in all other states. An obvious next question is the following: What kinds of superpositions can quantum computing systems create and how efficiently can it create them? This paper answers that question by making use of a generalization of quantum search that is described in the next paragraph.

A few years after the quantum search algorithm was discovered, it was observed that similar results are obtained by replacing the WH transform by almost any valid quantum mechanical operation (say U) and the state $\overline{0}$ by any basis state s. It was shown that by starting with the basis state s, and carrying out $O(1/|U_{ts}|)$ repetitions of the operation sequence $-I_s U^{-1} I_t U$, one could reach the t state [6] (similar results are also proved in [7]). This showed that one could use any starting point and unitary operation U and from these amplify the amplitude in a desired target state t. A new class of algorithms was thus invented. These extended far beyond search problems—in fact, it was shown that this framework could be used to enhance almost any quantum mechanical algorithm [8]. One constraint with these algorithms was that they work when the problem has exactly one t state. For many problems like game-tree search, this was a restriction and the algorithms either could not be shown to work with multiple solutions or needed complicated techniques [9]. The paper [10] mentioned the multisolution case as an open problem.

This paper shows how the basic generalized search algorithm can be extended to the multisolution case. From this an algorithm for generating an arbitrarily specified superposition is deduced. This can be used to sample according to a general probability distribution—the number of steps required is approximately the square root of that of the corresponding classical algorithm. The following is the organization of the rest of this paper: section III carries out a general analysis of the quantum search algorithm with arbitrary unitary transformations and multiple target

states; section IV shows how this can be used to generate an arbitrarily specified superposition.

(III) Generalized search (general analysis).—The following section analyzes the evolution of a superposition starting in a basis state s (s for source), after which a composite operation Q where $Q \equiv -I_s U^{-1} I_t U$ is applied a certain number of times. I_s is a diagonal matrix with all diagonal elements equal to 1 except the ssth elements which are -1. I_s may be written as $I_s = I - 2|s\rangle\langle s|$. Here I is the identity matrix and, following standard Dirac notation, $|s\rangle$ denotes the column vector with all except the sth element equal to zero, the sth element is 1, and $\langle s|$ denotes the corresponding row vector. Similarly I_t is a diagonal matrix with all diagonal elements equal to 1 except the ttth elements which are equal to -1. I_t can be written in the form $I_t = I - \sum_t 2|t\rangle\langle t|$, where $|t\rangle$ is the column vector with all elements equal to 0, except a single one of the t elements which is 1, $\langle t|$ is the corresponding row vector. U denotes an arbitrary unitary matrix and U^{-1} is its inverse. The following results hold for arbitrary U, section IV describes how to choose U based on the specified probability distribution.

In this notation,

$$Q|s\rangle \equiv -(I_s U^{-1} I_t U)\,|s\rangle = -(I - 2|s\rangle\langle s|)U^{-1}\Big(I - \sum_t 2|t\rangle\langle t|\Big)U|s\rangle$$
$$= -|s\rangle + 2|s\rangle + \sum_t 2\langle t|U|s\rangle U^{-1}|t\rangle - \sum_t 4|s\rangle\langle s|U^{-1}|t\rangle\langle t|U|s\rangle .$$

Note that $\langle t|U|s\rangle \equiv U_{ts}$. Also, since U is unitary, U^{-1} is equal to the transpose of the complex conjugate of U, and therefore $\langle s|U^{-1}|t\rangle = \langle t|U^*|s\rangle$ which is equal to U^*_{ts}. Therefore the above equation becomes

$$Q|s\rangle = |s\rangle + \sum_t (2U_{ts})U^{-1}|t\rangle - \sum_t 4|U_{ts}|^2|s\rangle .$$

Similarly, for any t state

$$QU^{-1}|t\rangle \equiv -(I_s U^{-1} I_t U)U^{-1}|t\rangle = -I_s U^{-1} I_t|t\rangle$$
$$= (I - 2|s\rangle\langle s|)U^{-1}|t\rangle = U^{-1}|t\rangle - 2U^*_{ts}|s\rangle .$$

The analysis so far shows that, if there are η t states, then the $(\eta + 1)$-dimensional space defined by the vector $|s\rangle$ and the η vectors $U^{-1}|t\rangle$ is preserved by the operator Q. There is indeed a simpler two-dimensional subspace that is also preserved, as we show in the next paragraph.

Multiply both sides of the second equation, $QU^{-1}|t\rangle = U^{-1}|t\rangle - 2U^*_{ts}|s\rangle$, by U_{ts} and sum over all t states. The equations now become

$$Q|s\rangle = |s\rangle\Big(1 - 4\sum_t |U_{ts}|^2\Big) + 2\sum_t U_{ts}U^{-1}|t\rangle \quad \text{(same as before)}$$

$$Q\sum_t U_{ts}U^{-1}|t\rangle = \sum_t U_{ts}U^{-1}|t\rangle - 2\sum_t |U_{ts}|^2|s\rangle .$$

Q is thus a transformation in the two-dimensional complex Hilbert space defined by $|s\rangle$ and $\sum_t U_{ts}U^{-1}|t\rangle$. Normalizing the vector $\sum_t U_{ts}U^{-1}|t\rangle$ and denoting $\sqrt{\sum_t |U_{ts}|^2}$ by u, the above transformation can be represented as

$$Q\begin{bmatrix} |s\rangle \\ \frac{1}{u}\sum_t U_{ts}U^{-1}|t\rangle \end{bmatrix} = \begin{bmatrix} (1 - 4u^2) & -2u \\ 2u & 1 \end{bmatrix} \times \begin{bmatrix} |s\rangle \\ \frac{1}{u}\sum_t U_{ts}U^{-1}|t\rangle \end{bmatrix}.$$

In order to find the effect of repeated applications of Q, we use standard matrix analysis which consists of finding the eigenvalues and eigenvectors of the transformation matrix. Assuming u to be small, the two eigenvalues and eigenvectors are approximately $\lambda_1 = 1 + 2iu$, $\nu_1 = \left[\begin{smallmatrix} 1 \\ -i \end{smallmatrix}\right]$ and $\lambda_2 = 1 - 2iu$, $\nu_2 = \left[\begin{smallmatrix} 1 \\ i \end{smallmatrix}\right]$.

The initial state vector is $|s\rangle$, which in terms of the eigenvectors may be written as $\frac{1}{2}(\nu_1 + \nu_2)$. After η applications of Q, this transforms into $\frac{1}{2}(\nu_1\lambda_1^\eta + \nu_2\lambda_2^\eta)$ which may be simplified to $\left[\begin{smallmatrix} \cos(2u\eta) \\ \sin(2u\eta) \end{smallmatrix}\right]$. Therefore, $\pi/4u$ applications of Q transform the state vector $|s\rangle$ into $\frac{1}{u}\sum_t U_{ts}U^{-1}|t\rangle$.

(IV) State vector engineering.—Suppose that we are required to synthesize a specified superposition. The amplitude in each of N basis states, denoted by $\overline{x}$, is required to be proportional to a given function $f(\overline{x})$. Assume, without loss of generality, that the maximum value of $|f(\overline{x})|$ is 1.

The following algorithm synthesizes the specified superposition in approximately $(\pi/4)\sqrt{N/\sum_{\overline{x}} |f(\overline{x})|^2}$ steps. One immediate application of this algorithm is in sampling according to an arbitrary probability distribution. For this a quantum superposition is generated with the amplitudes in each state having a magnitude equal to the square root of the probabilities and arbitrary phases. After this, through a measurement, a sample is obtained. Such an algorithm will require only $O(\sqrt{N})$ steps while a classical algorithm for general sampling will need at least $O(N)$ steps.

Solution: Define $N \equiv 2^n$ states by n qubits, denoted by $\overline{x}$. Include an additional ancilla qubit. Initialize the state so that all qubits are in the 0 state—the state of the whole system is denoted by $(0, \overline{0})$ (the first 0 denotes a single qubit in the 0 state and $\overline{0}$ denotes each of the other n qubits in the 0 state).

Next consider the following two unitary operations which constitute the building blocks of our algorithm. U_1: leave the first qubit unaltered and apply a WH transform to the other n qubits; and U_2: carry out a conditional rotation of the first qubit so that the state $(0, \overline{x})$ gets transformed into the superposition $[f(\overline{x})\,(0, \overline{x}) + \sqrt{1 - |f(\overline{x})|^2}\,(1, \overline{x})]$; i.e., the amplitude of the state $(0, \overline{x})$ is $f(\overline{x})$ and the amplitude of the state $(1, \overline{x})$ is $\sqrt{1 - |f(\overline{x})|^2}$.

This type of unitary operation has previously been used in quantum computing algorithms, e.g., in the mean

estimation algorithm in [8]. It can be accomplished by first transferring $f(\overline{x})$ into the phase through conditional phase rotation and then converting it into amplitude information through the same type of circuit as that for conditional phase inversion. It will be described in detail in [11].

(i) I_t: in case the first qubit is 0, invert the phase; if the first qubit is 1, leave it unchanged. In other words, states with the first qubit in the 0 state are t states.

(ii) I_s: in case all the qubits (including the first qubit) are 0, invert the phase; else leave it unchanged; i.e., $(0,\overline{0})$ is the s state.

Clearly if we start with the $(0,\overline{0})$ state and apply U_1 and then U_2, the amplitude in the $(0,\overline{x})$ state is $f(\overline{x})/\sqrt{N}$. In other words, if we define the composite operation $U \equiv U_2U_1$, the s state as $(0,\overline{0})$, and the t states as the $(0,\overline{x})$ states, then U_{ts}, the matrix element between s and the relevant t state, is $f(\overline{x})/\sqrt{N}$.

It follows from section III that by starting with the $(0,\overline{0})$ state, and applying the sequence of operations defined by $Q \equiv -(I_sU^{-1}I_tU)$, $(\pi/4)\sqrt{N/\sum_{\overline{x}}|f(\overline{x})|^2}$ times, followed by a single application of U, we get the first qubit in the 0 state and the remaining n qubits in a superposition with the amplitude of the $\overline{x}$ state as $f(\overline{x})/\sqrt{\sum_{\overline{x}}|f(\overline{x})|^2}$.

Observations:

(i) Let $f(\overline{x})$ be 1 at η points in the domain and zero everywhere else. Since $\sum_{\overline{x}}|f(\overline{x})|^2 = \eta$, the algorithm needs $(\pi/4)\sqrt{N/\eta}$ steps. After this it reaches the same superposition as reached in the basic quantum search algorithm with η solutions, and it needs exactly the same number of iterations as the quantum search algorithm [7] to reach this.

(ii) The number of steps required by the algorithm depends on $\sum_{\overline{x}}|f(\overline{x})|^2$. In case this quantity is not known in advance, the superposition can still be synthesized in $O(\sqrt{N})$ steps, by trying out the algorithm with a few carefully chosen run times. After this the ancilla qubit is measured. The algorithm is repeated until this is observed to be 0 (it can be shown that, with appropriately chosen run times, the probability of not getting even a single 0 falls exponentially with the square of the number of times the procedure is repeated [12]). Once a 0 is observed, the remaining n qubits immediately collapse into the desired superposition.

(iii) The algorithm assumed that the function $f(\overline{x})$ was normalized so that its maximum value was equal to 1. The algorithm as presented in this section is equally valid for different $f(\overline{x})$, provided the maximum value does not exceed 1. For example, if the desired probability at each $\overline{x}$ is specified [i.e., $|f(\overline{x})|^2$], the value of $\sum_{\overline{x}}|f(\overline{x})|^2$ becomes 1 and the algorithm needs exactly $(\pi/4)\sqrt{N}$ iterations to attain this distribution. In general, the number of steps required is smaller if we can choose a larger constant to scale $f(\overline{x})$. This is maximized for the choice made in this section where the maximum value of $f(\overline{x})$ is 1.

(V) Conclusion.—The quantum search algorithm is perhaps the simplest possible quantum mechanical algorithm that yields a significant advantage over a classical algorithm. It has inspired several new ideas and algorithms. This paper presents the most recent development which shows that the amplitude amplification class of algorithms can be made to work in the presence of multiple target states. One application of this is to generate a sample according to an arbitrary probability distribution in a number of steps which is only a square root of that required by a classical algorithm (accomplished by synthesizing a superposition with amplitudes which are the square roots of the classical probabilities and then carrying out a measurement). Another immediate application is in extending the framework of the generalized search class of algorithms of the type discussed in [6,7,13], so that they are no longer limited to problems with a single solution.

There might be other quantum algorithms that use synthesis of a quantum superposition as an intermediate step to accomplish some other end. Having pointed out an efficient technique for synthesizing superpositions, such applications are more conceivable.

Thanks to Ashwin Nayak, Norm Margolus, Hein Roehrig, and Charles Bennett for going through the paper and making valuable comments. This material is based upon work supported in part by the U.S. Army Research Office under Contract No. DAAG55-98-C-0040.

*Electronic address: lkgrover@bell-labs.com

[1] C. K. Law and J. H. Eberly, Phys. Rev. Lett. **76**, 1055–1058 (1996).

[2] D. Deutsch and R. Jozsa, Proc. R. Soc. London A **400**, 73–90 (1992).

[3] P. W. Shor, in *Proceedings of the 35th Annual Symposium on Fundamentals of Computer Science (FOCS), 1994* (IEEE Computer Society, Los Alamitos, CA, 1994), pp. 124–134.

[4] M. Boyer, G. Brassard, P. Hoyer, and A. Tapp, http://xxx.lanl.gov/abs/quant-ph/9605034.

[5] L. K. Grover, Phys. Rev. Lett. **78**, 325–328 (1997).

[6] L. K. Grover, Phys. Rev. Lett. **80**, 4329–4332 (1998).

[7] G. Brassard, P. Hoyer, and A. Tapp, http://xxx.lanl.gov/abs/quant-ph/9805082.

[8] L. K. Grover, in *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC), 1998* (ACM Press, New York, 1998), pp. 53–63.

[9] H. Buhrman, R. Cleve, and A. Wigderson, in *Proceedings of the 30th ACM Symposium of Theory of Computing (STOC), 1998* (Ref. [8]), pp. 63–69.

[10] L. K. Grover, Chaos Solitons Fractals **10**, 1695–1705 (1999); also available in *Lecture Notes in Computer Science* (Springer-Verlag, Berlin, 1999), Vol. 1509, pp. 126–139; http://xxx.lanl.gov/abs/quant-ph/9802035.

[11] H. Bernstein, H. N. Barnum, and L. K. Grover, "Implementing finite precision rapid sampling" (to be published).

[12] H. Buhrman, R. Cleve, R. De Wolf, and C. Zalka, *Proceedings of the 40th Annual Symposium on Fundamentals of Computer Science (FOCS), 1999* (IEEE Computer Society, Los Alamitos, CA, 1999).

[13] E. Farhi and S. Gutmann, lanl e-print http://xxx.lanl.gov/abs/quant-ph/9711035.