SCIENTIFIC PAPERS, UNIVERSITY OF LATVIA, 2012. Vol. 787
COMPUTER SCIENCE AND INFORMATION TECHNOLOGIES 136–146 P.

On fault-tolerance of Grover's algorithm

Nikolajs Nahimovs, Alexander Rivosh, Dmitry Kravchenko

Abstract

Grover's algorithm is a quantum search algorithm solving the unstructured search problem of size N in $O(\sqrt{N})$ queries, while any classical algorithm needs $O(N)$ queries [1].

However, if the query transformation might fail (with probability independent of the number of steps of the algorithm), then quantum speed-up disappears: no quantum algorithm can be faster than a classical exhaustive search by more than a constant factor [6].

In this paper we study the effect of a small number of failed queries. We show that k failed queries with a very high probability change the number of actually executed steps of Grover's algorithm from l to $O\left(\frac{l}{\sqrt{k}}\right)$.

1 Introduction

Grover's algorithm is a quantum search algorithm solving the unstructured search problem in about $\frac{\pi}{4}\sqrt{N}$ queries [1]. It has been analysed in great detail. The analysis has been mainly focused on the optimality and generalization of the algorithm [4, 2, 3], as well as on fault-tolerance of the algorithm to various kind of physical errors, such as decoherence or random imperfections in either diffusion transformations or black box queries [8, 7].

In this paper we study fault-tolerance of Grover's algorithm to logical faults, in our case failure of one or more query transformations. Regev and Schiff have shown [6] that if the query transformation fails with a fixed probability (independent of the number of steps of the algorithm), then quantum speed-up disappears: no quantum algorithm can be faster than a classical exhaustive search by more than a constant factor.

We find it interesting to understand what happens if only a small number of failed queries is allowed. We show that even a single failed query can stop the algorithm from finding any of marked elements. Remarkably, this property does not depend on a number of marked elements. This makes the quantum case completely different from the classical case.

A failure of a single or multiple query transformations results in a number of steps not being executed. We show that k failed queries with a very high probability change the number of actually executed steps of Grover's algorithm from l to $O\left(\frac{l}{\sqrt{k}}\right)$.

2 Grover's algorithm

Suppose we have an unstructured search space consisting of N elements $x_1, \ldots, x_N \in \{0, 1\}$. Our task is to find $x_i = 1$ or to conclude that no such x exists.

Grover's algorithm starts with a state $|\psi_{start}\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |i\rangle$. Each step of the algorithm consists of two transformations: Q – query transformation defined as $Q|i\rangle = (-1)^{x_i}|i\rangle$ and D – the inversion about average, defined as:

$$D = \begin{bmatrix} -1+\frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & -1+\frac{2}{N} & \cdots & \frac{2}{N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & -1+\frac{2}{N} \end{bmatrix}.$$

The state of the algorithm after t steps is $|\psi_t\rangle = (DQ)^t|\psi_{start}\rangle$.

Grover's algorithm has been analysed in detail and many facts about it are known. If there is one marked element, the probability of finding it by measuring $|\psi_t\rangle$ reaches $1 - o(1)$ for $t = O(\sqrt{N})$. If there are k marked elements, the probability of finding one of them by measuring $|\psi_t\rangle$ reaches $1 - o(1)$ for $t = O(\sqrt{N/k})$.

3 Grover's algorithm with errors

In this section we study the effect of small number of errors (omitted query transformations) on the transformation sequence of the algorithm. We show that an omission of a single or multiple query transformations is equivalent to replacing a number of steps (DQ transformation pairs) with an identity transformation, that is performing a smaller number of steps.

Let l be a number of steps of the algorithm. We show that $k \ll l$ uniformly distributed independent errors change the transformation sequence of the algorithm from $(DQ)^l$ to $(DQ)^L$, where L is the random variable with expectation $O(\frac{l}{k})$ and standard deviation $O\left(\frac{l}{\sqrt{k}}\right)$. Therefore, with a very high probability the length of the resulting transformation sequence is $O\left(\frac{l}{\sqrt{k}}\right)$ [5].

Our further analysis is focused on the omission of the query transformation Q. However a very similar analysis can be done for an omission of the D transformation.

3.1 Omitting a single query

The transformation sequence of Grover's algorithm is

$$DQ\,DQ\,\ldots\,DQ = (DQ)^l.$$

If we omit a single query transformation the sequence changes to

$$(DQ)^{l_1} D (DQ)^{l_2},$$

where $l_1 + l_2 + 1 = l$, or

$$D(QD)^{l_1}(DQ)^{l_2}.$$

As $DD = QQ = I$ (this follows from the definitions of D and Q transformations), the shortest subsequence will cancel a part of the longest subsequence. More precisely

$$l_1 \geq l_2: \quad D(QD)^{l_1}(DQ)^{l_2} = D(QD)^{l_1 - l_2}$$

$$l_1 < l_2: \quad D(QD)^{l_1}(DQ)^{l_2} = D(DQ)^{l_2 - l_1}.$$

Thus, a single omitted query transformation changes the transformation sequence of the algorithm from $(DQ)^l$ to $(DQ)^{O(|l_1 - l_2|)}$, decreasing the number of successful steps.

Suppose the query transformation can be omitted on a random algorithm step, that is l_1 is a uniformly distributed random variable. The length of the resulting transformation sequence will also be a random variable. Simple calculations show that it has mean $\frac{l}{2} + O(1)$ and variance $\frac{l^2}{12} + O(l)$.

Corollary

A single omitted query transformation on the average will twice decrease the number of successful steps of the algorithm (or will twice increase the average running time of the algorithm).

If the query transformation will be omitted right in the middle of the transformation sequence ($l_1 = l_2$), the number of successful steps will be 0. That is the algorithm will leave the initial state unchanged.

3.2 Omitting multiple queries

The transformation sequence of the algorithm is

$$DQ\,DQ\,\ldots\,DQ = (DQ)^l.$$

If we omit $k-1$ query transformations, the sequence changes to

$$(DQ)^{l_1}D(DQ)^{l_2}D\ldots(DQ)^{l_{k-1}}D(DQ)^{l_k},$$

where $l_1+l_2+\ldots+l_k+(k-1)=l$. By regrouping the brackets we will get

$$(DQ)^{l_1}DD(QD)^{l_2}(DQ)^{l_3}DD(DQ)^{l_4}\ldots =$$

$$(DQ)^{l_1}(QD)^{l_2}(DQ)^{l_3}(DQ)^{l_4}\ldots$$

Transformations Q and D have the following commutativity property:

$$(QD)^i(DQ)^j = (DQ)^j(QD)^i.$$

Thus, the sequence can be rewritten as

$$(DQ)^{l_1+l_3+\ldots}\,(QD)^{l_2+l_4+\ldots}\,.$$

Therefore, k omitted query transformations change the transformation sequence of the algorithm from $(DQ)^l$ to $(DQ)^{O(l_1-l_2+l_3-l_4+\ldots\pm l_k)}$.

We will show that in the continuous approximation case (positions of errors have continuous uniform distributions and $l_1+l_2+\ldots+l_k=l$) the length of the resulting transformation sequence is a random variable with mean 0 (even k) or $\frac{l}{k}$ (odd k) and variance $O\left(\frac{l^2}{k}\right)$. These values perfectly agree with numerical experiment results for discrete case.

Proof of the main result

Suppose we have $k-1$ independent random variables $X_1, X_2, \ldots, X_{k-1}$. Each X_i is uniformly distributed between 0 and l. That is the probability density function of X_i is

$$f_{X_i}(x) = \begin{cases} \frac{1}{l} & x \in [0,l] \\ 0 & x \notin [0,l] \end{cases}$$

and the cumulative distribution function is

$$F_{X_i}(x) = \begin{cases} 0 & x < 0 \\ \frac{x}{l} & x \in [0, l] \\ 1 & x > l \end{cases} .$$

The above random variables split the segment $[0, l]$ into k subsegments $l_1, l_2, \ldots, l_k$. The length of each subsegment is also a random variable.

Let us focus on the subsegment l_1. Probability that $l_1 \leq x$ is the probability that at least one of $X_i \leq x$. Thus, the cumulative distribution function of l_1 is

$$F_{l_1} = 1 - (1 - F_{X_1})(1 - F_{X_2}) \ldots (1 - F_{X_{k-1}})$$

or

$$F_{l_1}(x) = \begin{cases} 0 & x < 0 \\ 1 - (1 - \frac{x}{l})^{k-1} & x \in [0, l] \\ 1 & x > l \end{cases} .$$

The probability density function of l_1 is

$$f_{l_1}(x) = \begin{cases} \frac{k-1}{l}(1 - \frac{x}{l})^{k-2} & x \in [0, l] \\ 0 & x \notin [0, l] \end{cases} .$$

Knowing the probability density function of l_1, we can calculate its mean and variance by using the following formulae:

$$E[X] = \int_{-\infty}^{\infty} x \cdot f_X(x) dx$$

$$E[X^2] = \int_{-\infty}^{\infty} x^2 \cdot f_X(x) dx$$

$$Var[X] = E[X^2] - E[X]^2.$$

We leave out the details of calculation of integrals and give the results.

$$E[l_1] = \int_{-\infty}^{\infty} x \cdot f_{l_1}(x) dx = \int_0^l x \frac{k-1}{l} (1 - \frac{x}{l})^{k-2} dx = \frac{l}{k}$$

$$E[(l_1)^2] = \int_{-\infty}^{\infty} x^2 \cdot f_{l_1}(x)dx = \int_0^l x^2 \frac{k-1}{l}(1-\frac{x}{l})^{k-2}dx = \frac{2l^2}{k(k+1)}$$

$$Var[l_1] = \frac{2l^2}{k(k+1)} - \left(\frac{l}{k}\right)^2 = \frac{k-1}{k+1} \cdot \left(\frac{l}{k}\right)^2.$$

It is easy to see that all l_i subsegments have the same mean and variance. This follows from the fact that all X_i are independent and are uniformly distributed. We should also note that, although X_i are independent random variables, l_i are not independent (the length of one subsegment increases as other decreases and vice versa) .

Now let us focus on $L = l_1 - l_2 + l_3 - ... \pm l_k$. First we will calculate the mean of L. We will use the following well known formulae:

$$E[-X] = -E[X]$$

$$E[X_1 + \ldots + X_k] = E[X_1] + \ldots + E[X_k].$$

As all l_i have the same mean

$$E[L] = E[l_1] - E[l_2] + \ldots \pm E[l_k] = \begin{cases} 0 & k = 2m \\ \\ \frac{l}{k} & k = 2m+1 \end{cases}.$$

Now we will calculate the variance of L. As l_i subsegments are correlated, we have to use the following formula:

$$Var[X_1 + \ldots + X_k] = \sum_{i=1}^{k} Var[X_i] + \sum_{i \neq j} Cov[X_i, X_j]$$

The subsegment covariance can be easily calculated from the following trivial fact:

$$Var(l_1 + \ldots + l_k) = 0.$$

This is so because $l_1 + \ldots + l_k$ is always equal to l. Using the above formula, we will get:

$$Var[l_1 + \ldots + l_k] = \sum_{i=1}^{k} Var[l_i] + \sum_{i \neq j} Cov[l_i, l_j] = 0$$

or

$$\sum_{i=1}^{k} Var[l_i] = -\sum_{i \neq j} Cov[l_i, l_j].$$

As all l_i have the same mean and variance, they will also have the same covariances $Cov[l_i, l_j]$. Using this fact, we will get

$$k \cdot Var[l_i] = -k(k-1) \cdot Cov[l_i, l_j]$$

or

$$Cov[l_i, l_j] = -\frac{1}{k-1} \cdot Var[l_i] = -\frac{1}{k+1} \cdot \left(\frac{l}{k}\right)^2.$$

Now let us return to the variance of L:

$$Var[L] = \sum_{i=1}^{k} Var[l_i] \pm \sum_{i \neq j} Cov[l_i, l_j].$$

Covariance sign will depend on l_i and l_j signs (whether they are the same or not). More precisely:

$$Cov[-X, Y] = Cov[X, -Y] = -Cov[X, Y]$$

$$Cov[-X, -Y] = Cov[X, Y].$$

If $k = 2m$, then m subsegments have plus sign and m subsegments have minus sign. There are $2m(m-1)$ subsegment pairs with the same signs and $2m^2$ subsegment pairs with opposite signs (we should count both (l_i, l_j) and (l_j, l_i) pairs). Thus, we can rewrite the formula as:

$$Var[L] = k \cdot Var[l_i] + Cov[l_i, l_j] \cdot \left(2m(m-1) - 2m^2\right) =$$

$$= k \cdot Var[l_i] - k \cdot Cov[l_i, l_j] =$$

$$= k \cdot Var[l_i] + \frac{k}{k-1} \cdot Var[l_i] =$$

$$= k \cdot Var[l_i] \cdot \frac{k}{k-1}.$$

If $k = 2m+1$, then $m+1$ subsegments have plus sign and m subsegments have minus sign. There are $(m+1)m + m(m-1) = 2m^2$ subsegment pairs with the same signs and $2(m+1)m$ subsegment pairs with opposite signs. Thus, we can rewrite the formula as:

$$Var[L] = k \cdot Var[l_i] + Cov[l_i, l_j] \cdot \left(2m^2 - 2m(m-1)\right) =$$

$$= k \cdot Var[l_i] + (k-1) \cdot Cov[l_i, l_j] =$$
$$= k \cdot Var[l_i] - Var[l_i] =$$
$$= k \cdot Var[l_i] \cdot \frac{k-1}{k}.$$

Using O notation, we can rewrite both cases as $O(k) \cdot Var[l_i] = O\left(\frac{l^2}{k}\right)$. This ends the proof.

Corollary

We have shown that $k-1$ omitted query transformations change the length of the transformation sequence of the algorithm from l to a random variable with mean 0 (even k) or $\frac{l}{k}$ (odd k) and variance $O\left(\frac{l^2}{k}\right)$.

Using Chebyshev's inequality, we can show that with 96% probability L value will lie within five standard deviations from its mean [5]. For large k (but still $k \ll l$) even a more tight bound can be applied. In the next section we will show that the probability distribution of L for large k is close to the normal distribution. Thus, with 99.7% probability L will lie within three standard deviations from the mean.

Therefore, with a very high probability the length of the resulting transformation sequence changes from l to $O\left(\frac{l}{\sqrt{k}}\right)$.

4 Probability distribution of the median

In the previous section we have studied the following model. We have independent random variables $X_1, X_2, \ldots, X_{k-1}$. Each X_i is uniformly distributed between 0 and l. The random variables split the segment $[0, l]$ into k subsegments $l_1, l_2, \ldots, l_k$. Our task was to estimate $L = l_1 - l_2 + l_3 - l_4 + \ldots \pm l_k$. Due to symmetry of l_i, L is equal to $\frac{l}{2} - X_m$, where X_m is the median of $X_1, X_2, \ldots, X_{k-1}$, that is the point separating the higher half of the points from the lower half of the points.

In this section we will show that for a large number of uniformly distributed random variables (points), the probability distribution of the median is close to the normal distribution.

2k + 1 points

Let us consider a real number interval $[-N; N]$ and $2k+1$ random points, each having a uniform distribution. Median is the point number $k+1$.

Probability density of the median at position x, which is at the distance $|x|$ from 0, can be expressed by the formula

$$f_M(x) = \frac{(N-x)^k (N+x)^k}{(2N)^{2k+1}} \times \frac{(2k+1)!}{k!k!} = \frac{\left(N^2-x^2\right)^k (2k)!\,(2k+1)}{(2N)^{2k+1}\,k!k!} \quad (1)$$

Using the Stirling approximation we can rewrite (1):

$$F_M(x) \approx \frac{\left(N^2-x^2\right)^k \sqrt{4\pi k}\left(\frac{2k}{e}\right)^{2k}(2k+1)}{(2N)^{2k+1}\sqrt{2\pi k}\left(\frac{k}{e}\right)^k \sqrt{2\pi k}\left(\frac{k}{e}\right)^k} = \frac{\left(N^2-x^2\right)^k (2k+1)}{2N^{2k+1}\sqrt{\pi k}} \quad (2)$$

$$= \frac{\left(1-\frac{x^2}{N^2}\right)^k (2k+1)}{2N\sqrt{\pi k}} \quad (3)$$

For large k we can approximate $2k+1$ with $2k$:

$$f_M(x) \approx \frac{\left(1-\frac{x^2}{N^2}\right)^k \sqrt{k}}{N\sqrt{\pi}} \quad (4)$$

For small $\frac{x}{N}$ values (4) can be approximated (applying $1-z \approx e^{-z}$) by

$$f_M(x) \approx \frac{\left(e^{-\frac{x^2}{N^2}}\right)^k \sqrt{k}}{N\sqrt{\pi}} = \frac{\sqrt{k}}{N\sqrt{\pi}} e^{-k\frac{x^2}{N^2}} \quad (5)$$

which corresponds to the normal distribution with mean 0 and variance $\frac{N^2}{2k}$.

2k points

Let us consider a real number interval $[-N;N]$ and $2k$ random points, each having a uniform distribution. Median is the point number k.

Probability density of the median at position X, which is at the distance $|X|$ from 0, can be expressed by the formula

$$f_M(x) = \frac{(N-x)^{k-1} (N+x)^k}{(2N)^{2k}} \times \frac{(2k)!}{(k-1)!k!} = \frac{\left(N^2-x^2\right)^k k\,(2k)!}{(2N)^{2k}\,(N-x)k!k!} \quad (6)$$

Using the Stirling approximation we can rewrite (6):

$$f_M(x) \approx \frac{\left(N^2-x^2\right)^k k\sqrt{4\pi k}\left(\frac{2k}{e}\right)^{2k}}{(2N)^{2k}(N-x)\sqrt{2\pi k}\left(\frac{k}{e}\right)^k\sqrt{2\pi k}\left(\frac{k}{e}\right)^k} = \frac{\left(N^2-x^2\right)^k\sqrt{k}}{N^{2k}(N-x)\sqrt{\pi}} \quad (7)$$

$$= \frac{\left(1-\frac{x^2}{N^2}\right)^k\sqrt{k}}{\left(1-\frac{x}{N}\right)N\sqrt{\pi}} \quad (8)$$

For small $\frac{X}{N}$ values (8) can be approximated (applying $1-z \approx e^{-z}$) by

$$f_M(X) \approx \frac{\left(e^{-\frac{x^2}{N^2}}\right)^k\sqrt{k}}{\left(e^{-\frac{x}{N}}\right)N\sqrt{\pi}} = \frac{\sqrt{k}}{N\sqrt{\pi}}e^{-k\frac{x^2}{N^2}+\frac{x}{N}} \quad (9)$$

By multiplying (9) with $e^{-\frac{1}{4k}}$, which for large k is close to 1, we will get

$$f_M(x) \approx \frac{\sqrt{k}}{N\sqrt{\pi}}e^{-k\frac{x^2}{N^2}+\frac{x}{N}-\frac{1}{4k}} = \frac{\sqrt{k}}{N\sqrt{\pi}}e^{-k\frac{(x-\frac{N}{2k})^2}{N^2}} \quad (10)$$

which corresponds to the normal distribution with mean $\frac{N}{2k}$ and variance $\frac{N^2}{2k}$.

5 Conclusions

We have shown that even a single failed query can change the resulting transformation sequence of the algorithm to an identity transformation. On the average, a single failed query will twice decrease the length of the resulting transformation sequence. In case of k failed queries with a very high probability the length of the resulting transformation sequence will be decreased $O(\sqrt{k})$ times.

Similar argument can be applied to a wide range of other quantum query algorithms, such as amplitude amplification, some variants of quantum walks and NAND formula evaluation, etc. That is to any quantum query algorithm for which the transformation X applied between queries has the property $X^2 = I$.

References

[1] Lov Grover
A fast quantum mechanical algorithm for database search.
Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC), May 1996, pages 212-219
arXiv:quant-ph/9605043

[2] Cristof Zalka
Grovers quantum searching algorithm is optimal.
Physical Review A, 60:2746-2751, 1999.
arXiv:quant-ph/9711070

[3] G. Brassard et al.
Quantum Amplitude Amplification and Estimation.
arXiv:quant-ph/0005055

[4] C. Bennett et al.
Strengths and Weaknesses of Quantum Computing.
SIAM Journal on Computing (special issue on quantum computing) volume 26, number 5, pages 1510-1523.
arXiv:quant-ph/9701001

[5] R. Motwani, P. Raghavan
Randomized Algorithms.
Cambridge University Press, 1994.

[6] O. Regev, L. Schiff.
Impossibility of a Quantum Speed-up with a Faulty Oracle.
Proceedings of ICALP'2008, Lecture Notes in Computer Science, 5125:773-781, 2008.
`http://www.cs.tau.ac.il/~odedr/papers/faultygrover.pdf`

[7] D. Shapira et al.
Effect of unitary noise on Grover's quantum search algorithm.
Phys. Rev. A volume 67, issue 4, April 2003

[8] Pil H. Song, ILki Kim.
Computational leakage: Grover's algorithm with imperfections.
The European Physical Journal D - Atomic, Molecular, Optical and Plasma Physics volume 23, Number 2, May 2003, pages 299-303.
arXiv:quant-ph/0010075

[9] John Watrous
Quantum Computation.
Lecture course "CPSC 519/619", University of Calgary, 2006.
http://www.cs.uwaterloo.ca/ watrous/lecture-notes.html