Q

Quantization of Markov Chains
2004; Szegedy

PETER RICHTER, MARIO SZEGEDY
Department of Computer Science, Rutgers,
The State University of New Jersey, Piscataway, NJ, USA

Keywords and Synonyms

Quantum walks

Problem Definition

Spatial Search and Walk Processes

Spatial search by *quantum walk* is database search with the additional constraint that one must move through the search space via a *quantum walk* that obeys some locality structure (grid, hypercube, etc.). Quantum walks are analogues of classical random walks on graphs. The complexity of spatial search by quantum walk is essentially determined by the *quantum hitting time* [9] of the walk.

Let S with $|S| = N$ be a finite set of *states*, and let $P = (p_{x,y})_{x,y\in S}$ be the *transition probability matrix* of a *Markov chain* on S, also denoted by P. Assume that a subset $M \subseteq S$ of states are *marked*. The goal is either to find a marked state, given that $M \neq \emptyset$ (*search version*), or to determine whether M is nonempty (*decision version*). If the possible $x \to y$ moves (i. e., those with $p_{x,y} \neq 0$) form the edges of a (directed) graph G, it is said that the walk has *locality structure* G.

INPUT: Markov chain P on set S, marked subset $M \subseteq S$.
OUTPUT: A marked state with probability 0.1 iff one exists (search version), or a Boolean return value with one-sided error detecting $M \neq \emptyset$ with probability 0.1 (decision version).

If P is *irreducible* (i. e., if its underlying digraph is strongly connected), a marked state can be found with high probability in finite time by simulating a *classical* random walk using the coefficients of P. In the *quantum* case, this random walk process may be replaced by a *quantum walk* using the coefficients of P (in particular, respecting locality). The fundamental question is whether the quantum walk process finds a marked state *faster* than the classical random walk process.

The Quantum Walk Algorithm

Quantizing P is not so straightforward, since stochastic matrices have no immediate unitary equivalents. It turns out that one must either abandon the discrete-time nature of the walk [7] or define the walk operator on a space other than $\mathbf{C}^S$. Here the second route is taken, with notation as in [18]. On $\mathbf{C}^{S\times S}$, define the unitary $W_P := R_1 R_2$, where $R_1 = \sum_{x\in S}(2|p_x\rangle\langle p_x| - I) \otimes |x\rangle\langle x|$, $R_2 = \sum_{x\in S} |x\rangle\langle x| \otimes (2|p_x\rangle\langle p_x| - I)$, and $|p_x\rangle := \sum_{y\in S} \sqrt{p_{y,x}}|y\rangle$. W_P is the *quantization* of P, or the *discrete-time quantum walk operator* arising from P. One can "check" whether or not the current state is marked by applying the operator $O_M = \sum_{x\notin M} |x\rangle\langle x| - \sum_{x\in M} |x\rangle\langle x|$. Denote the cost of constructing W_P (in the units of the resource of interest) by U (*update cost*), the cost of constructing O_M by C (*checking cost*), and the cost of preparing the initial state, ϕ_0, by S (*setup cost*). Every time an operator is used, its cost is incurred. This abstraction, implicit in [2] and made explicit in [13], allows W_P and O_M to be treated as black-box operators and provides a convenient way to capture *time complexity* or, in the quantum query model, *query complexity*. The spatial search algorithm by quantum walk is described by:

ALGORITHM: A quantum circuit $X = X_m X_{m-1} \ldots X_1$, with "wires" (typically two) that carry $\mathbf{C}^S$, and control bits. Each X_i is either a W_P gate or an O_M gate, or a controlled version of one of these. X is applied to the initial state ϕ_0. The cost of the sequence is the sum of the costs of the individual operators. The *observation probability* is the probability that after measuring the final state, ϕ_m, in the standard basis, one of the wires outputs an element of M. If the observation probability is q, one must repeat the procedure $1/\sqrt{q}$ times using amplitude amplification (search version). In the decision version one can distinguish between M and M' if $|X\phi_0 - X'\phi_0| \geq 0.1$, where X arises from O_M and X' from $O_{M'}$.

Key Results

Earlier Results

Spatial search blends Grover's search algorithm [8], which finds a marked element in a database of size N in $\sqrt{N/|M|}$ steps, with quantum walks.

Quantum walks were first introduced by David Meyer and John Watrous to study quantum cellular automata and quantum log-space, respectively. Discrete-time quantum walks were investigated for their own sake by Nayak et al. [3,15] and Aharonov et al. [1] on the infinite line and the N-cycle, respectively. The central issues in the early development of quantum walks included definition of the walk operator, notions of mixing and hitting times, and speedups achievable compared to the classical setting. Exponential quantum speedup of the hitting time between antipodes of the hypercube was shown by Kempe [9], and Childs et al. [6] presented the first oracle problem solvable exponentially faster by a quantum walk based algorithm than by any (not necessarily walk-based) classical algorithm.

The first systematic studies of quantum hitting time on the hypercube and the d-dimensional torus were conducted by Shenvi et al. [17] and Ambainis et al. [4]. Improving upon the Grover search based spatial search algorithm of Aaronson and Ambainis, Ambainis et al. [4] showed that the d-dimensional torus (with $N = n^d$ nodes) can be searched by quantum walk with cost of order $S + \sqrt{N}(U + C)$ and observation probability $\Omega(1/\log N)$ for $d \geq 3$, and with cost of order $S + \sqrt{N \log N}(U + C)$ and observation probability $\Omega(1)$ for $d = 2$. The key difference between these results and those of [6,9] is that the walk is required to start from the uniform state, not from one which is somehow "related" to the state one wishes to hit. Only in the latter case is it possible to achieve an exponential speedup.

The first result that used a quantum walk to solve a natural algorithmic problem, the so-called *element distinctness problem*, was due to Ambainis [2]. Ambainis' algorithm uses a walk W on the *Johnson graph* $J(r, m)$ whose vertices are the r-size subsets of a universe of size m, with two subsets connected iff their symmetric difference has size two. The relevance of this graph follows from a nontrivial algorithmic idea whereby the three different costs (S, U, and C) are balanced in a novel way. In contrast, Grover's algorithm – though it inspired Ambainis' result – has no such option: its setup and update costs are zero in the query model.

Ambainis' main mathematical observation about the walk W on the Johnson graph is that $W^{\sqrt{r}}O_M$ behaves in much the same way as the Grover iteration DO_M, where D is the Grover diffusion operator. Recall that Grover's algorithm applies DO_M repeatedly, sending the uniform starting state ϕ_0 to the state $\phi_{good} = \sum_{x \in M} \sqrt{1/|M|}|x\rangle$ after $t = O(1/\alpha)$ iterations, where $\alpha := 2\sin^{-1}\langle\phi_{\text{good}}|\phi_0\rangle$ is the effective "rotation angle".

What do $W^{\sqrt{r}}$ and D have in common? Ambainis showed that the nontrivial eigenvalues of the matrix $W^{\sqrt{r}}$ in the (finite dimensional) subspace containing the orbit of ϕ_0 are separated away from 1 by a constant ε. Thus, $W^{\sqrt{r}}$ serves as a very good approximate reflection about the axis ϕ_0 – as good as Grover's in this application. This allows one to conclude the following: there exists a $t = O(1/\alpha)$ for which the overlap $\langle\phi_{\text{good}}|(W^{\sqrt{r}}O_M)^t|\phi_0\rangle = \Omega(1)$, so the output is likely in M.

Theorem 1 ([2]) *Let P be the random walk on the Johnson graph $J(r, m)$ with $r = o(m)$. Let M be either the empty set or the set of all r-size subsets containing a fixed subset $x_1, \ldots, x_k$ for constant $k \leq r$. Then there is a quantum algorithm that solves the hitting problem (search version) with cost of order $S + t(\sqrt{r} \cdot U + C)$, where $t = (\frac{m}{r})^{k/2}$. If the costs are $S = r$, $U = O(1)$, and $C = 0$, then the total cost has optimum $O(m^{k/(k+1)})$ at $r = O(m^{k/(k+1)})$.*

General Markov Chains

In [18], Szegedy investigates the hitting time of quantum walks arising from general Markov chains. His definitions (walk operator, hitting time) are abstracted directly from [2] and are consistent with prior literature, although slightly different in presentation.

For a Markov chain P, the (classical) *average hitting time* with respect to M can be expressed in terms of the *leaking walk matrix* P_M, which is obtained from P by deleting all rows and columns indexed by states of M. Let $h(x, M)$ denote the expected time to reach M from x and let $v_1, \ldots, v_{N-|M|}$, $\lambda_1, \ldots, \lambda_{N-|M|}$ be the (normalized) eigenvectors and associated eigenvalues of P_M. Let $d : S \to \mathbf{R}^+$ be a starting distribution and d' its restriction to $S \setminus M$. Then $h := \sum_{x \in S} d(x)h(x, M) = \sum_{k=1}^{N-|M|} \frac{|(v_k, d')|^2}{1-\lambda_k}$. Although the leaking walk matrix P_M is not stochastic, one can consider the *absorbing walk* matrix $P' = \left[\begin{smallmatrix} P_M & 0 \\ P'' & I \end{smallmatrix}\right]$, where P'' is the matrix obtained from P by deleting columns indexed by M and rows indexed by $S \setminus M$.P' behaves similarly to P but is absorbed by the first marked state it hits. Consider the quantization $W_{P'}$ of P' and define the *quantum hitting time*, H, of set M to be the smallest m for which $|W_{P'}^m\phi_0 - \phi_0| \geq 0.1$, where $\phi_0 := \sum_{x \in S} \sqrt{1/N}|x\rangle|p_x\rangle$ (which is stationary for W_P). Note that the construction cost of $W_{P'}$ is proportional to $U + C$.

Why is this definition of quantum hitting time interesting? The classical hitting time measures the number of iterations of the absorbing walk P' required to noticeably skew the uniform starting distribution. Similarly, the quantum hitting time bounds the number of iterations of the following quantum algorithm for detecting whether M is nonempty: At each step, apply operator $W_{P'}$. If M is empty, then $P' = P$ and the starting state is left invariant. If M is nonempty, then the angle between $W_{P'}^t \phi_0$ and $W_P^t \phi_0$ gradually increases. Using an additional *control register* to apply either $W_{P'}$ or W_P with quantum control, the divergence of these two states (should M be nonempty) can be detected. The required number of iterations is exactly H.

It remains to compute H. When P is symmetric and *ergodic*, the expression for the classical hitting time has a quantum analogue [18] ($|M| \leq N/2$ for technical reasons):

$$H \leq \sum_{k=1}^{N-|M|} \frac{v_k^2}{\sqrt{1-\lambda_k}}, \tag{1}$$

where v_k is the sum of the coordinates of v_k divided by $1/\sqrt{N}$. From (1) and the expression for h one can derive an amazing connection between the classical and quantum hitting times:

Theorem 2 ([18]) *Let P be symmetric and ergodic, and let h be the classical hitting time for marked set M and uniform starting distribution. Then the quantum hitting time of M is at most $\sqrt{h}$.*

One can further show:

Theorem 3 ([18]) *If P is state-transitive and $|M| = 1$, then the marked state is observed with probability at least N/h in $O(\sqrt{h})$ steps.*

Theorems 2 and 3 imply most quantum hitting time results of the previous section, *without calculation*, relying only on estimates of the corresponding classical hitting times. Expression (1) is based on a fundamental connection between the eigenvalues and eigenvectors of P and W_P. Notice that R_1 and R_2 are reflections on the subspaces generated by $\{|p_x\rangle \otimes |x\rangle \mid x \in S\}$ and $\{|x\rangle \otimes |p_x\rangle \mid x \in S\}$, respectively. Hence the eigenvalues of $R_1 R_2$ can be expressed in terms of the eigenvalues of the mutual Gram matrix of these systems. This matrix D, the *discriminant matrix* of P, is:

$$D(P) = \sqrt{P \circ P^T} \stackrel{\text{def}}{=} \left(\sqrt{p_{x,y} p_{y,x}}\right)_{x,y \in S}. \tag{2}$$

If P is symmetric then $D(P) = P$, and the formula remains fairly simple even when P is not symmetric. In particular, the absorbing walk P' has discriminant matrix $\left[\begin{smallmatrix} P_M & 0 \\ 0 & I \end{smallmatrix}\right]$. Finally, the relation between $D(P)$ and the spectral decomposition of W_P is given by:

Theorem 4 ([18]) *Let P be an arbitrary Markov chain on a finite state space S and let $\cos\theta_1 \geq \cdots \geq \cos\theta_l$ be those singular values of $D(P)$ lying in the open interval $(0,1)$, with associated singular vector pairs v_j, w_j for $1 \leq j \leq l$. Then the non-trivial eigenvalues of W_P (excluding 1 and -1) and their corresponding eigenvectors are $e^{-2i\theta_j}$, $R_1 w_j - e^{-i\theta_j} R_2 v_j$; $e^{2i\theta_j}$ and $R_j w_j - e^{i\theta_j} R_2 v_j$ for $1 \leq j \leq l$.*

Latest Development

Recently, Magniez et al. [12] have used Szegedy's quantization W_P of an ergodic walk P (rather than its absorbing version P') to obtain an efficient and general implementation of the abstract search algorithm of Ambainis et al. [4].

Theorem 5 ([12]) *Let P be reversible and ergodic with spectral gap $\delta > 0$. Let M have marked probability either zero or $\varepsilon > 0$. Then there is a quantum algorithm solving the hitting problem (search version) with cost $S + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}} U + C\right)$.*

Applications

Element Distinctness

Suppose one is given elements $x_1, \ldots, x_m \in \{1, \ldots, m\}$ and is asked if there exist i, j such that $x_i = x_j$. The classical query complexity of this problem is $\Theta(m)$. Ambainis [2] gave an (optimal) $O(m^{2/3})$ quantum query algorithm using a quantum walk on the Johnson graph of $m^{2/3}$-subsets of $\{1, \ldots, m\}$ with those subsets containing i, j with $x_i = x_j$ marked.

Triangle Finding

Suppose one is given the adjacency matrix A of a graph on n vertices and is required to determine if the graph contains a triangle (i. e., a clique of size 3) using as few queries as possible to the entries of A. The classical query complexity of this problem is $\Theta(n^2)$. Magniez, Santha, and Szegedy [13] gave an $\tilde{O}(n^{1.3})$ algorithm by adapting [2]. This was improved to $O(n^{1.3})$ by Magniez et al. [12].

Matrix Product Verification

Suppose one is given three $n \times n$ matrices A, B, C and is required to determine if $AB \neq C$ (i. e., if their exist i, j such that $\sum_k A_{ik} B_{kj} \neq C_{ij}$) using as few queries as possible to the entries of A, B, and C. This problem has classical

query complexity $\Theta(n^2)$. Buhrman and Spalek [5] gave an $O(n^{5/3})$ quantum query algorithm using [18].

Group Commutativity Testing

Suppose one is presented with a black-box group specified by its k generators and is required to determine if the group commutes using as few queries as possible to the group product operation (i. e., queries of the form "What is the product of elements g and h?"). The classical query complexity is $\Theta(k)$ group operations. Magniez and Nayak [11] gave an (essentially optimal) $\tilde{O}(k^{2/3})$ quantum query algorithm by walking on the product of two graphs whose vertices are (ordered) l-tuples of distinct generators and whose transition probabilities are nonzero only where the l-tuples at two endpoints differ in at most one coordinate.

Open Problems

Many issues regarding quantization of Markov chains remain unresolved, both for the hitting problem and the closely related mixing problem.

Hitting

Can the quadratic quantum hitting time speedup be extended from all symmetric Markov chains to all reversible ones? Can the lower bound of [18] on observation probability be extended beyond the class of state-transitive Markov chains with a unique marked state? What other algorithmic applications of quantum hitting time can be found?

Mixing

Another wide use of Markov chains in classical algorithms is in *mixing*. In particular, *Markov chain Monte Carlo* algorithms work by running an ergodic Markov chain with carefully chosen stationary distribution π until reaching its *mixing time*, at which point the current state is guaranteed to be distributed ε-close to uniform. Such algorithms form the basis of most randomized algorithms for approximating #P-complete problems. Hence, the problem is:

INPUT: Markov chain P on set S, tolerance $\varepsilon>0$.
OUTPUT: A state ε-close to π in total variation distance.

Notions of quantum mixing time were first proposed and analyzed on the line, the cycle, and the hypercube by Nayak et al. [3,15], Aharonov et al. [1], and Moore and Russell [14]. Recent work of Kendon and Tregenna [10] and Richter [16] has investigated the use of decoherence in improving mixing of quantum walks. Two fundamental questions about the quantum mixing time remain open: What is the "most natural" definition? And, when is there a quantum speedup over the classical mixing time?

Cross References

- ▶ Quantum Algorithm for Element Distinctness
- ▶ Quantum Algorithm for Finding Triangles

Recommended Reading

1. Aharonov, D., Ambainis, A., Kempe, J., Vazirani, U.: Quantum walks on graphs. In: Proc. STOC (2001)
2. Ambainis, A.: Quantum walk algorithm for element distinctness. SIAM J. Comput. **37**(1), 210–239 (2007). Preliminary version in Proc. FOCS 2004
3. Ambainis, A., Bach, E., Nayak, A., Vishwanath, A., Watrous, J.: One-dimensional quantum walks. In: Proc. STOC (2001)
4. Ambainis, A., Kempe, J., Rivosh, A.: Coins make quantum walks faster. In: Proc. SODA (2005)
5. Buhrman, H., Spalek, R.: Quantum verification of matrix products. In: Proc. SODA (2006)
6. Childs, A., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., Spielman, D.: Exponential algorithmic speedup by a quantum walk. In: Proc. STOC (2003)
7. Farhi, E., Gutmann, S.: Quantum computation and decision trees. Phys. Rev. A **58** (1998)
8. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proc. STOC (1996)
9. Kempe, J.: Discrete quantum walks hit exponentially faster. In: Proc. RANDOM (2003)
10. Kendon, V., Tregenna, B.: Decoherence can be useful in quantum walks. Phys. Rev. A. **67**, 42–315 (2003)
11. Magniez, F., Nayak, A.: Quantum complexity of testing group commutativity. Algorithmica **48**(3), 221–232 (2007) Preliminary version in Proc. ICALP 2005
12. Magniez, F., Nayak, A., Roland, J., Santha, M.: Search via quantum walk. In: Proc. STOC (2007)
13. Magniez, F., Santha, M., Szegedy, M.: Quantum algorithms for the triangle problem. SIAM J. Comput. **37**(2), 413–424 (2007) Preliminary version in Proc. SODA 2005
14. Moore, C., Russell, A.: Quantum walks on the hypercube. In: Proc. RANDOM (2002)
15. Nayak, A., Vishwanath, A.: Quantum walk on the line. quant-ph/0010117
16. Richter, P.C.: Quantum speedup of classical mixing processes. Phys. Rev. A **76**, 042306 (2007)
17. Shenvi, N., Kempe, J., Whaley, K.B.: A quantum random walk search algorithm. Phys. Rev. A **67**, 52–307 (2003)
18. Szegedy, M.: Quantum speed-up of Markov chain based algorithms. In: Proc. FOCS (2004)

Quantum Algorithm for Checking Matrix Identities

2006; Buhrman, Spalek

ASHWIN NAYAK
Department of Combinatorics and Optimization, University of Waterloo and Perimeter Institute for Theoretical Physics, Waterloo, ON, Canada

Keywords and Synonyms

Matrix product verification

Problem Definition

Let A, B, C be three given matrices of dimension $n \times n$ over a field, where C is claimed to be the matrix product AB. The straightforward method of checking whether $C = AB$ is to multiply the matrices A, B, and compare the entries of the result with those of C. This takes time $O(n^\omega)$, where ω is the "exponent of matrix multiplication". It is evident from the definition of the matrix multiplication operation that $2 \leq \omega \leq 3$. The best known bound on ω is 2.376 [4].

Here, and in the sequel, "time" is taken to mean "number of arithmetic operations" over the field (or other algebraic structure to which the entries of the matrix belong). Similarly, in stating space complexity, the multiplicative factor corresponding to the space required to represent elements of the algebraic structure is suppressed.

Surprisingly, matrix multiplication can be circumvented by using a randomized "fingerprinting" technique due to Freivalds [5], and the matrix product can be checked in time $O(n^2)$ with one-sided bounded probability of error. This algorithm extends, in fact, to matrices over any *integral domain* [3] and the number of random bits used may be reduced to $\log \frac{n}{\epsilon} + O(1)$ for an algorithm that makes one-sided probabilistic error at most ϵ [8]. (All logarithms in this article are taken to base 2.) The fingerprinting technique has found numerous other applications in theoretical computer science (see, for example [10]).

Buhrman and Špalek consider the complexity of checking matrix products on a quantum computer.

Problem 1 (Matrix product verification)
INPUT: *Matrices A, B, C of dimension $n \times n$ over an integral domain.*
OUTPUT: EQUAL *if $C = AB$, and* NOT EQUAL *otherwise.*

They also study the verification problem over the Boolean algebra $\{0, 1\}$ with operations $\{\vee, \wedge\}$, where the fingerprinting technique does not apply.

As an application of their verification algorithms, they consider multiplication of sparse matrices.

Problem 2 (Matrix multiplication)
INPUT: *Matrices A, B of dimension $n \times n$ over an integral domain or the Boolean algebra $\{0, 1\}$.*
OUTPUT: *The matrix product $C = AB$ over the integral domain or the Boolean algebra.*

Key Results

Ambainis, Buhrman, Høyer, Karpinski, and Kurur [2] first studied matrix product verification in the quantum mechanical setting. Using a recursive application of the Grover search algorithm [6], they gave an $O(n^{7/4})$ algorithm for the problem. Buhrman and Špalek improve this runtime by adapting search algorithms based on quantum walk that were recently discovered by Ambainis [1] and Szegedy [11].

Let $W = \{(i, j) | (AB - C)_{i,j} \neq 0\}$ be the set of coordinates where C disagrees with the product AB, and let W' be the largest independent subset of W. (A set of coordinates is said to be *independent* if no row or column occurs more than once in the set.) Define $q(W) = \max\{|W'|, \min\{|W|, \sqrt{n}\}\}$.

Theorem 1 *Consider Problem 1. There is a quantum algorithm that always returns* EQUAL *if $C = AB$, returns* NOT EQUAL *with probability at least 2/3 if $C \neq AB$, and has worst case run-time $O(n^{5/3})$, expected run-time $O(n^{2/3}/q(W)^{1/3})$, and space complexity $O(n^{5/3})$.*

Buhrman and Špalek state their results in terms of "black-box" complexity or "query complexity", where the entries of the input matrices A, B, C are provided by an oracle. The measure of complexity here is the number of oracle calls (queries) made. The query complexity of their quantum algorithm is the same as the run time in the above theorem. They also derive a lower bound on the query complexity of the problem.

Theorem 2 *Any bounded-error quantum algorithm for Problem 1 has query complexity $\Omega(n^{3/2})$.*

When the matrices A, B, C are Boolean, and the product is defined over the operations $\{\vee, \wedge\}$, an optimal algorithm with run-time/query complexity $O(n^{3/2})$ may be derived from an algorithm for AND-OR trees [7]. This has space complexity $O((\log n)^3)$.

All the quantum algorithms may be generalized to handle rectangular matrix product verification, with appropriate modification to the run-time and space complexity.

Applications

Using binary search along with the algorithms in the previous section, the position of a wrong entry in a matrix C (purported to be the product AB) can be located, and then corrected. Buhrman and Špalek use this in an iterative fashion to obtain a matrix multiplication algorithm, starting from the guess $C = 0$. When the product AB is

a sparse matrix, this leads to a quantum matrix multiplication scheme that is, for some parameters, faster than known classical schemes.

Theorem 3 *For any $n \times n$ matrices $A\,B$ over an integral domain, the matrix product $C = AB$ can be computed by a quantum algorithm with polynomially small error probability in expected time*

$$O(1) \cdot \begin{cases} n \log n \cdot n^{2/3} w^{2/3} & \text{when } 1 \le w \le \sqrt{n}\,, \\ n \log n \cdot \sqrt{n}\, w & \text{when } \sqrt{n} \le w \le n\,, \text{ and} \\ n \log n \cdot n\sqrt{w} & \text{when } n \le w \le n^2\,, \end{cases}$$

where w is the number of non-zero entries in C.

A detailed comparison of this quantum algorithm with classical ones may be found in [3].

A subsequent quantum walk based algorithm due to Magniez, Nayak, Roland, and Santha [9] finds a wrong entry in the same run-time as in Theorem 1, without the need for binary search. This improves the run-time of the quantum algorithm for matrix multiplication described above slightly.

Since Boolean matrix products can be verified faster, boolean matrix products can be computed in expected time $O(n^{3/2}w)$, where w is the number of '1' entries in the product.

All matrix product algorithms presented here may be used for multiplication of rectangular matrices as well, with appropriate modifications.

Cross References

- ► Quantization of Markov Chains
- ► Quantum Algorithm for Element Distinctness

Recommended Reading

1. Ambainis, A.: Quantum walk algorithm for Element Distinctness. In: Proceedings of the 45th Symposium on Foundations of Computer Science, pp. 22–31, Rome, Italy, 17–19 October 2004
2. Ambainis, A., Buhrman, H., Høyer, P., Karpinski, M., Kurur, P.: Quantum matrix verification. Unpublished manuscript (2002)
3. Buhrman, H., Špalek, R.: Quantum verification of matrix products. In: Proceedings of 17th ACM-SIAM Symposium on Discrete Algorithms, pp. 880–889, Miami, FL, USA, 22–26 January 2006
4. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. J. Symb. Comput. **9**(3), 251–280 (1990)
5. Freivalds, R.: Fast probabilistic algorithms. In: Proceedings of the 8th Symposium on Mathematical Foundations of Computer Science, pp. 57–69, Olomouc, Czechoslovakia, 3–7 September 1979
6. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th ACM Symposium on the Theory of Computing, pp. 212–219, Philadelphia, PA, USA, 22–24 May 1996
7. Høyer, P., Mosca, M., de Wolf, R.: Quantum search on bounded-error inputs. In: Proceedings of the 30th International Colloquium on Automata, Languages and Programming. Lecture Notes in Computer Science, vol. 2719, pp. 291–299, Eindhoven, The Netherlands, 30 June – 4 July 2003
8. Kimbrel, T., Sinha, R.K.: A probabilistic algorithm for verifying matrix products using $O(n^2)$ time and $\log_2 n + O(1)$ random bits. Inf. Proc. Lett. **45**(2), 107–110 (1993)
9. Magniez, F., Nayak, A., Roland, J., Santha, M.: Search via quantum walk. In: Proceeding of the 39th ACM Symposium on Theory of Computing, pp. 575–584, San Diego, CA, USA, 11–13 June 2007 (2007)
10. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press, Cambridge (1995)
11. Szegedy, M.: Quantum speed-up of Markov chain based algorithms. In: Proceedings of the 45th IEEE Symposium on Foundations of Computer Science, pp. 32–41, Rome, Italy, 17–19 October 2004 (2004)

Quantum Algorithm for the Collision Problem

1998; Brassard, Hoyer, Tapp

Alain Tapp
DIRO, University of Montréal, Montreal, QC, Canada

Problem Definition

A function F is said to be *r-to-one* if every element in its image has exactly r distinct preimages.
Input: an r-to-one function F.
Output: x_1 and x_2 such that $F(x_1) = F(x_2)$.

Key Results

The algorithm presented here finds collisions in arbitrary r-to-one functions F after only $O(\sqrt[3]{N/r})$ expected evaluations of F. The algorithm uses the function as a black box, that is, the only thing the algorithm requires is the capacity to evaluate the function. Again assuming the function is given by a black box, the algorithm is optimal [1] and it is more efficient than the best *possible* classical algorithm, which has query complexity $\Omega(\sqrt{N/r})$. The result is stated precisely in the following theorem and corollary.

Theorem 1 *Given an r-to-one function $F: X \to Y$ with $r \ge 2$ and an integer $1 \le k \le N = |X|$, algorithm* **Collision**(F, k) *returns a collision after an expected number of $O(k + \sqrt{N/(rk)})$ evaluations of F and uses space $\Theta(k)$. In particular, when $k = \sqrt[3]{N/r}$ then* **Collision**(F, k) *uses an expected number of $O(\sqrt[3]{N/r})$ evaluations of F and space $\Theta(\sqrt[3]{N/r})$.*

Corollary 2 *There exists a quantum algorithm that can find a collision in an arbitrary r-to-one function* $F: X \to Y$, *for any* $r \geq 2$, *using space S and an expected number of* $O(T)$ *evaluations of F for every* $1 \leq S \leq T$ *subject to*

$$ST^2 \geq |F(X)|,$$

where $F(X)$ *denotes the image of F.*

The algorithm uses as a procedure a version of Grover's search algorithm. Given a function H with domain size n and a target y, **Grover**(H, y) returns an x such that $H(x) = y$ in expected $O(\sqrt{n})$ evaluations of H.

Collision(F,k)

1. Pick an arbitrary subset $K \subseteq X$ of cardinality k. Construct a table L of size k where each item in L holds a distinct pair $(x, F(x))$ with $x \in K$.
2. Sort L according to the second entry in each item of L.
3. Check if L contains a collision, that is, check if there exist distinct elements $(x_0, F(x_0)), (x_1, F(x_1)) \in L$ for which $F(x_0) = F(x_1)$. If so, go to step 6.
4. Compute $x_1 =$ **Grover**$(H, 1)$, where $H : X \to \{0, 1\}$ denotes the function defined by $H(x) = 1$ if and only if there exists $x_0 \in K$ so that $(x_0, F(x)) \in L$ but $x \neq x_0$. (Note that x_0 is unique if it exists since we already checked that there are no collisions in L.)
5. Find $(x_0, F(x_1)) \in L$.
6. Output the collision $\{x_0, x_1\}$.

Applications

This problem is of particular interest for cryptology because some functions known as *hash functions* are used in various cryptographic protocols. The security of these protocols crucially depends on the presumed difficulty of finding collisions in such functions.

Cross References

► Greedy Set-Cover Algorithms

Recommended Reading

1. Aaronson, S., Shi, Y.: Quantum Lower Bounds for the Collision and the Element Distinctness Problems. J. ACM **51**(4), 595–605 (2004)
2. Boyer, M., Brassard, G., Høyer, P., Tapp A.: Tight bounds on quantum searching. Fortschr. Phys. **46**(4–5), 493–505 (1998)
3. Brassard, G., Høyer, P., Mosca, M., Tapp A.: Quantum Amplitude Amplification and Estimation. In: Lomonaco, S.J. (ed.) Quantum Computation & Quantum Information Science, AMS Contemporary Mathematics Series Millennium Volume, vol. 305, pp. 53–74. American Mathematical Society, Providence (2002)
4. Brassard, G., Høyer, P., Tapp, A.: Quantum Algorithm for the Collision Problem. 3rd Latin American Theoretical Informatics Symposium (LATIN'98). LNCS, vol. 1380, pp. 163–169. Springer (1998)
5. Carter, J.L., Wegman, M.N.: Universal classes of hash functions. J. Comput. Syst. Sci. **18**(2), 143–154 (1979)
6. Grover, L.K.: A fast quantum mechanical algorithm for database search. Proceedings of the 28th Annual ACM Symposium on Theory of Computing, pp. 212–219. ACM (1996)
7. Stinson, D.R.: Cryptography: Theory and Practice, CRC Press, Inc (1995)
8. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)

Quantum Algorithm for the Discrete Logarithm Problem

1994; Shor

Pranab Sen
School of Technology and Computer Science,
Tata Institute of Fundamental Research,
Mumbai, India

Keywords and Synonyms

Logarithms in groups

Problem Definition

Given positive real numbers $a \neq 1, b$, the logarithm of b to base a is the unique real number s such that $b = a^s$. The notion of the *discrete logarithm* is an extension of this concept to general groups.

Problem 1 (Discrete logarithm)
Input: *Group* G, $a, b \in G$ *such that* $b = a^s$ *for some positive integer s.*
Output: *The smallest positive integer s satisfying* $b = a^s$, *also known as the discrete logarithm of b to the base a in G.*

The usual logarithm corresponds to the discrete logarithm problem over the group of positive reals under multiplication. The most common case of the discrete logarithm problem is when the group $G = \mathbb{Z}_p^*$, the multiplicative group of integers between 1 and $p - 1$ modulo p, where p is a prime. Another important case is when the group G is the group of points of an elliptic curve over a finite field.

Key Results

The discrete logarithm problem in $\mathbb{Z}_p^*$, where p is a prime, as well as in the group of points of an elliptic curve over a finite field is believed to be intractable for ran-

domized classical computers. That is any, possibly randomized, algorithm for the problem running on a classical computer will take time that is superpolynomial in the number of bits required to describe an input to the problem. The best classical algorithm for finding discrete logarithms in $\mathbb{Z}_p^*$, where p is a prime, is Gordon's [4] adaptation of the number field sieve which runs in time $\exp(O((\log p)^{1/3}(\log\log p)^{2/3}))$.

In a breakthrough result, Shor [9] gave an efficient quantum algorithm for the discrete logarithm problem in any group G; his algorithm runs in time that is polynomial in the bit size of the input.

Result 1 ([9]) *There is a quantum algorithm solving the discrete logarithm problem in any group G on n-bit inputs in time $O(n^3)$ with probability at least 3/4.*

Description of the Discrete Logarithm Algorithm

Shor's algorithm [9] for the discrete logarithm problem makes essential use of an efficient quantum procedure for implementing a unitary transformation known as the *quantum Fourier transform*. His original algorithm gave an efficient procedure for performing the quantum Fourier transform only over groups of the form $\mathbb{Z}_r$, where r is a "smooth" integer, but nevertheless, he showed that this itself sufficed to solve the discrete logarithm in the general case. In this article, however, a more modern description of Shor's algorithm is given. In particular, a result by Hales and Hallgren [5] is used which shows that the quantum Fourier transform over any finite cyclic group $\mathbb{Z}_r$ can be efficiently approximated to inverse-exponential precision.

A description of the algorithm is given below. A general familiarity with quantum notation on the part of the reader is assumed. A good introduction to quantum computing can be found in the book by Nielsen and Chuang [8]. Let $(G, a, b, \bar{r})$ be an instance of the discrete logarithm problem, where $\bar{r}$ is a supplied upper bound on the order of a in G. That is, there exists a positive integer $r \leq \bar{r}$ such that $a^r = 1$. By using an efficient quantum algorithm for order finding also discovered by Shor [9], one can assume that the order of a in G is known, that is, the smallest positive integer r satisfying $a^r = 1$. Shor's order-finding algorithm runs in time $O((\log \bar{r})^3)$. Let $\epsilon > 0$. The discrete logarithm algorithm works on three registers, of which the first two are each t qubits long, where $t := O(\log r + \log(1/\epsilon))$, and the third register is big enough to store an element of G. Let U denote the unitary transformation

$$U\colon |x\rangle|y\rangle|z\rangle \mapsto |x\rangle|y\rangle|z \oplus (b^x a^y)\rangle\ ,$$

where $\oplus$ denotes bitwise XOR. Given access to a reversible oracle for group operations in G, U can be implemented reversibly in time $O(t^3)$ by repeated squaring.

Let $\mathbb{C}[\mathbb{Z}_r]$ denote the Hilbert space of functions from $\mathbb{Z}_r$ to complex numbers. The computational basis of $\mathbb{C}[\mathbb{Z}_r]$ consists of the delta functions $\{|l\rangle\}_{0\leq l\leq r-1}$, where $|\rangle$ is the function that sends the element l to 1 and the other elements of $\mathbb{Z}_r$ to 0. Let $\mathrm{QFT}_{\mathbb{Z}_r}$ denote the *quantum Fourier transform* over the cyclic group $\mathbb{Z}_r$ defined as the following unitary operator on $\mathbb{C}[\mathbb{Z}_r]$:

$$\mathrm{QFT}_{\mathbb{Z}_r}\colon |x\rangle \mapsto r^{-1/2} \sum_{y\in\mathbb{Z}_r} \mathrm{e}^{-2\pi i x y/r}|y\rangle\ .$$

It can be implemented in quantum time $O(t\log(t/\epsilon) + \log^2(1/\epsilon))$ up to an error of ϵ using one t-qubit register [5]. Note that for any $k \in \mathbb{Z}_r$, $\mathrm{QFT}_{\mathbb{Z}_r}$ transforms the state $r^{-1/2}\sum_{x\in\mathbb{Z}_r} \mathrm{e}^{2\pi i k x/r}|x\rangle$ to the state $|k\rangle$. For any integer $l, 0 \leq l \leq r-1$, define

$$|\hat{l}\rangle := r^{-1/2} \sum_{k=0}^{r-1} \mathrm{e}^{-2\pi i l k/r}|a^k\rangle\ . \tag{1}$$

Observe that $\{|\hat{l}\rangle\}_{0\leq l\leq r-1}$ forms an orthonormal basis of $\mathbb{C}[\langle a\rangle]$, where $\langle a\rangle$ is the subgroup generated by a in G and is isomorphic to $\mathbb{Z}_r$, and $\mathbb{C}[\langle a\rangle]$ denotes the Hilbert space of functions from $\langle a\rangle$ to complex numbers.

Algorithm 1 (Discrete logarithm)
Input: Elements $a, b \in G$, a quantum circuit for U, the order r of a in G.
Output: With constant probability, the discrete logarithm s of b to the base a in G.
Runtime: A total of $O(t^3)$ basic gate operations, including four invocations of $\mathrm{QFT}_{\mathbb{Z}_r}$ and one of U.

PROCEDURE:

1. Repeat Steps (a)–(e) twice, obtaining $(sl_1 \bmod r, l_1)$ and $(sl_2 \bmod r, l_2)$.

   (*a*) $|0\rangle|0\rangle|0\rangle$
   Initialization;

   (*b*) $\mapsto r^{-1} \sum_{x,y\in\mathbb{Z}_r} |x\rangle|y\rangle|0\rangle$
   Apply$\mathrm{QFT}_{\mathbb{Z}_r}$to the first two registers;

   (*c*) $\mapsto r^{-1} \sum_{x,y\in\mathbb{Z}_r} |x\rangle|y\rangle|b^x a^y\rangle$
   Apply U

(*d*) $\mapsto r^{-1/2} \sum_{l=0}^{r-1} |sl \bmod r\rangle |l\rangle |\hat{l}\rangle$

Apply $\mathrm{QFT}_{\mathbb{Z}_r}$ to the first two registers;

(*e*) $\mapsto (sl \bmod r, l)$

Measure the first two registers;

2. If l_1 is not coprime to l_2, abort.
3. Let k_1, k_2 be integers such that $k_1 l_1 + k_2 l_2 = 1$. Then, output $s = k_1(sl_1) + k_2(sl_2) \bmod r$.

The working of the algorithm is explained below. From Eq. (1), it is easy to see that

$$|b^x a^y\rangle = r^{-1/2} \sum_{l=0}^{r-1} \mathrm{e}^{2\pi i l(sx+y)/r} |\hat{l}\rangle \,.$$

Thus, the state in Step 1(c) of the above algorithm can be written as

$$\begin{aligned} & r^{-1} \sum_{x,y \in \mathbb{Z}_r} |x\rangle |y\rangle |b^x a^y\rangle \\ &= r^{-3/2} \sum_{l=0}^{r-1} \sum_{x,y \in \mathbb{Z}_r} \mathrm{e}^{2\pi i l(sx+y)/r} |x\rangle |y\rangle |\hat{l}\rangle \\ &= r^{-3/2} \sum_{l=0}^{r-1} \left[\sum_{x \in \mathbb{Z}_r} \mathrm{e}^{2\pi i s l x/r} |x\rangle \right] \cdot \left[\sum_{y \in \mathbb{Z}_r} \mathrm{e}^{2\pi i l y/r} |y\rangle \right] |\hat{l}\rangle . \end{aligned}$$

Now, applying $\mathrm{QFT}_{\mathbb{Z}_r}$ to the first two registers gives the state in Step 1(d) of the above algorithm. Measuring the first two registers gives $(sl \bmod r, l)$ for a uniformly distributed $l, 0 \le l \le r-1$ in Step 1(e). By elementary number theory, it can be shown that if integers l_1, l_2 are uniformly and independently chosen between 0 and $l-1$, they will be coprime with constant probability. In that case, there will be integers k_1, k_2 such that $k_1 l_1 + k_2 l_2 = 1$, leading to the discovery of the discrete logarithm s in Step 3 of the algorithm with constant probability. Since actually only an ϵ-approximate version of $\mathrm{QFT}_{\mathbb{Z}_r}$ can be applied, ϵ can be set to be a sufficiently small constant, and this will still give the correct discrete logarithm s in Step 3 of the algorithm with constant probability. The success probability of Shor's algorithm for the discrete logarithm problem can be boosted to at least 3/4 by repeating it a constant number of times.

Generalizations of the Discrete Logarithm Algorithm

The discrete logarithm problem is a special case of a more general problem called the *hidden subgroup problem* [8]. The ideas behind Shor's algorithm for the discrete logarithm problem can be generalized in order to yield an efficient quantum algorithm for hidden subgroups in Abelian groups (see [1] for a brief sketch). It turns out that finding the discrete logarithm of b to the base a in G reduces to the hidden subgroup problem in the group $\mathbb{Z}_r \times \mathbb{Z}_r$ where r is the order of a in G. Besides the discrete logarithm problem, other cryptographically important functions like integer factoring, finding the order of permutations, as well as finding self-shift-equivalent polynomials over finite fields can be reduced to instances of a hidden subgroup in Abelian groups.

Applications

The assumed intractability of the discrete logarithm problem lies at the heart of several cryptographic algorithms and protocols. The first example of public-key cryptography, namely, the Diffie–Hellman key exchange [2], uses discrete logarithms, usually in the group $\mathbb{Z}_p^*$ for a prime p. The security of the US national standard Digital Signature Algorithm (see [7] for details and more references) depends on the assumed intractability of discrete logarithms in $\mathbb{Z}_p^*$, where p is a prime. The ElGamal public-key cryptosystem [3] and its derivatives use discrete logarithms in appropriately chosen subgroups of $\mathbb{Z}_p^*$, where p is a prime. More recent applications include those in elliptic curve cryptography [6], where the group consists of the group of points of an elliptic curve over a finite field.

Cross References

- ▶ Abelian Hidden Subgroup Problem
- ▶ Quantum Algorithm for Factoring

Recommended Reading

1. Brassard, G., Høyer, P.: An exact quantum polynomial-time algorithm for Simon's problem. In: Proceedings of the 5th Israeli Symposium on Theory of Computing and Systems, pp. 12–23, Ramat-Gan, 17–19 June 1997
2. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theor. **22**, 644–654 (1976)
3. ElGamal, T.: A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theor. **31**(4), 469–472 (1985)
4. Gordon, D.: Discrete logarithms in GF(*p*) using the number field sieve. SIAM J. Discret. Math. **6**(1), 124–139 (1993)
5. Hales, L., Hallgren, S.: An improved quantum Fourier transform algorithm and applications. In: Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, pp. 515–525 (2000)
6. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, New York (2004)

7. Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1997)
8. Nielsen, M., Chuang, I.: Quantum computation and quantum information. Cambridge University Press, Cambridge (2000)
9. Shor, P.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)

Quantum Algorithm for Element Distinctness

2004; Ambainis

Andris Ambainis
Department of Computer Science, University of Latvia, Riga, Latvia

Problem Definition

In the *element distinctness* problem, one is given a list of N elements $x_1, \ldots, x_N \in \{1, \ldots, m\}$ and one must determine if the list contains two equal elements. Access to the list is granted by submitting queries to a black box, and there are two possible types of query.

Value queries. In this type of query, the input to the black box is an index i. The black box outputs x_i as the answer. In the quantum version of this model, the input is a quantum state that may be entangled with the workspace of the algorithm. The joint state of the query, the answer register, and the workspace may be represented as $\sum_{i,y,z} a_{i,y,z}|i, y, z\rangle$, with y being an extra register which will contain the answer to the query and z being the workspace of the algorithm. The black box transforms this state into $\sum_{i,y,z} a_{i,y,z}|i, (y + x_i) \bmod m, z\rangle$. The simplest particular case is if the input to the black box is of the form $\sum_i a_i|i, 0\rangle$. Then, the black box outputs $\sum_i a_i|i, x_i\rangle$. That is, a quantum state consisting of the index i is transformed into a quantum state, each component of which contains x_i together with the corresponding index i.

Comparison queries. In this type of query, the input to the black box consists of two indices i, j. The black box gives one of three possible answers: "$x_i > x_j$", "$x_i < x_j$" or "$x_i = x_j$". In the quantum version, the input is a quantum state consisting of basis states $|i, j, z\rangle$, with i, j being two indices and z being algorithm's workspace.

There are several reasons why the element distinctness problem is interesting to study. First of all, it is related to sorting. Being able to sort $x_1,\ldots,x_N$ enables one to solve the element distinctness by first sorting $x_1,\ldots,x_N$ in increasing order. If there are two equal elements $x_i = x_j$, then they will be next one to another in the sorted list. Therefore, after one has sorted $x_1,\ldots,x_N$, one must only check the sorted list to see if each element is different from the next one. Because of this relation, the element distinctness problem might capture some of the same difficulty as sorting. This has lead to a long line of research on classical lower bounds for the element distinctness problem (cf [6,8,15]. and many other papers).

Second, the central concept of the algorithms for the element distinctness problem is the notion of a collision. This notion can be generalized in different ways, and its generalizations are useful for building quantum algorithms for various graph-theoretic problems (e. g. triangle finding [12]) and matrix problems (e. g. checking matrix identities [7]).

A generalization of element distinctness is element k-distinctness [2], in which one must determine if there exist k different indices $i_1, \ldots, i_k \in \{1, \ldots, N\}$ such that $x_{i_1} = x_{i_2} = \cdots = x_{i_k}$. A further generalization is the k-subset finding problem [9], in which one is given a function $f(y_1, \ldots, y_k)$, and must determine whether there exist $i_1, \ldots, i_k \in \{1, \ldots, N\}$ such that $f(x_{i_1}, x_{i_2}, \ldots, x_{i_k}) = 1$.

Key Results

Element Distinctness: Summary of Results

In the classical (non-quantum) context, the natural solution to the element distinctness problem is done by sorting, as described in the previous section. This uses $O(N)$ value queries (or $O(N \log N)$ comparison queries) and $O(N \log N)$ time. Any classical algorithm requires $\Omega(N)$ value or $\Omega(N \log N)$ comparison queries. If the algorithm is restricted to $o(N)$ space, stronger lower bounds are known [15].

In the quantum context, Buhrman et al. [5] gave the first non-trivial quantum algorithm, using $O(N^{3/4})$ queries. Ambainis [2] then designed a new algorithm, based on a novel idea using quantum walks. Ambainis' algorithm uses $O(N^{2/3})$ queries and is known to be optimal: Aaronson and Shi [1,3,10] have shown that any quantum algorithm for element distinctness must use $\Omega(N^{2/3})$ queries.

For quantum algorithms that are restricted to storing r values x_i (where $r < N^{2/3}$), the best algorithm runs in $O(N/\sqrt{r})$ time.

All of these results are for value queries. They can be adapted to the comparison query model, with an $\log N$ factor increase in the complexity. The time complexity is within a polylogarithmic $O(\log^c N)$ factor of the query complexity, as long as the computational model is sufficiently general [2]. (Random access quantum memory is necessary for implementing any of the two known quantum algorithms.)

Element k-distinctness (and k-subset finding) can be solved with $O(N^{k/(k+1)})$ value queries, using $O(N^{k/(k+1)})$ memory. For the case when the memory is restricted to $r < N^{k/(k+1)}$ values of x_i, it suffices to use $O(r + (N^{k/2})/(r^{(k-1)/2}))$ value queries. The results generalize to comparison queries and time complexity, with a polylogarithmic factor increase in the time complexity (similarly to the element distinctness problem).

Element Distinctness: The Methods

Ambainis' algorithm has the following structure. Its state space is spanned by basic states $|T\rangle$, for all sets of indices $T \subseteq \{1, \ldots, N\}$ with $|T| = r$. The algorithm starts in a uniform superposition of all $|T\rangle$ and repeatedly applies a sequence of two transformations:

1. Conditional phase flip: $|T\rangle \to -|T\rangle$ for all T such that T contains i, j with $x_i = x_j$ and $|T\rangle \to |T\rangle$ for all other T;
2. Quantum walk: perform $O(\sqrt{r})$ steps of quantum walk, as defined in [2]. Each step is a transformation that maps each $|T\rangle$ to a combination of basis states $|T'\rangle$ for T' that differ from T in one element.

The algorithm maintains another quantum register, which stores all the values of x_i, $i \in T$. This register is updated with every step of the quantum walk.

If there are two elements i, j such that $x_i = x_j$, repeating these two transformations $O(N/r)$ times increases the amplitudes of $|T\rangle$ containing i, j. Measuring the state of the algorithm at that point with high probability produces a set T containing i, j. Then, from the set T, we can find i and j.

The basic structure of [2] is similar to Grover's quantum search, but with one substantial difference. In Grover's algorithm, instead of using a quantum walk, one would use Grover's *diffusion transformation*. Implementing Grover's diffusion requires $\Omega(r)$ updates to the register that stores x_i, $i \in T$. In contrast to Grover's diffusion, each step of quantum walk changes T by one element, requiring just one update to the list of x_i, $i \in T$. Thus, $O(\sqrt{r})$ steps of quantum walk can be performed with $O(\sqrt{r})$ updates, quadratically better than Grover's diffusion. And, as shown in [2], the quantum walk provides a sufficiently good approximation of diffusion for the algorithm to work correctly.

This was one of first uses of quantum walks to construct quantum algorithms. Ambainis, Kempe, Rivosh [4] then generalized it to handle searching on grids (described in another entry of this encyclopedia). Their algorithm is based on the same mathematical ideas, but has a slightly different structure. Instead of alternating quantum walk steps with phase flips, it performs a quantum walk with two different walk rules – the normal walk rule and the "perturbed" one. (The normal rule corresponds to a walk without a phase flip and the "perturbed" rule corresponds to a combination of the walk with a phase flip).

1. Initialize x to a state sampled from some initial distribution over the states of P.
2. t_2 times repeat:
   (a) If the current state y is marked, output y and stop;
   (b) Simulate t_1 steps of random walk, starting with the current state y.
3. If the algorithm has not terminated, output "no marked state".

Quantum Algorithm for Element Distinctness, Algorithm 1
Search by a classical random walk

Generalization to Arbitrary Markov Chains

Szegedy [14] and Magniez et al. [13] have generalized the algorithms of [4] and [2], respectively, to speed up the search of an arbitrary Markov chain. The main result of [13] is as follows.

Let P be an irreducible Markov chain with state space X. Assume that some states in the state space of P are *marked*. Our goal is to find a marked state. This can be done by a classical algorithm that runs the Markov chain P until it reaches a marked state (Algorithm 1).

There are 3 costs that contribute to the complexity of Algorithm 1:

1. **Setup cost** S**:** the cost to sample the initial state x from the initial distribution.
2. **Update cost** U**:** the cost to simulate one step of a random walk.
3. **Checking cost** C**:** the cost to check if the current state x is marked.

The overall complexity of the classical algorithm is then $S + t_2(t_1 U + C)$. The required t_1 and t_2 can be calculated from the characteristics of the Markov chain P. Namely,

Proposition 1 ([13]) *Let P be an ergodic, yet symmetric Markov chain. Let $\delta > 0$ be the eigenvalue gap of P and, assume that, whenever the set of marked states M is non-empty, we have $|M|/|X| \geq \epsilon$. Then there are $t_1 = O(1/\delta)$ and $t_2 = O(1/\epsilon)$ such that Algorithm 1 finds a marked element with high probability.*

Thus, the cost of finding a marked element classically is $O(S + 1/\epsilon(1/\delta U + C))$. Magniez et al. [13] construct a quantum algorithm that finds a marked element in

$O(S' + 1/\epsilon(1/\sqrt{\delta}U' + C'))$ steps, with S', U', C' being quantum versions of the setup, update and checking costs (in most of applications, these are of the same order as S, U and C). This achieves a quadratic improvement in the dependence on both ε and δ.

The element distinctness problem is solved by a particular case of this algorithm: a search on the Johnson graph. The Johnson graph is the graph whose vertices v_T correspond to subsets $T \subseteq \{1, \ldots, N\}$ of size $|T| = r$. A vertex v_T is connected to a vertex $v_{T'}$, if the subsets T and T' differ in exactly one element. A vertex v_T is *marked* if T contains indices i, j with $x_i = x_j$.

Consider the following Markov chain on the Johnson graph. The starting probability distribution s is the uniform distribution over the vertices of the Johnson graph. In each step, the Markov chain chooses the next vertex $v_{T'}$ from all vertices that are adjacent to the current vertex v_T, uniformly at random. While running the Markov chain, one maintains a list of all x_i, $i \in T$. This means that the costs of the classical Markov chain are as follows:

- Setup cost of $S = r$ queries (to query all x_i, $i \in T$ where v_T is the starting state).
- Update cost of $U = 1$ query (to query the value x_i, $i \in T' - T$, where v_T is the vertex before the step and $v_{T'}$ is the new vertex).
- Checking cost of $C = 0$ queries (the values x_i, $i \in T$ are already known to the algorithm, no further queries are needed).

The quantum costs S', U', C' are of the same order as S, U, C.

For this Markov chain, it can be shown that the eigenvalue gap is $\delta = O(1/r)$ and the fraction of marked states is $\epsilon = O((r^2)/(N^2))$. Thus, the quantum algorithm runs in time

$$O\left(S' + \frac{1}{\sqrt{\epsilon}}\left(\frac{1}{\sqrt{\delta}}U' + C'\right)\right) = O\left(S' + \sqrt{r}\left(\frac{N}{r}U' + C'\right)\right) = O\left(r + \frac{N}{\sqrt{r}}\right).$$

Applications

Magniez et al. [12] showed how to use the ideas from the element distinctness algorithm as a subroutine to solve the *triangle problem*. In the triangle problem, one is given a graph G on n vertices, accessible by queries to an oracle, and they must determine whether the graph contains a triangle (three vertices v_1, v_2, v_3 with $v_1\, v_2$, $v_1\, v_3$ and $v_2\, v_3$ all being edges). This problem requires $\Omega(n^2)$ queries classically. Magniez et al. [12] showed that it can be solved using $O(n^{1.3} \log^c n)$ quantum queries, with a modification of the element distinctness algorithm as a subroutine. This was then improved to $O(n^{1.3})$ by [13].

The methods of Szegedy [14] and Magniez et al. [13] can be used as subroutines for quantum algorithms for checking matrix identities [7,11].

Open Problems

1. How many queries are necessary to solve the element distinctness problem if the memory accessible to the algorithm is limited to r items, $r < N^{2/3}$? The algorithm of [2] gives $O(N/\sqrt{r})$ queries, and the best lower bound is $\Omega(N^{2/3})$ queries.
2. Consider the following problem:
   Graph collision [12]. The problem is specified by a graph G (which is arbitrary but known in advance) and variables $x_1, \ldots, x_N \in \{0, 1\}$, accessible by queries to an oracle. The task is to determine if G contains an edge uv such that $x_u = x_v = 1$. How many queries are necessary to solve this problem?
   The element distinctness algorithm can be adapted to solve this problem with $O(N^{2/3})$ queries [12], but there is no matching lower bound. Is there a better algorithm? A better algorithm for the graph collision problem would immediately imply a better algorithm for the triangle problem.

Cross References

- ▶ Quantization of Markov Chains
- ▶ Quantum Algorithm for Finding Triangles
- ▶ Quantum Search

Recommended Reading

1. Aaronson, S., Shi, Y.: Quantum lower bounds for the collision and the element distinctness problems. J. ACM **51**(4), 595–605 (2004)
2. Ambainis, A.: Quantum walk algorithm for element distinctness. SIAM J. Comput. **37**(1), 210–239 (2007)
3. Ambainis, A.: Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. Theor. Comput. **1**, 37–46 (2005)
4. Ambainis, A., Kempe, J., Rivosh, A.: In: Proceedings of the ACM/SIAM Symposium on Discrete Algorithms (SODA'06), 2006, pp. 1099–1108
5. Buhrman, H., Durr, C., Heiligman, M., Høyer, P., Magniez, F., Santha, M., de Wolf, R.: Quantum algorithms for element distinctness. SIAM J. Comput. **34**(6), 1324–1330 (2005)
6. Borodin, A., Fischer, M., Kirkpatrick, D., Lynch, N.: A time-space tradeoff for sorting on non-oblivious machines. J. Comput. Syst. Sci. **22**, 351–364 (1981)
7. Buhrman, H., Spalek, R.: Quantum verification of matrix products. In: Proceedings of the ACM/SIAM Symposium on Discrete Algorithms (SODA'06), 2006, pp. 880–889

8. Beame, P., Saks, M., Sun, X., Vee, E.: Time-space trade-off lower bounds for randomized computation of decision problems. J. ACM **50**(2), 154–195 (2003)
9. Childs, A.M., Eisenberg, J.M.: Quantum algorithms for subset finding. Quantum Inf. Comput. **5**, 593 (2005)
10. Kutin, S.: Quantum lower bound for the collision problem with small range. Theor. Comput. **1**, 29–36 (2005)
11. Magniez, F., Nayak, A.: Quantum complexity of testing group commutativity. In: Proceedings of the International Colloquium Automata, Languages and Programming (ICALP'05), 2005, pp. 1312–1324
12. Magniez, F., Santha, M., Szegedy, M.: Quantum algorithms for the triangle problem. SIAM J. Comput. **37**(2), 413–424 (2007)
13. Magniez, F., Nayak, A., Roland, J., Santha, M.: Search by quantum walk. In: Proceedings of the ACM Symposium on the Theory of Computing (STOC'07), 2007, pp. 575–584
14. Szegedy, M.: Quantum speed-up of Markov Chain based algorithms. In: Proceedings of the IEEE Conference on Foundations of Computer Science (FOCS'04), 2004, pp. 32–41
15. Yao, A.: Near-optimal time-space tradeoff for element distinctness. SIAM J. Comput. **23**(5), 966–975 (1994)

Quantum Algorithm for Factoring

1994; Shor

Sean Hallgren
Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, USA

Problem Definition

Every positive integer n has a unique decomposition as a product of primes $n = p_1^{e_1} \cdots p_k^{e_k}$, for primes numbers p_i and positive integer exponents e_i. Computing the decomposition $p_1, e_1, \ldots, p_k, e_k$ from n is the factoring problem.

Factoring has been studied for many hundreds of years and exponential time algorithms for it were found that include trial division, Lehman's method, Pollard's ρ method, and Shank's class group method [1]. With the invention of the RSA public-key cryptosystem in the late 1970s, the problem became practically important and started receiving much more attention. The security of RSA is closely related to the complexity of factoring, and in particular, it is only secure if factoring does not have an efficient algorithm. The first subexponential-time algorithm is due to Morrison and Brillhard [4] using a continued fraction algorithm. This was succeeded by the quadratic sieve method of Pomerance and the elliptic curve method of Lenstra [5]. The Number Field Sieve [2,3], found in 1989, is the best known classical algorithm for factoring and runs in time $\exp(c(\log n)^{1/3}(\log\log n)^{2/3})$ for some constant c. Shor's result is a polynomial-time quantum algorithm for factoring.

Key Results

Theorem 1 ([2,3]) *There is a subexponential-time classical algorithm that factors the integer n in time $\exp(c(\log n)^{1/3}(\log\log n)^{2/3})$.*

Theorem 2 ([6]) *There is a polynomial-time quantum algorithm that factors integers. The algorithm factors n in time $O((\log n)^2(\log n \log n)(\log\log\log n))$ plus polynomial in $\log n$ post-processing which can be done classically.*

Applications

Computationally hard number theoretic problems are useful for public key cryptosystems. The RSA public-key cryptosystem, as well as others, require that factoring not have an efficient algorithm. The best known classical algorithms for factoring can help determine how secure the cryptosystem is and what key sizes to choose. Shor's quantum algorithm for factoring can break these systems in polynomial-time using a quantum computer.

Open Problems

Whether there is a polynomial-time classical algorithm for factoring is open. There are problems which are harder than factoring such as finding the unit group of an arbitrary degree number field for which no efficient quantum algorithm has been found yet.

Cross References

- ▶ Quantum Algorithm for the Discrete Logarithm Problem
- ▶ Quantum Algorithms for Class Group of a Number Field
- ▶ Quantum Algorithm for Solving the Pell's Equation

Recommended Reading

1. Cohen, H.: A course in computational algebraic number theory. Graduate Texts in Mathematics, vol. 138. Springer (1993)
2. Lenstra, A., Lenstra, H. (eds.): The Development of the Number Field Sieve. Lecture Notes in Mathematics, vol. 1544. Springer (1993)
3. Lenstra, A.K., Lenstra, H.W. Jr., Manasse, M.S., Pollard, J.M.: The number field sieve. In: Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing, Baltimore, Maryland, 14–16 May 1990, pp. 564–572
4. Morrison, M., Brillhart, J.: A method of factoring and the factorization of F_7
5. Pomerance, C.: Factoring. In: Pomerance, C. (ed.) Cryptology and Computational Number Theory, Proceedings of Symposia in Applied Mathematics, vol. 42, p. 27. American Mathematical Society
6. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**, 1484–1509 (1997)

Quantum Algorithm for Finding Triangles

2005; Magniez, Santha, Szegedy

Peter Richter
Department of Computer Science, Rutgers, The State University of New Jersey, Piscataway, NJ, USA

Keywords and Synonyms

Triangle finding

Problem Definition

A *triangle* is a clique of size three in an undirected graph. Triangle finding is a fundamental computational problem whose time complexity is closely related to that of matrix multiplication. It has been the subject of considerable study recently as a basic search problem whose quantum query complexity is still unclear, in contrast to the unstructured search problem [4,10] and the element distinctness problem [1,3]. This survey concerns quantum query algorithms for triangle finding.

Notation and Constraints

A *quantum query algorithm* $Q_f : |\psi_0\rangle \mapsto |\psi_f\rangle$ computes a property P of a function f by mapping the initial state $|\psi_0\rangle = |0\rangle|0\rangle|0\rangle$ (in which its *query*, *answer*, and *workspace* registers are cleared) to a final state $|\psi_f\rangle = Q_f|\psi_0\rangle$ by applying a sequence $Q_f = U_k O_f U_{k-1} O_f \cdots U_1 O_f U_0$ of unitary operators on the complex vector space spanned by all possible basis states $|x\rangle|a\rangle|z\rangle$. The unitary operators are of two types: *oracle queries* $O_f : |x\rangle|a\rangle|z\rangle \mapsto |x\rangle|a \oplus f(x)\rangle|z\rangle$, which yield information about f, and *non-query steps* U_k, which are independent of f. The *quantum query complexity* of P is the minimum number of oracle queries required by a quantum query algorithm computing P with probability at least 2/3.

Consider the *triangle finding problem* for an unknown (simple, undirected) graph $G \subseteq \{(a,b) : a,b \in [n], a \neq b\}$ on vertices $[n] = \{1,\ldots,n\}$ and $m = |G|$ undirected edges, where $(a,b) = (b,a)$ by convention. The function f to query is the adjacency matrix of G and the property P to be computed is whether or not G contains a triangle.

Problem 1 (Triangle finding)
Input: *The adjacency matrix f of a graph G on n vertices.*
Output: *A triangle with probability $\geq 2/3$ if one exists (search version), or a boolean value indicating whether or not one exists with probability $\geq 2/3$ (decision version).*

A lower bound of $\Omega(n)$ on the quantum query complexity of the triangle finding problem was shown by Buhrman et al. [6]. The trivial upper bound of $O(n^2)$ is attainable by querying every entry of f classically.

Classical Results

The classical randomized query complexity of a problem is defined similarly to the quantum query complexity, only the operators U_k are stochastic rather than unitary; in particular, this means oracle queries can be made according to classical distributions but not quantum superpositions. It is easy to see that the randomized query complexity of the triangle finding problem (search and decision versions) is $\Theta(n^2)$.

Key Results

Improvement of the upper bound on the quantum query complexity of the triangle finding problem has stemmed from two lines of approach: increasingly clever utilization of structure in the search space (combined with standard quantum amplitude amplification) and application of quantum walk search procedures.

An $O(n + \sqrt{nm})$ Algorithm Using Amplitude Amplification

Since there are $\binom{n}{3}$ potential triangles (a,b,c) in G, a trivial application of Grover's quantum search algorithm [10] solves the triangle finding problem with $O(n^{3/2})$ quantum queries. Buhrman et al. [6] improved this upper bound in the special case where G is *sparse* (i. e., $m = o(n^2)$) by the following argument.

Suppose Grover's algorithm is used to find (a) an edge $(a,b) \in G$ among all $\binom{n}{2}$ potential edges, followed by (b) a vertex $c \in [n]$ such that (a,b,c) is a triangle in G. The costs of steps (a) and (b) are $O(\sqrt{n^2/m})$ and $O(\sqrt{n})$ quantum queries, respectively. If G contains a triangle Δ, then step (a) will find an edge (a,b) from Δ with probability $\Omega(1/m)$, and step (b) will find the third vertex c in the triangle $\Delta = (a,b,c)$ with constant probability. Therefore, steps (a) and (b) together find a triangle with probability $\Omega(1/m)$. By repeating the steps $O(\sqrt{m})$ times using amplitude amplification (Brassard et al. [5]), one can find a triangle with probability 2/3. The total cost is $O(\sqrt{m}(\sqrt{n^2/m} + \sqrt{n})) = O(n + \sqrt{nm})$ quantum queries. Summarizing:

Theorem 1 (Buhrman et al. [6]) *Using quantum amplitude amplification, the triangle finding problem can be solved in $O(n + \sqrt{nm})$ quantum queries.*

An $\tilde{O}(n^{10/7})$ Algorithm Using Amplitude Amplification

Let μ^2 be the complete graph on vertices $\mu \subseteq [n]$, $\nu_G(v)$ be the set of vertices adjacent to a vertex v, and $deg_G(v)$ be the degree of v. Note that for any vertex $v \in [n]$, one can either find a triangle in G containing v or verify that $G \subseteq [n]^2 \setminus \nu_G(v)^2$ with $\tilde{O}(n)$ quantum queries and success probability $1 - 1/n^3$, by first computing $\nu_G(v)$ classically and then using Grover's search logarithmically many times to find an edge of G in $\nu_G(v)^2$ with high probability if one exists. Szegedy et al [13,14]. use this observation to design an algorithm for the triangle finding problem that utilizes no quantum procedure other than amplitude amplification (just like the algorithm of Buhrman et al [6].) yet requires only $\tilde{O}(n^{10/7})$ quantum queries.

The algorithm of Szegedy et al. [13,14]. is as follows. First, select $k = \tilde{O}(n^{\epsilon})$ vertices $v_1, \ldots, v_k$ uniformly at random from $[n]$ without replacement and compute each $\nu_G(v_i)$. At a cost of $\tilde{O}(n^{1+\epsilon})$ quantum queries, one can either find a triangle in G containing one of the v_i or conclude with high probability that $G \subseteq G' := [n]^2 \setminus \cup_i \nu_G(v_i)^2$. Suppose the latter. Then it can be shown that with high probability, one can construct a partition (T, E) of G' such that T contains $O(n^{3-\epsilon'})$ triangles and $E \cap G$ has size $O(n^{2-\delta} + n^{2-\epsilon+\delta+\epsilon'})$ in $\tilde{O}(n^{1+\delta+\epsilon'})$ queries (or one will find a triangle in G in the process). Since $G \subseteq G'$, every triangle in G either lies within T or intersects E. In the first case, one will find a triangle in $G \cap T$ in $O(\sqrt{n^{3-\epsilon'}})$ quantum queries by searching G with Grover's algorithm for a triangle in T, which is known from the partitioning procedure. In the second case, one will find a triangle in G with an edge in E in $\tilde{O}(n + \sqrt{n^{3-\min\{\delta, \epsilon-\delta-\epsilon'\}}})$ quantum queries using the algorithm of Buhrman et al.[6] with $m = |G \cap E|$. Thus:

Theorem 2 (Szegedy et al. [13,14]) *Using quantum amplitude amplification, the triangle finding problem can be solved in* $\tilde{O}(n^{1+\epsilon} + n^{1+\delta+\epsilon'} + \sqrt{n^{3-\epsilon'}} + \sqrt{n^{3-\min\{\delta, \epsilon-\delta-\epsilon'\}}})$ *quantum queries.*

Letting $\epsilon = 3/7$ and $\epsilon' = \delta = 1/7$ yields an algorithm using $\tilde{O}(n^{10/7})$ quantum queries.

An $\tilde{O}(n^{13/10})$ Algorithm Using Quantum Walks

A more efficient algorithm for the triangle finding problem was obtained by Magniez et al. [13], using the quantum walk search procedure introduced by Ambainis [3] to obtain an optimal quantum query algorithm for the element distinctness problem.

Given oracle access to a function f defining a relation $C \subseteq [n]^k$ Ambainis' search procedure solves the *k-collision problem*: find a pair $(a_1, \ldots, a_k) \in C$ if one exists. The search procedure operates on three quantum registers $|A\rangle|D(A)\rangle|y\rangle$: the *set* register $|A\rangle$ holds a set $A \subseteq [n]$ of size $|A| = r$, the *data* register $|D(A)\rangle$ holds a data structure $D(A)$, and the *coin* register $|y\rangle$ holds an element $i \notin A$. By checking the data structure $D(A)$ using a quantum query procedure Φ with *checking cost* $c(r)$, one can determine whether or not $A^k \cap C \neq \emptyset$. Suppose $D(A)$ can be constructed from scratch at a *setup cost* $s(r)$ and modified from $D(A)$ to $D(A')$ where $|A \cap A'| = r - 1$ at an *update cost* $u(r)$. Then Ambainis' quantum walk search procedure solves the k-collision problem in $\tilde{O}(s(r) + (\frac{n}{r})^{k/2} \cdot (c(r) + \sqrt{r} \cdot u(r)))$ quantum queries. (For details, see the encyclopedia entry on element distinctness.)

Consider the *graph collision* problem on a graph $G \subseteq [n]^2$, where f defines the binary relation $C \subseteq [n]^2$ satisfying $C(u, u')$ if $f(u) = f(u') = 1$ and $(u, u') \in G$. Ambainis' search procedure solves the graph collision problem in $\tilde{O}(n^{2/3})$ quantum queries, by the following argument. Fix $k = 2$ and $r = n^{2/3}$ in the k-collision algorithm, and for every $U \subseteq [n]$ define $D(U) = \{(v, f(v)) : v \in U\}$ and $\Phi(D(U)) = 1$ if some $u, u' \in U$ satisfies C. Then $s(r) = r$ initial queries $f(v)$ are needed to set up $D(U)$, $u(r) = 1$ new query $f(v)$ is needed to update $D(U)$, and $c(r) = 0$ additional queries $f(v)$ are needed to check $\Phi(D(U))$. Therefore, $\tilde{O}(r + \frac{n}{r}(\sqrt{r})) = \tilde{O}(n^{2/3})$ queries are needed altogether.

Magniez et al. [13] solve the triangle finding problem by reduction to the graph collision problem. Again fix $k = 2$ and $r = n^{2/3}$. Let C be the set of edges contained in at least one triangle. Define $D(U) = G|_U$ and $\Phi(D(U)) = 1$ if some edge in $G|_U$ satisfies C. Then $s(r) = O(r^2)$ initial queries are needed to set up $D(U)$ and $u(r) = r$ new queries are needed to update $D(U)$. It remains to bound the checking cost $c(r)$. For any vertex $v \in [n]$, consider the graph collision oracle f_v on $G|_U$ satisfying $f_v(u) = 1$ if $(u, v) \in G$. An edge of $G|_U$ is a triangle in G if and only if the edge is a solution to the graph collision problem on $G|_U$ for some $v \in [n]$. This problem can be solved for a particular v in $\tilde{O}(r^{2/3})$ queries. Using $\tilde{O}(\sqrt{n})$ steps of amplitude amplification, one can find out if *any* $v \in [n]$ generates an accepting solution to the graph collision problem with high probability. Hence, the checking cost is $c(r) = \tilde{O}(\sqrt{n} \cdot r^{2/3})$ queries, from which it follows that:

Theorem 3 (Magniez et al. [13]) *Using a quantum walk search procedure, the triangle finding problem can be solved in* $\tilde{O}(r^2 + \frac{n}{r}(\sqrt{n} \cdot r^{2/3} + \sqrt{r} \cdot r))$ *quantum queries.*

Letting $r = n^{3/5}$ yields an algorithm using $\tilde{O}(n^{13/10})$ quantum queries.

In recent work Magniez et al. [12], using the quantum walk defined by Szegedy [15], have introduced a new quantum walk search procedure generalizing that of Ambainis [3]. Among the consequences is a quantum walk algorithm for triangle finding in $O(n^{13/10})$ quantum queries.

Applications

Extensions of the quantum walk algorithm for triangle finding have been used to find cliques and other fixed subgraphs in a graph and to decide monotone graph properties with small certificates using fewer quantum queries than previous algorithms.

Finding Cliques, Subgraphs, and Subsets

Ambainis' k-collision algorithm [3] can find a copy of any graph H with $k > 3$ vertices in $\tilde{O}(n^{2-2/(k+1)})$ quantum queries. In the case where H is a k-clique, Childs and Eisenberg [9] gave an $\tilde{O}(n^{2.5-6/(k+2)})$ query algorithm. A simple generalization of the triangle finding quantum walk algorithm of Magniez et al. [13] improves this to $\tilde{O}(n^{2-2/k})$.

Monotone Graph Properties

Recall that a *monotone graph property* is a boolean property of a graph whose value is invariant under permutation of the vertex labels and monotone under any sequence of edge deletions. Examples of monotone graph properties are connectedness, planarity, and triangle-freeness. A *1-certificate* is a minimal subset of edge queries proving that a property holds (e. g., three edges suffice to prove that a graph contains a triangle). Magniez et al. [13] show that their quantum walk algorithm for the triangle finding problem can be generalized to an $\tilde{O}(n^{2-2/k})$ quantum query algorithm deciding any monotone graph property with 1-certificates of size at most $k > 3$ vertices. The best known lower bound is $\Omega(n)$.

Open Problems

The most obvious remaining open problem is to resolve the quantum query complexity of the triangle finding problem; again, the best upper and lower bounds currently known are $O(n^{13/10})$ and $\Omega(n)$. Beyond this, there are the following open problems:

Quantum Query Complexity of Monotone Graph Properties

The best known lower bound for the quantum query complexity of (nontrivial) monotone graph properties is $\Omega(n^{2/3}\log^{1/6} n)$, observed by Andrew Yao to follow from the classical randomized lower bound $\Omega(n^{4/3}\log^{1/3} n)$ of Chakrabarti and Khot [8] and the quantum adversary technique of Ambainis [2]. Is an improvement to $\Omega(n)$ possible? If so, this would be tight, since one can determine whether the edge set of a graph is nonempty in $O(n)$ quantum queries using Grover's algorithm.

New Quantum Walk Algorithms

Quantum walks have been successfully applied in designing more efficient quantum search algorithms for several problems, including element distinctness [3], triangle finding [13], matrix product verification [7], and group commutativity testing [11]. It would be nice to see how far the quantum walk approach can be extended to obtain new and better quantum algorithms for various computational problems.

Cross References

- ► Quantization of Markov Chains
- ► Quantum Algorithm for Element Distinctness

Recommended Reading

1. Aaronson, S., Shi, Y.: Quantum lower bounds for the collision and the element distinctness problems. J. ACM **51**(4), 595–605, (2004), quant-ph/0112086
2. Ambainis, A.: Quantum lower bounds by quantum arguments. J. Comput. Syst. Sci. **64**, 750–767, (2002), quant-ph/0002066
3. Ambainis, A.: Quantum walk algorithm for element distinctness. SIAM J. Comput. **37**(1), 210–239, (2007) Preliminary version in Proc. FOCS, (2004), quant-ph/0311001
4. Bennett, C., Bernstein, E., Brassard, G., Vazirani, U.: Strengths and weaknesses of quantum computing. SIAM J. Comput. **26**(5), 1510–1523, (1997), quant-ph/9701001
5. Brassard, G., Høyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. In: Quantum Computation and Quantum Information: A Millennium Volume, AMS Contemporary Mathematics Series, vol. 305. (2002) quant-ph/0005055
6. Buhrman, H., Dürr, C., Heiligman, M., P.Høyer, Magniez, F., Santha, M., de Wolf, R.: Quantum algorithms for element distinctness. SIAM J. Computing **34**(6), 1324–1330, (2005). Preliminary version in Proc. CCC (2001) quant-ph/0007016
7. Buhrman, H., Spalek, R.: Quantum verification of matrix products. Proc. SODA, (2006) quant-ph/0409035
8. Chakrabarti, A., Khot, S.: Improved lower bounds on the randomized complexity of graph properties. Proc. ICALP (2001)
9. Childs, A., Eisenberg, J.: Quantum algorithms for subset finding. Quantum Inf. Comput. **5**, 593 (2005), quant-ph/0311038
10. Grover, L.: A fast quantum mechanical algorithm for database search. Proc. STOC (1996) quant-ph/9605043
11. Magniez, F., Nayak, A.: Quantum complexity of testing group commutativity. Algorithmica **48**(3), 221–232 (2007) Preliminary version in Proc. ICALP (2005) quant-ph/0506265

12. Magniez, F., Nayak, A., Roland, J., Santha, M.: Search via quantum walk. Proc. STOC (2007) quant-ph/0608026
13. Magniez, F., Santha, M., Szegedy, M.: Quantum algorithms for the triangle problem. SIAM J. Comput. **37**(2), 413–424, (2007). Preliminary version in Proc. SODA (2005) quant-ph/0310134
14. Szegedy, M.: On the quantum query complexity of detecting triangles in graphs. quant-ph/0310107
15. Szegedy, M.: Quantum speed-up of Markov chain based algorithms. Proc. FOCS (2004) quant-ph/0401053

Quantum Algorithm for the Parity Problem

1985; Deutsch

YAOYUN SHI
Department of Electrical Engineering and Computer Science, University of Michigan,
Ann Arbor, MI, USA

Keywords and Synonyms

Parity; Deutsch–Jozsa algorithm; Deutsch algorithm

Problem Definition

The *parity* of n bits $x_0, x_1, \cdots, x_{n-1} \in \{0,1\}$ is

$$x_0 \oplus x_1 \oplus \cdots \oplus x_{n-1} = \sum_{i=0}^{n-1} x_i \mod 2 .$$

As an elementary Boolean function, Parity is important not only as a building block of digital logic, but also for its instrumental roles in several areas such as error-correction, hashing, discrete Fourier analysis, pseudorandomness, communication complexity, and circuit complexity. The feature of Parity that underlies its many applications is its maximum sensitivity to the input: flipping any bit in the input changes the output. The computation of Parity from its input bits is quite straightforward in most computation models. However, two settings deserve attention.

The first is the circuit complexity of Parity when the gates are restricted to AND, OR, and NOT gates. It is known that Parity cannot be computed by such a circuit of a polynomial size and a constant depth, a groundbreaking result proved independently by Furst, Saxe, and Sipser [7], and Ajtai [1], and improved by several subsequent works.

The second, and the focus of this article, is in the decision tree model (also called the query model or the black-box model), where the input bits $x = x_0 x_1 \cdots x_{n-1} \in \{0,1\}^n$ are known to an oracle only, and the algorithm needs to ask questions of the type "x_i =?" to access the input. The complexity is measured by the number of queries. Specifically, a quantum query is the application of the following query gate

$$O_x : |i, b\rangle \mapsto |i, b \oplus x_i\rangle, \quad i \in \{0, \cdots, n-1\}, b \in \{0,1\} .$$

Key Results

Proposition 1 *There is a quantum query algorithm computing the parity of 2 bits with probability 1 using 1 query.*

Proof Denote by $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. The initial state of the algorithm is

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |-\rangle .$$

Apply a query gate, using the first register for the index slot and the second register for the answer slot. The resulting state is

$$\frac{1}{\sqrt{2}}((-1)^{x_0}|0\rangle + (-1)^{x_1}|1\rangle) \otimes |-\rangle .$$

Applying a Hadamard gate $H = |+\rangle\langle 0| + |-\rangle\langle 1|$ on the first register brings the state to

$$(-1)^{x_0}|x_0 + x_1\rangle \otimes |-\rangle .$$

Thus measuring the first register gives $x_0 + x_1$ with certainty. □

Corollary 2 *There is a quantum query algorithm computing the parity of n bits with probability 1 using $\lceil n/2 \rceil$ queries.*

The above quantum upper bound for Parity is tight, even if the algorithm is allowed to err with a probability bounded away from 1/2 [6]. In contrast, any classical randomized algorithm with bounded error probability requires n queries. This follows from the fact that on a random input, any classical algorithm not knowing all the input bits is correct with precisely 1/2 probability.

Applications

The quantum speedup for computing Parity was first observed by Deutsch [4]. His algorithm uses $|0\rangle$ in the answer slot, instead of $|-\rangle$. After one query, the algorithm has 3/4 chance of computing the parity, better than any classical algorithm (1/2 chance). The presented algorithm is actually a special case of the Deutsch–Jozsa Algorithm, which solves the following problem now referred to as the Deutsch–Jozsa Problem.

Problem 1 (Deutsch–Jozsa Problem) *Let $n \geq 1$ be an integer. Given an oracle function $f\colon \{0,1\}^n \to \{0,1\}$ that satisfies either (a) $f(x)$ is constant on all $x \in \{0,1\}^n$, or (b) $|\{x\colon f(x)=1\}| = |\{x\colon f(x)=0\}| = 2^{n-1}$, determine which case it is.*

When $n = 1$, the above problem is precisely Parity of 2 bits. For a general n, the Deutsch–Jozsa Algorithm solves the problem using only once the following query gate

$$O_f\colon |x,b\rangle \mapsto |x, f(x)\oplus b\rangle\,, \quad x\in\{0,1\}^n,\ b\in\{0,1\}\,.$$

The algorithm starts with

$$|0^n\rangle \otimes |-\rangle\,.$$

It applies $H^{\otimes n}$ on the index register (the first n qubits), changing the state to

$$\frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes |-\rangle\,.$$

The oracle gate is then applied, resulting in

$$\frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle \otimes |-\rangle\,.$$

For the second time, $H^{\otimes n}$ is applied on the index register, bringing the state to

$$\sum_{y\in\{0,1\}^n} \left(\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)+x\cdot y} \right) |y\rangle \otimes |-\rangle\,. \qquad (1)$$

Finally, the index register is measured in the computational basis. The Algorithm returns "Case (a)" if 0^n is observed, otherwise returns "Case (b)".

By direct inspection, the amplitude of $|0^n\rangle$ is 1 in Case (a), and 0 in Case (b). Thus the Algorithm is correct with probability 1. It is easy to see that any deterministic algorithm requires $n/2 + 1$ queries in the worst case, thus the Algorithm provides the first exponential quantum versus deterministic speedup.

Note that $O(1)$ expected number of queries are sufficient for randomized algorithms to solve the Deutsch–Jozsa Problem with a constant success probability arbitrarily close to 1. Thus the Deutsch–Jozsa Algorithm does not have much advantage compared with error-bounded randomized algorithms. One might also feel that the saving of one query for computing the parity of 2 bits by Deutsch–Jozsa Algorithm is due to the artificial definition of one quantum query. Thus the significance of the Deutsch–Jozsa Algorithm is not in solving a practical problem, but in its pioneering use of Quantum Fourier Transform (QFT), of which $H^{\otimes n}$ is one, in the pattern

$$\text{QFT} \to \text{Query} \to \text{QFT}\,.$$

The same pattern appears in many subsequent quantum algorithms, including those found by Bernstein and Vazirani [2], Simon [8], Shor.

The Deutsch–Jozsa Algorithm is also referred to as Deutsch Algorithm. The Algorithm as presented above is actually the result of the improvement by Cleve, Ekert, Macchiavello, and Mosca [3] and independently by Tapp (unpublished) on the algorithm in [5].

Cross References

► Greedy Set-Cover Algorithms

Recommended Reading

1. Ajtai, M.: $\sum_1^1$-formulae on finite structures. Ann. Pure Appl. Log. **24**(1), 1–48 (1983)
2. Bernstein, E., Vazirani, U.: Quantum complexity theory. SIAM J. Comput. **26**(5), 1411–1473 (1997)
3. Cleve, R., Ekert, A., Macchiavello, C., Mosca, M.: Quantum algorithms revisited. Proc. Royal Soc. London **A454**, 339–354 (1998)
4. Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. Proc. Royal Soc. London **A400**, 97–117 (1985)
5. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proc. Royal Soc. London **A439**, 553–558 (1992)
6. Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: A limit on the speed of quantum computation in determining parity. Phys. Rev. Lett. **81**, 5442–5444 (1998)
7. Furst, M., Saxe, J., Sipser, M.: Parity, circuits, and the polynomial time hierarchy. Math. Syst. Theor. **17**(1), 13–27 (1984)
8. Simon, D.R.: On the power of quantum computation. SIAM J. Comput. **26**(5), 1474–1483 (1997)

Quantum Algorithms for Class Group of a Number Field

2005; Hallgren

Sean Hallgren
Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, USA

Problem Definition

Associated with each number field is a finite abelian group called the class group. The order of the class group is called the class number. Computing the class number and the structure of the class group of a number field are among the main tasks in computational algebraic number theory [3].

A number field F can be defined as a subfield of the complex numbers $\mathbb{C}$ which is generated over the rational

numbers $\mathbb{Q}$ by an algebraic number, i. e. $F = \mathbb{Q}(\theta)$ where θ is the root of a polynomial with rational coefficients. The ring of integers $\mathcal{O}$ of F is the subset consisting of all elements that are roots of monic polynomials with integer coefficients. The ring $\mathcal{O} \subseteq F$ can be thought of as a generalization of $\mathbb{Z}$, the ring of integers in $\mathbb{Q}$. In particular, one can ask whether $\mathcal{O}$ is a principal ideal domain and whether elements in $\mathcal{O}$ have unique factorization. Another interesting problem is computing the unit group $\mathcal{O}^*$, which is the set of invertible algebraic integers inside F, that is, elements $\alpha \in \mathcal{O}$ such that α^{-1} is also in $\mathcal{O}$.

Ever since the class group was discovered by Gauss in 1798 it has been an interesting object of study. The class group of F is the set of equivalence classes of fractional ideals of F, where two ideals I and J are equivalent if there exists $\alpha \in F^*$ such that $J = \alpha I$. Multiplication of two ideals I and J is defined as the ideal generated by all products ab, where $a \in I$ and $b \in J$. Much is still unknown about number fields, such as whether there exist infinitely many number fields with trivial class group. The question of the class group being trivial is equivalent to asking whether the elements in the ring of integers $\mathcal{O}$ of the number field have unique factorization.

In addition to computing the class number and the structure of the class group, computing the unit group and determining whether given ideals are principal, called the principal ideal problem, are also central problems in computational algebraic number theory.

Key Results

The best known classical algorithms for the class group take subexponential time [1,3]. Assuming the GRH, computing the class group, the unit group, and solving the principal ideal problem are in NP∩CoNP [7].

The following theorems state that the three problems defined above have efficient quantum algorithms [4,6].

Theorem 1 *There is a polynomial-time quantum algorithm that computes the unit group of a constant degree number field.*

Theorem 2 *There is a polynomial-time quantum algorithm that solves the principal ideal problem in constant degree number fields.*

Theorem 3 *The class group and class number of a constant degree number field can be computed in quantum polynomial-time assuming the GRH.*

Computing the class group means computing the structure of a finite abelian group given a set of generators for it. When it is possible to efficiently multiply group elements and efficiently compute unique representations of each group element, then this problem reduces to the standard hidden subgroup problem over the integers, and therefore has an efficient quantum algorithm. Ideal multiplication is efficient in number fields. For imaginary number fields, there are efficient classical algorithms for computing group elements with a unique representation, and therefore there is an efficient quantum algorithm for computing the class group.

For real number fields, there is no known way to efficiently compute unique representations of class group elements. As a result, the classical algorithms typically compute the unit group and class group at the same time. A quantum algorithm [4] is able to efficiently compute the unit group of a number field, and then use the principal ideal algorithm to compute a unique quantum representation of each class group element. Then the standard quantum algorithm can be applied to compute the class group structure and class number.

Applications

There are factoring algorithms based on computing the class group of an imaginary number fields. One is exponential time and the other is subexponential-time [3].

Computationally hard number theoretic problems are useful for public key cryptosystems. Pell's equation reduces to the principal ideal problem, which forms the basis of the Buchmann-Williams key-exchange protocol [2]. Identification schemes have also been based on this problem by Hamdy and Maurer [5]. The classical exponential-time algorithms help determine which parameters to choose for the cryptosystem. Factoring reduces to Pell's equation and the best known algorithm for it is exponentially slower than the best factoring algorithm. Systems based on these harder problems were proposed as alternatives in case factoring turns out to be polynomial-time solvable. The efficient quantum algorithms can break these cryptosystems.

Open Problems

It remains open whether these problems can be solved in arbitrary degree number fields. The solution for the unit group can be thought of in terms of the hidden subgroup problem. That is, there exists a function on $\mathbb{R}^c$ which is constant on values that differ by an element of the unit lattice, and is one-to-one within the fundamental parallelepiped. However, this function cannot be evaluated efficiently since it has an uncountable domain, and instead an efficiently computable approximation must be used. To evaluate this discrete version of the function, a classical algorithm is used to compute reduced ideals near a given

point in $\mathbb{R}^c$. This algorithm is only polynomial-time for constant degree number fields as it computes the shortest vector in a lattice. Such an algorithm can be used to set up a superposition over points approximating the points in the a coset of the unit lattice. After setting up the superposition, it must be shown that Fourier sampling, i. e. computing the Fourier transform and measuring, suffices to compute the lattice.

Cross References

- ▶ Quantum Algorithm for Factoring
- ▶ Quantum Algorithm for Solving the Pell's Equation

Recommended Reading

1. Buchmann, J.: A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. In: Goldstein, C. (ed.) Séminaire de Théorie des Nombres, Paris 1988–1989, Progress in Mathematics, vol. 91, pp. 27–41. Birkhäuser (1990)
2. Buchmann, J.A., Williams, H.C.: A key exchange system based on real quadratic fields (extended abstract). In: Brassard, G. (ed.) Advances in Cryptology–CRYPTO '89. Lecture Notes in Computer Science, vol. 435, 20–24 Aug 1989, pp. 335–343. Springer (1990)
3. Cohen, H., A course in computational algebraic number theory, vol. 138 of Graduate Texts in Mathematics. Springer (1993)
4. Hallgren, S.: Fast quantum algorithms for computing the unit group and class group of a number field. In: Proceedings of the 37th ACM Symposium on Theory of Computing. (2005)
5. Hamdy, S., Maurer, M.: Feige-fiat-shamir identification based on real quadratic fields, Tech. Report TI-23/99. Technische Universität Darmstadt, Fachbereich Informatik. http://www.informatik.tu-darmstadt.de/TI/Veroeffentlichung/TR/ (1999)
6. Schmidt, A., Vollmer, U.: Polynomial time quantum algorithm for the computation of the unit group of a number field. In: Proceedings of the 37th ACM Symposium on Theory of Computing. (2005)
7. Thiel, C.: On the complexity of some problems in algorithmic algebraic number theory, Ph. D. thesis. Universität des Saarlandes, Saarbrücken, Germany (1995)

Quantum Algorithm for Search on Grids

2005; Ambainis, Kempe, Rivosh

Andris Ambainis
Department of Computer Science, University of Latvia, Riga, Latvia

Keywords and Synonyms

Spatial search

Problem Definition

Consider an $\sqrt{N} \times \sqrt{N}$ grid, with each location storing a bit that is 0 or 1. The locations on the grid are indexed by (i,j), where $i, j \in \{0, 1, \ldots, \sqrt{N}-1\}$. $a_{i,j}$ denotes the value stored at the location (i,j).

The task is to find a location storing $a_{i,j} = 1$. This problem is as an abstract model for search in a two-dimensional database, with each location storing a variable $x_{i,j}$ with more than two values. The goal is to find $x_{i,j}$ that satisfies certain constraints. One can then define new variables $a_{i,j}$ with $a_{i,j} = 1$ if $x_{i,j}$ satisfies the constraints and search for i, j satisfying $a_{i,j} = 1$.

The grid is searched by a "robot", which, at any moment of time is at one location i, j. In one time unit, the robot can either examine the current location or move one step in one of four directions (left, right, up or down).

In a probabilistic version of this model, the robot is probabilistic. It makes its decisions (querying the current location or moving) randomly according to pre-specified probability distributions. At any moment of time, such robot a is at a probability distribution over the locations of the grid. In the quantum case, one has a "quantum robot" [4] which can be in a quantum superposition of locations (i,j) and is allowed to perform transformations that move it at most one step at a time.

There are several ways to make this model of a "quantum robot" precise [1] and they all lead to similar results.

The simplest to define is the Z-local model of [1]. In this model, the robot's state space is spanned by states $|i, j, a\rangle$ with i, j representing the current location and a being the internal memory of the robot. The robot's state $|\psi\rangle$ can be any quantum superposition of those: $|\psi\rangle = \sum_{i,j,a} \alpha_{i,j,a} |i, j, a\rangle$, where $\alpha_{i,j,a}$ are complex numbers such that $\sum_{i,j,a} |\alpha_{i,j,a}|^2 = 1$. In one step, the robot can either perform a query of the value at the current location or a Z-local transformation.

A query is a transformation that leaves i, j parts of a state $|i, j, a\rangle$ unchanged and modifies the a part in a way that depends only on the value $a_{i,j}$. A Z-local transformation is a transformation that maps any state $|i, j, a\rangle$ to a superposition that involves only states with robot being either at the same location or at one of 4 adjacent locations ($|i, j, b\rangle$, $|i-1, j, b\rangle$, $|i+1, j, b\rangle$, $|i, j-1, b\rangle$ or $|i, j+1, b\rangle$ where the content of the robot's memory b is arbitrary).

The problem generalizes naturally to d-dimensional grid of size $N^{1/d} \times N^{1/d} \times \cdots \times N^{1/d}$, with robot being allowed to query or move one step in one of d directions in one unit of time.

Key Results

This problem was first studied by Benioff [4] who considered the use of the usual quantum search algorithm,

by Grover [8] in this setting. Grover's algorithm allows to search a collection of N items $a_{i,j}$ with $O(\sqrt{N})$ queries. However, it does not respect the structure of a grid. Between any two queries it performs a transformation that may require the robot to move from any location (i,j) to any other location (i',j'). In the robot model, where the robot in only allowed to move one step in one time unit, such transformation requires $O(\sqrt{N})$ steps to perform. Implementing Grover's algorithm, which requires $O(\sqrt{N})$ such transformations, therefore, takes $O(\sqrt{N})\times O(\sqrt{N}) = O(N)$ time, providing no advantage over the naive classical algorithm.

The first algorithm improving over the naive use of Grover's search was proposed by Aaronson and Ambainis [1] who achieved the following results:

- Search on $\sqrt{N}\times\sqrt{N}$ grid, if it is known that the grid contains exactly one $a_{i,j}=1$ in $O(\sqrt{N}\log^{3/2} N)$ steps.
- Search on $\sqrt{N}\times\sqrt{N}$ grid, if the grid may contain an arbitrary number of $a_{i,j}=1$ in $O(\sqrt{N}\log^{5/2} N)$ steps.
- Search on $N^{1/d}\times N^{1/d}\times\cdots\times N^{1/d}$ grid, for $d\geq 3$, in $O(\sqrt{N})$ steps.

They also considered a generalization of the problem, search on a graph G, in which the robot moves on the vertices v of the graph G and searches for a variable $a_v=1$. In one step, the robot can examine the variable a_v corresponding to the current vertex v or move to another vertex w adjacent to v. Aaronson and Ambainis [1] gave an algorithm for searching an arbitrary graph with grid-like expansion properties in $O(N^{1/2+o(1)})$ steps. The main technique in those algorithms was the use of Grover's search and its generalization, amplitude amplification [5], in combination with "divide-and-conquer" methods recursively breaking up a grid into smaller parts.

The next algorithms were based on quantum walks [3,6,7]. Ambainis, Kempe and Rivosh [3] presented an algorithm, based on a discrete time quantum walk, which searches the two-dimensional $\sqrt{N}\times\sqrt{N}$ in $O(\sqrt{N}\log N)$ steps, if the grid is known to contain exactly one $a_{i,j}=1$ and in $O(\sqrt{N}\log^2 N)$ steps in the general case. Childs and Goldstone [7] achieved a similar performance, using continuous time quantum walk. Curiously, it turned out that the performance of the walk crucially depended on the particular choice of the quantum walk, both in the discrete and continuous time and some very natural choices of quantum walk (e. g. one in [6]) failed.

Besides providing an almost optimal quantum speedup, the quantum walk algorithms also have an additional advantage: their simplicity. The discrete quantum walk algorithm of [3] uses just two bits of quantum memory. It's basis states are $|i,j,d\rangle$, where (i,j) is a location on the grid and d is one of 4 directions: $\leftarrow$, $\rightarrow$, $\uparrow$ and $\downarrow$. The basic algorithm consists of the following simple steps:

1. Generate the state $\sum_{i,j,d}\frac{1}{2\sqrt{N}}|i,j,d\rangle$.
2. $O(\sqrt{N\log N})$ times repeat
   (a) Perform the transformation

$$C_0=\begin{pmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

   on the states $|i,j,\leftarrow\rangle$, $|i,j,\rightarrow\rangle$, $|i,j,\uparrow\rangle$, $|i,j,\downarrow\rangle$, if $a_{i,j}=0$ and the transformation $C_1=-I$ on the same four states if $a_{i,j}=1$.
   (b) Move one step according to the direction register and reverse the direction:

$$\begin{aligned} |i,j,\rightarrow\rangle &\rightarrow |i+1,j,\leftarrow\rangle\,, \\ |i,j,\leftarrow\rangle &\rightarrow |i-1,j,\rightarrow\rangle\,, \\ |i,j,\uparrow\rangle &\rightarrow |i,j-1,\downarrow\rangle\,, \\ |i,j,\downarrow\rangle &\rightarrow |i,j+1,\uparrow\rangle\,. \end{aligned}$$

In case, if $a_{i,j}=1$ for one location (i,j), a significant part of the algorithm's final state will consist of the four states $|i,j,d\rangle$ for the location (i,j) with $a_{i,j}=1$. This can be used to detect the presence of such location. A quantum algorithm for search on a grid can be also derived by designing a classical algorithm that finds $a_{i,j}=1$ by performing a random walk on the grid and then applying Szegedy's general translation of classical random walks to quantum random chains, with a quadratic speedup over the classical random walk algorithm [12]. The resulting algorithm is similar to the algorithm of [3] described above and has the same running time.

For an overview on related quantum algorithms using similar methods, see [2,9].

Applications

Quantum algorithms for spatial search are useful for designing quantum communication protocols for the set disjointness problem. In the set disjointness problem, one has two parties holding inputs $x\in\{0,1\}^N$ and $y\in\{0,1\}^N$ and they have to determine if there is $i\in\{1,\ldots,N\}$ for which $x_i=y_i=1$. (One can think of x and y as representing subsets $X,Y\subseteq\{1,\ldots,N\}$ with $x_i=1(y_i=1)$ if $i\in X(i\in Y)$. Then, determining if $x_i=y_i=1$ for some i is equivalent to determining if $X\cap Y\neq\emptyset$.)

The goal is to solve the problem, communicating as few bits between the two parties as possible. Classically, $\Omega(N)$ bits of communication are required [10]. The optimal quantum protocol [1] uses $O(\sqrt{N})$ quantum bits of

communication and its main idea is to reduce the problem to spatial search. As shown by the $\Omega(\sqrt{N})$ lower bound of [11], this algorithm is optimal.

Cross References

- ▶ Quantization of Markov Chains
- ▶ Quantum Search

Recommended Reading

1. Aaronson, S., Ambainis, A.: Quantum search of spatial regions. In: Proc. 44th Annual IEEE Symp. on Foundations of Computer Science (FOCS), 2003, pp. 200–209
2. Ambainis, A.: Quantum walks and their algorithmic applications. Int. J. Quantum Inf. **1**, 507–518 (2003)
3. Ambainis, A., Kempe, J., Rivosh, A.: Coins make quantum walks faster. In: Proc. of SODA'05, pp 1099–1108
4. Benioff, P.: Space searches with a quantum robot. In: Quantum computation and information (Washington, DC, 2000). Contemp. Math., vol. 305, pp. 1–12. Amer. Math. Soc. Providence, RI (2002)
5. Brassard, G., Høyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. In: Quantum computation and information (Washington, DC, 2000). Contemp. Math., vol. 305, pp. 53–74. American Mathematical Society, Providence, RI (2002)
6. Childs, A.M., Goldstone, J.: Spatial search by quantum walk. Phys. Rev. A **70**, 022314 (2004)
7. Childs, A.M., Goldstone, J.: Spatial search and the Dirac equation. Phys. Rev. A. **70**, 042312 (2004)
8. Grover, L.: A fast quantum mechanical algorithm for database search. In: Proc. 28th STOC, Philadelphia, Pennsylvania, pp 212–219. ACM Press, New York, (1996)
9. Kempe, J.: Quantum random walks – an introductory overview. Contemp. Phys. **44**(4), 302–327 (2003)
10. Razborov, A.: On the Distributional Complexity of Disjointness. Theor. Comput. Sci. **106**(2), 385–390 (1992)
11. Razborov, A.A.: Quantum communication complexity of symmetric predicates. Izvestiya of the Russian Academy of Science, Mathematics, **67**, 145–159 (2002)
12. Szegedy, M.: Quantum speed-up of Markov Chain based algorithms. In: Proceedings of FOCS'04, pp. 32–41

Quantum Algorithm for Solving the Pell's Equation

2002; Hallgren

Sean Hallgren

Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, USA

Problem Definition

Pell's equation is one of the oldest studied problem in number theory. For a positive square-free integer d, Pell's equation is $x^2 - dy^2 = 1$, and the problem is to compute integer solutions x, y of the equation [7,9]. The earliest algorithm for it uses the continued fraction expansion of $\sqrt{d}$ and dates back to 1000 a.d. by Indian mathematicians. Lagrange showed that there are an infinite number of solutions of Pell's equation. All solutions are of the form $x_n + y_n\sqrt{d} = (x_1 + y_1\sqrt{d})^n$, where the smallest solution, (x_1, y_1), is called the fundamental solution. The solution (x_1, y_1) may have exponentially many bits in general in terms of the input size, which is $\log d$, and so cannot be written down in polynomial time. To resolve this difficulty, the computational problem is recast as computing the integer closest to the regulator $R = \ln(x_1 + y_1\sqrt{d})$. In this representation solutions of Pell's equation are positive integer multiples of R.

Solving Pell's equation is a special case of computing the unit group of number field. For a positive non-square integer Δ congruent to 0 or 1 mod 4, $K = \mathbb{Q}(\sqrt{\Delta})$ is a real quadratic number field. Its subring $\mathcal{O} = \mathbb{Z}[\frac{\Delta+\sqrt{\Delta}}{2}] \subseteq Q(\sqrt{\Delta})$ is called the quadratic order of discriminant Δ. The unit group is the set of invertible elements of $\mathcal{O}$. Units have the form $\pm\varepsilon^k$, where $k \in \mathbb{Z}$, for some $\varepsilon > 1$ called the fundamental unit. The fundamental unit ε can have exponentially many bits, so an approximation of the regulator $R = \ln \varepsilon$ is computed. In this representation the unit group consists of integer multiples of R. Given the integer closest to R there are classical polynomial-time algorithms to compute R to any precision. There are also efficient algorithms to test if a given number is a good approximation to an integer multiple of a unit, or to compute the least significant digits of $\varepsilon = e^R$ [1,3].

Two related and potentially more difficult problems are the principal ideal problem and computing the class group of a number field. In the principal ideal problem, a number field and an ideal I of $\mathcal{O}$ are given, and the problem is to decide if the ideal is principal, i. e. whether there exists α such that $I = \alpha\mathcal{O}$. If it is principal, then one can ask for an approximation of $\ln \alpha$. There are efficient classical algorithms to verify that a number is close to $\ln \alpha$ [1,3]. The class group of a number field is the finite abelian group defined by taking the set of fractional ideals modulo the principal fractional ideals. The class number is the size of the class group. Computing the unit group, computing the class group, and solving the principal ideal problems are three of the main problems of computational algebraic number theory [3]. Assuming the GRH, they are in NP∩CoNP [8].

Key Results

The best known classical algorithms for the problems defined in the last section take subexponential time, but there are polynomial-time quantum algorithms for them [4,6].

Theorem 1 *Given a quadratic discriminant Δ, there is a classical algorithm that computes an integer multiple of the regulator to within one. Assuming the GRH, this algorithm computes the regulator to within one and runs in expected time* $\exp(\sqrt{(\log \Delta) \log\log \Delta})^{O(1)}$.

Theorem 2 *There is a polynomial-time quantum algorithm that, given a quadratic discriminant Δ, approximates the regulator to within δ of the associated order $\mathcal{O}$ in time polynomial in* $\log \Delta$ *and* $\log \delta$ *with probability exponentially close to one.*

Corollary 1 *There is a polynomial-time quantum algorithm that solves Pell's equation.*

The quantum algorithm for Pell's equation uses the existence of a periodic function on the reals which has period R and is one-to-one within each period [4,6]. There is a discrete version of this function that can be computed efficiently. This function does not have the same periodic property since it cannot be evaluated at arbitrary real numbers such as R, but it does approximate the situation well enough for the quantum algorithm. In particular, computing the approximate period of this function gives R to the closest integer, or in other words, computes a generator for the unit group.

Theorem 3 *There is a polynomial-time quantum algorithm that solves the principal ideal problem in real quadratic number fields.*

Corollary 2 *There is a polynomial-time quantum algorithm that can break the Buchmann–Williams key-exchange protocol in real quadratic number fields.*

Theorem 4 *The class group and class number of a real quadratic number field can be computed in quantum polynomial-time assuming the GRH.*

Applications

Computationally hard number theoretic problems are useful for public key cryptosystems. There are reductions from factoring to Pell's equation and Pell's equation to the principal ideal problem, but no reductions are known in the opposite direction. The principal ideal problem forms the basis of the Buchmann–Williams key-exchange protocol [2]. Identification schemes based on this problem have been proposed by Hamdy and Maurer [5]. The classical exponential-time algorithms help determine which parameters to choose for the cryptosystem. The best known algorithm for Pell's equation is exponentially slower than the best factoring algorithm. Systems based on these harder problems were proposed as alternatives in case factoring turns out to be polynomial-time solvable. The efficient quantum algorithms can break these cryptosystems.

Open Problems

It remains open whether these problems can be solved in arbitrary degree number fields. The solution for Pell's equation can be thought of in terms of the hidden subgroup problem. That is, there exists a periodic function on the reals which has period $R \in \mathbb{R}$ and is one-to-one within each period. However, this function cannot be evaluated efficiently since it has an uncountable domain, and instead an efficiently computable approximation must be used. To evaluate this discrete version of the function, a classical algorithm is used to compute reduced ideals near a given point in $\mathbb{R}$. This algorithm is only polynomial-time for constant degree number fields as it computes the shortest vector in a lattice. Such an algorithm can be used to set up a superposition over points approximating the points in the a coset of the unit lattice. After setting up the superposition, it must shown Fourier sampling, i. e. computing the Fourier transform and measuring, suffices to compute the lattice.

Cross References

- ▶ Quantum Algorithm for Factoring
- ▶ Quantum Algorithms for Class Group of a Number Field

Recommended Reading

1. Buchmann, J., Thiel, C., Williams, H.C.: Short representation of quadratic integers. In: Bosma, W., van der Poorten A.J. (eds.) Computational Algebra and Number Theory, Sydney 1992. Mathematics and its Applications, vol. 325, pp. 159–185. Kluwer Academic Publishers (1995)
2. Buchmann, J.A., Williams, H.C.: A key exchange system based on real quadratic fields (extended abstract). In: Brassard, G. (ed.) Advances in Cryptology–CRYPTO '89. Lecture Notes in Computer Science, vol. 435, pp. 335–343. Springer 1990, 20–24 Aug (1989)
3. Cohen, H.: A course in computational algebraic number theory, vol. 138 of Graduate Texts in Mathematics. Springer (1993)
4. Hallgren, S.: Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. J. ACM **54**(1), 1–19 (2007)
5. Hamdy, S., Maurer, M.: Feige-fiat-shamir identification based on real quadratic fields, Tech. Report TI-23/99. Technische Universität Darmstadt, Fachbereich Informatik, http://www.informatik.tu-darmstadt.de/TI/Veroeffentlichung/TR/ (1999)
6. Jozsa, R.: Notes on Hallgren's efficient quantum algorithm for solving Pell's equation, tech. report, quant-ph/0302134 (2003)
7. Lenstra Jr, H.W.: Solving the Pell equation. Not. Am. Math. Soc. **49**, 182–192 (2002)

8. Thiel, C.: On the complexity of some problems in algorithmic algebraic number theory, Ph. D. thesis. Universität des Saarlandes, Saarbrücken, Germany (1995)
9. Williams, H.C.: Solving the Pell equation. In: Proc. Millennial Conference on Number Theory, pp. 397–435 (2002)

Quantum Approximation of the Jones Polynomial
2005; Aharonov, Jones, Landau

Zeph Landau
Department of Mathematics, City College of CUNY, New York, NY, USA

Keywords and Synonyms

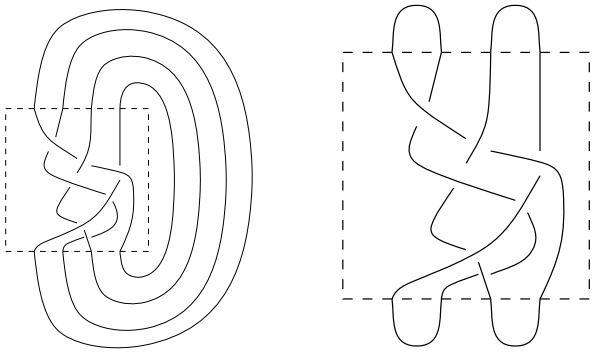
AJL algorithm

Problem Definition

A knot invariant is a function on knots (or links –i. e. circles embedded in R^3) which is invariant under isotopy of the knot, i. e., it does not change under stretching, moving, tangling, etc., (cutting the knot is not allowed). In low dimensional topology, the discovery and use of *knot invariants* is of central importance. In 1984, Jones [12] discovered a new knot invariant, now called the Jones polynomial $V_L(t)$, which is a Laurent polynomial in $\sqrt{t}$ with integer coefficients, and which is an invariant of the link L. In addition to the important role it has played in low dimensional topology, the Jones polynomial has found applications in numerous fields, from DNA recombination [16], to statistical physics [20].

From the moment of the discovery of the Jones polynomial, the question of how hard it is to compute became important. There is a very simple inductive algorithm (essentially due to Conway [5]) to compute it by changing crossings in a link diagram, but, naively applied, this takes exponential time in the number of crossings. It was shown [11] that the computation of $V_L(t)$ is #P-hard for all but a few values of t where $V_L(t)$ has an elementary interpretation. Thus a polynomial time algorithm for computing $V_L(t)$ for any value of t other than those elementary ones is unlikely. Of course, the #P-hardness of the problem does not rule out the possibility of good approximations. Still, the best classical algorithms to approximate the Jones polynomial at all but trivial values are exponential. Simply stated, the problem becomes:

Problem 1 *For what values of t and for what level of approximation can the Jones polynomial $V_L(t)$ be approximated in time polynomial in the number of crossings and links of the link L?*



Quantum Approximation of the Jones Polynomial, Figure 1
The trace closure (*left*) and plat closure (*right*) of the same 4-strand braid

Key Results

As mentioned above, exact computation of the Jones polynomial for most t is #P-hard and the best known classical algorithms to approximate the Jones polynomial are exponential. The key results described here consider the above problem in the context of quantum rather than classical computation.

The results concern the approximation of links that are given as closures of braids. (All links can be described this way.) Briefly, a braid of n strands and m crossings is described pictorially by n strands hanging alongside each other, with m crossings, each of two adjacent strands. A braid B may be "closed" to form a link by tying its ends together in a variety of ways, two of which are the *trace closure* (denoted by B^{tr}) which joins the ith strand from the top right to the ith strand from the bottom right (for each i), and the *plat closure* (denoted by B^{pl}) which is defined only for braids with an even number of strands by connecting pairs of adjacent strands (beginning at the rightmost strand) on both the top and bottom. Examples of the trace and plat closure of the same 4-strand braid are given in Fig. 1.

For such braids, the following results have been shown by Aharonov, Jones, and Landau:

Theorem 2.1 [3] *For a given braid B in B_n with m crossings, and a given integer k, there is a quantum algorithm which is polynomial in n,m,k which with all but exponentially (in n,m,k) small probability, outputs a complex number r with $|r - V_{B^{\mathrm{tr}}}(e^{2\pi i/k})| < \epsilon d^{n-1}$ where $d = 2\cos(\pi/k)$, and ϵ is inverse polynomial in n,k,m.*

Theorem 2.2 [3] *For a given braid B in B_n with m crossings, and a given integer k, there is a quantum algorithm which is polynomial in n,m,k which with all but exponentially (in n,m,k) small probability, outputs a complex*

number r with $|r - V_{B^{pl}}(e^{2\pi i/k})| < \epsilon d^{n/2-1}$ *where* $d = 2\cos(\pi/k)$ *and* ϵ *is inverse polynomial in n,k,m.*

The original connection between quantum computation and the Jones polynomial was made earlier in the series of papers [6,7,8,9]. A model of quantum computation based on Topological Quantum Field Theory (*TQFT*) and Chern–Simons theory was defined in [6,7], and Kitaev, Larsen, Freedman and Wang showed that this model is polynomially equivalent in computational power to the standard quantum computation model in [8,9]. These results, combined with a deep connection between *TQFT* and the value of the Jones polynomial at particular roots of unity discovered by Witten 13 years earlier [18], implicitly implied (without explicitly formulating) an efficient quantum algorithm for the approximation of the Jones polynomial at the value $e^{2\pi i/5}$.

The approximation given by the above algorithms are additive, namely the result lies in a given window, whose size is independent of the actual value being approximated. The formulation of this kind of additive approximation was given in [4]; this is much weaker than a multiplicative approximation, which is what one might desire (again, see discussion in [4]). One might wonder if under such weak requirements, the problem remains meaningful at all. It turns out that, in fact, this additive approximation problem is hard for quantum computation, a result originally shown by Freedman, Kitaev, and Wang:

Theorem 2.3 Adapted from [9] *The problem of approximating the Jones polynomial of the plat closure of a braid at* $e^{2\pi i/k}$ *for constant k, to within the accuracy given in Theorem 2.2, is BQP-hard.*

A different proof of this result was given in [19], and the result was strengthened by Aharonov and Arad [1] to any k which is polynomial in the size of the input, namely, for all the plat closure cases for which the algorithm is polynomial in the size of the braid.

Understanding the Algorithm

The structure of the solution described by Theorems 2.1 and 2.2 consists of four steps:

1. *Mapping the Jones polynomial computation to a computation in the Temperley–Lieb algebra.* There exists a homomorphism of the braid group inside the so called Temperley–Lieb algebra (this homomorphism was the connection that led to the original discovery of the Jones polynomial in [12]). Using this homomorphism, the computation of the Jones polynomial of either the plat or trace closure of a braid can be mapped to the computation of a particular linear functional (called the Markov trace) of the image of the braid in the Temperley–Lieb algebra (for an essential understanding of a geometrical picture of the Temperley–Lieb algebra, see [14]).
2. *Mapping the Temperley–Lieb algebra calculation into a linear algebra calculation.* Using a representation of the Temperley–Lieb algebra, called the path model representation, the computation in step 1 is shown to be equal to a particular weighted trace of the matrix corresponding to the Temperley–Lieb algebra element coming from the original braid.
3. *Choosing the parameter t corresponding to unitary matrices.* The matrix in step 2 is a product of basic matrices corresponding to individual crossings in the braid group; an important characteristic of these basic matrices is that they have a local structure. In addition, by choosing the values of t as in Theorems 2.1 and 2.2, the matrices corresponding to individual crossings become unitary. The result is that the original problem has been turned into a weighted trace calculation of a matrix formed from a product of local unitary matrices–a problem well suited to a quantum computer.
4. *Implementing the quantum algorithm.* Finally the weighted trace calculation of a matrix described in step 3 is formally encoded into a calculation involving local unitary matrices and qubits.

A nice exposition of the algorithm is given in [15].

Applications

Since the publication [3], a number of interesting results have ensued investigating the possibility of quantum algorithms for other combinatorial/topological questions. Quantum algorithms have been developed for the case of the HOMFLYPT two-variable polynomial of the trace closure of a braid at certain pairs of values [19]. (This paper also extends the results of [3] to a class of more generalized braid closures; it is recommended reading as a complement to [3] or [15] as it gives the representation theory of the Jones-Wentzl representations thus putting the path model representation of the Temperley–Lieb algebra in a more general context). A quantum algorithm for the colored Jones polynomial is given in [10].

Recently, significant progress was made on the question of approximating the partition function of the Tutte polynomial of a graph [2]. This polynomial, at various parameters, captures important combinatorial features of the graph. Intimately associated to the Tutte polynomial is the Potts model, a model originating in statistical physics as a generalization of the Ising model to more than 2 states [17,20]; approximating the partition function of the Tutte polynomial of a graph is a very important question

in statistical physics. The work of [2] develops a quantum algorithm for additive approximation of the Tutte polynomial for all planar graphs at all points in the Tutte plane and shows that for a significant set of these points (though not those corresponding to the Potts model) the problem of approximating is a complete problem for a quantum computer. Unlike previous results, these results use non-unitary representations.

Open Problems

There remain many unanswered questions related to the computation of the Jones polynomial from both a classical and quantum computational point of view.

From a classical computation point of view, the originally stated Problem 1 remains wide open for all but trivial choices of t. A result as strong as Theorem 2.2 for a classical computer seems unlikely since it would imply (via Theorem 2.3) that classical computation is as strong as quantum computation. A recent result by Jordan and Shor [13] shows that the approximation given in Theorem 2.1 solves a complete problem for a presumed (but not proven) weaker quantum model called the one clean qubit model. Since this model seems weaker than the full quantum computation model, a classical result as strong as Theorem 2.1 for the trace closure of a braid is perhaps in the realm of possibility.

From a quantum computational point of view, various open directions seem worthy of pursuit. Most of the quantum algorithms known as of the writing of this entry are based on the quantum Fourier transform, and solve problems which are algebraic and number theoretical in nature. Arguably, the greatest challenge in the field of quantum computation, (together with the physical realization of large scale quantum computers), is the design of new quantum algorithms based on substantially different techniques. The quantum algorithm to approximate the Jones polynomial is significantly different from the known quantum algorithms in that it solves a problem which is combinatorial in nature, and it does so without using the Fourier transform. These observations suggest investigating the possibility of quantum algorithms for other combinatorial/topological questions. Indeed, the results described in the applications section above address questions of this type. Of particular interest would be progress beyond [2] in the direction of the Potts model; specifically either showing that the approximation given in [2] is non-trivial or providing a different non-trivial algorithm.

Cross References

► Fault-Tolerant Quantum Computation
► Quantum Error Correction

Recommended Reading

1. Aharonov, D., Arad, I.: The BQP-hardness of approximating the Jones Polynomial. arxiv: quant-ph/0605181 (2006)
2. Aharonov, D., Arad, I., Eban, E., Landau, Z.: Polynomial Quantum Algorithms for Additive approximations of the Potts model and other Points of the Tutte Plane. arxiv:quant-ph/0702008 (2007)
3. Aharonov, D., Jones, V., Landau, Z.: A polynomial quantum algorithm for approximating the Jones polynomial. Proceedings of the 38th ACM Symposium on Theory of Computing (STOC) Seattle, Washington, USA, arxiv:quant-ph/0511096 (2006)
4. Bordewich, M., Freedman, M., Lovasz, L., Welsh, D.: Approximate counting and Quantum computation, Combinatorics. Prob. Comput. **14**(5–6), 737–754 (2005)
5. Conway, J.H.: An enumeration of knots and links, and some of their algebraic properties. Computational Problems in Abstract Algebra (Proc. Conf., Oxford, 1967), 329–358 (1970)
6. Freedman, M.: P/NP and the quantum field computer. Proc. Natl. Acad. Sci. USA **95**, 98–101 (1998)
7. Freedman, M., Kitaev, A., Larsen, M., Wang, Z.: Topological quantum computation. Mathematical challenges of the 21st century. (Los Angeles, CA, 2000). Bull. Amer. Math. Soc. (N.S.) **40**(1), 31–38 (2003)
8. Freedman, M.H., Kitaev, A., Wang, Z.: Simulation of topological field theories by quantum computers. Commun. Math. Phys. **227**, 587–603 (2002)
9. Freedman, M.H., Kitaev, A., Wang, Z.: A modular Functor which is universal for quantum computation. Commun. Math. Phys. **227**(3), 605–622 (2002)
10. Garnerone, S., Marzuoli, A., Rasetti, M.: An efficient quantum algorithm for colored Jones polynomials arXiv.org:quant-ph/0606167 (2006)
11. Jaeger, F., Vertigan, D., Welsh, D.: On the computational complexity of the Jones and Tutte polynomials. Math. Proc. Cambridge Philos. Soc. **108**(1), 35–53 (1990)
12. Jones, V.F.R.: A polynomial invariant for knots via von Neumann algebras. Bull. Am. Math. Soc. **12**(1), 103–111 (1985)
13. Jordan, S., Shor, P.: Estimating Jones polynomials is a complete problem for one clean qubit. http://arxiv.org/abs/0707.2831 (2007)
14. Kauffman, L.: State models and the Jones polynomial. Topology **26**, 395–407 (1987)
15. Kauffman, L., Lomonaco, S.: Topological Quantum Computing and the Jones Polynomial, arXiv.org:quant-ph/0605004 (2006)
16. Podtelezhnikov, A., Cozzarelli, N., Vologodskii, A.: Equilibrium distributions of topological states in circular DNA: interplay of supercoiling and knotting. (English. English summary) Proc. Natl. Acad. Sci. USA **96**(23), 12974–129 (1999)
17. Potts, R.: Some generalized order - disorder transformations, Proc. Camb. Phil. Soc. **48**, 106–109 (1952)
18. Witten, E.: Quantum field theory and the Jones polynomial. Commun. Math. Phys. **121**(3), 351–399 (1989)
19. Wocjan, P., Yard, J.: The Jones polynomial: quantum algorithms and applications in quantum complexity theory. In: Quantum Information and Computation, vol. 8, no. 1 & 2, 147–180 (2008). arXiv.org:quant-ph/0603069 (2006)
20. Wu, F.Y.: Knot Theory and statistical mechanics. Rev. Mod. Phys. **64**(4), 1099–1131 (1992)

Quantum Dense Coding

1992; Bennett, Wiesner

Barbara M. Terhal
IBM Research, Yorktown Heights, NY, USA

Keywords and Synonyms

Super dense coding; Dense coding

Problem Definition

Quantum information theory distinguishes classical bits from quantum bits or qubits. The quantum state of n qubits is represented by a complex vector in $(\mathbb{C}^2)^{\otimes n}$, where $(\mathbb{C}^2)^{\otimes n}$ is the tensor product of n 2-dimensional complex vector spaces. Classical n-bit strings form a basis for the vector space $(\mathbb{C}^2)^{\otimes n}$. Column vectors in $(\mathbb{C}^2)^{\otimes n}$ are denoted as $|\psi\rangle$ and row vectors are denoted as $|\psi\rangle^\dagger = |\psi\rangle^{*T} \equiv \langle\psi|$. The complex inner-product between vectors $|\psi\rangle$ and $|\phi\rangle$ is conveniently written as $\langle\psi|\phi\rangle$.

Entangled quantum states $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$ are those quantum states that cannot be written as a product of some vectors $|\psi_i\rangle \in \mathbb{C}^2$, that is $|\psi\rangle \neq \bigotimes_i |\psi_i\rangle$. The Bell states are four orthogonal (maximally) entangled states defined as

$$|\Psi_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\ , \quad |\Psi_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)\ ,$$
$$|\Psi_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)\ , \quad |\Psi_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)\ .$$

The Pauli-matrices X, Y and Z are three unitary, Hermitian 2×2 matrices. They are defined as $X = |0\rangle\langle 1| + |1\rangle\langle 0|$, $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ and $Y = iXZ$.

Quantum states can evolve dynamically under inner-product preserving unitary operations U ($U^{-1} = U^\dagger$). Quantum information can be mapped onto observable classical information through the formalism of quantum measurements. In a quantum measurement on a state $|\psi\rangle$ in $(\mathbb{C}^2)^{\otimes n}$ a basis $\{|x\rangle\}$ in $(\mathbb{C}^2)^{\otimes n}$ is chosen. This basis is made observable through an interaction of the qubits with a macroscopic measurement system. A basis vector x is thus observed with probability $\mathbb{P}(x) = |\langle x|\psi\rangle|^2$.

Quantum information theory or more narrowly quantum Shannon theory is concerned with protocols which enable distant parties to efficiently transmit quantum or classical information, possibly aided by the sharing of quantum entanglement between the parties. For a detailed introduction to quantum information theory, see the book by Nielsen & Chuang [10].

Key Results

Super Dense Coding [3] is the protocol in which two classical bits of information are sent from sender Alice to receiver Bob. This is accomplished by sharing a Bell state $|\Psi_{00}\rangle_{AB}$ between Alice and Bob and the transmission of one qubit. The protocol is illustrated in Fig. 1. Given two bits b_1, b_2 Alice performs the following unitary transformation on her half of the Bell state:

$$P_{b_1 b_2} \otimes I_B |\Psi_{00}\rangle = |\Psi_{b_1 b_2}\rangle\ , \tag{1}$$

i. e. one of the four Bell states. Here $P_{00} = I, P_{01} = X$, $P_{10} = Z$ and $P_{11} = XZ = -iY$. Alice then sends her qubit to Bob. This allows Bob to do a measurement in the Bell basis. He distinguishes the four states $|\Psi_{b_1 b_2}\rangle$ and learns the value of the two bits b_1, b_2.

The protocol demonstrates the interplay between classical information and quantum information. No information can be communicated by merely sharing an entangled state such as $|\Psi_{00}\rangle$ without the actual transmission of physical information carriers. On the other hand it is a consequence of Holevo's theorem [8] that one qubit can encode at most one classical bit of information. The protocol of dense coding shows that the two resources of entanglement and qubit transmission *combined* give rise to a *super-dense coding* of classical information. Dense Coding is thus captured by the following resource inequality

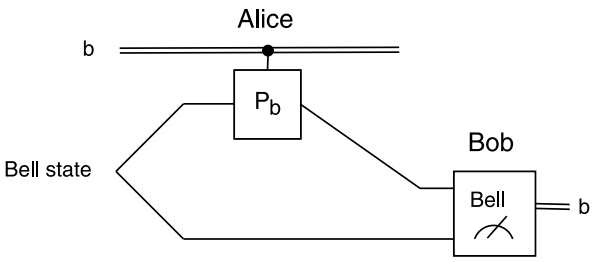
$$1\,\text{ebit} + 1\,\text{qubit} \succeq 2\,\text{cbits}\ . \tag{2}$$

In words, one bit of quantum entanglement (one ebit) in combination with the transmission of one qubit is sufficient for the transmission of two classical bits or cbits.

Quantum Teleportation [1] is a protocol that is dual to Dense Coding. In quantum teleportation, 1 ebit (a Bell state) is used in conjunction with the transmission of two classical bits to send one qubit from Alice to Bob. Thus the resource relation for Quantum Teleportation is

$$1\,\text{ebit} + 2\,\text{cbits} \succeq 1\,\text{qubit}\ . \tag{3}$$

The relation with quantum teleportation allows one to argue that dense coding is optimal. It is not possible to encode $2k$ classical bits in less than $m < k$ quantum bits even in the presence of shared quantum entanglement. Let us assume the opposite and obtain a contradiction. One uses quantum teleportation to convert the transmission of k quantum bits into the transmission of $2k$ classical bits. Then one can use the assumed super-dense coding scheme to encode these $2k$ bits into $m < k$ qubits. As a result one can send k quantum bits by effectively transmitting $m < k$ quantum bits (and sharing quantum entanglement) which is known to be impossible.



Quantum Dense Coding, Figure 1
Dense Coding. Alice and Bob use a shared Bell state to transmit two classical bits $b = (b_1, b_2)$ by sending one qubit. *Double lines* are classical bits and *single lines* represent quantum bits

Applications

Harrow [7] has introduced the notion of a coherent bit, or cobit. The notion of a cobit is useful in understanding resource relations and trade-offs between quantum and classical information. The noiseless transmission of a qubit from Alice to Bob can be viewed as the linear map $S_q : |x\rangle_A \to |x\rangle_B$ for a set of basis states $\{|x\rangle\}$. The transmission of a classical bit can be viewed as the linear map $S_c : |x\rangle_A \to |x\rangle_B |x\rangle_E$ where E stands for the environment Eve. Eve's copy of every basis state $|x\rangle$ can be viewed as the output of a quantum measurement and thus Bob's state is classical. The transmission of a cobit corresponds to the linear map $S_{co} : |x\rangle_A \to |x\rangle_A |x\rangle_B$. Since Alice keeps a copy of the transmitted data, Bob's state is classical. On the other hand, the cobit can also be used to generate a Bell state between Alice and Bob. Since no qubit can be transmitted via a cobit, a cobit is weaker than a qubit. A cobit is stronger than a classical bit since entanglement can be generated using a cobit.

One can define a *coherent* version of super-dense coding and quantum teleportation in which measurements are replaced by unitary operations. In this version of dense coding Bob replaces his Bell measurement by a rotation of the states $|\Psi_{b_1 b_2}\rangle$ to the states $|b_1 b_2\rangle_B$. Since Alice keeps her input bits, the coherent protocol implements the map $|x_1 x_2\rangle_A \to |x_1 x_2\rangle_A |x_1 x_2\rangle_B$. Thus one can strengthen the dense coding resource relation to

$$1\,\text{ebit} + 1\,\text{qubit} \succeq 2\,\text{cobits}\,. \tag{4}$$

Similarly, the coherent execution of quantum teleportation gives rise to the modified relation 2 cobits + 1 ebit $\succeq$ 1 qubit + 2 ebits. One can omit 1 ebit on both sides of the inequality by using ebits catalytically, i. e. they can be borrowed and returned at the end of the protocol. One can then combine both coherent resource inequalities and obtain a resource *equality*

$$2\,\text{cobits} = 1\,\text{qubit} + 1\,\text{ebit}\,. \tag{5}$$

A different extension of dense coding is the notion of super-dense coding of quantum states proposed in [6]. Instead of dense coding classical bits, the authors in [6] propose to code quantum bits *whose quantum states are known to the sender Alice*. This last restriction is usually referred to as the remote preparation of qubits, in contrast to the transmission of qubits whose states are unknown to the sender. In remote preparation of qubits the sender Alice can use the additional knowledge about her states in the choice of encoding. In [6] it is shown that one can obtain the asymptotic resource relation

$$1\,\text{ebit} + 1\,\text{qubit} \succeq 2\,\text{remotely prepared qubit(s)}\,. \tag{6}$$

Such relation would be impossible if the r.h.s. were replaced by 2 qubits. In that case the inequality could be used repeatedly to obtain that 1 qubit suffices for the transmission of an arbitrary number of qubits which is impossible.

The "non-oblivious" super-dense coding of quantum states should be compared with the non-oblivious and asymptotic variant of quantum teleportation which was introduced in [2]. In this protocol, referred to as remote state preparation (using classical bits), the quantum teleportation inequality, Eq. (3) is tightened to

$$1\,\text{ebit} + 1\,\text{cbit} \succeq 1\,\text{remotely prepared qubit(s)}\,. \tag{7}$$

These various resource (in)equalities and their underlying protocols can be viewed as the first in a comprehensive theory of resources inequalities. The goal of such theory [4] is to provide a unified and simplified approach to quantum Shannon theory.

Experimental Results

In [9] a partial realization of dense coding was given using polarization states of photons as qubits. The Bell state $|\Psi_{01}\rangle$ can be produced by parametric down-conversion; this state was used in the experiment as the shared entanglement between Alice and Bob. With current experimental techniques it is not possible to carry out a low-noise measurement in the Bell basis which uniquely distinguishes the four Bell states. Thus in [9] one of three messages, *a trit*, is encoded into the four Bell states. Using two-particle interferometry Bob learns the value of the trit by distinguishing two of the four Bell states uniquely and obtaining a third measurement signal for the two other Bell states.

In perfect dense coding the channel capacity is 2 bits. For the trit-scheme of [9] the ideal channel capacity is $\log 3 \approx 1.58$. Due to the noise in the operations and measurements the authors of [9] estimate the experimentally achieved capacity as 1.13 bits.

In [11] the complete protocol of dense coding was carried out using two $^9Be^+$ ions confined to an electromagnetic trap. A qubit is formed by two internal hyperfine levels of the $^9Be^+$ ion. Single qubit and two-qubit operations are carried out using two polarized laser beams. A single qubit measurement is performed by observing a weak/strong fluorescence of $|0\rangle$ and $|1\rangle$. The authors estimate that the noise in the unitary transformations and measurements leads to an overall error rate on the transmission of the bits b of 15%. This results in an effective channel capacity of 1.16 bits.

In [5] dense coding was carried out using NMR spectroscopy. The two qubits were formed by the nuclear spins of ^1H and ^{13}C of chloroform molecules $^{13}CHCL_3$ in liquid solution at room temperature. The full dense coding protocol was implemented using the technique of temporal averaging and the application of coherent RF pulses, see [10] for details. The authors estimate an overall error-rate on the transmission of the bits b of less than 10%.

Cross References

► Teleportation of Quantum States

Recommended Reading

1. Bennett, C.H., Brassard, G., Crepeau, C., Jozsa, R., Peres, A., Wootters, W.K.: Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. Phys. Rev. Lett. **70**, 1895–1899 (1993)
2. Bennett, C.H., DiVincenzo, D.P., Smolin, J.A., Terhal, B.M., Wootters, W.K.: Remote state preparation. Phys. Rev. Lett. **87**, 077902 (2001)
3. Bennett, C.H., Wiesner, S.J.: Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states. Phys. Rev. Lett. **69**, 2881–2884 (1992)
4. Devetak, I., Harrow, A., Winter, A.: A resource framework for quantum Shannon theory. Tech. Report CSTR-05-008, CS Department, University of Bristol, December (2005)
5. Fang, X., Zhu, X., Feng, M., Mao, X., Du, F.: Experimental implementation of dense coding using nuclear magnetic resonance. Phys. Rev. A **61**, 022307 (2000)
6. Harrow, A., Hayden, P., Leung, D.: Superdense coding of quantum states. Phys. Rev. Lett. **92**, 187901 (2004)
7. Harrow, A.W.: Coherent communication of classical messages. Phys. Rev. Lett. **92**, 097902 (2004)
8. Holevo, A.S.: Bounds for the quantity of information transmitted by a quantum communication channel. Problemy Peredachi Informatsii, **9**, 3–11 (1973). English translation in: Probl. Inf. Transm. **9**, 177–183 (1973)
9. Mattle, K., Weinfurter, H., Kwiat, P.G., Zeilinger, A.: Dense coding in experimental quantum communication. Phys. Rev. Lett. **76**, 4656–4659 (1996)
10. Nielsen, M.A., Chuang, I.L.: Quantum computation and quantum information. Cambridge University Press, Cambridge, U.K. (2000)
11. Schaetz T., Barrett, M.D., Leibfried, D., Chiaverini, J., Britton, J., Itano, W.M., Jost, J.D., Langer, C., Wineland, D.J.: Quantum Dense Coding with Atomic Qubits. Phys. Rev. Lett. **93**, 040505 (2004)

Quantum Error Correction

1995; Shor

Martin Rötteler
NEC Laboratories America, Princeton, NJ, USA

Keywords and Synonyms

Quantum error-correcting codes; Quantum codes; Stabilizer codes

Problem Definition

Quantum systems can never be considered isolated from an environment which permanently causes disturbances of the state of the system. This noise problem threatens quantum computers and their great promise, namely to provide a computational advantage over classical computers for certain problems (see also the cross references in the Sect."Cross References"). Quantum noise is usually modeled by the notion of a *quantum channel* which generalizes the classical case, and, in particular, includes scenarios for communication (space) and storage (time) of quantum information. For more information about quantum channels and quantum information in general, see [12]. A basic channel is the quantum mechanical analog of the classical binary symmetric channel [11]. This quantum channel is called the *depolarizing channel* and depends on a parameter p. Its effect is to randomly apply one of the Pauli spin matrices X, Y, and Z to the state of the system, mapping a quantum state ρ of one qubit to $(1-p)\rho + p/3(X\rho X + Y\rho Y + Z\rho Z)$. It should be noted that it is always possible to map any quantum channel to a depolarizing channel by twirling operations. The basic problem of quantum error correction is to devise a mechanism which allows to perfectly recover quantum information which has been sent through a quantum channel, in particular the depolarizing channel.

Key Results

For a long time, it was not known whether it would be possible to protect quantum information against noise. Even some indication in the form of the no-cloning theorem was put forward to support the view that it might be impossible. The no-cloning theorem essentially says that an unknown quantum state cannot be copied perfectly, thereby dashing the hopes that a simple triple-replication and majority voting mechanism (which works well classically) could be used for the quantum case. Therefore it came as a surprise when Shor [13] found a quantum code which encodes one qubit into nine qubits in such a way that the resulting state has the ability to be protected against arbitrary single-qubit errors on each of these nine qubits. The idea is to use a concatenation of two threefold repetition codes. One of them protects against bit-flip errors while the other protects against phase-flip errors. The quantum code is a two-dimensional subspace of the 2^9 dimensional Hibert space $(\mathbb{C}^2)^{\otimes 9}$. Two orthogonal basis vectors of this space are identified with the logical 0 and 1 states, respectively, usually called $|\underline{0}\rangle$ and $|\underline{1}\rangle$. Explicitly, the code is given by

$$
\begin{aligned}
|\underline{0}\rangle &= \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle) \quad \otimes (|000\rangle + |111\rangle) \\
&\qquad\qquad\qquad\qquad \otimes (|000\rangle + |111\rangle)\,, \\
|\underline{1}\rangle &= \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle) \quad \otimes (|000\rangle - |111\rangle) \\
&\qquad\qquad\qquad\qquad \otimes (|000\rangle - |111\rangle)\,.
\end{aligned}
$$

The state $\alpha|0\rangle + \beta|1\rangle$ of one qubit is encoded to the state $\alpha|\underline{0}\rangle + \beta|\underline{1}\rangle$ of the nine qubit system. The reason why this code can correct one arbitrary quantum error is as follows.

First, suppose that a bit-flip error has happened, which in quantum mechanical notation is given by the operator X. Then a majority vote of each block of three qubits $1-3, 4-6$, and $7-9$ can be computed and the bit-flip can be corrected. To correct against phase-flip errors, which are given by the operator Z, the fact is used that the code can be written as $|\underline{0}\rangle = |{+}{+}{+}\rangle + |{-}{-}{-}\rangle$, $|\underline{1}\rangle = |{+}{+}{+}\rangle - |{-}{-}{-}\rangle$, where $|\pm\rangle = \frac{1}{\sqrt{2}}(|000\rangle \pm |111\rangle)$. By measuring each block of three in the basis $\{|+\rangle, |-\rangle\}$, the majority of the phase-flips can be detected and one phase-flip error can be corrected. Similarly, it can be shown that Y, which is a combination of a bit-flip and a phase-flip, can be corrected.

Discretization of Noise

Even though the above procedure seemingly only takes care of bit-flips and phase-flip errors, it actually is true that an *arbitrary* error affecting a single qubit out of the nine qubits can be corrected. In particular, and perhaps surprisingly, this is also the case if one of the nine qubits is completely destroyed. The linearity of quantum mechanics allows this method to work. Linearity implies that whenever operators A and B can be corrected, so can their sum $A + B$ [6,13,15]. Since the (finite) set $\{\mathbf{1}_2, X, Y, Z\}$ forms a vector space basis for the (continuous) set of all one-qubit errors, the nine-qubit code can correct an arbitrary single qubit error.

Syndrome Decoding and the Need for Fresh Ancillas

A way to do the majority vote quantum-mechanically is to introduce two new qubits (also called ancillas) that are initialized in $|0\rangle$. Then, the results of the two parity checks for the repetition code of length three can be computed into these two ancillas. This syndrome computation for the repetition code can be done using the so-called controlled not (CNOT) gates [12] and Hadamard gates. After this, the qubits holding the syndrome will factor out (i. e., they have no influence on future superpositions or interferences of the computational qubits), and can be discarded. Quantum error correction demands a large supply of fresh qubits for the syndrome computations which have to be initialized in a state $|0\rangle$. The preparation of many such states is required to fuel active quantum error correcting cycles, in which syndrome measurements have to be applied repeatedly. This poses great challenges to any concrete physical realization of quantum error-correcting codes.

Conditions for General Quantum Codes

Soon after the discovery of the first quantum code, general conditions required for the existence of codes, which protect quantum systems against noise, were sought after. Here the noise is modeled by a general quantum channel, given by a set of error operators E_i. The Knill–Laflamme conditions [8] yield such a characterization. Let C be the code subspace and let P_C be an orthogonal projector onto C. Then the existence of a recovery operation for the channel with error operators E_i is equivalent to the equation

$$P_C E_i^\dagger E_j P_C = \lambda_{i,j} P_C\,,$$

for all i and j, where $\lambda_{i,j}$ are some complex constants. This recently has been extended to the more general framework of subsystem codes (also called operator quantum error correcting codes) [10].

Constructing Quantum Codes

The problem of deriving general constructions of quantum codes was addressed in a series of ground-breaking papers by several research groups in the mid 90s. Techniques were developed which allow classical coding theory to be imported to an extent that is enough to provide many families of quantum codes with excellent error correction properties.

The IBM group [2] investigated quantum channels, placed bounds on the quantum channels' capacities, and showed that for some channels it is possible to compute the capacity (such as for the quantum erasure channel). Furthermore, they showed the existence of a five qubit quantum code that can correct an arbitrary error, thereby being much more efficient than Shor's code. Around the same time, Calderbank and Shor [4] and Steane [14] found a construction of quantum codes from any pair C_1, C_2 of classical linear codes satisfying $C_2^{\perp} \subseteq C_1$. Named after their inventors, these codes are known as CSS codes.

The AT&T group [3] found a general way of defining a quantum code. Whenever a classical code over the finite field $\mathbb{F}_4$ exists that is additively closed and self-orthogonal with respect to the Hermitian inner product, they were able to find even more examples of codes. Independently, D. Gottesman [6,7] developed the theory of stabilizer codes. These are defined as the simultaneous eigenspaces of an abelian subgroup of the group of tensor products of Pauli matrices on several qubits. Soon after this, it was realized that the two constructions are equivalent.

A stabilizer code which encodes k qubits into n qubits and has distance d is denoted by $[[n, k, d]]$. It can correct up to $\lfloor (d-1)/2 \rfloor$ errors of the n qubits. The rate of the code is defined as $r = k/n$. Similar to classical codes, bounds on quantum error-correcting codes are known; i. e., the Hamming, Singleton, and linear programming bounds.

Asymptotically Good Codes

Matching the developments in classical algebraic coding theory, an interesting question deals with the existence of asymptotically good codes; i. e., families of quantum codes with parameters $[[n_i, k_i, d_i]]$, where $i \geq 0$, which have asymptotically non-vanishing rate $\lim_{i\to\infty} k_i/n_i > 0$ and non-vanishing relative distance $\lim_{i\to\infty} d_i/n_i > 0$. In [4], the existence of asymptotically good codes was established using random codes. Using algebraic geometry (Goppa) codes, it was later shown by Ashikhmin, Litsyn, and Tsfasman that there are also explicit families of asymptotically good quantum codes. Currently, most constructions of quantum codes are from the above mentioned stabilizer/additive code construction, with notable exception of a few non-additive codes and some codes which do not fit into the framework of Pauli error bases.

Applications

Besides their canonical application to protect quantum information against noise, quantum error correcting codes have been used for other purposes as well. The Preskill/Shor proof of the security of the quantum key distribution scheme BB84 relies on an entanglement purification protocol, which in turn uses CSS codes. Furthermore, quantum codes have been used for quantum secret sharing, quantum message authentication, and secure multiparty quantum computations. Properties of stabilizer codes are also germane for the theory of fault-tolerant quantum computation.

Open Problems

The literature of quantum error correction is fast growing, and the list of open problems is certainly too vast to be surveyed here in detail. The following short list is highly influenced by the preference of the author.

It is desirable to find quantum codes for which all stabilizer generators have low weight. This would be the quantum equivalent of low-density parity check (LDPC) codes. Since the weights directly translate into the complexity of the syndrome computation circuitry, it would be highly desirable to find examples of such codes. So far, only few sporadic constructions are known.

It is an open problem to find new families of quantum codes which improve on the currently known estimates on the threshold for fault-tolerant quantum computing. An advantage might be to use subsystem codes, since they allow for simple error correction circuits. It would be useful to find more families of subsystem codes, thereby generalizing the Bacon/Shor construction.

Most quantum codes are designed for the depolarizing channel, where – roughly speaking – the error probability is improved from p to $p^{d/2}$ for a distance d code. The independence assumption underlying this model might not always be justified and therefore it seems imperative to consider other, e. g., non-Markovian, error models. Under some assumptions on the decay of the interaction strengths, threshold results for such channels have been shown. However, good constructions of codes for such types of noise are still out of reach.

Approximate quantum error-correcting codes have found applications in quantum authentication and recently for secure multiparty quantum computations [1].

Here the Knill–Laflamme conditions do not have to be satisfied exactly, but some error is allowed. This gives much more freedom in defining subspaces and if some error can be tolerated, quantum codes with much better error correction capabilities become feasible. However, not many constructions of such codes are known.

Experimental Results

Active quantum error-correcting codes, such as those codes which require syndrome measurements and correction operations, as well as passive codes (i. e., codes in which the system stays in an simultaneous invariant subspace of all error operators for certain types of noise), have been demonstrated for some physical systems. The most advanced physical demonstration in this respect are the nuclear magnetic resonance (NMR) experiments [9]. The three-qubit repetition code, which protects one qubit against phase-flip error Z, was demonstrated in an ion-trap for beryllium ion qubits [5].

Data Sets

M. Grassl maintains http://www.codetables.de, which contains tables of the best known quantum codes, some entries of which extend [3, Table III]. It also contains bounds on the minimum distance of quantum codes for given lengths and dimensions, and contains information about the construction of the codes. In principle, this can be used to get explicit generator matrices (see also the following section, "URL to Code").

URL to Code

The computer algebra system Magma (http://magma.maths.usyd.edu.au/magma/) has functions and data structures for defining and analyzing quantum codes. Several quantum codes are already defined in a database of quantum codes. For instance, the command `QECC(F,n,k)` returns the best known quantum code (i. e., the one of highest distance) over the field F, of length n, and dimension k. It allows the user to define new quantum codes, to study their properties (such as the weight distribution, automorphism), and several predefined methods for obtaining new codes from old ones.

Cross References

- ▶ Abelian Hidden Subgroup Problem
- ▶ Fault-Tolerant Quantum Computation
- ▶ Quantization of Markov Chains
- ▶ Quantum Algorithm for Element Distinctness
- ▶ Quantum Algorithm for Factoring
- ▶ Quantum Algorithm for Finding Triangles
- ▶ Quantum Algorithm for Solving the Pell's Equation
- ▶ Quantum Key Distribution
- ▶ Teleportation of Quantum States

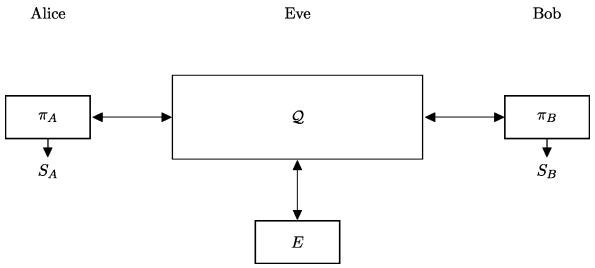
Recommended Reading

1. Ben-Or, M., Crépeau, C., Gottesman, D., Hassidim, A., Smith, A.: Secure multiparty quantum computation with (only) a strict honest majority. In: Proceedings of the 47th Symposium on Foundations of Computer Science (FOCS'06), 2006, pp. 249–260
2. Bennett, C.H., DiVincenzo, D.P., Smolin, J.A., Wootters, W.K.: Mixed-state entanglement and quantum error correction. Phys. Rev. A **54**, 3824–3851 (1996)
3. Calderbank, A.R., Rains, E.M., Shor, P.W., Sloane, N.J.A.: Quantum error correction via codes over GF(4). IEEE Trans. Inform. Theory **44**, 1369–1387 (1998)
4. Calderbank, A.R., Shor, P.W.: Good quantum error-correcting codes exist. Phys. Rev. A **54**, 1098–1105 (1996)
5. Chiaverini, J., Leibfried, D., Schaetz, T., Barrett, M.D., Blakestad, R.B., Britton, J., Itano, W.M., Jost, J.D., Knill, E., Langer, C., Ozeri, R., Wineland, D.J.: Realization of quantum error correction. Nature **432**, 602–605 (2004)
6. Gottesman, D.: Class of quantum error-correcting codes saturating the quantum Hamming bound. Phys. Rev. A **54**, 1862–1868 (1996)
7. Gottesman, D.: Stabilizer codes and quantum error correction, Ph. D. thesis, Caltech. (1997) See also: arXiv preprint quant-ph/9705052
8. Knill, E., Laflamme, R.: Theory of quantum error-correcting codes. Phys. Rev. A **55**, 900–911 (1997)
9. Knill, E., Laflamme, R., Martinez, R., Negrevergne, C.: Benchmarking quantum computers: the five-qubit error correcting code. Phys. Rev. Lett. **86**, 5811–5814 (2001)
10. Kribs, D., Laflamme, R., Poulin, D.: Unified and generalized approach to quantum error correction. Phys. Rev. Lett. **94**(4), 180501 (2005)
11. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error–Correcting Codes. North–Holland, Amsterdam (1977)
12. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
13. Shor, P.W.: Scheme for reducing decoherence in quantum computer memory. Phys. Rev. A **52**, R2493–R2496 (1995)
14. Steane, A.: Error correcting codes in quantum theory. Phys. Rev. Lett. **77**, 793–797 (1996)
15. Steane, A.: Multiple-particle interference and quantum error correction. Proc. R. Soc. London A **452**, 2551–2577 (1996)

Quantum Key Distribution

1984; Bennett, Brassard
1991; Ekert

Renato Renner
ETH, Institute for Theoretical Physics, Zurich, Switzerland



Quantum Key Distribution, Figure 1
A QKD protocol π consists of algorithms π_A and π_B for Alice and Bob, respectively. The algorithms communicate over a quantum channel $\mathcal{Q}$ that might be coupled to a system E controlled by an adversary. The goal is to generate identical keys S_A and S_B which are independent of E

Keywords and Synonyms

Quantum key exchange, Quantum key growing

Problem Definition

Secret keys, i. e., random bitstrings not known to an adversary, are a vital resource in cryptography (they can be used, e. g., for message encryption or authentication). The *distribution* of secret keys among distant parties, possibly only connected by insecure communication channels, is thus a fundamental cryptographic problem. *Quantum key distribution (QKD)* is a method to solve this problem using quantum communication. It relies on the fact that any attempt of an adversary to wiretap the communication would, by the laws of quantum mechanics, inevitably introduce disturbances which can be detected.

For the technical definition, consider a setting consisting of two honest parties, called *Alice* and *Bob*, as well as an adversary, *Eve*. Alice and Bob are connected by a quantum channel $\mathcal{Q}$ which might be coupled to a (quantum) system E controlled by Eve (see Fig. 1). In addition, it is assumed that Alice and Bob have some means to exchange classical messages *authentically*, that is, they can make sure that Eve is unable to (undetectably) alter classical messages during transmission. If only insecure communication channels are available, Alice and Bob can achieve this using an *authentication scheme* [15] which, however, requires a short initial key. This is why QKD is sometimes called *Quantum Key Growing*.

A *QKD protocol* $\pi = (\pi_A, \pi_B)$ is a pair of algorithms for Alice and Bob, producing classical outputs S_A and S_B, respectively. S_A and S_B take values in $S \cup \{\perp\}$ where S is called *key space* and $\perp$ is a symbol (not contained in S) indicating that no key can be generated. A QKD protocol π with key space S is said to be *perfectly secure on a channel* $\mathcal{Q}$ if, after its execution using communication over $\mathcal{Q}$, the following holds:

- $S_A = S_B$;
- if $S_A \neq \perp$ then S_A and S_B are uniformly distributed on S and independent of the state of E.

More generally, π is said to be *ε-secure on $\mathcal{Q}$* if it satisfies the above conditions except with probability (at most) ε. Furthermore, π is said to be *ε-robust on $\mathcal{Q}$* if the probability that $S_A = \perp$ is at most ε.

In the standard literature on QKD, protocols are typically parametrized by some positive number k quantifying certain resources needed for its execution (e. g., the amount of communication). A protocol $\pi = (\pi_k)_{k\in\mathbb{N}}$ is said to be *secure* (*robust*) on a channel $\mathcal{Q}$ if there exists a sequence $(\varepsilon_k)_{k\in\mathbb{N}}$ which approaches zero exponentially fast such that π_k is ε_k-secure (ε_k-robust) on $\mathcal{Q}$ for any $k \in \mathbb{N}$. Moreover, if the key space of π_k is denoted by S_k, the *key rate* of $\pi = (\pi_k)_{k\in\mathbb{N}}$ is defined by $r = \lim_{k\to\infty} \frac{\ell_k}{k}$ where $\ell_k := \log_2 |S_k|$ is the key length.

The ultimate goal is to construct QKD protocols π which are secure against general attacks, i. e., on *all* possible channels $\mathcal{Q}$. This ensures that an adversary cannot get any information on the generated key even if she fully controls the communication between Alice and Bob. At the same time, a protocol π should be robust on certain realistic (possibly noisy) channels $\mathcal{Q}$ in the absence of an adversary. That is, the protocol must always produce a key, unless the disturbances in the channel exceed a certain threshold. Note that, in contrast to security, robustness cannot be guaranteed in general (i. e., on all $\mathcal{Q}$), as an adversary could, for instance, interrupt the entire communication between Alice and Bob (in which case key generation is obviously impossible).

Key Results

Protocols

On the basis of the pioneering work of Wiesner [16], Bennett and Brassard, in 1984, invented QKD and proposed a first protocol, known today as the *BB84 protocol* [2]. The idea was then further developed by Ekert, who established a connection to quantum entanglement [7]. Later, in an attempt to increase the efficiency and practicability of QKD, various extensions to the BB84 protocol as well as alternative types of protocols have been proposed.

QKD protocols can generally be subdivided into (at least) two subprotocols. The purpose of the first, called *distribution protocol*, is to generate a *raw key pair*, i. e., a pair of correlated classical values X and Y known to Alice and Bob, respectively. In most protocols (including BB84), Alice chooses $X = (X_1, \ldots, X_k)$ at random, encodes each of

the X_i into the state of a quantum particle, and then sends the k particles over the quantum channel to Bob. Upon receiving the particles, Bob applies a measurement to each of them, resulting in $Y = (Y_1, \ldots, Y_k)$. The crucial idea now is that, by virtue of the laws of quantum mechanics, the secrecy of the raw key is a function of the strength of the correlation between X and Y; in other words, the more information (on the raw) key an adversary tries to acquire, the more disturbances she introduces.

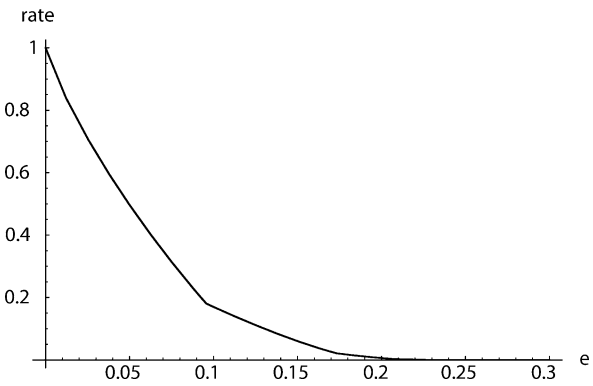
This is exploited in the second subprotocol, called *distillation protocol*. Roughly speaking, Alice and Bob estimate the statistics of the raw key pair (X, Y). If the correlation between their respective parts is sufficiently strong, they use classical techniques such as *information reconciliation* (error correction) and *privacy amplification* (see [3] for the case of a classical adversary which is relevant for the analysis of security against individual attacks and [12,13] for the quantum-mechanical case which is relevant in the context of collective and general attacks) to turn (X, Y) into a pair (S_A, S_B) of identical and secure keys.

Key Rate as a Function of Robustness and Security

The performance (in terms of the key rate) of a QKD protocol strongly depends on the desired level of security and robustness it is supposed to provide, as illustrated in Fig. 2. (The robustness is typically measured in terms of the *maximum tolerated channel noise*, i. e., the maximum noise of a channel $\mathcal{Q}$ such that the protocol is still robust on $\mathcal{Q}$ according to the above definition.) The results summarized below apply to protocols of the form described above where, for the analysis of robustness, it is assumed that the quantum channel $\mathcal{Q}$ connecting Alice and Bob is *memoryless* and *time-invariant*, i. e., each transmission is subject to the same type of disturbances. Formally, such channels are denoted by $\mathcal{Q} = \bar{\mathcal{Q}}^{\otimes k}$ where $\bar{\mathcal{Q}}$ describes the action of the channel in a single transmission.

Security Against Individual Attacks A QKD protocol π is said to be *secure against individual attacks* if it is secure on any channel $\mathcal{Q}$ of the form $\bar{\mathcal{Q}}^{\otimes k}$ where the coupling to E is purely classical. Note that this notion of security is relatively weak. Essentially, it only captures attacks where the adversary applies identical and independent measurements to each of the particles sent over the channel.

The following general statement can be derived from a classical argument due to Csiszár and Körner [5]. Let τ be a distribution protocol as described above, i. e., τ generates a raw key pair (X, Y). Moreover, let S be a set of quantum channels $\bar{\mathcal{Q}}$ suitable for τ. Then there exists a QKD protocol π (parametrized by k), consisting of k executions



Quantum Key Distribution, Figure 2
Key rate of an extended version of the BB84 QKD protocol depending on the maximum tolerated channel noise (measured in terms of the bit-flip probability *e*) [12]

of the subprotocol τ followed by an appropriate distillation protocol such that the following holds: π is robust on $\mathcal{Q} = \bar{\mathcal{Q}}^{\otimes k}$ for any $\bar{\mathcal{Q}} \in S$, secure against individual attacks, and has key rate at least

$$r \geq \min_{\bar{\mathcal{Q}} \in S} H(X|Z) - H(X|Y) \,, \tag{1}$$

where the Shannon entropies on the r.h.s. are evaluated for the joint distribution $P^{\bar{\mathcal{Q}}}_{XYZ}$ of the raw key (X, Y) and the (classical) value Z held by Eve's system E after one execution of τ on the channel $\bar{\mathcal{Q}}$. Evaluating the right hand side for the BB84 protocol on a channel with bit-flip probability e shows that the rate is non-negative if $e \leq 14.6\%$ [8].

Security Against Collective Attacks A QKD protocol π is said to be *secure against collective attacks* if it is secure on any channel $\mathcal{Q}$ of the form $\bar{\mathcal{Q}}^{\otimes k}$ with arbitrary coupling to E. This notion of security is strictly stronger than security against individual attacks, but it still relies on the assumption that an adversary does not apply joint operations to the particles sent over the channel.

As shown by Devetak and Winter [6], the above statement for individual attacks extends to collective attacks when replacing inequality (1) by

$$r \geq \min_{\bar{\mathcal{Q}} \in S} S(X|E) - H(X|Y) \tag{2}$$

where $S(X|E)$ is the conditional von Neumann entropy evaluated for the classical value X and the quantum state of E after one execution of τ on $\bar{\mathcal{Q}}$. For the standard BB84 protocol, the rate is positive as long as the bit-flip probability e of the channel satisfies $e \leq 11.0\%$ [14] (see Fig. 2 for a graph of the performance of an extended version of the protocol).

Security Against General Attacks A QKD protocol π is said to be *secure against general attacks* if it is secure on any arbitrary channel $\mathcal{Q}$. This type of security is sometimes also called *full* or *unconditional security* as it does not rely on any assumptions on the type of attacks or the resources needed by an adversary.

The first QKD protocol to be proved secure against general attacks was the BB84 protocol. The original argument by Mayers [11] was followed by various alternative proofs. Most notably, based on a connection to the problem of entanglement purification [4] established by Lo and Chau [10], Shor and Preskill [14] presented a general argument which applies to various versions of the BB84 protocol.

More recently it has been shown that, for virtually *any* QKD protocol, security against collective attacks implies security against general attacks [12]. In particular, the above statement about the security of QKD protocols against collective attacks, including formula 2 for the key rate, extends to security against general attacks.

Applications

Because the notion of security described above is *composable* [13] (see [1,12] for a general discussion of composability of QKD) the key generated by a secure QKD protocol can in principle be used within any application that requires a secret key (such as one-time pad encryption). More precisely, let $\mathcal{A}$ be a scheme which, when using a *perfect* key S (i. e., a uniformly distributed bitstring which is independent of the adversary's knowledge), has some failure probability δ (according to some arbitrary failure criterion). Then, if the perfect key S is replaced by the key generated by an ε-secure QKD protocol, the failure probability of $\mathcal{A}$ is bounded by $\delta + \varepsilon$ [13].

Experimental Results

Most known QKD protocols (including BB84) only require relatively simple quantum operations on Alice and Bob's side (e. g., preparing a two-level quantum system in a given state or measuring the state of such a system). This makes it possible to realize them with today's technology. Experimental implementations of QKD protocols usually use photons as carriers of quantum information, because they can easily be transmitted (e. g., through optical fibers). A main limitation, however, is noise in the transmission, which, with increasing distance between Alice and Bob, reduces the performance of the protocol (see Fig. 2). We refer to [9] for an overview on quantum cryptography with a focus on experimental aspects.

Cross References

- ▶ Quantum Error Correction
- ▶ Teleportation of Quantum States

Recommended Reading

1. Ben-Or, M., Horodecki, M., Leung, D.W., Mayers, D., Oppenheim, J.: The universal composable security of quantum key distribution. In: Second Theory of Cryptography Conference TCC. Lecture Notes in Computer Science, vol. 3378, pp. 386–406. Springer, Berlin (2005). Also available at http://arxiv.org/abs/quant-ph/0409078
2. Bennett, C.H., Brassard, G.: Quantum cryptography: Public-key distribution and coin tossing. In: Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, pp. 175–179. IEEE Computer Society Press, Los Alamitos (1984)
3. Bennett, C.H., Brassard, G., Crépeau, C., Maurer, U.: Generalized privacy amplification. IEEE Trans. Inf. Theory **41**(6), 1915–1923 (1995)
4. Bennett, C.H., Brassard, G., Popescu, S., Schumacher, B., Smolin, J., Wootters, W.: Purification of noisy entanglement and faithful teleportation via noisy channels. Phys. Rev. Lett. **76**, 722–726 (1996)
5. Csiszár, I., Körner, J.: Broadcast channels with confidential messages. IEEE Trans. Inf. Theory **24**, 339–348 (1978)
6. Devetak, I., Winter, A.: Distillation of secret key and entanglement from quantum states. Proc. R. Soc. Lond. A **461**, 207–235 (2005)
7. Ekert, A.K.: Quantum cryptography based on Bell's theorem. Phys. Rev. Lett. **67**, 661–663 (1991)
8. Fuchs, C.A., Gisin, N., Griffiths, R.B., Niu, C., Peres, A.: Optimal eavesdropping in quantum cryptography, I. Information bound and optimal strategy. Phys. Rev. A **56**, 1163–1172 (1997)
9. Gisin, N., Ribordy, G., Tittel, W., Zbinden, H.: Quantum cryptography. Rev. Mod. Phys. **74**, 145–195 (2002)
10. Lo, H.-K., Chau, H.F.: Unconditional security of quantum key distribution over arbitrarily long distances. Science **283**, 2050–2056 (1999)
11. Mayers, D.: Quantum key distribution and string oblivious transfer in noisy channels. In: Advances in Cryptology – CRYPTO '96. Lecture Notes in Computer Science, vol. 1109, pp. 343–357. Springer (1996)
12. Renner, R.: Security of Quantum Key Distribution. Ph. D. thesis, Swiss Federal Institute of Technology (ETH) Zurich, Also available at http://arxiv.org/abs/quant-ph/0512258 (2005)
13. Renner, R., König, R.: Universally composable privacy amplification against quantum adversaries. In: Second Theory of Cryptography Conference TCC. Lecture Notes in Computer Science, vol. 3378, pp. 407–425. Springer, Berlin (2005). Also available at http://arxiv.org/abs/quant-ph/0403133
14. Shor, P.W., Preskill, J.: Simple proof of security of the BB84 quantum key distribution protocol. Phys. Rev. Lett. **85**, 441 (2000)
15. Wegman, M.N., Carter, J.L.: New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.* **22**, 265–279 (1981)
16. Wiesner, S.: Conjugate coding. Sigact News **15**(1), 78–88 (1983)

Quantum Search

1996; Grover

Lov K. Grover[1], Ben W. Reichardt[2]
[1] Bell Labs, Alcatel-Lucent, Murray Hill, NJ, USA
[2] Institute for Quantum Information, California Institute of Technology, Pasadena, CA, USA

Keywords and Synonyms

Quantum unsorted database search

Problem Definition

Informal Description

The search problem can be described informally as, given a set of N items, identify an item satisfying a given property. Assume that it is easy to query whether a specific item satisfies the property or not. Now, the set of N items is not sorted and so there appears to be no shortcut to the brute-force method of checking each item one by one until the desired item is found. However, that intuition is only correct for classical computers; quantum computers can be in multiple states simultaneously and can examine multiple items at the same time. There is no obvious lower bound to how fast search can be run by a quantum computer, but nor is there an obvious technique faster than brute-force search. It turns out, though, that there is an efficient quantum mechanical search algorithm that makes only $O(\sqrt{N})$ queries, and this is optimal.

This quantum algorithm works very different from searching with a classical computer [5]. The optimal classical strategy is to check the items one at a time in random order. After η items are picked, the probability that the search hasn't yet succeeded is $(1 - 1/N)(1 - 1/(N - 1))\cdots(1 - 1/(N - \eta + 1))$. For $\eta \ll N$, the success probability is therefore roughly $1-(1-1/N)^{\eta} \approx \eta/N$. Increasing the success probability to a constant requires the number of items picked, η, to be $\Omega(N)$.

In contrast, the quantum search algorithm through a series of quantum mechanical operations steadily increases the amplitude on the target item. In η steps it increases this amplitude to roughly $\eta/\sqrt{N}$, and hence the success probability (on measuring the state) to η^2/N. Boosting this to $\Omega(1)$ requires only $O(\sqrt{N})$ steps, approximately the square-root of the number of steps required by any classical algorithm.

The reason the quantum search algorithm has been of so much interest in a variety of fields is that it can be adapted to different settings, giving a new class of quantum algorithms extending well beyond search problems.

Formal Statement

Given oracle access to a bit string $x \in \{0,1\}^N$, find an index i such that $x_i = 1$, if such index exists. In particular, determine if $x = 0^N$ or not – i. e., evaluate the OR function $x_1 \vee x_2 \vee \cdots \vee x_N$. To understand this, think of the indices i as combinatorial objects of some sort, and x_i indicates whether i satisfies a certain property or not – with x_i efficiently computable given i. The problem is to find an object satisfying the property. This search problem is *unstructured* because the solution may be arbitrary. Ordered search of a *sorted* list, on the other hand, may be abstracted as: given access to a string promised to be of the form $x = 0^m 1^{N-m}$, find m.

Classically, oracle access means that one has a black-box subroutine that given i returns x_i. The cost of querying the oracle is taken to be one per query. The hardest inputs to search are clearly those x that are all zeros except in a single position – when there is a single solution – a single "needle in a haystack." (For the OR function, such inputs are hard to distinguish from $x = 0^N$.) For any deterministic search algorithm, there exists such an input on which the algorithm makes at least N oracle queries; brute-force search is the best strategy. Any randomized search algorithm with ε probability of success must make N/ε queries.

Quantumly, one is allowed black-box access to a unitary oracle U_x that can query the oracle in a superposition and get an answer in a superposition. U_x is defined as a controlled reflection about indices i with $x_i = 0$:

$$U_x|c, i\rangle = (-1)^{cx_i}|c, i\rangle\,, \tag{1}$$

where $|c\rangle$ is a control qubit. This can be implemented using U'_x satisfying $U'_x(|c, i, b\rangle) = |c, i, (cx_i) \oplus b\rangle$ – where $b \in \{0,1\}$ and $\oplus$ is addition mod two – by setting the second register to $(1/\sqrt{2})(|0\rangle - |1\rangle)$.

For example, if ϕ is a 3-SAT formula on n variables, $i \in \{1, 2, \ldots, N = 2^n\}$ represents a setting for the variables, and x_i indicates if assignment i satisfies ϕ; then is ϕ satisfiable? (Another common example is unstructured database search: i is a record and x_i a function of that record. However, this example is complicated because records need to be stored in a physical memory device. If it is easier to access nearby records, then spatial relationships come into play.)

More generally, say there is a subroutine that returns an efficiently verifiable answer to some problem with probability ε. To solve the problem with constant probability, the subroutine can be run $\Omega(1/\varepsilon)$ times. Quantumly, if the subroutine is a unitary process that returns the right answer with *amplitude* $\sqrt{\varepsilon}$, is there a better technique than measuring the outcome and running the subroutine

$\Omega(1/\varepsilon)$ times? It turns out that this question is closely related to search, because the uniform superposition over indices $(1/\sqrt{N})\sum_i |i\rangle$ has amplitude of returning the right answer as $1/\sqrt{N}$. Thus, an algorithm for this problem immediately implies a search algorithm.

Key Results

Grover [13] showed that there exists a quantum search algorithm that is quadratically faster than the optimal classical randomized algorithm:

Theorem 1 (Grover search) *There is a quantum black-box unstructured search algorithm with success probability ε, using $O(\sqrt{N\varepsilon})$ queries and $O(\sqrt{N\varepsilon}\cdot \log\log N)$ time. If promised that the Hamming weight of x is $|x| \geq k$, then one of the i such that $x_i = 1$ can be found using $O(\sqrt{N\varepsilon/k})$ queries.*

The Grover search algorithm has its simplest form if given the promise that $|x| = 1$. Then the single "marked item" i^* with $x_{i^*} = 1$ can be found by preparing the uniform superposition over indices $|\Psi\rangle = (1/\sqrt{N})\sum_i |i\rangle$, then repeating $\sqrt{N}$ times:

1. Apply the oracle U_x from Eq. (1), with the control bit $c = 1$, to reflect about i^*.
2. Reflect about $|\Psi\rangle$ by applying $U_D = I - 2|\Psi\rangle\langle\Psi|$.

Finally, measure and output i.

It turns out that $i = i^*$ with constant probability. The analysis is straightforward because the quantum state $|\varphi\rangle$ stays in the two-dimensional subspace spanned by $|i^*\rangle$ and $|\Psi\rangle$. Initially, the amplitude on i^* is $\langle i^*|\Psi\rangle = 1/\sqrt{N}$, and the angle between $|i^*\rangle$ and the initial state $|\varphi_0\rangle = |\Psi\rangle$ is $\pi/2 - \theta$, with $\theta = \arcsin 1/\sqrt{N} \approx 1/\sqrt{N}$. Each pair of reflection steps decreases the angle between the $|\varphi\rangle$ and $|i^*\rangle$ by exactly θ, so $\sqrt{N}$ steps suffice to bound the angle away from $\pi/2$. (Using the small angle approximation, after t steps of alternating reflections the amplitude $\langle i^*|\varphi_t\rangle$ is about $t/\sqrt{N}$, making the success probability about t^2/N.)

The reflection about the uniform superposition, $U_D = I - 2|\Psi\rangle\langle\Psi|$, is known as a Grover diffusion step. If the indices are represented in binary, with $N = 2^n$, it can be implemented as transversal Hadamard gates $H^{\otimes n}$, where $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix}1 & 1\\ 1 & -1\end{smallmatrix}\right)$, followed by reflection about $|0^n\rangle$, followed by $H^{\otimes n}$ again. This operation can also be interpreted as an inversion about the average of the amplitudes $\{\langle i|\varphi_t\rangle\}$. Note that if one measures i before each query to the oracle, then the algorithm collapses to effectively classical search by random guessing.

Brassard and Høyer [6], and Grover [14] both realized that quantum search can be applied on top of nearly any quantum algorithm for any problem. Roughly, the *amplitude amplification* technique says that given one quantum algorithm that solves a problem with small probability ε, then by using $O(m)$ calls to that algorithm the success probability can be increased to about $m^2\varepsilon$. (Classically, the success probability could only be increased to about $m\varepsilon$.) More formally,

Theorem 2 (Amplitude amplification, [1, Lemma 9]) *Let $\mathcal{A}$ be a quantum algorithm that outputs a correct answer and witness with probability $\delta \leq \varepsilon$ where ε is known. Furthermore, let*

$$m \leq \frac{\pi}{4\arcsin\sqrt{\varepsilon}} - \frac{1}{2}\,.$$

Then there is an algorithm $\mathcal{A}'$ that uses $2m+1$ calls to $\mathcal{A}$ and $\mathcal{A}^{-1}$ and outputs a correct answer and a witness with probability

$$\delta_{new} \geq \left(1 - \frac{(2m+1)^2}{3}\delta\right)(2m+1)^2\delta\,.$$

Here, one is "searching" for an answer to some problem. The "oracle" is implemented by a routine that conditionally flips the phase based on whether or not the answer is correct (checked using the witness). The reflection about the initial state is implemented by inverting $\mathcal{A}$, applying a reflection about $|0\rangle$, and then reapplying $\mathcal{A}$ (similarly to how the reflection about $|\Psi\rangle$ can be implemented using Hadamard gates). See also [7].

The square-root speedup in quantum search is *optimal*; Bennett, Bernstein, Brassard and Vazirani [4] gave an $\Omega(\sqrt{N})$ lower bound on the number of oracle queries required for a quantum search algorithm. Therefore, quantum computers cannot give an exponential speedup for arbitrary unstructured problems; there exists an oracle relative to which BQP $\not\subseteq$ NP (an NP machine can guess the answer and verify it with one query). In fact, under the promise that $|x| = 1$, the algorithm is precisely optimal and cannot be improved by even a single query [22].

Grover's search algorithm is robust in several ways:

- It is robust against changing both initial state and the diffusion operator:

Theorem 3 ([2]) *Assume $|x| = 1$ with $x_{i^*} = 1$. Assume the initial state $|\varphi_0\rangle$ has real amplitudes $\langle i|\varphi_0\rangle$, with $\langle i^*|\varphi_0\rangle = \alpha$. Let the reflection oracle be $U_x = I - 2|i^*\rangle\langle i^*|$. Let the diffusion operator U_D be a real unitary matrix in the basis $\{|i\rangle\}$. Assume $U_D|\varphi_0\rangle = |\varphi_0\rangle$ and that $U_D|\psi\rangle = e^{i\theta_\psi}|\psi\rangle$ for $\theta_\psi \in [\varepsilon, 2\pi - \varepsilon]$ (where $\varepsilon > 0$ is a constant) for all eigenvectors $|\psi\rangle$ orthogonal to $|\varphi_0\rangle$. Then, there exists $t = O(1/\alpha)$ such that $|\langle i^*|(U_D U_x)^t|\varphi_0\rangle| = \Omega(1)$. (The constant under $\Omega(1)$ is independent of α but can depend on ε.)*

Therefore, there is in fact an entire class of related algorithms that use different "diffusion" operators. This robustness is useful in applications, and may help to explain why Grover search ideas appear so frequently in quantum algorithms.

- Høyer, Mosca and de Wolf [16] showed that quantum search can be implemented so as to be robust also against faulty oracles, a problem known as Bounded-Error Search:

Theorem 4 *Suppose* $U''_x|i,b\rangle = \sqrt{p_i}|i, x_i \oplus b\rangle + \sqrt{1-p_i}|i,(1-x_i)\oplus b\rangle$, *with each* $p_i \geq 9/10$ *(i. e., there is a bounded coherent error rate in the oracle). Search can still be implemented in* $O(\sqrt{N})$ *time.*

Applications

An early application of the Grover search algorithm was to finding collisions; given oracle access to a 2-to-1 function f, find distinct points x, y such that $f(x) = f(y)$. Brassard, Høyer and Tapp [8] developed an $O(N^{1/3})$-time collision algorithm. Finding $x \neq y$ such that $f(x) = f(y)$ for a general function f is known as the Element-Distinctness problem. Buhrman et al. [9] later developed an $O(N^{3/4}\log N)$-time algorithm for Element Distinctness, using amplitude amplification. In a breakthrough, Ambainis [2] gave an optimal, $O(N^{2/3})$-time algorithm for Element Distinctness, which has also led to other applications [18]. Ambainis's algorithm extends quantum search by using a certain quantum walk to replace the Grover diffusion step, and uses Theorem 3 for its analysis.

Grover search has also proved useful in communication complexity. For example, a straightforward distributed implementation of the search algorithm solves the Set Intersection problem – Alice and Bob have respective inputs $x, y \in \{0,1\}^N$, and want to find an index i such that $x_i = y_i = 1$ – with $O(\sqrt{N}\log N)$ qubits of communication. Recently, this technique has led to an exponential classical/quantum separation in the memory required to evaluate a certain total function with a streaming input [17].

Unlike the usual Grover search that has an oscillatory behavior, fixed-point quantum search algorithms converge monotonically to the solution. These algorithms replace the reflection operation – a phase shift of π – with selective phase shifts of $\pi/3$.

Theorem 5 ([15]) *Let* R_s *and* R_t *be selective* $\pi/3$ *phase shifts of the source and target state(s), respectively. If* $\|\langle t|U\rangle|s\rangle\|^2 = 1-\varepsilon$, *then* $\left\|\langle t|\, UR_sU^\dagger R_tU\,|s\rangle\right\|^2 = 1-\varepsilon^3$.

In other words, the deviation of the final state from the desired final state reduces to the cube of what it was for the original transformation. (Classically only an $O(\varepsilon^2)$ improvement is possible.) This clearly gives a monotonic improvement towards the solution state, which is useful when the number of solutions is very high. The technique has also been applied to composite pulses [19]. However, it does not give a square-root speedup.

Another extension of unstructured search is to game-tree evaluation, which is a recursive search problem. Classically, using the alpha-beta pruning technique, the value of a balanced binary AND-OR tree can be computed with zero error in expected time $O(N^{\log_2[(1+\sqrt{33})/4]}) = O(N^{0.754}\,)$[20], and this is optimal even for bounded-error algorithms [21]. Applying quantum search recursively, a depth-d regular AND-OR tree can be evaluated with constant error in time $\sqrt{N}\cdot O(\log N)^{d-1}$, where the log factors come from amplifying the success probability of inner searches to be close to one. Bounded-error quantum search, Theorem 4, allows eliminating these log factors, so the time becomes $O(\sqrt{N}\cdot c^d)$, for some constant c. Very recently, an $N^{1/2+o(1)}$-time algorithm has been discovered for evaluating an arbitrary AND-OR tree on N variables [3,11,12].

Open Problems

As already mentioned, search of a sorted list may be abstracted as, given $x = 0^m1^{N-m}$, find m. Classically, $\lceil\log_2 N\rceil$ queries are necessary and sufficient to find m, with binary search achieving the optimum. Quantumly, $\Theta(\log N)$ queries are also necessary and sufficient, but the constant is unknown. The best lower bound on an exact algorithm (i. e., which succeeds with probability one after a fixed number of queries) is about $0.221\log_2 N$ and the best exact algorithm uses about $0.443\log_2 N$ queries (although there is a quantum Las Vegas algorithm that uses expected $0.32\log_2 N$ queries) [10].

Cross References

- ► Quantum Algorithm for Element Distinctness
- ► Routing

Recommended Reading

1. Aaronson, S., Ambainis A.: Quantum search of spatial regions. Theor. Comput. **1**, 47–79 (2005)
2. Ambainis, A.: Quantum walk algorithm for element distinctness. SIAM J. Comput. **37**(1), 210–239 (2007)
3. Ambainis, A.: A nearly optimal discrete query quantum algorithm for evaluating NAND formulas, arXiv:0704.3628 (2007)
4. Bennett, C.H., Bernstein, E., Brassard, G., Vazirani, U.: Strengths and weaknesses of quantum computing. SIAM J. Comput. **26**(5), 1510–1523 (1997)

5. Brassard, G.: Searching a quantum phone book. Science **275**(5300), 627–628 (1997)
6. Brassard, G., Høyer, P.: An exact quantum polynomial-time algorithm for Simon's problem. In: Proc. 5th Israeli Symp. on Theory of Computing and Systems (ISTCS), pp. 12–23. IEEE Computer Society Press, Hoboken (1997)
7. Brassard, G., Høyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. In: Quantum Computation and Quantum Information Science. AMS Contemporary Mathematics Series, vol. 305 Contemporary Mathematics, pp. 53–74, Providence (2002)
8. Brassard, G., Høyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. In: Proc. 3rd Latin American Theoretical Informatics Conference (LATIN). Lecture Notes in Computer Science, vol. 1380, pp. 163–169. Springer, New York (1998)
9. Buhrman, H., Dürr, C., Heiligman, M., Høyer, P., Magniez, F., Santha, M., de Wolf, R. Quantum algorithms for element distinctness, quant-ph/0007016 (2000)
10. Childs, A.M., Landahl A.J., Parrilo, P.A.: Improved quantum algorithms for the ordered search problem via semidefinite programming. Phys. Rev. A **75**, 032335 (2007)
11. Childs, A.M., Reichardt, B.W., Špalek, R., Zhang, S.: Every NAND formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer, quant-ph/0703015 (2007)
12. Farhi, E., Goldstone, J., Gutmann, S.: A quantum algorithm for the Hamiltonian NAND tree. quant-ph/0702144 (2007)
13. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proc. 28th ACM Symp. on Theory of Computing (STOC), pp. 212–219. Philadelphia, 22–24 May 1996
14. Grover, L.K.: A framework for fast quantum mechanical algorithms. In: Proc. 30th ACM Symp. on Theory of Computing (STOC), pp. 53–62. Dallas, 23–26 May 1998
15. Grover, L.K.: Fixed-point quantum search. Phys. Rev. Lett. **95**, 150501 (2005)
16. Høyer, P., Mosca, M., de Wolf, R.: Quantum search on bounded-error inputs. In: Proc. 30th International Colloquium on Automata, Languages and Programming (ICALP 03), Eindhoven, The Netherlands, pp. 291–299 (2003)
17. Le Gall, F.: Exponential separation of quantum and classical online space complexity. In: Proc. ACM Symp. on Parallel Algorithms and Architectures (SPAA), Cambride, 30 July–1 August (2006)
18. Magniez, F., Nayak, A., Roland, J., Santha, M.: Search via quantum walk. quant-ph/0608026. In: Proc. of 39th ACM Symp. on Theory of Computing (STOC), San Diego, 11–13 June, pp. 575–584 (2007)
19. Reichardt, B.W., Grover, L.K.: Quantum error correction of systematic errors using a quantum search framework. Phys. Rev. A **72**, 042326 (2005)
20. Saks, M., Wigderson, A.: Probabilistic Boolean decision trees and the complexity of evaluating game trees. In: Proc. of 27th IEEE Symp. on Foundation of Computer Science (FOCS), Toronto, 27–29 October, pp. 29–38 (1986)
21. Santha, M.: On the Monte Carlo decision tree complexity of read-once formulae. Random Struct. Algorit. **6**(1), 75–87 (1995)
22. Zalka, C.: Grover's quantum searching algorithm is optimal. Phys. Rev. A **60**(4), 2746–2751 (1999)

Quickest Route

► All Pairs Shortest Paths in Sparse Graphs
► Single-Source Shortest Paths

Quorums

1985; Garcia-Molina, Barbara

Dahlia Malkhi
Microsoft, Silicon Valley Campus,
Mountain View, CA, USA

Keywords and Synonyms

Quorum systems; Voting systems; Coteries

Problem Definition

Quorum systems are tools for increasing the availability and efficiency of replicated services. A *quorum system* for a universe of servers is a collection of subsets of servers, each pair of which intersect. Intuitively, each quorum can operate on behalf of the system, thus increasing its availability and performance, while the intersection property guarantees that operations done on distinct quorums preserve consistency.

The motivation for quorum systems stems from the need to make critical missions performed by machines that are reliable. The only way to increase the reliability of a service, aside from using intrinsically more robust hardware, is via replication. To make a service robust, it can be installed on multiple identical servers, each one of which holds a copy of the service state and performs read/write operations on it. This allows the system to provide information and perform operations even if some machines fail or communication links go down. Unfortunately, replication incurs a cost in the need to maintain the servers consistent. To enhance the availability and performance of a replicated service, Gifford and Thomas introduced in 1979 [3,14] the usage of *votes* assigned to each server, such that a majority of the sum of votes is sufficient to perform operations. More generally, quorum systems are defined formally as follows:

Quorum system: Assume a *universe* U of servers, $|U| = n$, and an arbitrary number of clients. A *quorum system* $\mathcal{Q} \subseteq 2^U$ is a set of subsets of U, every pair of which intersect. Each $Q \in \mathcal{Q}$ is called a *quorum*.

Access Protocol

To demonstrate the usability of quorum systems in constructing replicated services, quorums are used here to implement a multi-writer multi-reader atomic shared variable. Quorums have also been used in various *mutual exclusion* protocols, to achieve Consensus, and in commit protocols.

In the application, clients perform read and write operations on a variable x that is replicated at each server in the universe U. A copy of the variable x is stored at each server, along with a timestamp value t. Timestamps are assigned by a client to each replica of the variable when the client writes the replica. Different clients choose distinct timestamps, e. g., by choosing integers appended with the name of c in the low-order bits. The read and write operations are implemented as follows.

Write: For a client c to write the value v, it queries each server in some quorum Q to obtain a set of value/timestamp pairs $A = \{\langle v_u, t_u\rangle\}_{u\in Q}$; chooses a timestamp $t \in T_c$ greater than the highest timestamp value in A; and updates x and the associated timestamp at each server in Q to v and t, respectively.

Read: For a client to read x, it queries each server in some quorum Q to obtain a set of value/timestamp pairs $A = \{\langle v_u, t_u\rangle\}_{u\in Q}$. The client then chooses the pair $\langle v, t\rangle$ with the highest timestamp in A to obtain the result of the read operation. It writes back $\langle v, t\rangle$ to each server in some quorum Q'.

In both read and write operations, each server updates its local variable and timestamp to the received values $\langle v, t\rangle$ only if t is greater than the timestamp currently associated with the variable. The above protocol correctly implements the semantics of a multi-writer multi-reader atomic variable (see ► Linearizability).

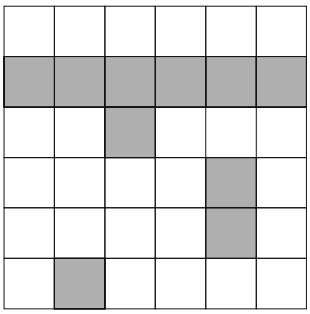
Key Results

Perhaps the two most obvious quorum systems are the singleton, and the set of majorities, or more generally, weighted majorities suggested by Gifford [3].

Singleton: The set system $\mathcal{Q} = \{\{u\}\}$ for some $u \in U$ is the singleton quorum system.

Weighted Majorities: Assume that every server s in the universe U is assigned a number of votes w_s. Then, the set system $\mathcal{Q} = \{Q \subseteq U : \sum_{q\in Q} w_q > (\sum_{q\in U} w_q)/2\}$ is a quorum system called Weighted Majorities. When all the weights are the same, simply call this the system of Majorities.

An example of a quorum system that cannot be defined by voting is the following Grid construction:



Quorums, Figure 1
The Grid quorum system of 6 × 6, with one quorum shaded

Grid: Suppose that the universe of servers is of size $n = k^2$ for some integer k. Arrange the universe into a $\sqrt{n} \times \sqrt{n}$ grid, as shown in Fig. 1. A quorum is the union of a full row and one element from each row below the full row. This yields the Grid quorum system, whose quorums are of size $O(\sqrt{n})$.

Maekawa suggests in [6] a quorum system that has several desirable symmetry properties, and in particular, that every pair of quorums intersect in exactly one element:

FPP: Suppose that the universe of servers is of size $n = q^2 + q + 1$, where $q = p^r$ for a prime p. It is known that a finite projective plane exists for n, with $q + 1$ pairwise intersecting subsets, each subset of size $q + 1$, and where each element is contained in $q + 1$ subsets. Then the set of finite projective plane subsets forms a quorum system.

Voting and Related Notions

Since generally it would be senseless to access a large quorum if a subset of it is a quorum, a good definition may avoid such anomalies. Garcia-Molina and Barbara [2] call such well-formed systems *coteries*, defined as follows:

Coterie: A *coterie* $\mathcal{Q} \subseteq 2^U$ is a quorum system such that for any $Q, Q' \in \mathcal{Q} : Q \not\subseteq Q'$.

Of special interest are quorum systems that cannot be reduced in size (i. e., that no quorum in the system can be reduced in size). Garcia-Molina and Barbara [2] use the term "dominates" to mean that one quorum system is always superior to another, as follows:

Domination: Suppose that $\mathcal{Q}, \mathcal{Q}'$ are two coteries, $\mathcal{Q} \neq \mathcal{Q}'$, such that for every $Q' \in \mathcal{Q}'$, there exists a $Q \in \mathcal{Q}$ such that $Q \subseteq Q'$. Then $\mathcal{Q}$ *dominates* $\mathcal{Q}'$.$\mathcal{Q}'$ is *dominated* if there exists a coterie $\mathcal{Q}$ that dominates it, and is *non-dominated* if no such coterie exists.

Voting was mentioned above as an intuitive way of thinking about quorum techniques. As it turns out, vote assignments and quorums are not equivalent. Garcia-

Molina and Barbara [2] show that quorum systems are strictly more general than voting, i. e. each vote assignment has some corresponding quorum system but not the other way around. In fact, for a system with n servers, there is a double-exponential ($2^{2^{cn}}$) number of non-dominated coteries, and only $O(2^{n^2})$ different vote assignments, though for $n \leq 5$, voting and non-dominated coteries are identical.

Measures

Several measures of quality have been identified to address the question of which quorum system works best for a given set of servers; among these, *load* and *availability* are elaborated on here.

Load A measure of the inherent performance of a quorum system is its *load*. Naor and Wool define in [10] the load of a quorum system as the probability of accessing the busiest server in the *best* case. More precisely, given a quorum system $\mathcal{Q}$, an *access strategy* w is a probability distribution on the elements of $\mathcal{Q}$; i. e., $\sum_{Q\in\mathcal{Q}} w(Q) = 1$. $w(Q)$ is the probability that quorum Q will be chosen when the service is accessed. Load is then defined as follows:

Load: Let a strategy w be given for a quorum system $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$ over a universe U. For an element $u \in U$, the load induced by w on u is $l_w(u) = \sum_{Q_i \ni u} w(Q_i)$. The load induced by a strategy w on a quorum system $\mathcal{Q}$ is

$$L_w(\mathcal{Q}) = \max_{u\in U}\{l_w(u)\}.$$

The *system load* (or just *load*) on a quorum system $\mathcal{Q}$ is

$$L(\mathcal{Q}) = \min_{w}\{L_w(\mathcal{Q})\},$$

where the minimum is taken over all strategies.

The load is a best-case definition, and will be achieved only if an optimal access strategy is used, and only in the case that no failures occur. A strength of this definition is that load is a property of a quorum system, and not of the protocol using it.

The following theorem was proved in [10] for all quorum systems.

Theorem 1 *Let $\mathcal{Q}$ be a quorum system over a universe of n elements. Denote by $c(\mathcal{Q})$ the size of the smallest quorum of $\mathcal{Q}$. Then $L(\mathcal{Q}) \geq \max\{\frac{1}{c(\mathcal{Q})}, \frac{c(\mathcal{Q})}{n}\}$. Consequently, $L(\mathcal{Q}) \geq \frac{1}{\sqrt{n}}$.*

Availability The resilience f of a quorum system provides one measure of how many crash failures a quorum system is *guaranteed* to survive.

Resilience: The *resilience* f of a quorum system $\mathcal{Q}$ is the largest k such that for every set $K \subseteq U$, $|K| = k$, there exists $Q \in \mathcal{Q}$ such that $K \cap Q = \emptyset$.

Note that, the resilience f is at most $c(\mathcal{Q}) - 1$, since by disabling the members of the smallest quorum every quorum is hit. It is possible, however, that an f-resilient quorum system, though vulnerable to a few failure configurations of $f + 1$ failures, can survive many configurations of more than f failures. One way to measure this property of a quorum system is to assume that each server crashes independently with probability p and then to determine the probability F_p that no quorum remains completely alive. This is known as *failure probability* and is formally defined as follows:

Failure probability: Assume that each server in the system crashes independently with probability p. For every quorum $Q \in \mathcal{Q}$ let $\mathcal{E}_Q$ be the event that Q is *hit*, i. e., at least one element $i \in Q$ has crashed. Let crash($\mathcal{Q}$) be the event that all the quorums $Q \in \mathcal{Q}$ were hit, i. e., crash($\mathcal{Q}$) = $\bigwedge_{Q\in\mathcal{Q}} \mathcal{E}_Q$. Then the system failure probability is $F_p(\mathcal{Q}) = \Pr(\text{crash}(\mathcal{Q}))$.

Peleg and Wool study the availability of quorum systems in [11]. A good failure probability $F_p(\mathcal{Q})$ for a quorum system $\mathcal{Q}$ has $\lim_{n\to\infty} F_p(\mathcal{Q}) = 0$ when $p < \frac{1}{2}$. Note that, the failure probability of any quorum system whose resilience is f is at least $e^{-\Omega(f)}$. Majorities has the best availability when $p < \frac{1}{2}$; for $p = \frac{1}{2}$, there exist quorum constructions with $F_p(\mathcal{Q}) = \frac{1}{2}$; for $p > \frac{1}{2}$, the singleton has the best failure probability $F_p(\mathcal{Q}) = p$, but for most quorum systems, $F_p(\mathcal{Q})$ tends to 1.

The Load and Availability of Quorum Systems

Quorum constructions can be compared by analyzing their behavior according to the above measures. The singleton has a load of 1, resilience 0, and failure probability $F_p = p$. This system has the best failure probability when $p > \frac{1}{2}$, but otherwise performs poorly in both availability and load.

The system of Majorities has a load of $\lceil\frac{n+1}{2n}\rceil \approx \frac{1}{2}$. It is resilient to $\lfloor\frac{n-1}{2}\rfloor$ failures, and its failure probability is $e^{-\Omega(n)}$. This system has the highest possible resilience and asymptotically optimal failure probability, but poor load.

Grid's load is $O(\frac{1}{\sqrt{n}})$, which is within a constant factor from optimal. However, its resilience is only $\sqrt{n} - 1$ and it has poor failure probability which tends to 1 as n grows.

The resilience of a FPP quorum system is $q \approx \sqrt{n}$. The load of FPP was analyzed in [10] and shown to be $L(\text{FPP}) = \frac{q+1}{n} \approx 1/\sqrt{n}$, which is optimal. However, its failure probability tends to 1 as n grows.

As demonstrated by these systems, there is a tradeoff between load and fault tolerance in quorum systems, where the resilience f of a quorum system $\mathcal{Q}$ satisfies $f \leq nL(\mathcal{Q})$. Thus, improving one must come at the expense of the other, and it is in fact impossible to simultaneously achieve both optimally. One might conclude that good load conflicts with low failure probability, which is not necessarily the case. In fact, there exist quorum systems such as the Paths system of Naor and Wool [10] and the Triangle Lattice of Bazzi [1] that achieve asymptotically optimal load of $O(1/\sqrt{n})$ and have close to optimal failure probability for their quorum sizes. Another construction is the CWlog system of Peleg and Wool [12], which has unusually small quorum sizes of $\log n - \log\log n$, and for systems with quorums of this size, has optimal load, $L(\text{CWlog}) = O(1/\log n)$, and optimal failure probability.

Byzantine Quorum Systems

For the most part, quorum systems were studied in environments where failures may simply cause servers to become unavailable (benign failures). But what if a server may exhibit arbitrary, possibly malicious behavior? Malkhi and Reiter [7] carried out a study of quorum systems in environments prone to arbitrary (Byzantine) behavior of servers. Intuitively, a quorum system tolerant of Byzantine failures is a collection of subsets of servers, each pair of which intersect in a set containing sufficiently many *correct* servers to mask out the behavior of faulty servers. More precisely, Byzantine quorum systems are defined as follows:

Masking quorum system: A quorum system $\mathcal{Q}$ is a *b-masking quorum system* if it has resilience $f \geq b$, and each pair of quorums intersect in at least $2b + 1$ elements.

The masking quorum system requirements enable a client to obtain the correct answer from the service despite up to b Byzantine server failures. More precisely, a write operation remains as before; to obtain the correct value of x from a read operation, the client reads a set of value/timestamp pairs from a quorum Q and sorts them into clusters of identical pairs. It then chooses a value/timestamp pair that is returned from at least $b + 1$ servers, and therefore must contain at least one correct server. The properties of masking quorum systems guarantee that at least one such cluster exists. If more than one such cluster exists, the client chooses the one with the highest timestamp. It is easy to see that any value so obtained was written before, and moreover, that the most recently written value is obtained. Thus, the semantics of a multi-writer multi-reader safe variable are obtained (see ► Linearizability) in a Byzantine environment.

For a b-masking quorum system, the following lower bound on the load holds:

Theorem 2 *Let $\mathcal{Q}$ be a b-masking quorum system. Then $L(\mathcal{Q}) \geq \max\{\frac{2b+1}{c(\mathcal{Q})}, \frac{c(\mathcal{Q})}{n}\}$, and consequently $L(\mathcal{Q}) \geq \sqrt{\frac{2b+1}{n}}$.*

This bound is tight, and masking quorum constructions meeting it were shown.

Malkhi and Reiter explore in [7] two variations of masking quorum systems. The first, called *dissemination quorum systems*, is suited for services that receive and distribute *self-verifying* information from correct clients (e. g., digitally signed values) that faulty servers can fail to redistribute but cannot undetectably alter. The second variation, called *opaque masking quorum systems*, is similar to regular masking quorums in that it makes no assumption of self-verifying data, but it differs in that clients do not need to know the failure scenarios for which the service was designed. This somewhat simplifies the client protocol and, in the case that the failures are maliciously induced, reveals less information to clients that could guide an attack attempting to compromise the system. It is also shown in [7] how to deal with faulty clients in addition to faulty servers.

Probabilistic Quorum Systems

The resilience of any quorum system is bounded by half of the number of servers. Moreover, as mentioned above, there is an inherent tradeoff between low load and good resilience, so that it is in fact impossible to simultaneously achieve both optimally. In particular, quorum systems over n servers that achieve the optimal load of $\frac{1}{\sqrt{n}}$ can tolerate at most $\sqrt{n}$ faults.

To break these limitations, Malkhi et al. propose in [8] to relax the intersection property of a quorum system so that "quorums" chosen according to a specified strategy intersect only with very high probability. They accordingly name these *probabilistic quorum systems*. These systems admit the possibility, albeit small, that two operations will be performed at non-intersecting quorums, in which case consistency of the system may suffer. However, even a small relaxation of consistency can yield dramatic improvements in the resilience and failure probability of the system, while the load remains essentially unchanged. Probabilistic quorum systems are thus most suitable for use when availability of operations despite the presence of faults is more important than certain consistency. This might be the case if the cost of inconsistent operations is high but not irrecoverable, or if obtaining the most up-to-

date information is desirable but not critical, while having no information may have heavier penalties.

The family of constructions suggested in [8] is as follows:

$\mathbf{W(n, \ell)}$ Let U be a universe of size n. $W(n, \ell)$, $\ell \geq 1$, is the system $\langle \mathcal{Q}, w \rangle$ where $\mathcal{Q}$ is the set system $\mathcal{Q} = \{Q \subseteq U : |Q| = \ell\sqrt{n}\}$; w is an access strategy w defined by $\forall Q \in \mathcal{Q}, w(Q) = \frac{1}{|\mathcal{Q}|}$.

The probability of choosing according to w two quorums that do not intersect is less than $e^{-\ell^2}$, and can be made sufficiently small by appropriate choice of ℓ. Since every element is in $\binom{n-1}{\ell\sqrt{n}-1}$ quorums, the load $L(W(n, \ell))$ is $\frac{\ell}{\sqrt{n}} = O(\frac{1}{\sqrt{n}})$. Because only $\ell\sqrt{n}$ servers need be available in order for some quorum to be available, $W(n, \ell)$ is resilient to $n - \ell\sqrt{n}$ crashes. The failure probability of $W(n, \ell)$ is less than $e^{-\Omega(n)}$ for all $p \leq 1 - \frac{\ell}{\sqrt{n}}$, which is asymptotically optimal. Moreover, if $\frac{1}{2} \leq p \leq 1 - \frac{\ell}{\sqrt{n}}$, this probability is provably better than any (non-probabilistic) quorum system.

Relaxing consistency can also provide dramatic improvements in environments that may experience Byzantine failures. More details can be found in [8].

Applications

Just about any fault tolerant distributed protocol, such as Paxos [5] or consensus [1] implicitly builds on quorums, typically majorities. More concretely, scalable data repositories were built, such as Fleet [9], Rambo [4], and Rosebud [13].

Cross References

► Concurrent Programming, Mutual Exclusion

Recommended Reading

1. Dwork, C., Lynch, N., Stockmeyer, L.: Consensus in the presence of partial synchrony. J. Assoc. Comput. Mach. **35**, 288–323 (1988)
2. Garcia-Molina, H., Barbara, D.: How to assign votes in a distributed system. J. ACM **32**, 841–860 (1985)
3. Gifford, D.K.: Weighted voting for replicated data. In: Proceedings of the 7th ACM Symposium on Operating Systems Principles, 1979, pp. 150–162
4. Gilbert, S.: Lynch, N., Shvartsman, A., Rambo ii: Rapidly reconfigurable atomic memory for dynamic networks. pp. 259–268. In: Proceedings if the IEEE 2003 International Conference on Dependable Systems and Networks (DNS). San Francisco, USA (2003)
5. Lamport, L.: The part-time parliament. ACM Trans. Comput. Syst. **16**, 133–169 (1998)
6. Maekawa, M.: A $\sqrt{n}$ algorithm for mutual exclusion in decentralized systems. ACM Trans. Comput. Syst. **3**(2), 145–159 (1985)
7. Malkhi, D., Reiter, M.: Byzantine quorum systems. Distrib. Comput. **11**, 203–213 (1998)
8. Malkhi, D., Reiter, M., Wool, A., Wright, R.: Probabilistic quorum systems. Inf. Comput. J. **170**, 184–206 (2001)
9. Malkhi, D., Reiter, M.K.: An architecture for survivable coordination in large-scale systems. IEEE Trans. Knowl. Data Engineer. **12**, 187–202 (2000)
10. Naor, M., Wool, A.: The load, capacity and availability of quorum systems. SIAM J. Comput. **27**, 423–447 (1998)
11. Peleg, D., Wool, A.: The availability of quorum systems. Inf. Comput. **123**, 210–223 (1995)
12. Peleg, D., Wool, A.: Crumbling walls: A class of practical and efficient quorum systems. Distrib. Comput. **10**, 87–98 (1997)
13. Rodrigues, R., Liskov, B.: Rosebud: A scalable byzantine-fault tolerant storage architecture. In: Proceedings of the 18th ACM Symposium on Operating System Principles, San Francisco, USA (2003)
14. Thomas, R.H.: A majority consensus approach to concurrency control for multiple copy databases. ACM Trans. Database Syst. **4**, 180–209 (1979)