Entanglement and its Role in Shor's Algorithm

Vivien M. Kendon[1], William J. Munro
Trusted Systems Laboratory
HP Laboratories Bristol
HPL-2005-215
December 5, 2005*

entanglement,
Shor's algorithm

Entanglement has been termed a critical resource for quantum information processing and is thought to be the reason that certain quantum algorithms, such as Shor's factoring algorithm, can achieve exponentially better performance than their classical counterparts. The nature of this resource is still not fully understood: here we use numerical simulation to investigate how entanglement between register qubits varies as Shor's algorithm is run on a quantum computer. The patterns in the entanglement are found to correlate with the choice of basis for the quantum Fourier transform rather than with any crucially quantum aspect of the algorithm.

* Internal Accession Date Only
[1]School of Physics and Astronomy, University of Leeds, Leeds, LS2 9JT, UK
Approved for External Publication

Entanglement and its Role in Shor's Algorithm

Vivien M. Kendon[1,3] and William J. Munro[2,3]

[1] School of Physics and Astronomy, University of Leeds, LS2 9JT, UK,
V.Kendon@leeds.ac.uk
[2] Hewlett-Packard Laboratories, Filton Road, Stoke Gifford, Bristol, BS34 8QZ, UK
[3] QOLS, Optics, Blackett Laboratory, Imperial College London, SW7 2BW, UK.

Abstract. Entanglement has been termed a critical resource for quantum information processing and is thought to be the reason that certain quantum algorithms, such as Shor's factoring algorithm, can achieve exponentially better performance than their classical counterparts. The nature of this resource is still not fully understood: here we use numerical simulation to investigate how entanglement between register qubits varies as Shor's algorithm is run on a quantum computer. The patterns in the entanglement are found to correlate with the choice of basis for the quantum Fourier transform rather than with any crucially quantum aspect of the algorithm.

1 Introduction

Quantum computation is generally regarded as being significantly more powerful than classical computation. There are numerous possible routes forward for quantum hardware [1], however, progress in the development of algorithms has been slow, in part because we don't yet fully understand how the quantum advantage works. Few quantum algorithms promise an exponential speed up over classical algorithms, of those that do, Shor's algorithm [2] is perhaps the most important because it can be used to factor large numbers and hence has implications for classical security methods.

The improvement that Shor's algorithm provides is generally attributed to entanglement enabling the algorithm to run efficiently. In fact, there are two key characteristics of the quantum resources used for computation. The first is that a general superposition of 2^n levels may be represented in n 2-level systems [3], allowing the the physical resource to grow only *linearly* with n (quantum parallelism). The second aspect is best explained by considering the classical computational cost of simulating a typical step in a quantum computation. If entanglement is absent then the algorithm can be simulated with an equivalent amount of classical resources. In recent work, Jozsa and Linden [4] have proven that, if a quantum algorithm that cannot be simulated classically using resources only polynomial in the size of the input data, then it must have multipartite entanglement involving unboundedly many of its qubits – if it is run on a quantum computer using pure quantum states. However, the presence of multipartite entanglement is not a sufficient condition for a pure state quantum computer to be hard to simulate classically. If the quantum computer is described using stabilizer formalism [5, 6], there are many highly entangled states that have simple classical descriptions, for instance the Bell and GHZ states. Moreover, a quantum computer using mixed states may still require exponential classical resources to simulate even if its qubits are not entangled, and it is not known whether such states may be used to perform efficient quantum computation. Parker and Plenio [7] have presented a version of Shor's algorithm using only one pure qubit, the rest may start in any mixed state. They found that entanglement was present when the algorithm ran efficiently for factoring 15 and 21 (tested numerically).

There is also no proof that an equally efficient classical algorithm cannot exist for Shor's algorithm[4], though for quantum walks an algorithm with a proven exponential speed up is known [8]. It is difficult to draw general conclusions about how such a speed-up comes about with few examples to work from. Nonetheless, a better understanding of how quantum algorithms and the entanglement within them work may help us to design new quantum algorithms. Given that entanglement is necessary (though not sufficient) for pure state quantum computation with an exponential speed-up over classical computation, we might instead ask whether the entanglement provides a quantitative resource during the computational process. We try to answer this question by investigating the level of entanglement within Shor's algorithm as it proceeds, gate by gate.

2 Shor's Algorithm

We begin with a brief overview of how Shor's algorithm works. We wish to factor a number $N = pq$ where p and q are prime numbers. Classical number theory provides a way to determine these primes with high probability (not unity generally) by finding the period r of the function $f_a(x) = a^x (\mathrm{mod}\ N)$ where a is an integer chosen to be less than N and co-prime to it. It is easy to check whether a is co-prime to N using Euclid's algorithm. If a happens not to be co-prime then their common factor gives a factor of N and the job is done but this happens only rarely for large N. Once the period r is found the numbers

$$m_{\pm} = a^{r/2} \pm 1 \tag{1}$$

generally share either p or q with N as a common factor. The remaining factor can be found efficiently using standard techniques. Not all choices of a give periods r which yields a factor p or q. For instance, sometimes the period r will be odd, whence the numbers from Eq. (1) can be non-integer. When the chosen a does not lead to a valid factor, the procedure can be repeated with a different choice a small number of times until a factor is found.

The hard part of the algorithm is determining the period r of the function $f_a(x) = a^x (\mathrm{mod}\ N)$. Shor found a very elegant and efficient means of doing this quantum mechanically, depicted schematically in Fig. 1. Consider that one has

[4] A polynomial factoring algorithm is now known but there is still no classically efficient order-finding algorithm, which is the core of Shor's algorithm.

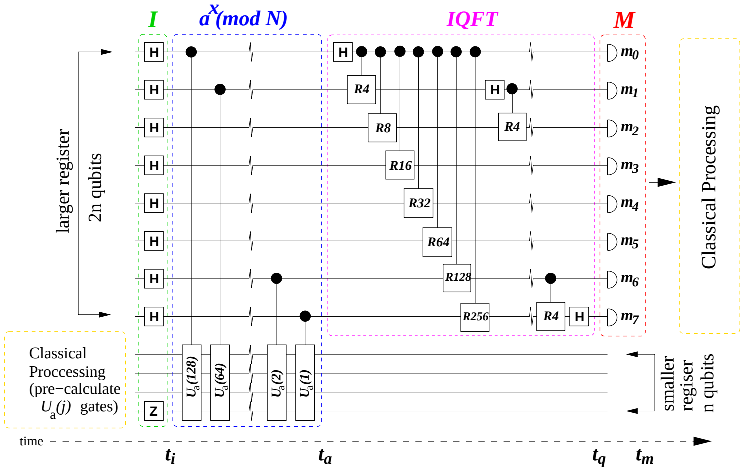

Fig. 1. Schematic circuit diagram of Shor's algorithm for factoring 15 implemented on a 12 qubit quantum register. The initialisation I is done with single qubit Hadamard (H) and bit-flip (Z) gates. Controlled-$U(j)$ gates are used to produce $a^x(\text{mod } N)$. The inverse quantum Fourier transform uses controlled rotations (Rm) The last quantum step is the measurement (M), which is followed by classical post-processing to obtain a factor of N.

two quantum registers (one of size $2n$ where $n = \lceil \log_2 N \rceil$ qubits and the second of size n qubits. We will denote the basis states of a quantum register by $|x\rangle$, with $x \in \{0 \ldots 2n-1\}$. The binary representation of x indicates which register qubits are in state $|0\rangle$ and which are in state $|1\rangle$. A general state of a $2n$ qubit register $|\Psi(t)\rangle$ at time t can thus be written as a superposition of basis states,

$$|\Psi(t)\rangle = \sum_{x=0}^{2n} a_x(t)|x\rangle, \tag{2}$$

where $a_x(t)$ is a complex number, normalised such that $\sum a_x(t)^2 = 1$. The algorithm begins by preparing the larger quantum register in an equal superposition $\sum_{x=0}^{2^{2n}-1} |x\rangle$ of all possible $2n$ basis states while the smaller register is prepared in the definite state $|1\rangle$. The total initial state of the system is hence

$$|\Psi(t_i)\rangle = \frac{1}{2^n} \sum_{x=0}^{2^{2n}-1} |x\rangle|1\rangle \tag{3}$$

The next step is a unitary transformation which acts on both registers according to $U|x\rangle|b\rangle = |x\rangle|ba^x(\text{mod } N)\rangle$ giving the output state

$$|\Psi(t_a)\rangle = \frac{1}{2^n} \sum_{x=0}^{2^{2n}-1} |x\rangle|a^x(\text{mod } N)\rangle \tag{4}$$

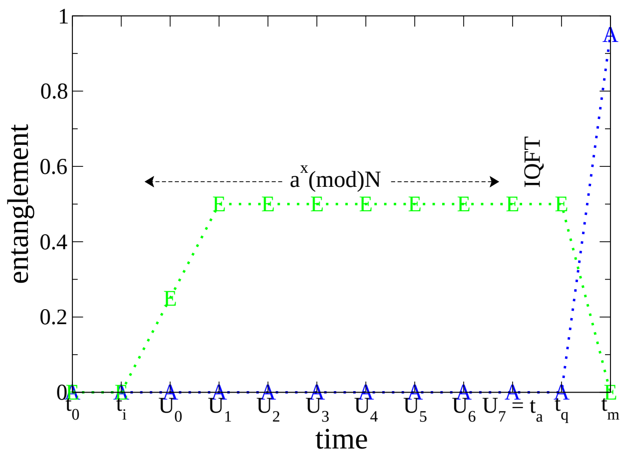

Fig. 2. Entanglement in Shor's 12 qubit algorithm as a function of gates sequence according to Fig (1) with the co-prime chosen as $a = 13$. E is the entanglement between the registers and A is the entanglement within the smaller register. The entanglement within the larger register is zero throughout.

Then an inverse quantum Fourier transform (IQFT) defined by

$$Q^{-1}|y\rangle = \frac{1}{2^n} \sum_{z=0}^{2^{2n}-1} e^{-2\pi i y z/2^{2n}} |z\rangle \tag{5}$$

is applied, which transform the state $|\Psi(t_a)\rangle$ from Eq. (4) into

$$|\Psi(t_q)\rangle = \frac{1}{2^{2n}} \sum_{x=0}^{2^{2n}-1} \sum_{z=0}^{2^{2n}-1} e^{-2\pi i x z/2^{2n}} |x\rangle |a^x (\text{mod } N)\rangle. \tag{6}$$

By measuring the larger register in the computational basis we obtain an integer number c. Now $c/2^{2n}$ is closely approximated by the fraction j/r and so r can be obtained classically using continued fractions. Choosing the larger register to be $2n$ qubits provides a high enough accuracy for c such that r can be determined from a single measurement on all $2n$ qubits. It is possible to use fewer qubits in this first register but the probability of determining r decreases, and the algorithm may need to be repeated correspondingly many more times.

3 Factoring 15

We start our analysis of the entanglement by studying the circuit for factoring 15 (3x5), though we need to be careful when drawing conclusions as it is not necessarily typical of factoring larger numbers. With the benefit of hindsight,

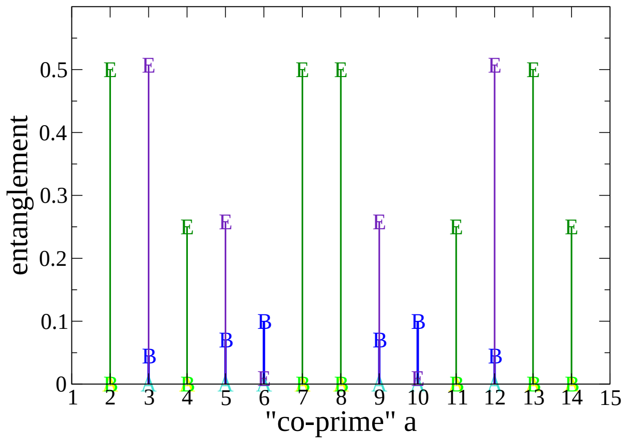

Fig. 3. Maximum amount of entanglement in Shor's algorithm for factoring 15 on a 12 qubit quantum computer versus co-prime a. Values of a co-prime to 15 are shown in green, while $a = 3, 5, 6, 9, 10, 12$ (not co-prime) are shown in blue/violet. Symbols are E for entanglement between the registers and B for entanglement in the larger register. The lower register has no entanglement within it at this stage of the computation.

rather than tracking the entanglement as each basic gate is applied, we choose to look at certain key points in the algorithm, since many gates make no changes to the entanglement. We restrict our attention to controlled composite gates: the $U(j)$ gate which is implements the operation $a^j (\mathrm{mod}\ N)$ for $j \in \{1, 2 \ldots 2^{2n}\}$, and the rotations in the IQFT. Details of how to efficiently construct these composite gates from a universal set of one and two qubit gates may be found in, for example, [5]. There are 8 of the $U(j)$ gates (in general $2n$, one for each larger register qubit), which is manageable, but for the IQFT there are 27 (in general $(2n+1)(n-1)$ for a $2n$ qubit register) rotation gates, so we have treated the whole IQFT as one unit. Along with single qubit gates as necessary, the circuit using these composite gates is depicted in Fig. 1.

As we are only considering the evolution of pure states we can measure the entanglement between the two registers using the entropy of the subsystems

$$E_c = -\sum_i \lambda_i \log \lambda_i, \tag{7}$$

where the $\{\lambda_i\}$ are the eigenvalues of the reduced density matrix of either of the registers (both have the same eigenvalues). To quantify the entanglement *within* each register is not so straightforward. To restrict our attention to the qubits within a single register, we first trace out the other register leaving a mixed state, ρ_L or ρ_S. Most entanglement measures for mixed states are computationally intractable in practice for more than a few qubits; we also need to consider all the possible divisions of the qubits into different subsets in order to locate all of the entanglement. A reasonable approximation to quantifying the entanglement

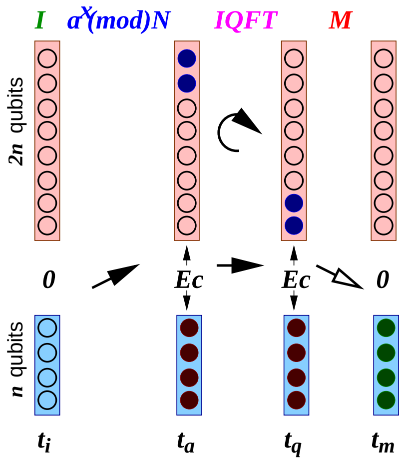

Fig. 4. Pattern of entanglement during Shor's algorithm factoring $N = 15$ with co-prime $a = 13$. After the $U(j)$ gates the top two qubits in the larger register (blue) are entangled with the four qubits in the smaller register (red). After the IQFT, the entanglement is transfered to the lower two qubits in the larger register. Qubits represented by open circles are not entangled.

within a register can be obtained by applying a partial transpose to each subset of qubits and calculating the negativity [9, 10] given by $\eta = \mathrm{Tr}|\rho^T| - 1$ i.e., the sum of the negative eigenvalues of the transposed matrix ρ_L^T or ρ_S^T. If the negativity is zero for all possible subsets of qubits in the register, then we can say that at most the register has bound entanglement [11], which is not generally considered useful for quantum information tasks (though see [12]). Non-zero negativity definitely implies the presence of entanglement.

In Fig. 2 we plot the entanglement in Shor's algorithm using the entropy of the subsystem where possible (full state is pure) and the negativity where the single register state is mixed. The negativity turns out to be zero for both registers throughout the algorithm (except the measurement leaves the smaller register entangled, but this cannot be useful for the remaining classical steps of the algorithm). The entanglement between the registers builds up to a maximum during the first two $U(j)$ gates, then stays constant until the measurement.

Next we ask whether this pattern/degree of entanglement is affected by the choice of co-prime, or even (for comparison) whether a is actually co-prime or not. The maximum entanglement between the two registers at time t_a, after applying the $U(j)$ gates but before applying the IQFT, is plotted in Fig. 3. Also shown is the average negativity for the larger register (for the lower register it is always zero at this stage of the algorithm). When a is not co-prime, there is entanglement within the larger register but when a is co-prime there is none. Otherwise, it is clear that there is no real pattern between the maximum amount of entanglement and the chosen value of a.

size of subsystem	small register	large register after U	large register after IQFT	large register difference ΔE
1 qubit	0.811	1.000	0.938	-0.062
2 qubits	1.538	1.600	1.599	-0.001
3 qubits	2.151	1.843	2.020	+0.177
4 qubits	2.585	1.972	2.283	+0.311
5 qubits	2.585	2.081	2.447	+0.366
6 qubits		2.184	2.547	+0.363
7 qubits		2.285	2.602	+0.318
8 qubits		2.385	2.589	+0.204
9 qubits		2.485	2.619	+0.134

Table 1. Average entropy of subsystems for factoring 21 with $a = 2$.

Even though the entanglement between the registers, as measured by the entropy of the subsystems, does not change during the IQFT, the distribution of the entanglement between the individual qubits does change. If we return to our first example, factoring 15 with $a = 13$, and look at the entropy of subsets of qubits from the larger register, we can deduce that only two of the eight qubits are entangled with the four qubits in the smaller register. During the action of the IQFT, this entanglement is transfered from the top two qubits to the bottom two in the larger register. We represent this in a diagram shown in Fig. 4.

However, we cannot draw general conclusions from the process of factoring 15 since 15 is actually extremely easy to factor. It is straightforward to see that at least one of $a^{r/2} \pm 1$ is divisible by 3 or 5 for nearly all choices of $a, r > 1$, regardless of whether a is co-prime to N or even whether r is the period of $x^a(\mathrm{mod}\ N)$. We need to look at larger N.

4 Factoring 21

We next look at factoring 21 (3×7). To do this on a quantum computer in the same manner as the circuit for factoring 15 shown in Fig. 1 requires a total of 15 qubits, 10 in the larger register and 5 in the smaller. For co-prime $a = 13$, we find a similar pattern of entanglement to that shown in Fig. 4 for 15 with $a = 13$, except that for 21 there is only entanglement between one qubit in the larger register and two qubits in the lower register. Again, the IQFT step shifts the entanglement from the top qubit to the bottom qubit in the larger register.

The larger register is now at the limit of our computational resources for calculating the full analysis of the negativity. By using random samples of cuts, instead of calculating all possible cuts, we find that for co-prime $a = 13$ there is no entanglement within either register, but for other choices of co-prime such as $a = 2$ and $a = 4$, entanglement is generated within the larger register during the IQFT. For these co-primes we also find a more complex pattern in the entropies of the subsystems: the entanglement now involves all the register qubits. The details are shown in Table 1: essentially the entanglement becomes more multipartite

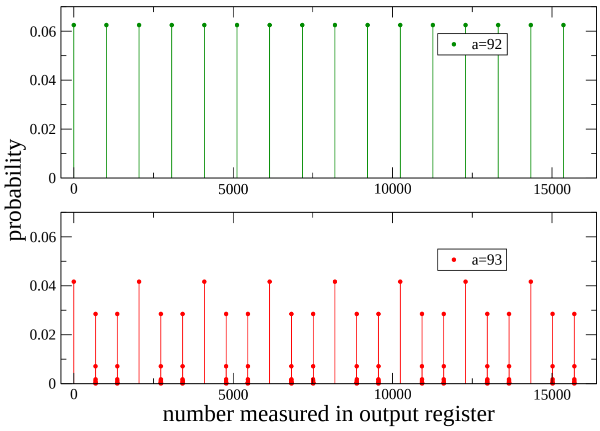

Fig. 5. Distribution of measurement outcomes for factoring 119 with $a = 92$ (upper) and $a = 93$ (lower) with periods $r = 16$ and 24 respectively. Circles show the probability of measuring the number on the ordinate as the outcome of the algorithm; drop lines are for clarity. The upper figure has just 16 circles, while the lower figure with period 24 has significant probability for measuring neighbouring numbers to the minor peaks.

(the average reduces for one and two qubit cuts, while for larger cuts it increases). We will provide an explanation for this observation in the next section where we examine larger examples.

5 Factoring larger numbers

We also studied semi-primes $32 < N < 64$ and $64 < N < 128$, which require 18 and 21 qubits respectively for the quantum registers. In these cases, though we cannot easily calculate a full entanglement analysis, we have calculated the entropy between one qubit and the rest of the qubits in both registers, this corresponds to the quantities in the top line of Table 1. The difference in the average entropy ΔE_1 before and after the IQFT (corresponding to the last column in Table 1), is shown in Table 2 grouped by the period r.

There is a clear pattern for ΔE_1: the closer the period r is to a power of 2, the smaller the value of ΔE_1. For $r = 2^m$, the IQFT is exact giving $\Delta E_1 = 0$ in all cases. This can be understood by looking at the measurement results on the larger register, from which the period r is calculated. Figure 5 shows the probability of measuring each possible number c in the larger register at the end of running the algorithm for two examples factoring 119, with co-prime 92 (period $r = 16$) and co-prime 93 (period $r = 24$). When the period is exactly a power of two, the fraction $c/2^{2n}$ gives the period r exactly, whereas when r is not a power of two, the peak probability tries to fall between two possible numbers and thus spreads the wavefunction over several adjacent numbers. This spread increases the entanglement in the upper register.

N	r (number of co-primes with this r)								
$p \times q$	$-\langle \Delta E_1 \rangle$								
15	2 (3)	4 (4)							
3×5	0.0	0.0							
21	2 (3)	3 (2)	6 (6)						
3×7	0.0	0.706	0.624						
33	2 (3)	5 (4)	10 (12)						
3×11	0.0	0.285	0.256						
35	2 (3)	3 (2)	4 (4)	6 (6)	12 (8)				
5×7	0.0	0.869	0.0	0.788	0.706				
39	2 (3)	3 (2)	4 (4)	6 (6)	12 (8)				
3×13	0.0	0.869	0.0	0.788	0.706				
51	2 (3)	4 (4)	8 (8)	16 (16)					
3×17	0.0	0.0	0.0	0.0					
55	2 (3)	4 (4)	5 (4)	10 (12)	20 (16)				
5×11	0.0	0.0	0.285	0.256	0.226				
57	2 (3)	3 (2)	6 (6)	9 (6)	18 (18)				
3×19	0.0	0.869	0.788	0.080	0.071				
77	2 (3)	3 (2)	5 (4)	6 (6)	10 (12)	15 (8)	30 (24)		
7×11	0.0	1.033	0.343	0.951	0.314	0.034	0.031		
91	2 (3)	3 (8)	4 (4)	6 (24)	12 (32)				
7×13	0.0	1.033	0.0	0.951	0.869				
119	2 (3)	3 (2)	4 (4)	6 (6)	8 (8)	12 (8)	16 (16)	24 (16)	48 (32)
7×17	0.0	1.033	0.0	0.951	0.0	0.869	0.0	0.788	0.706

Table 2. Average decrease in entanglement $-\langle \Delta E_1 \rangle$ between one qubit and the rest during the IQFT.

6 Discussion

We have found that, as expected, the quantum registers become highly entangled during Shor's algorithm. The entanglement increases during the progress of the algorithm, and during the IQFT in particular it can become more multipartite in nature if the period being found is not an exact power of two. Thus, if we performed the IQFT to some other base than two – for example, in base three, perhaps using a quantum register made up of qutrits (three-state quantum systems) rather than qubits – the entanglement pattern would change completely. Therefore the entanglement cannot be correlated in a quantitative way with the success of the algorithm. While entanglement is certainly generated in significant quantities during pure state quantum computation, this is best understood as a by-product of exploiting the full Hilbert space for quantum parallelism: the majority of quantum states are known to be highly entangled [13, 14].

We find no evidence that the entanglement generated during the execution of the algorithm is used up in a quantitative way to fuel the computation process, indeed, the overall entanglement does not decrease at any point before the measurement. This is in complete contrast to quantum communications tasks where maximally entangled pairs of qubits can perform a specific amount of communication, during which the entanglement is used up. Entanglement is used quantitatively in many practical proposals for implementations of a quantum computer, notably [15], this use can be attributed to the communications tasks carried out to move the quantum data around in the physical qubits.

We thank Stephen Parker, Martin Plenio and Ben Travaglione for valuable discussions. VK was funded by the UK Engineering and Physical Sciences Research Council grant number GR/N2507701 (to Sept 2003) and now by a Royal Society University Research Fellowship. A Royal Society Research Grant provided additional computing facilities. Partly funded by the European Union.

Bibliography

[1] Braunstein, S., Lo, H.K., eds. In Braunstein, S., Lo, H.K., eds.: Fortschr. Phys. Volume 48. (2000)

[2] Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Sci Statist. Comput. **26** (1997) 1484

[3] Blume-Kohout, R., Caves, C.M., Deutsch, I.H.: Climbing mount scalable: Physical-resource requirements for a scalable quantum computer. Found. Phys. **32** (2002) 1641–1670

[4] Jozsa, R., Linden, N.: On the role of entanglement in quantum computational speed-up. Proc. Roy. Soc. Lond. A Mat. 459 **459** (2003) 2011–2032

[5] Nielsen, M.A., Chuang, I.J.: Quantum Computation and Quantum Information. Cambridge University Press, Cambs. UK (2000)

[6] Gottesman, D.: Stabiliser codes and quantum error correction. PhD thesis, California Institute of Technology, Pasadena, CA (1997)

[7] Parker, S., Plenio, M.B.: Entanglement simulations of shor's algorithm. J. Mod. Optic **49** (2001) 1325–1353

[8] Childs, A.M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., Spielman, D.A.: Exponential algorithmic speedup by a quantum walk. In: Proc. 35th Annual ACM Symposium on Theory of Computing (STOC 2003), Assoc. for Comp. Machinery, New York (2003) 59–68

[9] Peres, A.: Separability criterion for density matrices. Phys. Rev. Lett. **77** (1996) 1413–1415

[10] Życzkowski, K., Horodecki, P., Sanpera, A., Lewenstein, M.: On the volume of the set of mixed entangled states. Phys. Rev. A **58** (1998) 883–892

[11] Horodecki, M., Horodecki, P., Horodecki, R.: Mixed-state entanglement and distillation: is there a "bound" entanglement in nature? Phys. Rev. Lett. **80** (1998) 5239–5242

[12] Horodecki, P., Horodecki, M., Horodecki, R.: Bound entanglement can be activated. Phys. Rev. Lett. **82** (1999) 1056–1059

[13] Kendon, V.M., Życzkowski, K., Munro, W.J.: Bounds on entanglement in qudit systems. Phys. Rev. A **66** (2002) 062310

[14] Hayden, P., Leung, D.W., Winter, A.: Aspects of generic entanglement. quant-ph/0407049 (2004)

[15] Raussendorf, R., Briegel, H.J.: A one-way quantum computer. Phys. Rev. Lett. **86** (2001) 5188–5191