WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Deutsch
Español
فارسی
Français
Italiano
עברית
Lietuvių
Magyar
日本語
Polski
Русский
Српски / srpski
Tiếng Việt
Edit links

Grover's algorithm

From Wikipedia, the free encyclopedia

Grover's algorithm is a quantum algorithm that finds with high probability the unique input to a black box function that produces a particular output value, using just $O(N^{1/2})$ evaluations of the function, where N is the size of the function's domain.

The analogous problem in classical computation cannot be solved in fewer than $O(N)$ evaluations (because, in the worst case, the correct input might be the last one that is tried). At roughly the same time that Grover published his algorithm, Bennett, Bernstein, Brassard, and Vazirani published a proof that no quantum solution to the problem can evaluate the function fewer than $O(N^{1/2})$ times, so Grover's algorithm is asymptotically optimal.[1]

Unlike other quantum algorithms, which may provide exponential speedup over their classical counterparts, Grover's algorithm provides only a quadratic speedup. However, even quadratic speedup is considerable when N is large. Grover's algorithm could brute force a 128-bit symmetric cryptographic key in roughly 2^{64} iterations, or a 256-bit key in roughly 2^{128} iterations. As a result, it is sometimes suggested that symmetric key lengths be doubled to protect against future quantum attacks.

Like many quantum algorithms, Grover's algorithm is probabilistic in the sense that it gives the correct answer with a probability of less than 1. Though there is technically no upper bound on the number of repetitions that might be needed before the correct answer is obtained, the expected number of repetitions is a constant factor that does not grow with N.

Grover's original paper described the algorithm as a database search algorithm, and this description is still common. The database in this analogy is a table of all of the function's outputs, indexed by the corresponding input.

Contents [hide]

1 Applications
2 Setup
3 Algorithm steps
4 The first iteration
5 Description of U_ω
6 Geometric proof of correctness
7 Algebraic proof of correctness
8 Extension to space with multiple targets
9 Quantum partial search
10 Optimality
11 Applicability and Limitations
12 See also
13 Notes
14 References
15 External links

Applications [edit]

Although the purpose of Grover's algorithm is usually described as "searching a database", it may be more accurate to describe it as "inverting a function". Roughly speaking, if we have a function *y = f(x)* that can be evaluated on a quantum computer, Grover's algorithm allows us to calculate *x* when given *y*. Inverting a function is related to the searching of a database because we could come up with a function that produces one particular value of *y* ("true" for instance) if *x* matches a desired entry in a database, and another value of *y* ("false") for other values of *x*.

Grover's algorithm can also be used for estimating the mean and median of a set of numbers, and for solving the Collision problem. The algorithm can be further optimized if there is more than one matching entry and the number of matches is known beforehand.

Setup [edit]

Consider an unsorted database with *N* entries. The algorithm requires an *N*-dimensional state space *H*, which can be supplied by $n = \log_2 N$ qubits. Consider the problem of determining the index of the database entry which satisfies some search criterion. Let *f* be the function which maps database entries to *0* or *1*, where *f(ω)=1* if and only if *ω* satisfies the search criterion. We are provided with (quantum black box) access to a subroutine in the form of a unitary operator, U_ω, which acts as follows (for the *ω* for which *f(ω)=1*):

$$U_\omega|\omega\rangle = -|\omega\rangle$$
$$U_\omega|x\rangle = |x\rangle \qquad \text{for all } x \neq \omega$$

Our goal is to identify the index $|\omega\rangle$.

Algorithm steps [edit]

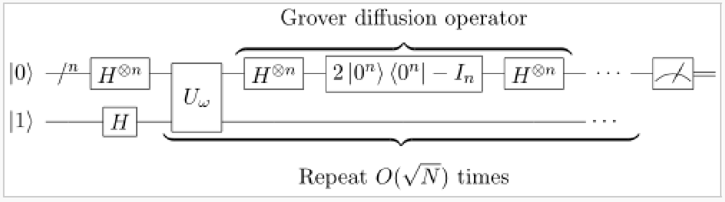
The steps of Grover's algorithm are given as follows. Let $|s\rangle$ denote the uniform superposition over all states

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=1}^{N} |x\rangle.$$

Then the operator

$$U_s = 2\,|s\rangle\,\langle s| - I$$

is known as the Grover diffusion operator.



Grover diffusion operator
$|0\rangle$
$/^n$
$H^{\otimes n}$
U_ω
$H^{\otimes n}$
$2\,|0^n\rangle\,\langle 0^n| - I_n$
$H^{\otimes n}$
$|1\rangle$
H
Repeat $O(\sqrt{N})$ times

Quantum circuit representation of Grover's algorithm

Here is the algorithm:

1. Initialize the system to the state
$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=1}^{N} |x\rangle$$
2. Perform the following "Grover iteration" *r(N)* times. The function *r(N),* which is asymptotically *O(N*$^{1/2}$*)*, is described below.
    1. Apply the operator U_ω.
    2. Apply the operator U_s.
3. Perform the measurement Ω. The measurement result will be eigenvalue λ_ω with probability approaching 1 for N≫1. From λ_ω, ω may be obtained.

The first iteration [edit]

A preliminary observation, in parallel with our definition

$$U_s = 2\,|s\rangle\,\langle s| - I,$$

is that U_ω can be expressed in an alternate way:

$$U_\omega = I - 2\,|\omega\rangle\,\langle\omega|.$$

To prove this it suffices to check how U_ω acts on basis states:

$$(I - 2|\omega\rangle\langle\omega|)|\omega\rangle = |\omega\rangle - 2|\omega\rangle\langle\omega|\omega\rangle = -|\omega\rangle = U_\omega|\omega\rangle.$$
$$(I - 2|\omega\rangle\langle\omega|)|x\rangle = |x\rangle - 2|\omega\rangle\langle\omega|x\rangle = |x\rangle = U_\omega|x\rangle \text{ for all } x \neq \omega.$$

The following computations show what happens in the first iteration:

$$\langle\omega|s\rangle = \langle s|\omega\rangle = \frac{1}{\sqrt{N}}.$$
$$\langle s|s\rangle = N\frac{1}{\sqrt{N}} \cdot \frac{1}{\sqrt{N}} = 1.$$
$$U_\omega|s\rangle = (I - 2|\omega\rangle\langle\omega|)|s\rangle = |s\rangle - 2|\omega\rangle\langle\omega|s\rangle = |s\rangle - \frac{2}{\sqrt{N}}|\omega\rangle.$$
$$U_s(|s\rangle - \frac{2}{\sqrt{N}}|\omega\rangle) = (2|s\rangle\langle s| - I)\left(|s\rangle - \frac{2}{\sqrt{N}}|\omega\rangle\right) = 2|s\rangle\langle s|s\rangle - |s\rangle \quad - \quad \frac{4}{\sqrt{N}}|s\rangle\langle s|\omega\rangle + \frac{2}{\sqrt{N}}|\omega\rangle =$$
$$= 2|s\rangle - |s\rangle - \frac{4}{\sqrt{N}} \cdot \frac{1}{\sqrt{N}}|s\rangle + \frac{2}{\sqrt{N}}|\omega\rangle = |s\rangle - \frac{4}{N}|s\rangle + \frac{2}{\sqrt{N}}|\omega\rangle = \frac{N-4}{N}|s\rangle + \frac{2}{\sqrt{N}}|\omega\rangle.$$

After application of the two operators (U_ω and U_s), the amplitude of the searched-for element has increased from $|\langle\omega|s\rangle|^2 = 1/N$ to $|\langle\omega|U_s U_\omega s\rangle|^2 \approx 9/N$.

Description of U_ω [edit]

Grover's algorithm requires a "quantum oracle" operator U_ω which can recognize solutions to the search problem and give them a negative sign. In order to keep the search algorithm general, we will leave the inner workings of the oracle as a black box, but will explain how the sign is flipped. The oracle contains a function f which returns $f(x) = 1$ if $|x\rangle$ is a solution to the search problem and $f(x) = 0$ otherwise. The oracle is a unitary operator which operates on two qubits, the index qubit $|x\rangle$ and the oracle qubit $|q\rangle$:

$$|x\rangle|q\rangle \xrightarrow{U_\omega} |x\rangle|q \oplus f(x)\rangle$$

As usual, $\oplus$ denotes addition modulo 2. The operation flips the oracle qubit if $f(x) = 1$ and leaves it alone otherwise. In Grover's algorithm we want to flip the sign of the state $|x\rangle$ if it labels a solution. This is achieved by setting the oracle qubit in the state $(|0\rangle - |1\rangle)/\sqrt{2}$, which is flipped to $(|1\rangle - |0\rangle)/\sqrt{2}$ if $|x\rangle$ is a solution:

$$|x\rangle\,(|0\rangle - |1\rangle)\,/\sqrt{2} \xrightarrow{U_\omega} (-1)^{f(x)}|x\rangle\,(|0\rangle - |1\rangle)\,/\sqrt{2}$$

We regard $|x\rangle$ as flipped, thus the oracle qubit is not changed, so by convention the oracle qubits are usually not mentioned in the specification of Grover's algorithm. Thus the operation of the oracle U_ω is simply written as:

$$|x\rangle \xrightarrow{U_\omega} (-1)^{f(x)}|x\rangle$$

Geometric proof of correctness [edit]

Consider the plane spanned by $|s\rangle$ and $|\omega\rangle$; equivalently, the plane spanned by $|\omega\rangle$ and the perpendicular ket $|s'\rangle = \frac{1}{\sqrt{N-1}}\sum_{x\neq\omega}|x\rangle$.

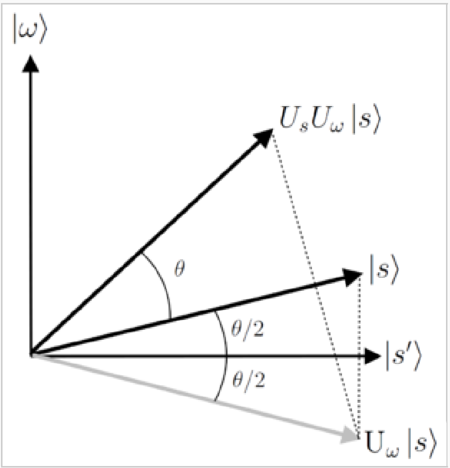
We will consider the first iteration, acting on the initial ket $|s\rangle$. Since $|\omega\rangle$ is one of the basis vectors in $|s\rangle$ the overlap is

$$\langle s'|s\rangle = \sqrt{\frac{N-1}{N}}$$

In geometric terms, the angle $\theta/2$ between $|s\rangle$ and $|s'\rangle$ is given by:

$$\sin\theta/2 = \frac{1}{\sqrt{N}}$$

The operator U_ω is a reflection at the hyperplane orthogonal to $|\omega\rangle$ for vectors in the plane spanned by $|s'\rangle$ and $|\omega\rangle$; i.e. it acts as a reflection across $|s'\rangle$. The operator U_s is a reflection through $|s\rangle$. Therefore, the state vector remains in the plane spanned by $|s'\rangle$ and $|\omega\rangle$ after each application of the operators U_s and U_ω, and it is straightforward to check that the operator U_sU_ω of each Grover iteration step rotates the state vector by an angle of $\theta = 2\arcsin\frac{1}{\sqrt{N}}$.



Picture showing the geometric interpretation of the first iteration of Grover's algorithm. The state vector $|s\rangle$ is rotated towards the target vector $|\omega\rangle$ as shown.

We need to stop when the state vector passes close to $|\omega\rangle$; after this, subsequent iterations rotate the state vector *away* from $|\omega\rangle$, reducing the probability of obtaining the correct answer. The exact probability of measuring the correct answer is:

$$\sin^2\left(\left(r+\frac{1}{2}\right)\theta\right)$$

where *r* is the (integer) number of Grover iterations. The earliest time that we get a near-optimal measurement is therefore $r \approx \pi\sqrt{N}/4$.

Algebraic proof of correctness [edit]

To complete the algebraic analysis we need to find out what happens when we repeatedly apply U_sU_ω. A natural way to do this is by eigenvalue analysis of a matrix. Notice that during the entire computation, the state of the algorithm is a linear combination of s and ω. We can write the action of U_s and U_ω in the space spanned by $\{|s\rangle, |\omega\rangle\}$ as:

$$U_s : a|\omega\rangle + b|s\rangle \mapsto (|\omega\rangle\ |s\rangle)\begin{pmatrix}-1 & 0\\ 2/\sqrt{N} & 1\end{pmatrix}\begin{pmatrix}a\\ b\end{pmatrix}.$$

$$U_\omega : a|\omega\rangle + b|s\rangle \mapsto (|\omega\rangle\ |s\rangle)\begin{pmatrix}-1 & -2/\sqrt{N}\\ 0 & 1\end{pmatrix}\begin{pmatrix}a\\ b\end{pmatrix}.$$

So in the basis $\{|\omega\rangle, |s\rangle\}$ (which is neither orthogonal nor a basis of the whole space) the action U_sU_ω of applying U_ω followed by U_s is given by the matrix

$$U_sU_\omega = \begin{pmatrix}-1 & 0\\ 2/\sqrt{N} & 1\end{pmatrix}\begin{pmatrix}-1 & -2/\sqrt{N}\\ 0 & 1\end{pmatrix} = \begin{pmatrix}1 & 2/\sqrt{N}\\ -2/\sqrt{N} & 1-4/N\end{pmatrix}.$$

This matrix happens to have a very convenient Jordan form. If we define $t = \arcsin(1/\sqrt{N})$, it is

$$U_sU_\omega = M\begin{pmatrix}\exp(2it) & 0\\ 0 & \exp(-2it)\end{pmatrix}M^{-1} \text{ where } M = \begin{pmatrix}-i & i\\ \exp(it) & \exp(-it)\end{pmatrix}.$$

It follows that *r*th power of the matrix (corresponding to *r* iterations) is

$$(U_sU_\omega)^r = M\begin{pmatrix}\exp(2rit) & 0\\ 0 & \exp(-2rit)\end{pmatrix}M^{-1}$$

Using this form we can use trigonometric identities to compute the probability of observing *ω* after *r* iterations mentioned in the previous section,

$$\left|\begin{pmatrix}\langle\omega|\omega\rangle & \langle\omega|s\rangle\end{pmatrix}(U_sU_\omega)^r\begin{pmatrix}0\\ 1\end{pmatrix}\right|^2 = \sin^2\left((2r+1)t\right).$$

Alternatively, one might reasonably imagine that a near-optimal time to distinguish would be when the angles *2rt* and *-2rt* are as far apart as possible, which corresponds to $2rt \approx \pi/2$ or $r = \pi/4t = \pi/4\arcsin(1/\sqrt{N}) \approx \pi\sqrt{N}/4$. Then the system is in state

$$(|\omega\rangle\ |s\rangle)(U_sU_\omega)^r\begin{pmatrix}0\\ 1\end{pmatrix} \approx (|\omega\rangle\ |s\rangle)M\begin{pmatrix}i & 0\\ 0 & -i\end{pmatrix}M^{-1}\begin{pmatrix}0\\ 1\end{pmatrix} = |w\rangle\frac{1}{\cos(t)} - |s\rangle\frac{\sin(t)}{\cos(t)}.$$

A short calculation now shows that the observation yields the correct answer *ω* with error *O(1/N)*.

Extension to space with multiple targets [edit]

If, instead of 1 matching entry, there are *k* matching entries, the same algorithm works but the number of iterations must be $\pi(N/k)^{1/2}/4$ instead of $\pi N^{1/2}/4$. There are several ways to handle the case if *k* is unknown. For example, one could run Grover's algorithm several times, with

$$\pi\frac{N^{1/2}}{4}, \pi\frac{(N/2)^{1/2}}{4}, \pi\frac{(N/4)^{1/2}}{4}, \ldots$$

iterations. For any *k*, one of the iterations will find a matching entry with a sufficiently high probability. The total number of iterations is at most

$$\pi\frac{N^{1/2}}{4}\left(1+\frac{1}{\sqrt{2}}+\frac{1}{2}+\cdots\right)=\pi\frac{N^{1/2}}{4}\left(2+\sqrt{2}\right)$$

which is still O($N^{1/2}$). It can be shown that this can be improved. If the number of marked items is *k*, where *k* is unknown, there is an algorithm that finds the solution in $\sqrt{N/k}$ queries. This fact is used in order to solve the collision problem. [2][3]

Quantum partial search [edit]

A modification of Grover's algorithm called quantum partial search was described by Grover and Radhakrishnan in 2004.[4] In partial search, one is not interested in finding the exact address of the target item, only the first few digits of the address. Equivalently, we can think of "chunking" the search space into blocks, and then asking "in which block is the target item?". In many applications, such a search yields enough information if the target address contains the information wanted. For instance, to use the example given by L.K. Grover, if one has a list of students organized by class rank, we may only be interested in whether a student is in the lower 25%, 25-50%, 50-70% or 75-100% percentile.

To describe partial search, we consider a database separated into K blocks, each of size $b = N/K$. Obviously, the partial search problem is easier. Consider the approach we would take classically - we pick one block at random, and then perform a normal search through the rest of the blocks (in set theory language, the complement). If we don't find the target, then we know it's in the block we didn't search. The average number of iterations drops from $N/2$ to $(N-b)/2$.

Grover's algorithm requires $\pi/4\sqrt{N}$ iterations. Partial search will be faster by a numerical factor which depends on the number of blocks K. Partial search uses n_1 global iterations and n_2 local iterations. The global Grover operator is designated G_1 and the local Grover operator is designated G_2.

The global Grover operator acts on the blocks. Essentially, it is given as follows:

1. Perform j_1 standard Grover iterations on the entire database.
2. Perform j_2 local Grover iterations. A local Grover iteration is a direct sum of Grover iterations over each block.
3. Perform one standard Grover iteration

The optimal values of j_1 and j_2 are discussed in the paper by Grover and Radhakrishnan. One might also wonder what happens if one applies successive partial searches at different levels of "resolution". This idea was studied in detail by Korepin and Xu, who called it binary quantum search. They proved that it is not in fact any faster than performing a single partial search.

Optimality [edit]

It is known that Grover's algorithm is optimal. That is, any algorithm that accesses the database only by using the operator U_ω must apply U_ω at least as many times as Grover's algorithm.[1] This result is important in understanding the limits of quantum computation. If the Grover's search problem was solvable with $\log^c N$ applications of U_ω, that would imply that NP is contained in BQP, by transforming problems in NP into Grover-type search problems. The optimality of Grover's algorithm suggests (but does not prove) that NP is not contained in BQP.

The number of iterations for *k* matching entries, $\pi(N/k)^{1/2}/4$, is also optimal.[2]

Applicability and Limitations [edit]

When applications of Grover's algorithm are considered, it should be emphasized that the database is not represented explicitly. Instead, an oracle is invoked to evaluate an item by its index. Reading a full data-base item by item and converting it into such a representation may take a lot longer than Grover's search. To account for such effects, Grover's algorithm can be viewed as solving an equation or satisfying a constraint. In such applications, the oracle is a way to check the constraint and is not related to the search algorithm. This separation usually prevents algorithmic optimizations, whereas conventional search algorithms often rely on such optimizations and avoid exhaustive search. These and other considerations about using Grover's algorithm are discussed in [5]

See also [edit]

- Amplitude amplification
- Shor's algorithm

Notes [edit]

1. ^ *a* *b* Bennett C.H., Bernstein E., Brassard G., Vazirani U., *The strengths and weaknesses of quantum computation*. SIAM Journal

on Computing 26(5): 1510–1523 (1997). Shows the optimality of Grover's algorithm.
2. ^ a b Michel Boyer; Gilles Brassard; Peter Høyer; Alain Tapp (1998), "Tight Bounds on Quantum Searching", *Fortsch. Phys.* **46**: 493–506, arXiv:quant-ph/9605034, Bibcode:1998ForPh..46..493B, doi:10.1002/3527603093.ch10
3. ^ Andris Ambainis (2004), "Quantum search algorithms", *SIGACT News* **35** (2): 22–35, arXiv:quant-ph/0504012, Bibcode:2005quant.ph..4012A, doi:10.1145/992287.992296
4. ^ L.K. Grover and J. Radhakrishnan,*Is partial quantum search of a database any easier?*. quant-ph/0407122
5. ^ Viamontes G.F.; Markov I.L.; Hayes J.P. (2005), "Is Quantum Search Practical?" (PDF), *IEEE/AIP Computing in Science and Engineering* **7** (3): 62–70

References [edit]

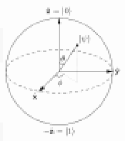
- Grover L.K.: *A fast quantum mechanical algorithm for database search*, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (May 1996) p. 212
- Grover L.K.: *From Schrödinger's equation to quantum search algorithm*, American Journal of Physics, 69(7): 769-777, 2001. Pedagogical review of the algorithm and its history.
- Nielsen, M.A. and Chuang, I.L. *Quantum computation and quantum information*. Cambridge University Press, 2000. Chapter 6.
- What's a Quantum Phone Book?, Lov Grover, Lucent Technologies
- Grover's Algorithm: Quantum Database Search, C. Lavor, L.R.U. Manssur, R. Portugal
- Grover's algorithm on arxiv.org

External links [edit]

- [1] simulation and circuit diagram.
- Grover's Quantum Search Algorithm animated explanation.
- [2] simulation and circuit diagram with cats.
- Wolfram Demonstration Project: Quantum Circuit Implementing Grover's Search Algorithm

Quantum information science [hide]

General		Quantum computer · Qubit · Quantum information · Quantum programming · Timeline of quantum computing
Quantum communication		Quantum capacity · Classical capacity · Entanglement-assisted classical capacity · Quantum channel (Quantum network) · Quantum cryptography (Quantum key distribution) · Quantum energy teleportation · Quantum teleportation · Superdense coding · LOCC · Entanglement distillation
Quantum algorithms		Universal quantum simulator · Deutsch–Jozsa algorithm · **Grover's algorithm** · Quantum Fourier transform · Shor's algorithm · Simon's problem · Quantum phase estimation algorithm · Quantum annealing · Algorithmic cooling
Quantum complexity theory		Quantum Turing machine · BQP · QMA · PostBQP
Quantum computing models		Quantum circuit (Quantum gate) · One-way quantum computer (cluster state) · Adiabatic quantum computation · Topological quantum computer
Decoherence prevention		Quantum error correction · Stabilizer codes · Entanglement-Assisted Quantum Error Correction · Quantum convolutional codes
Physical implementations	**Quantum optics**	Cavity QED · Circuit QED · Linear optical quantum computing
	Ultracold atoms	Trapped ion quantum computer · Optical lattice
	Spin-based	Nuclear magnetic resonance QC · Kane QC · Loss–DiVincenzo QC · Nitrogen-vacancy center
	Superconducting quantum computing	Charge qubit · Flux qubit · Phase qubit



Categories: Quantum algorithms | Search algorithms