International Journal of Pure and Applied Mathematics
Volume 60 No. 1 2010, 83-88

SIMULATION AND THE EXACT COMPLEXITY OF GROVER'S SEARCH ALGORITHM

Eugen Grycko[1] §, Werner Kirsch[2], Tobias Mühlenbruch[3]
[1,2,3]Department of Mathematics and Computer Science
University of Hagen
125, Lützowstr., Hagen, D-58084, GERMANY
[1]email: eugen.grycko@fernuni-hagen.de
[2]email: werner.kirsch@fernuni-hagen.de
[3]email: tobias.muehlenbruch@fernuni-hagen.de

Abstract: We explore a possibility of simulation of Grover's search algorithm on a classical computer. To access the exact (asymptotic) complexity of this algorithm a structural study of the iteration procedure is presented.

AMS Subject Classification: 60-04, 68U20
Key Words: tensor product, Hilbert space, unitary operator, search space

1. Introduction

Quantum computation is a growing interdisciplinary branch of science. An important advantage of many quantum algorithms is their efficiency as compared with classical ones. A main disadvantage of quantum computation today is the lack of comfortable quantum hardware which would be able to carry out the invented algorithms.

Grover (1998) introduced a quantum algorithm for searching an item in an unstructured list. The algorithm offers an efficiency gain when run on a quantum computer.

The aim of the present contribution is pointing out a possibility of simula-

Received: February 17, 2010

§Correspondence author

tion of a quantum computer on a classical one. The main idea can be exemplified by presenting a description of Grover's search algorithm enabling us to implement the simulation procedure in an imperative programming language (Section 2). A structural study of Grower's iteration procedure yields the exact complexity of the algorithm (Section 3).

2. Simulation of Grover's Algorithm on a Classical Computer

Let us suppose we are searching for an element s_0 in the search space $\{0,1\}^n$ consisting of $N = 2^n$ binary tuples of length n. Let us assume that s_0 is coded by the function $f : \{0,1\}^n \to \{0,1\}$ given by

$$f(s) := \begin{cases} 1 & \text{for } s = s_0, \\ 0 & \text{otherwise.} \end{cases}$$

In the quantum computational set-up we consider the the n-fold tensor power

$$\mathbb{H} := (\mathbb{C}^2)^{\otimes n} = \mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2$$

of the Hilbert space $\mathbb{C}^2$ whose standard (orthonormal) basis is given by

$$(e_0, e_1) := \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right).$$

A classical tuple $(i_1, \ldots, i_n) \in \{0,1\}^n$ is represented by the quantum state

$$e_{i_1} \otimes \ldots \otimes e_{i_n} \in \mathbb{H}$$

which can be realized on a quantum computer. Note that the system

$$B := (e_{i_1} \otimes \ldots \otimes e_{i_n})_{(i_1,\ldots,i_n)\in\{0,1\}^n}$$

is an orthonormal basis of Hilbert space $\mathbb{H}$.

Let $H : \mathbb{C}^2 \to \mathbb{C}^2$ be the Hadamard operator which is described by the unitary matrix

$$H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Grover's algorithm starts with the state

$$e_0 \otimes \ldots \otimes e_0 \in \mathbb{H}$$

which can be transformed into the entangled state

$$x^{(0)} := H_n(e_0 \otimes \ldots \otimes e_0) := (He_0 \otimes \ldots \otimes He_0) \tag{2.1}$$

by the application of the n-fold tensor power $H_n := H \otimes \ldots \otimes H$ of Hadamard operator H. According to the introduced identification of tuples in $\{0,1\}^n$ with

tensors in the orthonormal basis B, the prepared state $x^{(0)} \in \mathbb{H}$ corresponds to the uniform distribution on $\{0,1\}^n$.

To describe an iteration step of Grover's algorithm let us enumerate elements of basis B in their lexical order by the numbers $0, 1, \ldots, N-1$. Let $i_s \in \{0, 1, \ldots, N-1\}$ correspond to tuple s_0. We define the unitary operator $V_f : \mathbb{H} \to \mathbb{H}$ by

$$V_f(e_{i_1} \otimes \ldots \otimes e_{i_n}) := \begin{cases} -e_{i_1} \otimes e_{i_2} \otimes \ldots \otimes e_{i_n} & \text{for } (i_1, \ldots, i_n) = s_0, \\ e_{i_1} \otimes \ldots \otimes e_{i_n} & \text{otherwise.} \end{cases}$$

V_f is represented by the $N \times N$-matrix $R_N = (r_{ij})_{i,j=0,\ldots,N-1}$, where

$$r_{ij} = \begin{cases} -1 & \text{for } i = j = i_s, \\ 1 & \text{for } i = j \neq i_s, \\ 0 & \text{otherwise.} \end{cases}$$

Note that operator V_f can be efficiently implemented on a quantum computer when function f is involved.

Let I_N denote the $N \times N$ identity matrix and E_N the $N \times N$-matrix of ones. Obviously,

$$D_N := -I_N + \frac{2}{N} E_N$$

is an unitary matrix. Grover's iteration can be introduced as the repeated application of matrix $D_N R_N$:

$$x^{(k+1)} := D_N R_N \, x^{(k)} \qquad (k \geq 0), \tag{2.2}$$

where initial state $x^{(0)}$ is defined in (2.1), cf. Homeister (2008).

Plausibility considerations entail that after some number l of steps performed according to (2.2) component $x_{i_0}^{(l)}$ of vector $x^{(l)}$ indicates a high probability $(x_{i_0}^{(l)})^2$ of retrieval of s_0 by measuring state $x^{(l)}$ w.r.t. orthonormal basis B.

The presented description of Grover's algorithm suggests that it can be simulated on a classical computer by representing the state tensor and the iterative application of matrix $D_N R_N$ using standard data structures and commands of an imperative programming language.

Our simulation experiments confirm the fact that probability $(x_{i_0}^{(k)})^2$ increases after k steps for $0 \leq k \leq l$ for an appropriate l depending on n; for $k > l$ the probability of retrieval of item s_0 decreases performing an oscillatory behavior. This observation motivates the search for an optimal number of steps ensuring retrieval with a prescribed probability.

3. The Structure of Grover's Iteration Procedure

According to Section 2 the complexity of Grover's algorithm can be determined based on a simulation of a quantum computer on a classical one. In the present section we show that the complexity can be also accessed by a structural study of Grover's iteration procedure.

The initial state $x^{(0)}$ of the iteration has the representation

$$x^{(0)} = \frac{1}{\sqrt{N}} \cdot u$$

w.r.t. basis B where tuples $(i_1, \ldots, i_n)$ are coded by integers $0, 1, \ldots, N-1$ and $u = (1, \ldots, 1)^t \in \mathbb{C}^N$ (where x^t denotes the transpose of vector x). Let e_{i_s} denote the canonical unit vector of $\mathbb{C}^N$ whose i_s-th component is 1. Put

$$u_{s_0} := \frac{1}{\sqrt{N-1}} \cdot (u - e_{i_s}).$$

Obviously, (e_{i_s}, u_{s_0}) is an orthonormal pair of vectors such that

$$x^{(0)} = \frac{1}{\sqrt{N}} \cdot e_{i_s} + \sqrt{\frac{N-1}{N}} \cdot u_{s_0} = \cos\varphi_0 \cdot e_{i_s} + \sin\varphi_0 \cdot u_{s_0}$$

holds where

$$\varphi_0 := \arctan\sqrt{N-1}.$$

Put

$$U := \mathrm{LH}(e_{i_s}, u_{s_0}),$$

where LH denotes the linear hull.

It is readily checked that

$$D_N R_N e_{i_s} = \frac{N-2}{N} \cdot e_{i_s} - \frac{2\sqrt{N-1}}{N} \cdot u_{s_0}$$

and

$$D_N R_N u_{s_0} = \frac{2\sqrt{N-1}}{N} \cdot e_{i_s} + \frac{N-2}{N} \cdot u_{s_0}.$$

It follows that U is invariant under $D_N R_N$ and $x^{(k)} \in U$ for $k = 0, 1, \ldots$. Moreover, the operation of matrix $D_N R_N$ on U can be represented by the rotation matrix

$$A_N := \begin{pmatrix} \frac{N-2}{N} & -\frac{2\sqrt{N-1}}{N} \\ \frac{2\sqrt{N-1}}{N} & \frac{N-2}{N} \end{pmatrix} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$$

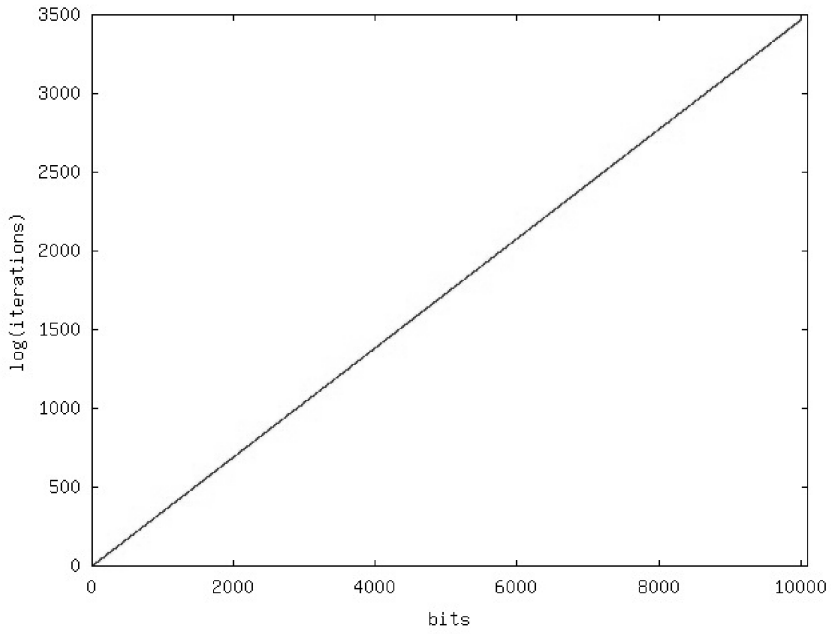

Figure 1: Number of steps versus size of the search space in bits

w.r.t. the orthonormal basis (e_{i_s}, u_{s_0}), where

$$\varphi := \arctan\left(\frac{2\sqrt{N-1}}{N-2}\right).$$

The powers of matrix A_N are given by

$$A_N^k = \begin{pmatrix} \cos k\varphi & -\sin k\varphi \\ \sin k\varphi & \cos k\varphi \end{pmatrix} \qquad (k = 1, 2, \ldots).$$

It follows that

$$x^{(k)} = \cos\varphi_k \cdot e_{i_s} + \sin\varphi_k \cdot u_{s_0},$$

where $\varphi_k = \varphi_0 + k\varphi$ for $k = 0, 1, \ldots$.

An indicator of efficiency of Grower's algorithm is the number l_N of iteration steps entailing

$$\cos^2\varphi_{l_N} \geq \frac{1}{2}$$

for the probability $\cos^2\varphi_{l_N}$ of retrieval of item i_s. Standard reasoning implies

that

$$l_N = \left\lceil \frac{0.75 \cdot \pi - \varphi_0}{\varphi} \right\rceil$$

where $\lceil x \rceil$ denotes the smallest integer $\geq x$.

In Figure 1 the horizontal axis corresponds to the number n of bits required for the representation of the search space. The vertical axis corresponds to the (natural) logarithm of the number l_N of Grover iterations required for the retrieval of item i_s with probability $\geq 1/2$. The diagram shows a linear dependence between n and $\log l_N$. The regression ansatz

$$l_N = \beta \cdot N^{\alpha}$$

with parameters α and β entails the least squares estimates

$$\widehat{\alpha} = 0.4999897 \qquad \text{and} \qquad \widehat{\beta} = 0.392887$$

that clarify the exact complexity of Grower's algorithm.

Acknowledgments

The authors greatly appreciate valuable comments of Martin Könenberg from Hagen concerning the present contribution.

References

[1] L. Grover, A fast quantum-mechanical algorithm for database search, In: *Proc. 28 STOC* (1998), 212-219.

[2] M. Homeister, *Quantum Computing Verstehen*, Second Edition, Vieweg, Wiesbaden (2008).

[3] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (2007).