ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/228574906

Quantum bit string comparator: Circuits and applications

Article · January 2007

CITATIONS
47

READS
4,003

2 authors, including:

Rubens Ramos
Universidade Federal do Ceará
162 PUBLICATIONS **616** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Quantum information processing View project

Quantum physics and number theory View project

All content following this page was uploaded by Rubens Ramos on 31 May 2014.

The user has requested enhancement of the downloaded file.

David Sena Oliveira and Rubens Viana Ramos

Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará - DETI/UFC
C.P. 6007 – Campus do Pici - 60755-640 Fortaleza-Ce Brasil
E-mail: davidsena@deti.ufc.br; rubens@deti.ufc.br

Quantum bit string comparator: circuits and applications

Received July 03, 2007

Quantum computation has attracted much attention since it was shown by Shor and Grover the possibility to implement quantum algorithms able to realize, respectively, factoring and searching in a faster way than any other known classical algorithm. In particular, it is possible to use Grover's algorithm, taking profit of its ability to find a specific value in an unordered database, to find, for example, the zero of a logical function or the minimal or maximal value in a database. Here we show quantum algorithms to solve those cited problems. The solution requires the use of a quantum bit string comparator. This quantum circuit compares two quantum states and identifies if they are equal or, otherwise, which of them is the largest. Moreover, we also show the quantum bit string comparator allow us to implement conditional statements in quantum computation, a fundamental structure for designing of algorithms.

1. Introduction

The Grover's quantum search algorithm is a celebrated result in quantum computation that proves that quantum information properties (superposition) can improve the speedup of finding a specific value within an unordered database. In this case, no technique using data structures can be used and only sequential tentative can be realized. Computationally, the quantum search is proved to get in average $O(N^{1/2})$ operations (in comparison with the $O(N)$ classical operations), which indicates a quadratic speed-up [1–3]. Even though this improvement can be considered minor than other quantum algorithms, Shor [4] and Deutsch-Jozsa[5] algorithms are exponentially better than their classical counterparts, the fact is that searching is fundamental in computer science having a large amount of applications. In addition, no classical algorithm can be more efficient than Grover algorithm, that is, the quantum search algorithm is as efficient as the best search algorithm could be. The basic reason that allows this performance is the smart use of the quantum superposition which means that all states can be processed at once (in contrast with the combinatory explosion of the classical alternatives). Basically, during the processing, the database, initially an equally weighted superposition of all possible states, converges to a state that can also be a superposition, but containing only the states that are solutions of the problem, named marked states. There are several works on variations of Grover's algorithm [6,7], entanglement measures based on the Grover' algorithm [8,9] and implementation of

Grover's algorithm [10,11]. Here, our goal is to show how to solve some interesting mathematical problems using the Grover's algorithm with an oracle based in a quantum circuit that compares two quantum states, representing binary strings, and identifies if they are equal or not and, in this last case, which of them is the largest (or the lowest). The circuit that makes the comparison is named quantum bit string comparator, QBSC. Furthermore, we show how to use the QBSC to construct quantum algorithms that employ conditional statements.

2. Quantum circuits for the quantum bit string comparator

Given two n-partite of qubits quantum states $|a\rangle|b\rangle$ the quantum bit string comparator is a unitary evolution U_{CMP} that works as shown in (1)

$$U_{CMP}|a\rangle|b\rangle|0^{\otimes m}\rangle|0\rangle|0\rangle = |a\rangle|b\rangle|\psi\rangle|x\rangle|y\rangle . \tag{1}$$

In (1) there are m+2 ancillas at the input, $|\psi\rangle$ is a m qubit output state that has not important information and the last two qubits carry the comparison information. For example, if a=b then x=y=0, if a>b then x=1 and y=0, and if a<b then x=0 and y=1. The evolution shown in (1) can be realized using the quantum circuit shown in Fig. 1 (for three qubits strings). It is able to compare two binary strings (having the same number of bits) identifying, by the measurement of two qubits, if they are equal or, if they are different, which of them is the largest (or the lowest).

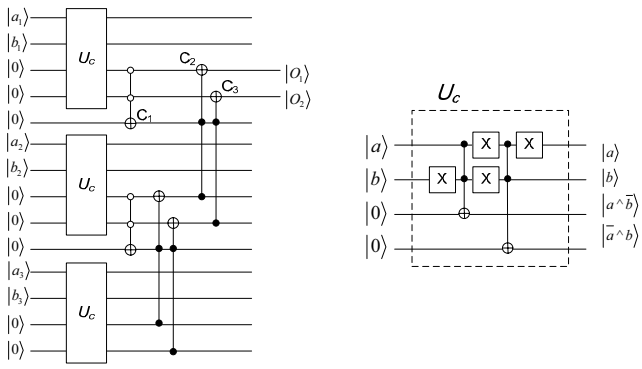

Fig. 1. Quantum circuit for comparison of two strings of three qubits: $|a\rangle$=$|a_1\rangle|a_2\rangle|a_3\rangle$ and $|b\rangle$=$|b_1\rangle|b_2\rangle|b_3\rangle$

The quantum circuit proposed makes the comparison of two strings of three qubits, but the generalization to any number of qubits is straightforward. Basically, the quantum circuit compares the strings bit-to-bit from the left (most significant bit) to the right (less significant bit). In a measurement of the outputs (O_1 and O_2), if O_1=1 and O_2=0 then a>b; if O_1=0 and O_2=1 then a<b; at last, if O_1=0 and O_2=0 then a=b. Initially, the comparison between the first bit of each string is dominant, that is, if they are different, then the outputs will be O_1=a_1 and O_2=b_1. If they are equal (a_1=b_1) the comparison between the second bit of each string will be dominant, that is, if they are different, then the outputs will be O_1=a_2 and

O_2=b_2. If the second bits are also equal, the comparison between the third bits of each string will be dominant and so on. In the circuit of Fig. 1 the transfer of dominion from one position of the string to the next is realized by the Toffoli gate C_1 (activated in zero) and the Toffoli gates C_2 and C_3. Obviously, only the less significant bit does not have the dominion transfer circuit. The following examples, shown in Table 1, will make clear the functioning of the circuit (for simplicity it will be considered the comparison of two states of two qubits, but the result is directly generalized for any number of qubits).

Table 1. Examples of the output of the quantum bit string comparator for two strings of two qubits at the inputs.

$\|a\rangle$	$\|10\rangle$	$\|00\rangle$	$\|10\rangle$	$\|11\rangle$	$\|01\rangle$
$\|b\rangle$	$\alpha\|00\rangle+\beta\|01\rangle$	$\alpha\|01\rangle+\beta\|10\rangle$	$\alpha\|00\rangle+\beta\|11\rangle$	$\alpha\|01\rangle+\beta\|11\rangle$	$\alpha\|01\rangle+\beta\|11\rangle$
O_1	$\|1\rangle$ (1)	$\|0\rangle$ (1)	$\|1\rangle$ ($\|\alpha\|^2$)	$\|1\rangle$ ($\|\alpha\|^2$)	$\|0\rangle$ (1)
O_2	$\|0\rangle$ (1)	$\|1\rangle$ (1)	$\|1\rangle$ ($\|\beta\|^2$)	$\|0\rangle$ (1)	$\|1\rangle$ ($\|\beta\|^2$)

In Table 1, the number inside the parenthesis besides the qubit means the probability of the output to be that qubit. For example, comparing $|a\rangle=|11\rangle$ with $|b\rangle=\alpha|01\rangle+\beta|11\rangle$, with probability $|\alpha|^2$ $a>b$ and, hence, $|O_1O_2\rangle=|10\rangle$. On the other hand, with probability $|\beta|^2$ $a=b$ and, hence, $|O_1O_2\rangle=|00\rangle$. Hence, in this case, the output state is $|O_1O_2\rangle=(\alpha|1\rangle_1+\beta|0\rangle_1)|0\rangle_2$. Another quantum circuit for the QBSC, based on subtractions, named NKO, can be as shown in Fig. 2 [12].

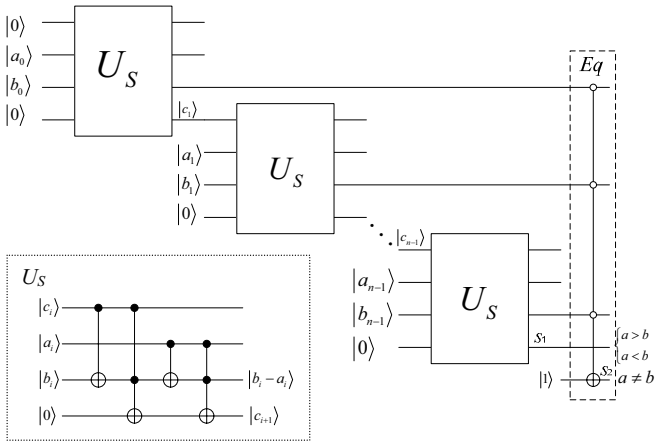

Figure 2 - Quantum circuit for comparison of two strings of qubits using subtractions

In order to compare the quantum circuits shown in Figs. 1 and 2, we realized a complexity analysis. Since the construction of any quantum circuit can be done using single qubit gates (*Sq*) and CNOT gates (*Cn*), the cost of the circuits QBSC and NKO is calculated based on the unitary cost of those universal gates. The unitary costs are defined by: 1) Cost of *Sq* = 1*u*. 2) Cost of *Cn* = 1*d*. For example, the Swap and Toffoli gates have the following costs:

- The Swap (*Sw*) gate can be constructed using three CNOT gates, hence, it costs 3*d*;
- The Toffoli (*Tof*) gate can be constructed without ancilla [13] as shown in Fig. 3. It costs Three *Cn*, two *Sw* and four *Sq*, then, it costs 9*d*+4*u*.

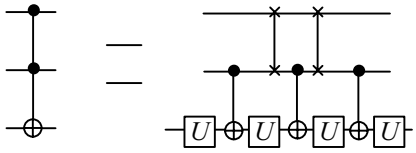

Figure 3. Toffoli gate implemented using only single-qubit gates and CNOT gates, without ancillas.

Thus, the final cost to construct a *n*-qubit QBSC is presented in Table 2.

Table 2. Construction cost of QBSC circuit.

Gate	Components	Unitary Cost	Quantity
Uc	2 *Tof* + 2 *Sw* + 4 *Pu*	24*d*+12*u*	*n*
C_1	1 *Tof* + 4 *Pu*	9*d*+8*u*	*n*-1
C_2	8 *Sw* + 1 *Tof*	33*d*+4*u*	*n*-1
C_3	8 *Sw* + 1 *Tof*	33*d*+4*u*	*n*-1

The total *n*-qubit QBSC cost is [99(*n*-1)+24] CNOTs and [28(*n*-1)+12] single-qubit gates.

To realize the NKO complexity analysis, we assume that the quantum gate to verify the equality (*Eq*) is constructed using Swap gates to order the qubits and one MCNOT gate with *n*-control qubits. The two low cost way to implement the MCNOT with *n*-control qubits [14] are shown in Table 3 with their respective costs.

Table 3. Cost to implement a MCNOT gate with *n*-control qubits.

Gate	Ancilla qubits	Number of Toffolis	Total cost
MCNOT1	1	32*n*-96	288*n*·*d*-864*d*+128*n*·*u*-384*u*
MCNOT2	*n*-2	16*n*-32	144*n*·*d*-288*d*+64*n*·*u*-128*u*

One can construct the NKO circuit using MCNOT1 or MCNOT2. The construction cost of the NKO components are presented in Table 4.

Table 4. Cost to implement a NKO circuit

Gate	Components	Unitary cost	Quantity
Us	2 *Sw* + 2 *Cn* + 2 *Tof*	17*d*+4*u*	*n*
*Eq*1	n^2 *Sw* + 2*n Pu* + McN1	$3n^2d$+288*n*·*d*-864*d*+130*n*·*u*-384*u*	1
*Eq*2	n^2 Sw + 2*n Pu* + McN2	$3n^2d$ +144*n*·*d*-288*d*+66*n*·*u*-128*u*	1

Using Table 3, we see that NKO1 (NKO with *Eq*1, that uses MCNOT1) has total cost of [$3n^2$+305*n*-864] CNOTs and [134*n*-384] single-qubit gates. The NKO2 (NKO with *Eq*2, that uses MCNOT2) has total cost of [$3n^2$ +161*n*-288] CNOTs and [70*n*-128] single-qubit gates. Therefore, the creation of NKO circuit requires more resources than the creation of QBSC circuit. This happen because of the MCNOT used by NKO. The Figures 4 and 5 show, respectively, the graphics of the number of CNOT gates and single qubit gates required by QBSC and KNO in order to build a *n*-qubit comparison quantum circuit.

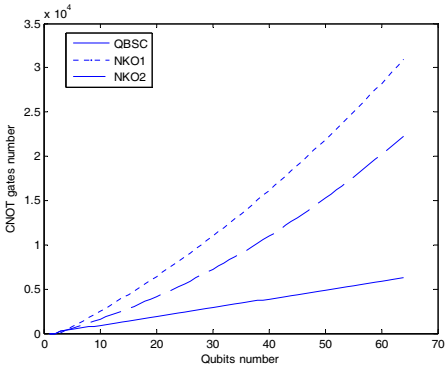

Figure 4. Number of CNOTs versus number of qubits for QBSC, NKO1 and KNO2.

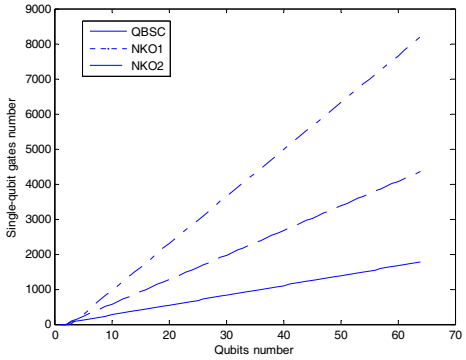

Figure 5. Number of single-qubit gates versus number of qubits for QBSC, NKO1 and KNO2.

Another point to be considered is the number of ancillas needed. For the QBSC are necessary $3n$-1, as can be calculated observing Fig. 1. The NKO1 needs n+3 ancillas. Observing Fig. 2 one counts n+2, the last one comes from the implementation of the MCNOT and it is not shown in Fig. 2. Finally, the NKO2 uses $2n$ ancillas.

The last parameter to be considered is the degree of parallelism, that is, how compact the comparators can be constructed. The larger the number of gates that can be simultaneously used the faster is the program execution. In QBSC, the comparison between the pairs of qubits can be parallelized, that is, every gates Uc and C_1 can be processed at the same time.

2. Applications of the quantum bit string comparator as an oracle in Grover's algorithm

There are several important applications of the comparison of binary strings in quantum computation. Let us firstly discuss the use of the QBSC as an oracle in the Grover's quantum search algorithm. An example of 4 qubits is shown in Fig. 6, but a generalization for any number of qubits is straightforward.

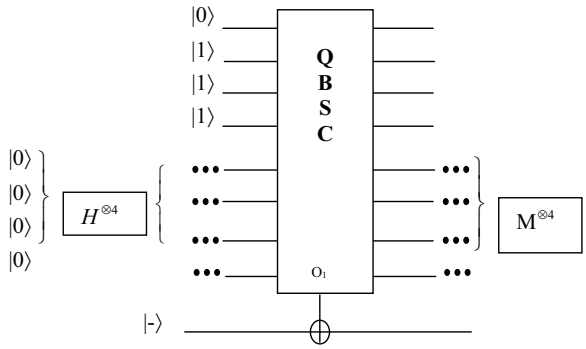

Figure 6. QBSC circuit as an oracle in a Grover search algorithm of four qubits. *H*-Hadamard gate. M – Measurer.

For the circuit shown in Fig. 6, the task is to search in the database words of four bits larger than "0111". This is clearly a typical case of multiple marked states and, hence, the output of Grover's algorithm will be a superposition of the all possible solutions. The reference state $|0111\rangle$ works as string $|a\rangle$ while the database is $|b\rangle$. If the initial state of Grover's algorithm is $(1/4)\sum_{i=0}^{15}|i\rangle$ then the output state will be:

$$\frac{\left(|8\rangle+|9\rangle+|10\rangle+|11\rangle+|12\rangle+|13\rangle+|14\rangle+|15\rangle\right)}{2\sqrt{2}} \tag{2}$$

where the decimal representation has been used for simplification. If instead of search for states larger than $|0111\rangle$ one was looking for states lower than $|0111\rangle$, then O_2 would be used instead of O_1 (the reference is $|a\rangle$). Let us now suppose that the goal is to find the minimal value in the database. In order to find the minimal value the quantum circuit shown in Fig. 4 has to be used to activate the lowest CNOT of Grover's quantum circuit.

Using the circuit of Fig. 7, the oracle will recognize strings lower or equal than the reference. The algorithm to find the minimum is as follows [15]: Initially, one value of the database is randomly chosen. This value will be used for comparison (string $|a\rangle$). The algorithm runs and, at end, the result of the measurement will be one of the members of the database lower or equal than the initial value used. The result of the measurement will now be used as the new value to be compared. The process is repeated till the result of the measurement does not change anymore. If one is looking for the minimal of a function f, represented by the unitary evolution U_f, then the quantum circuit shown in Fig. 8 represents the complete oracle circuit.

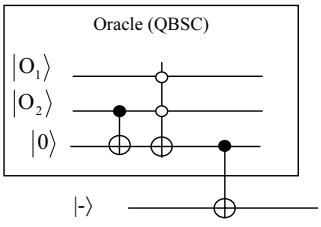

Figure 7. Control of the lowest CNOT of Grover's quantum circuit in order to find the minimal value in a database using the quantum comparator circuit as oracle.

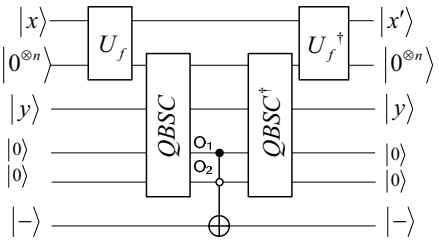

Figure 8. Oracle circuit with QBSC for finding the minimal value of the function *f*.

In Fig. 8, $|x\rangle$ is the database, $|y\rangle$ is the present "minimal value" that works as reference and the operation U_f acts in the following way: $U_f|x\rangle|0^{\otimes n}\rangle=|x\rangle|f(x)\rangle$.

Another interesting application is the problem of inverting a function, that is, given y what is the x such that $y=f(x)$. The unitary transformation U_f represents the function whose argument one wishes to find. In order to solve this problem, almost the same quantum circuit of Fig. 8 can be used, the difference is the Tofolli gate, that now has to be activated only when $O_1=O_2=0$ as shown in Fig. 9.

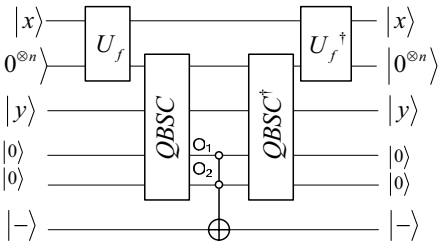

Figure 9. Oracle circuit with QBSC used to invert the function *f*.

Given a y, the oracle in Fig. 9 marks only those states $|x\rangle$ the obeys the condition $f(x)=y$. In particular, if $|y\rangle=|0^{\otimes n}\rangle$, then the oracle will recognize only the zero of the function *f*.

Another important question is the search of intervals. For example, given the constants y, α and β, what are the values of x for which $y-\beta<f(x)<y+\alpha$? The oracle to be used in the solution of this kind of problem is shown in Fig. 10.

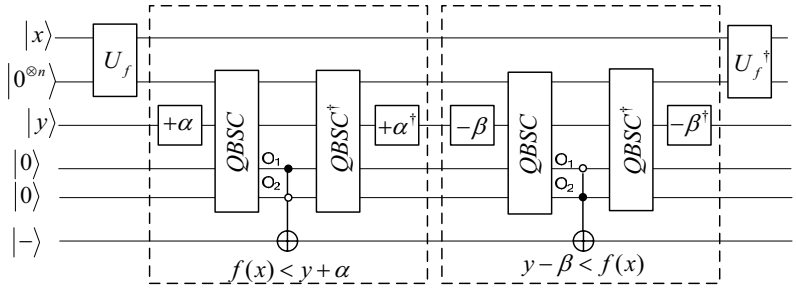

Figure 10. Oracle with QBSC for searching of the values of x such $y-\beta< f(x) < y+\alpha$.

In the quantum circuit in Fig. 10, the $+\alpha$ and $-\beta$ gates, whose ancillas are not shown, realize, respectively, the sum of α and subtraction of β. In Fig. 11.a-c it is shown for a fictitious example the change of the sign of the amplitudes before the first QBSC (*a*), after the first QBSC (*b*) and after the second QBSC (*c*).

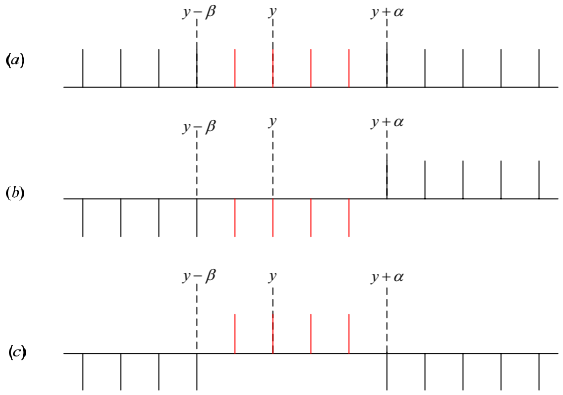

Figure 11. Change of the signs of the amplitudes during operation of the oracle shown in Fig. 10.

3. Conditional statements in quantum computation

In general, the QBSC expands the notion of controlled operation. Using the QBSC, controlled operations of the type U^C, where C is a conditional statement, can be constructed. In this case, the operator U is applied to a set of qubits only if the conditional statement C is true. This last can be anyone of the type $a>b$, $a<b$, $a\geq b$, $a\leq b$, $a=b$ and $a\neq b$. For instance, using again the Grover's algorithm as scenario, we can implement the following piece of software: If $|a\rangle>|b\rangle$ then search for solution S_1, otherwise, search for solution S_2. For an oracle based on *n*-CNOTS, the quantum circuit for a four bit problem is as shown in Fig. 12.

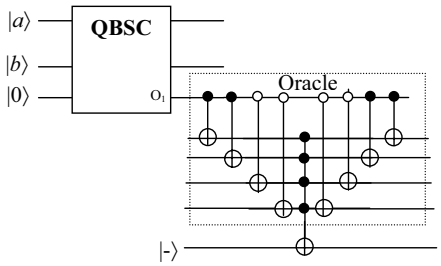

Figure 12. Implementation of the conditional statement: If $a>b$ then search for solution $S_1=|0011\rangle$, otherwise, search for solution $S_2=|1100\rangle$.

In Fig. 12, if $a>b$ the Grover algorithm will search for $|0011\rangle$, otherwise the algorithm will search for $|1100\rangle$.

4. Conclusions

The quantum bit string comparator enables the implementation of quantum algorithms using conditional statements, a fundamental structure for designing of algorithms. This enlarges the number of applications where quantum algorithms can be used and, at the same time, it brings close to quantum programmers successful techniques used in classical computation based on comparisons. Furthermore, the use of the QBSC with Grover algorithm gives us power to solve some mathematical problems of the type presented in this work, as well open the possibility to create quantum algorithm with very specific tasks. For example, constructing a database composed only of prime numbers it is possible, using the QBSC and Grover algorithm, to search for an even number that does not satisfy Goldbach's conjecture (all even number larger than two can be written as the sum of two prime numbers).

Acknowledgements

This work was supported by the Brazilian agency FUNCAP.

References

[1] L. V. Grover, A fast quantum mechanical algorithm for database search, in Proc., 28th Annual ACM Symposium on the Theory of Computing, New York, (1996) pp. 212.
[2] L. V. Grover, Quantum mechanics helps in searching for a needle in a haystack, Phys. Rev. Lett., 79, (1997) 325-329.
[3] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge Univ. Press, Cambridge, ch. 4 and 6 (2000).
[4] P. W. Shor, Algorithms For Quantum Computation: Discrete Logs and Factoring, in Proc. 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, Nov. (1994) 20-22.
[5] D. Deutsch, R. Jozsa, Rapid solutions of problems by quantum computation, Proc. of the Royal Soc. of London, 439, (1992) 553.
[6] L. V. Grover, A different kind of quantum search. Preprint at (http://www.arXiv.org/quant-ph/0503205) (2005).
[7] T. Tulsi, L. V. Grover, and A. Patel, A new algorithm for directed quantum search. Preprint at (http://www.arXiv.org/quant-ph/0505007) (2005).
[8] O. Biham, M. A. Nielsen, and T. J. Osborne, Entanglement monotone derived from Grover's algorithm, Phys. Rev. A, 65, (2202) 062312.
[9] Y. Shimoni, D. Shapira, and O. Biham, Characterisation of pure states of multiple qubits using the Groverian entanglement measure, Phys. Rev. A, 69, (2004) 062303.
[10] M. O. Scully, and M. S. Zubairy, Quantum optical implementation of Grover's algorithm, PNAS, 98**,** 17, (2001) 9490-9493.
[11] K.-A. Brickman, P. C. Haljan, P. J. Lee, M. Acton, L. Deslauriers, and C. Monroe, Implementation of Grover's quantum search algorithm in a scalable system, Phys. Rev. A, 72, (2005) 050306.
[12] A. L. Nascimento, L. A. B. Kowada and W. R. de Oliveira, "A reversible ULA", First Workshop-school in Quantum Information and Computation, WECIQ 2006, Brazil.

[13] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin and H. Weinfurter, "Elementary gates for quantum computation" submitted to Phys. Rev. A 1995.
[14] D. Maslov, G. W. Dueck, "Improved quantum cost for n-bit Toffoli gates", Electronics Letters, v. 39, i. 25, p. 1790 – 1791, 2003.
[15] C. Dürr, and P. Hoyer, A quantum algorithm for finding the minimum, Preprint at (http://www.arXiv.org/ quant-ph/9607014v2) (1999).

View publication stats