Optimization of partial search

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **38** (2005) L731–L738 doi:10.1088/0305-4470/38/44/L02

LETTER TO THE EDITOR

Optimization of partial search

Vladimir E Korepin

C N Yang Institute for Theoretical Physics, State University of New York at Stony Brook,
Stony Brook, NY 11794-3840, USA

E-mail: korepin@insti.physics.sunysb.edu

Received 15 July 2005, in final form 6 September 2005
Published 19 October 2005
Online at stacks.iop.org/JPhysA/38/L731

Abstract

A quantum Grover search algorithm can find a target item in a database faster than any classical algorithm. One can trade accuracy for speed and find a part of the database (a block) containing the target item even faster; this is partial search. A partial search algorithm was recently suggested by Grover and Radhakrishnan. Here we optimize it. Efficiency of the search algorithm is measured by the number of queries to the oracle. The author suggests a new version of the Grover–Radhakrishnan algorithm which uses a minimal number of such queries. The algorithm can run on the same hardware that is used for the usual Grover algorithm.

PACS numbers: 03.67.−a, 03.67.Lx

1. Introduction

Database search has many applications and is widely used. Grover discovered a quantum algorithm that searches faster than a classical algorithm [1]. It consists of repetition of the Grover iteration G_1. We shall call it global iteration (see (5)). The number of repetitions is

$$j_{\text{full}} = \frac{\pi}{4}\sqrt{N} \tag{1}$$

for a database with large number of entries N. After j_{full} the algorithm finds the target item.

Sometimes it is sufficient to find an approximate location of the target item. A partial search considers the following problem: a database is separated into K blocks, of a size $b = N/K$. We want to find a block with the target item, not the target item itself. The first quantum algorithm for a partial search was suggested by Grover and Radhakrishnan [7]. They showed that classical partial search takes $\sim(N - b)$ queries, but a quantum algorithm takes only $\sim(\sqrt{N} - \mathbf{c}\sqrt{b})$ queries. Here $\mathbf{c}$ is a positive coefficient. This algorithm uses several global iterations $G_1^{j_1}$ and then several local iterations $G_2^{j_2}$ (see (8)). Local searches are made in each block separately in parallel. Here we optimize this algorithm: the number of queries to the oracle is minimized, the coefficient $\mathbf{c}$ is increased. An exact expression for the number of queries necessary to find the target block is given by formulae (17), (20) and (23) below. We also consider some other partial search algorithms: different sequences of local and global searches. The efficiency of search algorithms is measured by the number of queries to the oracle; we call this the number of iterations. The lower bound is at the end of the paper. Partial search can use the same hardware as the full search. The preliminary version of this paper can be found on the internet archive [8].

2. Partial search

2.1. Global iterations

First, let us recall the full Grover search. We consider a database with one target item. The aim of the Grover algorithm is to identify a target state $|t\rangle$ among an unordered set of N states. This is achieved by repeating global iteration which is defined in terms of two operators. The first changes the sign of the target state $|t\rangle$ only:

$$I_t = \hat{I} - 2|t\rangle\langle t|, \qquad \langle t|t\rangle = 1, \tag{2}$$

where $\hat{I}$ is the identity operator. The second operator,

$$I_{s_1} = \hat{I} - 2|s_1\rangle\langle s_1|, \tag{3}$$

changes the sign of the uniform superposition of all basis states $|s_1\rangle$,

$$|s_1\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle, \qquad \langle s_1|s_1\rangle = 1. \tag{4}$$

The *global iteration* is defined as a unitary operator

$$G_1 = -I_{s_1}I_t. \tag{5}$$

We shall use the eigenvectors of G_1:

$$G_1\left|\psi_1^{\pm}\right\rangle = \lambda_1^{\pm}\left|\psi_1^{\pm}\right\rangle, \qquad \lambda_1^{\pm} = \exp[\pm 2\mathrm{i}\theta_1],$$
$$\left|\psi_1^{\pm}\right\rangle = \frac{1}{\sqrt{2}}|t\rangle \pm \frac{\mathrm{i}}{\sqrt{2}}\left(\sum_{\substack{x=0\\x\neq t}}^{N-1}\frac{|x\rangle}{\sqrt{(N-1)}}\right). \tag{6}$$

They were found in [4]. The angle θ_1 is defined by

$$\sin^2\theta_1 = \frac{1}{N}. \tag{7}$$

2.2. Grover–Radhakrishnan algorithm for partial search

The partial search algorithm is designed to find a block with the target item: the target block. We shall call other blocks non-target blocks. The algorithm uses j_1 global iteration and j_2 local iterations. Local iterations are Grover iterations for each block:

$$G_2 = -I_{s_2}I_t. \tag{8}$$

I_t is given by (2), but I_{s_2} is different. The action of the operator I_{s_2} in an individual block can be represented as

$$I_{s_2}\big|_{\text{block}} = \hat{I}|_{\text{block}} - 2|s_2\rangle\langle s_2|, \qquad |s_2\rangle = \frac{1}{\sqrt{b}}\sum_{\text{the block}}|x\rangle. \tag{9}$$

For the whole database we should write I_{s_2} as a direct sum of the operators (9) over all blocks.

Relevant eigenvectors of G_2 are

$$G_2\big|\psi_2^{\pm}\big\rangle = \lambda_2^{\pm}\big|\psi_2^{\pm}\big\rangle, \qquad \lambda_2^{\pm} = \exp[\pm 2\mathrm{i}\theta_2], \qquad \big|\psi_2^{\pm}\big\rangle = \frac{1}{\sqrt{2}}|t\rangle \pm \frac{\mathrm{i}}{\sqrt{2}}|\mathrm{ntt}\rangle. \tag{10}$$

Here the $|\mathrm{ntt}\rangle$ is a normalized sum of all non-target items in the target block:

$$|\mathrm{ntt}\rangle = \frac{1}{\sqrt{b-1}} \sum_{\substack{x\neq t\\ \text{target block}}} |x\rangle, \qquad \langle\mathrm{ntt}|\mathrm{ntt}\rangle = 1. \tag{11}$$

We shall need an angle θ_2 given by

$$\sin^2\theta_2 = \frac{K}{N} = \frac{1}{b}. \tag{12}$$

Local iteration does not change non-target blocks. Inside the target block it acts similar to the usual Grover search. After several global iterations and several locals we still have to apply one more global iteration. The partial search algorithm creates a vector

$$|d\rangle = G_1 G_2^{j_2} G_1^{j_1} |s_1\rangle. \tag{13}$$

In the state $|d\rangle$ the amplitudes of all items in non-target blocks are the same. Using the eigenvectors of local (10) and global iterations from (6), we can calculate this amplitude and require that it vanish:

$$\begin{aligned}\frac{-N}{\sqrt{N-1}}\left(\frac{1}{2}-\frac{1}{K}\right)&\cos((2j_1+1)\theta_1)\\ &= \cos(2j_2\theta_2)\sin((2j_1+1)\theta_1) + \sqrt{\frac{b-1}{N-1}}\sin(2j_2\theta_2)\cos((2j_1+1)\theta_1)\\ &\quad - \sqrt{b-1}\sin(2j_2\theta_2)\sin((2j_1+1)\theta_1) + \frac{b-1}{\sqrt{N-1}}\cos(2j_2\theta_2)\cos((2j_1+1)\theta_1).\end{aligned} \tag{14}$$

This equation guarantees that the amplitude of each item in each non-target block vanishes. Now we can *measure*. In the simplest case $N = 2^n$ and $K = 2^k$, so we can label blocks by k qubits (items inside of a block are labelled by $n-k$ qubits). We measure only k block qubits and *find the target block*. We shall choose the numbers of iterations j_1 and j_2 by minimizing the total number of iterations $j_1 + j_2$.

To see universal features we consider the limit when each block is very large, $b \to \infty$; this makes the total number of items in the whole database also large $N = Kb \to \infty$. The expression for angles (7) and (12) simplifies:

$$\theta_1 = 1/\sqrt{N}, \qquad \theta_2 = 1/\sqrt{b}.$$

It was shown in [7] that the numbers of iterations scale as

$$j_1 = \frac{\pi}{4}\sqrt{N} - \eta\sqrt{b}, \qquad j_2 = \alpha\sqrt{b}, \qquad \mathbf{c} = \eta - \alpha. \tag{15}$$

Here η and α are parameters of the order of 1 (they have a limit). For large blocks $b \to \infty$ equation (14) can be simplified to

$$\tan\left(\frac{2\eta}{\sqrt{K}}\right) = \frac{2\sqrt{K}\sin 2\alpha}{K - 4\sin^2\alpha}. \tag{16}$$

2.3. Minimization of total number of iterations

Let us minimize the number of queries to the oracle (number of iterations): $S = j_1 + j_2 + 1 \to \frac{\pi}{4}\sqrt{N} - \mathbf{c}\sqrt{b}$. Here $\mathbf{c} = \eta - \alpha$. To optimize the algorithm we have to minimize $(\alpha - \eta)$ having in mind constraint (16). The author found the optimal values of α and η, they depend on K; let us distinguish them by a subindex α_K and η_K. The minimum number of queries is achieved at

$$\tan\frac{2\eta_K}{\sqrt{K}} = \frac{\sqrt{3K-4}}{K-2}, \qquad \cos 2\alpha_K = \frac{K-2}{2(K-1)}, \qquad \mathbf{c} = \eta_K - \alpha_K. \tag{17}$$

This describes the optimal version of the Grover–Radhakrishnan algorithm.

Let us study the dependence on the number of blocks K; α_K monotonically decreases with K:

$$\alpha_2 = \frac{\pi}{4} \geqslant \alpha_K \geqslant \frac{\pi}{6} = \alpha_\infty \qquad K = 2 \quad \to \quad K = \infty. \tag{18}$$

In the case of two large blocks $K = 2$ minimization of the number of queries of partial search algorithm gives $\alpha_2 = \pi/4$, $\eta_2 = \pi/2\sqrt{2}$. This means that for $K = 2$ algorithm skips the global iterations and makes a full local search in each block: $j_1 = 0$, $j_2 = (\pi/4)\sqrt{b}$. For three blocks or more, $3 \leqslant K$, the algorithm makes less than full search of each block (locally). Now let us analyse the number of global iterations:

$$j_1 = \left(\frac{\pi}{4} - \frac{\eta_K}{\sqrt{K}}\right)\sqrt{N} > 0, \qquad \frac{\mathrm{d}}{\mathrm{d}K}\left(\frac{\eta_K}{\sqrt{K}}\right) < 0, \qquad \frac{\mathrm{d}j_1}{\mathrm{d}K} > 0, \qquad \text{for} \quad 3 \leqslant K. \tag{19}$$

The parameter η_K decreases monotonically from $\eta_2 = \pi/(2\sqrt{2})$ to $\eta_\infty = \sqrt{3/4}$, when K increases.

The difference $\alpha_K - \eta_K$ monotonically decreases with K. Numerical values of α_K and and η_K for different numbers of blocks are

$$\begin{array}{lll}
\alpha_2 \approx 0.7854, & \eta_2 \approx 1.1107, & \alpha_2 - \eta_2 \approx -0.3253 \\
\alpha_3 \approx 0.659\,06, & \eta_3 \approx 0.9961, & \alpha_3 - \eta_3 \approx -0.337\,04 \\
\alpha_4 \approx 0.6155, & \eta_4 \approx 0.9553, & \alpha_4 - \eta_4 \approx -0.3398 \\
\alpha_5 \approx 0.5932, & \eta_5 \approx 0.9341, & \alpha_5 - \eta_5 \approx -0.3409 \\
\alpha_\infty \approx 0.5236, & \eta_\infty \approx 0.866, & \alpha_\infty - \eta_\infty \approx -0.3424.
\end{array}$$

These are solutions of equation (17). These parameters define the number of the iterations:

$$j_1 = \frac{\pi}{4}\sqrt{N} - \eta_K\sqrt{b}, \qquad j_2 = \alpha_K\sqrt{b}, \qquad S_K \approx j_1 + j_2 \to \frac{\pi}{4}\sqrt{N} + (\alpha_K - \eta_K)\sqrt{b}. \tag{20}$$

We can compare this with the full search in *randomly* picked $K - 1$ blocks, which takes

$$R_K = \frac{\pi}{4}\sqrt{\frac{K-1}{K}}\sqrt{N} \tag{21}$$

iterations (see (1)). For two blocks partial search and random pick takes the same number of queries: $R_2 = S_2 = [\pi/4]\sqrt{N/2}$. For more blocks partial search is faster:

$$\begin{array}{ll}
R_3 = 0.641\sqrt{N}, & S_3 = 0.59\sqrt{N}, \\
R_4 = 0.68\sqrt{N}, & S_4 = 0.586\sqrt{N}, \\
R_5 = 0.702\sqrt{N}, & S_5 = 0.63\sqrt{N}.
\end{array}$$

Here we compared the random pick algorithm with the partial search algorithm using: $S_K = (\pi/4 + [\alpha_K - \eta_K]/\sqrt{K})\sqrt{N}$. We see that starting from $K = 3$ partial search algorithm works faster than the random pick. As the number of blocks increases the advantage becomes more essential.

But for a large K we should compare the partial search algorithm with its *interrupted* version: if we make only global iterations of the partial search algorithm and measure the wavefunction of the database, the probability to find the target item is

$$p_t = \sin^2((2j_1 + 1)\theta_1) = \frac{(K-2)^2}{K(K-1)}. \tag{22}$$

It monotonically increases with K.

Let us solve equation (17) explicitly for a large K:

$$\alpha_K \to \frac{\pi}{6} + \frac{1}{2\sqrt{3}K} + \frac{5\sqrt{3}}{(6K)^2}, \qquad \eta_K \to \frac{\sqrt{3}}{2} + \frac{1}{2\sqrt{3}K} + \frac{11\sqrt{3}}{90K^2}, \qquad K \to \infty.$$

Corrections to these expressions are of the order $1/K^3$. The total number of queries is

$$S_K \to \frac{\pi}{4}\sqrt{N} + (\alpha_K - \eta_K)\sqrt{b}, \qquad -\mathbf{c} = \alpha_K - \eta_K = \frac{\pi}{6} - \sqrt{\frac{3}{4}} + \frac{1}{5\sqrt{3}(2K)^2} < 0. \tag{23}$$

Random pick (21) takes more queries:

$$R_K \to \frac{\pi}{4}\sqrt{N} - \left(\frac{\pi}{8\sqrt{K}}\right)\sqrt{b}, \qquad K \to \infty. \tag{24}$$

As for the *interrupted* version of the algorithm in the limit of $K \to \infty$, the probability to find the target item by measuring after global iterations is close to certainty: $p_t = 1 - 3/K$, $K \to \infty$ (see (22)). The partial search algorithm is efficient for a limited number of blocks only: $3 \leqslant K \leqslant 3/(1 - p_t)$. If we choose the probability $p_t = 0.9$, then the partial search algorithm works well in the region:

$$3 \leqslant K \leqslant 30. \tag{25}$$

The version of the partial search algorithm described here is little faster than the original Grover–Radhakrishnan algorithm [7]: in the expression for the total number of iterations S_K the coefficient $\mathbf{c} = \eta_K - \alpha_K$ in (23) and (20) is from 1% to 3% larger (depending on K). But our version uses the absolute minimum of queries to the oracle.

3. Other sequences of searches

The Grover–Radhakrishnan algorithm used global–local sequence of searches. It is interesting to consider other partial search algorithms, for example other sequences of local and global searches. First, let us note that if we start from the uniform superposition of all items in the whole database $|s_1\rangle$, see (4), then after several local and global iterations the vector describing a current state of the data base will stay in a three-dimensional subspace. To clarify: (i) the amplitudes of all items in non-target blocks will always remain the same; (ii) the amplitudes of all non-target items in the target block also will remain the same. The orthonormal basis in the three-dimensional space is formed by the target item $|t\rangle$, the sum of all non-target items in the target block $|\text{ntt}\rangle$, defined in (11) and $|u\rangle$:

$$|u\rangle = \frac{1}{\sqrt{b(K-1)}} \sum_{\substack{\text{all items in all} \\ \text{non-target blocks}}} |x\rangle. \tag{26}$$

In this basis j_2 repetitions of the local iteration (8) can be represented as a three-dimensional matrix:

$$G_2^{j_2} = \begin{pmatrix} \cos(2j_2\theta_2) & \sin(2j_2\theta_2) & 0 \\ -\sin(2j_2\theta_2) & \cos(2j_2\theta_2) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{This is a local search.} \tag{27}$$

Now let us turn our attention to global iterations (5); j_1 repetitions of the global iterations can be represented as

$$G_1^{j_1} = \begin{pmatrix} \cos(2j_1\theta_1), & \sin(2j_1\theta_1)\sin\gamma, & \sin(2j_1\theta_1)\cos\gamma \\ -\sin(2j_1\theta_1)\sin\gamma, & (-1)^{j_1}\cos^2\gamma + \cos(2j_1\theta_1)\sin^2\gamma, & \sin\gamma\cos\gamma((-1)^{j_1+1} + \cos(2j_1\theta_1)) \\ -\sin(2j_1\theta_1)\cos\gamma, & \sin\gamma\cos\gamma((-1)^{j_1+1} + \cos(2j_1\theta_1)), & (-1)^{j_1}\sin^2\gamma + \cos(2j_1\theta_1)\cos^2\gamma \end{pmatrix}. \tag{28}$$

This is a global search. Now we can represent a sequence of local and global searches as a product of three-dimensional matrices. Let us consider two examples:

- Global–Local–Global sequence of searches
  We start with the uniform superposition of all items in the database $|s_1\rangle$ (4), apply j_1 global iterations (5), then j_2 local iterations (8) and then j_3 global iterations. The state of the database will become:

  $$G_1^{j_3} G_2^{j_2} G_1^{j_1} |s_1\rangle. \tag{29}$$

  Let us require that the amplitudes of all items in non-target blocks vanish: $\langle u|G_1^{j_3} G_2^{j_2} G_1^{j_1}|s_1\rangle = 0$. Explicitly multiplying matrices we can write this equation in the form:

  $$\begin{aligned} \cos(2j_2\theta_2)\{&-\cos\gamma\sin(2j_3\theta_1)\sin(2j_1\theta_1) + \sin^2\gamma\cos(2j_1\theta_1)\cos\gamma[(-1)^{j_3+1} \\ &+ \cos(2j_3\theta_1)]\} + \sin(2j_2\theta_2)\{-\sin\gamma\cos\gamma\sin(2j_3\theta_1)\cos(2j_1\theta_1) \\ &- \sin\gamma\cos\gamma\sin(2j_1\theta_1)[(-1)^{j_3+1} + \cos(2j_3\theta_1)]\} \\ &+ \cos\gamma\cos(2j_1\theta_1)[(-1)^{j_3}\sin^2\gamma + \cos^2\gamma\cos(2j_3\theta_1)] = 0. \end{aligned} \tag{30}$$

  Here we used $\sin\gamma = 1/\sqrt{K}$, $(0 \leqslant \gamma \leqslant \pi/4)$. Now we have to minimize the number of queries to the oracle $S_{glg} = j_1 + j_2 + j_3$, having in mind (30) as a constraint. After we calculate the minimum of S_{glg} we have to compare it with the optimized version of the Grover–Radhakrishnan algorithm (20), (17) and find out which one is faster.

  Equation (30) can be solved numerically for $2 \leqslant K \leqslant 100$ and $S_{glg} = j_1 + j_2 + j_3$ also can be minimized numerically. The minimum of S_{glg} is larger than the optimized version of the Grover–Radhakrishnan algorithm. For many blocks, $100 \leqslant K$, the parameter γ is small, so the equation can be studied analytically and minimum also can be found analytically. This version of partial search algorithm does not provide acceleration compared to the Grover–Radhakrishnan. The partial search algorithm suggested in this paper is still optimal.

- Local–Global–Local
  Let us try another version of partial search. It differs from the previous by the sequence of iterations. We start with $|s_1\rangle$, apply j_0 local iterations, then j_1 global iterations (8) and then j_2 local iterations. The state of the database will become $G_1 G_2^{j_2} G_1^{j_1} G_2^{j_0}|s_1\rangle$. The amplitudes of all items in non-target blocks should vanish: $\langle u|G_1 G_2^{j_2} G_1^{j_1} G_2^{j_0}|s_1\rangle = 0$. Explicitly multiplying the matrices we can represent this equation in the form

$$\begin{aligned}\cos\gamma\cos 2\gamma\cos(z) + \sin^2\gamma\cos\gamma\cos 2\gamma\cos(z)[\cos(x) - 1]\\ + (-1)^{j_1}\sin^2\gamma\cos\gamma\cos 2\gamma[-\cos(x) + 1]\\ - \sin\gamma\cos\gamma\cos 2\gamma\sin(z)\sin(x) + \sin 2\gamma\cos(2\theta_2 j_2)\{\sin\gamma\cos(z)\\ + \sin^3\gamma\cos(z)[\cos(x) - 1] - \sin^2\gamma\sin(z)\sin(x)\\ + (-1)^{j_1}\sin\gamma\cos^2\gamma[\cos(x) - 1]\} = \sin 2\gamma\sin(2\theta_2 j_2)\{\sin(z)\\ + \sin\gamma\sin(x)\cos(z) + \sin^2\gamma\sin(z)[\cos(x) - 1]\}.\end{aligned} \tag{31}$$

Now we have to minimize the total number of the queries to the oracle $S_{lgl} = j_0 + j_1 + j_2$ and compare with the optimized version of the Grover–Radhakrishnan algorithm (20) and (17) to find out which one is faster. This equation was solved and S_{lgl} minimized numerically for $2 \leqslant K \leqslant 100$. For a large number of blocks, $100 \leqslant K$, parameter γ is small, so the solution and minimization can be done analytically. This version of partial search does not provide acceleration as well. The version suggested in this paper, equation (17), is still optimal.

Details of these two versions of partial search will be published elsewhere.

3.1. Lower bound

A lower bound for number of queries to the oracle was found in [7]:

$$S \geqslant \frac{\pi}{4}\sqrt{N} - \frac{\pi}{4}\sqrt{b}. \tag{32}$$

It is based on the lower bound for the full search [2, 6]. One can first search for the block and then for the target item in the block. We can improve the lower bound for the algorithms that have the same final state for the target block. After we run the partial search algorithm the wavefunction of the database (13) has nonzero components only in the target block. The calculations show

$$|d\rangle = \sin\alpha_K|t\rangle + \cos\alpha_K|\mathrm{ntt}\rangle \tag{33}$$

(see (17) and (11)). We can represent it as a result of the application of j_e Grover iterations to uniform superposition of all basis states in the target block:

$$|d\rangle = G_2^{j_e}|s_2\rangle, \qquad j_e = \frac{\alpha_K}{2}\sqrt{b} \tag{34}$$

(see (8) and (9)). It will take only $\tilde{j}_{\mathrm{full}} = (\pi/4 - \alpha_K/2)\sqrt{b}$ iterations to find the target item in the target block. We can bound S from the following: $S + \tilde{j}_{\mathrm{full}} \geqslant \pi\sqrt{N}/4$. The lower bound depends on the number of blocks (see (18)). Replacing α_K by its minimum (18) we get a tighter lower bound:

$$S \geqslant \frac{\pi}{4}\sqrt{N} + \left(-\frac{\pi}{4} + \frac{\alpha_K}{2}\right)\sqrt{b} \geqslant \frac{\pi}{4}\sqrt{N} - \frac{\pi}{6}\sqrt{b}. \tag{35}$$

4. Summary

We optimized the Grover–Radhakrishnan method of partial search. We conjecture that our version of partial search is optimal in a wider class of partial search algorithms (arbitrary sequences of local and global iterations).

Acknowledgments

The author is grateful to L K Grover and J Radhakrishnan for productive discussions and J Liao for numerical calculations.

References

[1] Grover L K 1996 *Proc. 28th ACM Symp. on the Theory of Computing* (New York: ACM) pp 212–9

[2] Bennett C H, Bernstein E, Brassard G and Vazirani U 1997 *SIAM J. Comput.* **26** 1510–23 (*Preprint* quant-ph/9701001)

[3] Boyer M, Brassard G, Hoyer P and Tapp A 1998 *Fortschr. Phys.* **46** 493–506 (*Preprint* quant-ph/9605034)

[4] Brassard G, Hoyer P, Mosca M and Tapp A 2002 *Contemp. Math.* **305** 53

[5] Xiao Li and Jones Jonathan A 2005 *Preprint Preprint* quant-ph/0504054

[6] Zalka C 1999 *Phys. Rev.* A **60** 2746

[7] Grover L K and Radhakrishnan J 2004 *Preprint Preprint* quant-ph/0407122

[8] Korepin V E 2005 *Preprint Preprint* quant-ph/0503238