Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2008), *22*, 285–297. Printed in the USA.

doi:10.1017/S089006040800019X

Evolving blackbox quantum algorithms using genetic programming

RALF STADELHOFER,[1] WOLFGANG BANZHAF,[2] AND DIETER SUTER[3]

[1]Department of Computer Science, University of Dortmund, Dortmund, Germany
[2]Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada
[3]Department of Physics, University of Dortmund, Dortmund, Germany

(RECEIVED February 21, 2005; ACCEPTED June 29, 2007)

Abstract

Although it is known that quantum computers can solve certain computational problems exponentially faster than classical computers, only a small number of quantum algorithms have been developed so far. Designing such algorithms is complicated by the rather nonintuitive character of quantum physics. In this paper we present a genetic programming system that uses some new techniques to develop and improve quantum algorithms. We have used this system to develop two formerly unknown quantum algorithms. We also address a potential deficiency of the quantum decision tree model used to prove lower bounds on the query complexity of the parity problem.

Keywords: Genetic Programming; Quantum Computation; Quantum Information Processing

1. INTRODUCTION

Recently, the concept of using quantum physical devices to perform computation has received much attention caused by the celebrated factoring algorithm proposed by Shor (1994). This algorithm provides an exponential speed-up in comparison to all known classical factoring algorithms. Unfortunately, increased efforts following this promising result did not return any further developments as impressive. More recent results have cooled expectations that proved to be too optimistic (Bennett et al., 1997; Bernstein & Vazirani, 1997; Beals et al., 2001), and it still remains unclear whether quantum computers (QCs) will provide an alternative superior to classical computation.

Yet the concept of computational power provides a fundamentally new language for studying the relationship between classical and quantum physics. Therefore, even modest speed-ups of quantum algorithms (QAs) compared to classical algorithms might provide further insight into the fundamental differences between classical and quantum physics. This can be helpful in estimating the computational benefit of QCs.

Unfortunately, the design and development of QAs is a very cumbersome task, mainly because of the nonintuitive character of quantum physics. Thus, it is reasonable to investigate automated algorithm design techniques in the development of new QAs. The usage of genetic programming (GP) was pioneered by Williams and Gray (1998), who used this approach to decompose a given quantum transformation into a sequence of elementary quantum gates. Another approach, used by Spector et al. (1999), enabled the development of quantum circuits without knowing the quantum transformation in advance. Since then, several related strategies using GP to aid the development of QAs have been proposed and investigated by several authors (e.g., Leier & Banzhaf, 2003*a*, 2003*b*; Massey et al., 2004).

Here we present our GP system that we used to develop two formerly unknown better-than-classical QAs. We start with an introduction to quantum computation by means of the blackbox model of computation in Section 2. In Section 3 we present our GP system and discuss the conditions a fitness function should fulfill to facilitate the development of QAs. The QAs that we developed are presented in Section 4. Section 5 concludes.

2. QUANTUM COMPUTATION AND THE BLACKBOX MODEL OF COMPUTATION

An informative introduction to the complex topic of quantum computation can be found in Rieffel and Polak (2000). For a comprehensive overview we defer the reader to the book of Nielsen and Chuang (2000).

Reprint requests to: Ralf Stadelhofer, Department of Computer Science, LS XI, University of Dortmund, Dortmund 44221, Germany. E-mail: ralf.stadelhofer@udo.edu

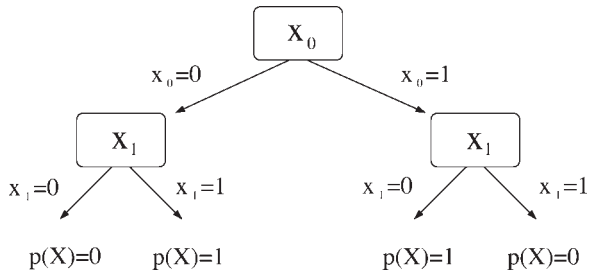

Fig. 1. A classical decision tree that computes the parity $p(X)$ of the blackbox.

A large class of QAs can be formulated using the blackbox model of computation (Beals et al., 2001). In this contribution we restrict our investigations to blackbox problems only.

2.1. Blackbox model of computation

A blackbox $X = (x_0, x_1, \ldots, x_{N-1})$, also referred to as an oracle, consists of N Boolean variables x_i with $x_i \in \{0, 1\}$. On input i, the oracle returns the Boolean variable x_i. Usually one wants to compute a Boolean property $p(X)$ of such a blackbox using as few oracle queries as possible. The number of these queries is also called query complexity, which is the relevant complexity measure in this context.[1] A classical algorithm that computes property $p(X)$ of a blackbox X can be represented by a decision tree with the kth query depending on the outcome of the $k - 1$ previous queries (see Fig. 1).

Consider, for example, the parity problem: one wants to know if a binary string $x_{N-1}x_{N-2}\cdots x_0$ of length N contains an even or an odd number of entries x_i with $x_i = 1$.

As can be seen in Figure 1, a classical decision tree that computes the parity of the blackbox $X = (x_0, x_1)$ has to query the two elements one after the other.

Before we introduce the notation and essential ideas of quantum computing, we want to stress the main resource used by QCs to reduce the number of oracle queries: it is the exploitation of superpositions and interference of states that enables QCs to query several blackbox elements simultaneously.

From now on we distinguish between single-issue QCs and ensemble QCs. A single-issue QC denotes a single quantum system like the ions of an ion trap that is used for computation. An ensemble QC denotes a large number of quantum systems that are used for computation. The NMR sample of a liquid state NMR-QC, for example, contains ~10^{18} molecules, each a valid QC.

2.2. Single-issue QCs

A bit is defined to be either in the state 0 or in the state 1, whereas a quantum bit (qubit) can be in a superposition of both states. In quantum physics a physical state j (here $j \in \{0, 1\}$) is represented by a vector denoted by $|j\rangle$. With this notation the superposition $|\psi\rangle$ over both states represents a qubit and looks as follows:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \text{with} \quad \alpha, \beta \in \mathbb{C} \quad \text{and} \quad |\alpha|^2 + |\beta|^2 = 1.$$

The probability amplitudes α, β are used to calculate the probabilities $|\alpha|^2$, $|\beta|^2$ that a measurement returns the state $|0\rangle$, $|1\rangle$, respectively. The probability to measure either state $|0\rangle$ or state $|1\rangle$ has to be $|\alpha|^2 + |\beta|^2 = 1$.

A quantum system composed of several qubits is called a quantum register. The notation $|i_1 i_0\rangle \equiv |i_1\rangle|i_0\rangle \equiv |i_1\rangle \otimes |i_0\rangle$ with i_0, $i_1 \in \{0, 1\}$ describes a two-qubit register; $\otimes$ denotes the tensor product. This notation can easily be extended to describe n-qubit registers. A state like $|\psi\rangle = 1/\sqrt{2}(|00\rangle + |11\rangle)$ that is not decomposable into a tensor product of the subsystems' states is called entangled:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq \underbrace{(\alpha_1|0\rangle + \beta_1|1\rangle)}_{\text{qubit 1}} \otimes \underbrace{(\alpha_0|0\rangle + \beta_0|1\rangle)}_{\text{qubit 0}},$$

where α_1, α_0, β_1, $\beta_0 \in \mathbb{C}$. Entanglement is a resource necessary to query blackbox elements in parallel using superpositions and binary encoding (see Lloyd, 2000). The states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ define the computational basis for the vector space of two-qubit states. With $(1, 0, 0, 0)^t \equiv |00\rangle$, $(0, 1, 0, 0)^t \equiv |01\rangle$, and so forth, the state $|\psi\rangle$ above is represented by the vector $(1/\sqrt{2}, 0, 0, 1/\sqrt{2})^t$.

It is well known that any arbitrary quantum operation (unitary transformation) can be decomposed into a sequence of unitary quantum operations on one and two qubits (Nielsen & Chuang, 2000). This is somehow similar to the fact that any Boolean function can be decomposed into a sequence of elementary logic gates (e.g., NOT, AND, OR). Because quantum gates are described by continuous parameters it is, however, only possible to approximate such an operation by a sequence of elementary one- and two-qubit gates, taken from a discrete set, up to an error ε (Nielsen & Chuang, 2000). The elementary one-qubit gates we used in our investigation are the Hadamard gate $\mathbf{H}$ and a discrete set of the rotation gates $\mathbf{R}_x(\alpha_k)$ and $\mathbf{R}_y(\alpha_k)$. The discrete value α_k is calculated by $\alpha_k = -\pi + k \cdot 2\pi/\xi$ with $k \in \{1, \ldots, \xi\}$, where $\xi \in \mathbb{N}$ is chosen dependent on the desired accuracy of the approximation. With the Pauli matrices σ_x, σ_y, σ_z and the identity matrix $\mathbb{I}$ defined by

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -\imath \\ \imath & 0 \end{pmatrix},$$
$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

the one-qubit gates $\mathbf{R}_x(\alpha_k)$ and $\mathbf{R}_y(\alpha_k)$ are given by

$$\mathbf{R}_x(\alpha_k) = \exp\left(-\imath\frac{\alpha_k}{2}\sigma_x\right) = \cos\left(\frac{\alpha_k}{2}\right)\mathbb{I} - \imath\sin\left(\frac{\alpha_k}{2}\right)\sigma_x.$$
$$\mathbf{R}_y(\alpha_k) = \exp\left(-\imath\frac{\alpha_k}{2}\sigma_y\right) = \cos\left(\frac{\alpha_k}{2}\right)\mathbb{I} - \imath\sin\left(\frac{\alpha_k}{2}\right)\sigma_y.$$

[1] The total number of gates is usually not considered.

The Hadamard gate is defined by

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

Hadamard gates are used by us only for convenience since the one-qubit gates $\mathbf{R}_x(\alpha_k)$, and $\mathbf{R}_y(\alpha_k)$ would be already sufficient to create any arbitrary one-qubit gate $\mathbf{U}$ via (see Nielsen & Chuang, 2000):

$$\mathbf{U} = e^{\imath\alpha}\mathbf{R}_x(\beta)\mathbf{R}_y(\gamma)\mathbf{R}_x(\delta) \quad \text{with} \quad \alpha, \beta, \gamma, \delta \in [0, 2\pi].$$

A single discrete two-qubit operation called **CNOT**-gate, together with the discrete set of one-qubit gates defined above is sufficient to provide a complete set of unitary operations that is capable of approximating any arbitrary unitary operation up to the desired accuracy (Nielsen & Chuang, 2000). The **CNOT**-gate is defined by its action on a two-qubit state:

$$\mathbf{CNOT}|l\rangle|m\rangle = |l\rangle|m \oplus l\rangle \quad \text{with} \quad l, m \in \{0, 1\}. \qquad (1)$$

Here, $\oplus$ denotes the XOR operation between the Boolean variables l and m. Analogous to a Boolean circuit it is possible to define a quantum circuit as a directed acyclic graph whose nodes are labeled by elementary quantum gates like the ones defined above. Because of reversibility of quantum operations and the no-cloning theorem (Nielsen & Chuang, 2000) any elementary quantum gate has as many inputs as outputs. An example of such a quantum circuit on two qubits together with the final measurements can be found in the upper part of Figure 2.

2.3. Ensemble QCs

Up to now, we only considered single quantum systems whose state is known. In this case, one usually speaks of a pure state of the quantum system. Even in single-issue QCs, unavoidable decoherence processes turn the initially pure states into mixed states. Such states cannot be described by a single quantum state. The appropriate formalism for their description is the density operator that describes systems where one only knows the probability p_i that a certain pure quantum state $|\psi_i\rangle$ occurs.

Before we can define this operator we have to introduce the dual state $\langle\psi|$ of a quantum state $|\psi\rangle$. Using this notation the scalar product of two quantum states $|\psi\rangle$ and $|\phi\rangle$ is given by $\langle\psi|\phi\rangle$. With $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \equiv (\alpha, \beta)^t$ one gets $\langle\psi| \equiv \alpha^*\langle 0| + \beta^*\langle 1| \equiv (^*\alpha^*, \beta^*)$. Using this notation the normalization condition can be formulated as $\langle\psi|\psi\rangle = \alpha\alpha^* + \beta\beta^* = 1$. Another necessary concept used in quantum physics is the concept of the measurement operator, which usually can be represented by a Hermitian matrix. The expectation value $\langle\mathbf{M}\rangle$ of a measurement described by the measurement operator $\mathbf{M}$ for a quantum system in the pure state $|\psi\rangle$ is calculated by

$$\langle\mathbf{M}\rangle = \langle\psi|\mathbf{M}|\psi\rangle = \mathrm{tr}\{\mathbf{M} \cdot |\psi\rangle\langle\psi|\} = \mathrm{tr}\{\mathbf{M}\rho\}, \quad \text{with} \quad \rho = |\psi\rangle\langle\psi|.$$

Here ρ denotes the density operator of the pure quantum state $|\psi\rangle$. With $|i\rangle$ denoting a basis state of the vector space of our quantum system, the trace $\mathrm{tr}\{\mathbf{A}\}$ of an operator is defined by $\mathrm{tr}\{\mathbf{A}\} = \sum_i \langle i|\mathbf{A}|i\rangle$ with $i \in \{0, 1, \ldots, 2^n - 1\}$ where n denotes the number of qubits. A measurement can be described by a projection operator $\mathbf{P}$. Consider, for example, the state $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|11\rangle$. If one wants to know the probability to measure the leftmost qubit to be in state $|0\rangle$ the corresponding measurement operator looks as follows:

$$\mathbf{P}_0 = |0\rangle\langle 0| \otimes \mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Calculation of $\langle\mathbf{P}_0\rangle$ returns $\langle\mathbf{P}_0\rangle = |\alpha|^2 + |\beta|^2$. This procedure provides the formal framework to calculate measurement probabilities for pure states. Nevertheless, as we only

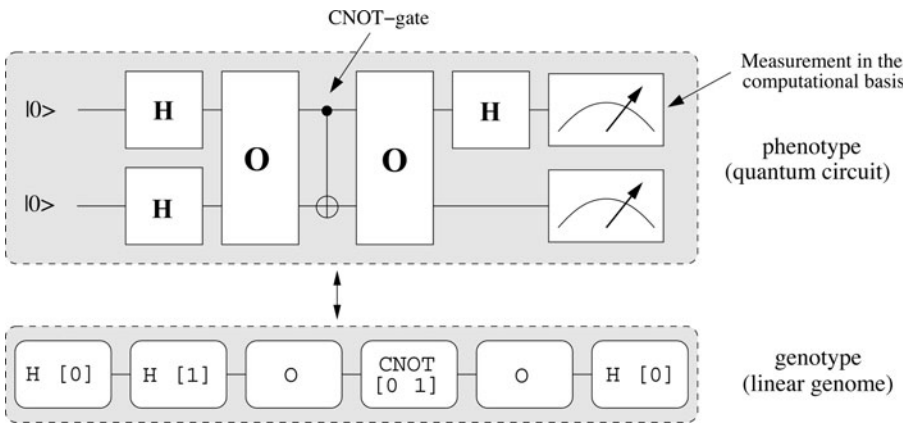

Fig. 2. A quantum circuit is represented by a linear list (linear genome) of the quantum gates used. The sequence of quantum gates is obtained by reading this linear list from left to right. H [1] denotes a Hadamard gate **H** that is applied to qubit 1. CNOT [0 1] denotes a **CNOT** gate where qubit 0 is the control qubit and qubit 1 is the target qubit.

consider measurements in the computational basis we added the absolute squares of the probability amplitudes directly, as it is obvious which probability amplitudes are involved without using this time-consuming calculation scheme.

A mixed state that is in the state $|\psi_i\rangle$ with probability p_i is represented by $\rho = \Sigma_i\, p_i|\psi_i\rangle\langle\psi_i|$. The expectation value $\langle\mathbf{M}\rangle$ for a mixed state is calculated by

$$\langle\mathbf{M}\rangle = \sum_i p_i\langle\psi_i|\mathbf{M}|\psi_i\rangle = \mathrm{tr}\{\mathbf{M}\cdot\rho\}.$$

Let the unitary operator $\mathbf{A}$ describe a quantum computation and let ρ_{init} describe the initial state of the quantum register. The final quantum state ρ_{fin} can be calculated by $\rho_{\mathrm{fin}} = \mathbf{A}\rho_{\mathrm{init}}\mathbf{A}^+$, where $\mathbf{A}^+$ is the transposed and complex conjugate of the matrix $\mathbf{A}$: $(\mathbf{A}^+)_{ij} = (\mathbf{A}^{-1})_{ij} = \mathbf{A}^*_{ji}$. In our investigations concerning ensemble QCs we only considered liquid-state NMR-QCs (an introduction to NMR-QCs can be found in Nielsen & Chuang, 2000). Here, the initial state is the thermal state

$$\rho_{\mathrm{th}} \approx \frac{1}{2^n}(\mathbb{I} - \mathcal{H}) \approx \frac{1}{2^n}\left(\mathbb{I} - \sum_{i=0}^{n-1}\omega_i\mathbf{I}_z^{(i)}\right), \tag{2}$$

where we have set $\hbar/k_BT = 1$ and invoked the high-temperature approximation. Here, $\mathcal{H}$ denotes the Hamiltonian of the spin system, ω_i is the Larmor frequency of the ith spin (qubit), and $\mathbf{I}_z^{(i)}$ the corresponding spin operator:

$$\mathbf{I}_z^{(i)} = \mathbb{I}\otimes\cdots\otimes\mathbb{I}\otimes\underbrace{\frac{\sigma_z}{2}}_{i\text{th spin}}\otimes\mathbb{I}\otimes\cdots\otimes\mathbb{I}.$$

In liquid-state NMR quantum computation one measures the magnetization of the ith nuclear spin in the molecules of the NMR sample along the x and y axis. We restricted ourselves to measurements along the x axis, which are described by measurement operators of the form

$$\mathbf{I}_x^{(i)} = \mathbb{I}\otimes\cdots\otimes\mathbb{I}\otimes\underbrace{\frac{\sigma_x}{2}}_{i\text{th spin}}\otimes\mathbb{I}\otimes\cdots\otimes\mathbb{I}.$$

3. EVOLVING BLACKBOX QAs USING GP

In this section we take a closer look at the GP system in the development of QAs that will be presented in Section 4. Our GP system was used to evolve quantum circuits for small problem instances on a few qubits only. These circuits were examined for scalability to find the corresponding uniform circuit family and thus the algorithm that solves the problem. At first we describe how a quantum circuit is represented by our GP system. Then we present the different oracle gates used. Because we evolved quantum circuits for single issue QCs as well as for ensemble QCs the section about fitness functions is divided into two parts.

3.1. Representation of quantum decision trees in our GP system

As already mentioned in the last section, a computation of a blackbox's property can be visualized by a decision tree. According to Buhrman and de Wolf (2002) a quantum decision tree is defined by a sequence of oracle gates $\mathbf{O}$ that represent blackbox calls, alternating with unitary transformations $\mathbf{U}_i$:

$$\mathbf{A} = \mathbf{U}_T\cdot\mathbf{O}\cdot\mathbf{U}_{T-1}\cdot\mathbf{O}\cdots\mathbf{U}_1\cdot\mathbf{O}\cdot\mathbf{U}_0, \quad \text{with} \quad T\in\mathbb{N}. \tag{3}$$

Such a sequence can also be used to simulate classical decision trees (see Buhrman & de Wolf, 2002). Consider a single-issue QC; here, the sequence denoted by the unitary transformation $\mathbf{A}$ is applied to the initial state $|\psi_{\mathrm{init}}\rangle = |0\cdots 0\rangle$. Measuring the readout-qubit of the final state $|\psi_{\mathrm{final}}\rangle = \mathbf{A}|\psi_{\mathrm{init}}\rangle$ returns a binary value. If this binary value equals $p(X)$ for all blackboxes one says that the quantum decision tree computes the property $p(X)$ of the blackbox encoded by the oracle gate $\mathbf{O}$.

It is adequate to represent quantum decision trees by quantum circuits. To do that one needs a complete set of one-qubit and two-qubit gates. In addition, one needs the set of oracle gates $\mathbf{O}$ representing the blackboxes X. In principle, the measurement of a readout qubit should be sufficient to decide the property $p(X)$ of a blackbox X; nevertheless, algorithms like the Deutsch–Jozsa (DJ) algorithm answer the problem by measuring several qubits (Deutsch & Jozsa, 1992). This procedure makes it possible to dispense with the additional output qubit and additional quantum operations necessary to encode the answer into this output qubit. Now that the answer to the posed problem cannot be obtained any more by measuring the state of a single qubit (output qubit), the QA has to return different measurement results for those blackboxes X that differ in their property $p(X)$. How this can be realized is shown in Section 3.3.1. We decided to perform all measurements possible on the final state $|\psi_{\mathrm{final}}\rangle$ to check if one of these measurements returns different results for blackboxes X that differ in their property $p(X)$.[2] If the state $|\psi_{\mathrm{init}}\rangle$ is encoded by n qubits we thus perform all n one-qubit measurements, all $n(n-1)/2$ two-qubit measurements and so on.[3]

As in the work of Spector et al. (1999), we have chosen a linear genome to represent quantum circuits in our GP system (see Fig. 2). Because a quantum gate is specified by several parameters like rotation angles, control-qubits, and so forth, one has to decide where these additional data are to be stored and how they are to be manipulated by the evolutionary process. The most natural method is to consider a quantum gate and these additional parameters as a unit, allowing the evolutionary process to only modify the unit as a whole. Because any QA starts with the initial state $|\psi_{\mathrm{init}}\rangle$ and qubits to be measured are specified by a global variable, we did not encode

[2] We only consider measurements in the computational basis.

[3] This corresponds to a total of 2^n measurements. This is done to evaluate the quantum circuit, and therefore has no influence on the scalability of the quantum circuit.

these parameters into the genome. As stated in the caption of Figure 2, the sequence of quantum gates is obtained by reading the genome from left to right.

3.2. Oracle gates

To see if a quantum circuit correctly calculates the property $p(X)$ of blackbox X one has to check this circuit for all blackboxes. The blackboxes are encoded into a quantum system via the oracle gates $\mathbf{O}$, so these gates provide the fitness cases the quantum circuit is to be tested with. Depending on the property a quantum circuit has to compute, the number of blackboxes (oracles) that are to be tested can grow superexponentially. Thus, not only the effort to simulate a quantum system grows exponentially with the number of qubits, but also the number of oracles that are to be tested increases quickly rendering an investigation of quantum circuits with many qubits impractical. One should, therefore, strive to reduce the number of oracles to be tested.

One possibility to do that is to use an encoding of the blackbox values into a quantum state that enables a reduction in the number of oracles to be tested. The DJ problem for a single query qubit can be used to illustrate this procedure.[4] Here, the constant blackboxes $X_1 = (0, 0)$, $X_2 = (1, 1)$ with $p(X_{1,2}) = 0$ are to be distinguished from the balanced blackboxes $X_3 = (0, 1)$, $X_4 = (1, 0)$ with $p(X_{3,4}) = 1$. If one employs the usual definition of the oracle gate

$$\mathbf{O}|k\rangle|0\rangle = |k\rangle|x_k\rangle, \quad k, x_k \in \{0, 1\}, \tag{4}$$

one has to test each of the four different oracles. Because the DJ problem can also be solved by using the following oracle gate (Collins et al., 1998):

$$\mathbf{O}|k\rangle = (-1)^{x_k}|k\rangle, \tag{5}$$

it is possible to find a quantum circuit that solves this problem by testing two oracles only. Using the latter definition returns the same oracle, up to a global phase shift, for the two constant blackboxes: $\mathbf{O}_{1,2} = \pm\mathbb{I}$. For the two balanced blackboxes one gets: $\mathbf{O}_{3,4} = \pm\sigma_z$. Because global phase shifts are not measurable, it is not necessary to test all four different oracles but only the oracle $\mathbf{O}_1 = \mathbf{I}$ and $\mathbf{O}_3 = \sigma_z$. A further advantage of this method is that one needs no additional output register as would be the case with the definition in Eq. (4).

Nevertheless, it depends on the property $p(X)$ one is interested in, whether an encoding of the blackbox entry x_k into the phase $(-1)^{x_k}$ is possible. If, for instance, one wants to calculate the property $p(X_1) = 0$ and $p(X_2) = 1$ with $X_1 = (0, 0)$ and $X_2 = (1, 1)$ it is impossible to encode the blackbox entries into phase shifts, because then both cases become indistinguishable for a quantum system as they only differ by a global phase shift. To see if an encoding into a local phase shift via Eq. (5) is possible one thus has to check if for every pair of blackboxes $X = (x_0, x_1, \ldots, x_n)$ and $\bar{X} = (\bar{x}_0, \bar{x}_1, \ldots, \bar{x}_n)$ the condition $p(X) = p(\bar{X})$ holds. Here, $\bar{x}_k$ denotes the negation of the binary variable x_k.

Up to now we only considered blackboxes X whose entries were binary values. In Simon's problem (Simon, 1994) one investigates blackboxes $X = (x_0, x_1, \ldots, x_{N-1})$ with $x_k \in \{0, 1, \ldots, l-1\}$ and $l \in \mathbb{N}$. If one applies the approach of Eq. (4), one needs an output register size of $\log_2(l)$ qubits. Up to now, both, in simulation and in experiments, only a few qubits are realizable, and thus this approach is not reasonable for $l \gg 1$. Therefore, we used a combined approach where the blackbox entries with $l \in \mathbb{N}$ are encoded partially into a phase shift and partially into an m-qubit output register. We write the decomposition $x_k^{(\text{phase})}$ of $\underline{x_k}^{(\text{XOR})}$ for blackbox entry x_k if it is decomposed into part $x_k^{(\text{phase})}$ encoded into a phase shift, and part $\underline{x_k}^{(\text{XOR})}$ encoded into an output register. Consider, for example, blackbox entries $x_k \in \{0, \ldots, 15\}$. With two output qubits one only can encode four different values. Therefore, one needs another four different phase shifts to encode unambiguously. The value $x_k = 13$ would thus be decomposed into $x_k^{(\text{phase})} = 3$ and $\underline{x_k}^{(\text{XOR})} = 01$. This decomposition follows from the binary representation of 13: 1101. The two rightmost values are the assignment for $\underline{x_k}^{(\text{XOR})}$, the integer represented by the two leftmost qubits is the value of the phase shift.

In general, the oracle gate looks like:

$$\mathbf{O}|\underline{k}\rangle|\underline{b}\rangle = (\zeta)^{x_k^{(\text{phase})}}|\underline{k}\rangle|\underline{b} \oplus \underline{x_k}^{(\text{XOR})}\rangle, \quad \zeta = \exp\left(\frac{2\pi\imath}{l/2^m}\right), \tag{6}$$

with $k \in \{0, 1, \ldots, N-1\}$ and $b, x_k \in \{0, 1\}^m$. Here, we write $\underline{b} \oplus \underline{\bar{x}_k}^{(\text{XOR})}$ to denote the bitwise XOR-operation between the m-bit value $\underline{b}$ and the m-bit value $\underline{\bar{x}_i}^{(\text{XOR})}$. We write $\underline{k}$ to denote the binary decomposition of the integer k. For $m = 0$, the blackbox entries are entirely encoded into phase shifts, which saves $\log_2(l)$ qubits. Nevertheless, as before, not every blackbox problem can be encoded using this approach. For $m = \log_2(l)$ the blackbox entries are completely encoded into the output register.

In Section 4 we will present a QA that was evolved using oracle gates as defined in Eq. (6).

3.3. Fitness functions

The definition of the fitness function is the most sensitive part in setting up the GP system because here one has to make assumptions that might turn out to be ill-suited for evolving quantum circuits. The mathematical structure of quantum physics allows to define a metric on the space of unitary operators, that is, quantum circuits, enabling the assignment of a step length to mutation and crossover operators. Small mutations, according to this measure, lead to small changes in the measured expectation values (see Nielsen & Chuang, 2000). Because the fitness function has to depend on these expectation values it is desirable that it retains this property. We

[4] The term query qubits denotes qubits that are used to address blackbox elements.

would like to emphasize, however, that all QAs presented here were evolved using a fitness function that did not fully obey this condition.

3.3.1. Fitness function for single issue QCs

We will motivate the definition of the fitness function by means of the DJ problem for blackboxes with four elements. Nevertheless, the problems we evolved QAs for that are presented in this article are the parity problem and a special case of the hidden subgroup problem and not the DJ problem. Both problems were treated by our GP system using fitness functions similar to the one we now discuss.

For four blackbox elements the DJ problem is to distinguish between the constant blackboxes $X_1 = (0, 0, 0, 0)$, $X_2 = \bar{X}_1 = (1, 1, 1, 1)$ with $p(X_{1,2}) = 0$ and the balanced blackboxes:

$$X_3 = (0,0,1,1), \quad X_4 = (0,1,0,1), \quad X_5 = (0,1,1,0)$$
$$X_6 = \bar{X}_3 = (1,1,0,0), \quad X_7 = \bar{X}_4 = (1,0,1,0), \quad X_8 = \bar{X}_5 = (1,0,0,1),$$

with $p(X_{i\geq 3}) = 1$. According to Section 3.2 this problem can be solved by encoding the blackbox entries into phase shifts via the oracle gates defined by Eq. (5). Thus fitness depends on the measurement results for the four blackboxes $X_1 = (0, 0, 0, 0)$, $X_3 = (0, 0, 1, 1)$, $X_4 = (0, 1, 0, 1)$ and $X_5 = (0, 1, 1, 0)$. The oracle gates of the remaining blackboxes differ from the oracle gates $\mathbf{O}_1$, $\mathbf{O}_3$, $\mathbf{O}_4$, and $\mathbf{O}_5$, which represent the four blackboxes mentioned above by a global phase shift only (see also Section 3.2).

Table 1 shows measurement probabilities for a fictive quantum circuit applied to input state $|\psi_{\text{init}}\rangle = |00\rangle$. How well does this circuit decide between constant and balanced blackboxes? We consider a quantum state $|i_1 i_0\rangle$ to be measurable with a sufficient probability if $\text{prob}(|i_1 i_0\rangle) > 1/N$, where N is the number of quantum states we want to distinguish (here, $N = 4$) and $\text{prob}(|i_1 i_0\rangle)$ is the probability to measure state $|i_1 i_0\rangle$. A binary variable $b_{i_1 i_0}$ indicates whether quantum state $|i_1 i_0\rangle$ fulfills this condition (see Table 2).

Using $b_{i_1 i_0}$ for each blackbox X, one has to check whether $b_{i_1 i_0} = 1$ for constant and balanced blackboxes simultaneously. In this case, the quantum circuit is not able to distinguish between the different blackboxes. If, on the other hand, there is a difference in the outcome, we consider the circuit a promising candidate for solving the problem and the error probability of this circuit is calculated to check how well the circuit distinguishes between the two kinds of blackboxes.

Table 1. *Measurement probabilities for a fictive quantum circuit*

	prob(|00⟩)	prob(|01⟩)	prob(|10⟩)	prob(|11⟩)
$X_1 = (0, 0, 0, 0)$	0.8	0.05	0.05	0.1
$X_3 = (0, 0, 1, 1)$	0.15	0.25	0.25	0.35
$X_4 = (0, 1, 0, 1)$	0.1	0.15	0.15	0.6
$X_5 = (0, 1, 1, 0)$	0.1	0.5	0.3	0.1

$\text{prob}(|i_1 i_0\rangle)$, the probability of measuring quantum state $|i_1 i_0\rangle$ after applying the quantum circuit to $|00\rangle$.

Table 2. *Results obtained from Table 1 by setting the binary value $b_{i_1 i_0} = 1$ if $p(|i_1 i_0\rangle) > 1/4$*

	b_{00}	b_{01}	b_{10}	b_{11}
$X_1 = (0, 0, 0, 0)$	1	0	0	0
$X_3 = (0, 0, 1, 1)$	0	0	0	1
$X_4 = (0, 1, 0, 1)$	0	0	0	1
$X_5 = (0, 1, 1, 0)$	0	1	1	0

According to Table 2, only the constant blackbox X_1 is mapped to state $|\psi_{\text{final}}\rangle = |00\rangle$. The quantum circuit is therefore able to distinguish between constant and balanced blackboxes. However, as indicated in Table 1, there is a certain probability that a blackbox, for instance $X_3 = (0, 0, 1, 1)$, could be classified as constant (with a probability of 15%). Inspection of Table 1 shows further that the probability of misclassifying the balanced blackboxes $X_4 = (0, 1, 0, 1)$ and $X_5 = (0, 1, 1, 0)$ is 10% in both cases. It can also be calculated that the error for a constant blackbox is 20%.

It is better, therefore, to have the fitness function depend on at least two parameters. The first parameter is called clash, and quantifies how often it is not possible to distinguish between constant and balanced blackboxes. Another parameter, `worst_error`, denotes the highest probability of a misclassification. In the example we have `worst_error` = 0.2, the error probability that a constant blackbox is classified to be balanced. As further parameters we used `avg_error` denoting the average error, `oracles` to denote the number of oracle gates, and `length` to denote the total number of quantum gates used by the quantum circuit.

Similar to the approach used by Spector et al. (1999), these parameters were used to create a lexicographic fitness function represented by a vector of the form

$$f = (\texttt{clash}, \texttt{worst\_error}, \texttt{avg\_error}, \texttt{oracles}, \texttt{length}).$$

The position of the parameters in this vector represents their priority, decreasing from left to right.

This fitness function is to be contrasted with that of Spector et al. (1999), who measure the probability of the state of a single qubit. The advantage of our approach is that an evolved quantum circuit that solves the DJ problem has the possibility to resemble the original one, which decides the property of a blackbox with 2^n elements by n measurements. Therefore, we do not need additional quantum gates necessary to solve the problem by the measurement of a single qubit, only. Such additional quantum gates would make it difficult to extract the functionality and thus the scalability of a quantum circuit.

3.3.2. Fitness function for ensemble QCs

A liquid-state NMR-QC realizes qubits by the spin states of the molecule's spin-1/2 nuclei. Because a liquid-state

NMR-QC performs its computations on an ensemble of molecules, the fitness function has to be derived from expectation values $\langle \mathbf{I}_x^{(i)} \rangle$ of the measurement operator $\mathbf{I}_x^{(i)}$ as described in Section 2.3.

Consider, for example, a two-qubit system. Here, the initial state is described by

$$\rho_{\text{th}} = \frac{1}{4}\left(\omega_0 \mathbf{I}_z^{(0)} + \omega_1 \mathbf{I}_z^{(1)}\right). \tag{7}$$

In contrast to Eq. (2), we skipped the term proportional to the identity matrix as it does not contribute to the expectation values $\langle \mathbf{I}_x^{(i)} \rangle$.

A quantum circuit described by the unitary transformation $\mathbf{A}$ maps this state to $\rho = \mathbf{A}\rho_{\text{th}}\mathbf{A}^+$. The measurement of the ith spin's magnetization along the x axis is calculated by

$$\langle \mathbf{I}_x^{(i)} \rangle = \text{tr}\{\mathbf{I}_x^{(i)}\rho\}. \tag{8}$$

Analogous to the single-issue QC, all combinations of measuring the spins' magnetization are performed to check if one of these returns different results for blackboxes X that differ in their property $p(X)$. In our example, one therefore has to calculate $\langle \mathbf{M} \rangle$ for $\mathbf{M} = \mathbf{I}_x^{(0)}$, $\mathbf{M} = \mathbf{I}_x^{(1)}$, and $\mathbf{M} = \mathbf{I}_x^{(0)} + \mathbf{I}_x^{(1)}$.

To quantify how well a promising circuit solves the problem, we divided the interval $[-|\langle \mathbf{M} \rangle|, \ldots, |\langle \mathbf{M} \rangle|]$ of possible measurement results in several disjoint subintervals.[5] If the measurement results for blackboxes with $p(X) = 0$ belong to different subintervals than the measurement results for blackboxes with $p(X) = 1$ we consider the circuit a promising candidate for solving the problem. To decide how well this circuit can distinguish between both types of blackboxes we calculate the minimal distance between the corresponding measurement results. The variable `min_dist` denotes this minimal distance, whereas the variable `avg_dist` denotes the average distance.

The fitness function is represented by the vector

$$f = (\texttt{clash}, \texttt{min\_dist}, \texttt{avg\_dist}, \texttt{orcales}, \texttt{length}).$$

Analogous to the fitness function introduced in the last section we used a lexicographic ordering of the vector components with `clash`, the most significant one. Once again, an optimal circuit would have minimal values in all components of its fitness function.[6] Nevertheless, we still have the problem that the component `clash` does not respect the continuity of the expectation values in the step length of mutation operators. When `clash` $= 0$, the parameters `min_distance` and `avg_distance` become significant, and these parameters respect the continuity of the expectation values in the step length of mutation operators.

[5] By defining an operator norm via $\|\mathbf{M}\| = \sqrt{\text{tr}\{\mathbf{M}^+\mathbf{M}\}}$ and by applying Cauchy–Schwarz's inequality one gets $|\langle \mathbf{M} \rangle| \leq ||\mathbf{M}|| \cdot ||\rho_{\text{th}}||$. Thus, the interval of possible measurement results can easily be calculated.

[6] As the variables `min_distance` and `avg_distance` are to be maximized we used their negative values in the definition of the fitness function to remain consistent with the condition that an optimal fitness function has minimal values in all of its components.

4. QAS DEVELOPED BY GP

In this section we present two new better than classical QAs that were developed with the help of our GP system using the parameters shown in Table 3.

Both algorithms were found by applying the GP system to the smallest instances of the corresponding problem. The circuits returned by the GP system were examined for scalability to find the corresponding uniform circuit family and thus the algorithm.

Figure 3 shows the length distribution of the best individual found in each generation for 60 independent runs of the GP system for the $n = 3$ qubit parity problem. This result indicates that 50% of the runs find the optimal quantum circuit after approximately 375 generations. The genome of an optimal solution is

```
H[0]  0  Ry[1 95]  0  Rx[2 31]  0  Rx[1 95]  0  H[0]
```

A run (500 generations) takes about 27 min on a single core of an AMD Opteron 870 processor. Here, `Ry[1 95]` denotes that the one-qubit gate $\mathbf{R}_y(2 \cdot \alpha_{95})$ is applied to qubit 1, where $\alpha_{95} = \pi/2$. The quantum circuit in the lower part of Figure 4 can be derived from this one by replacing `Ry[1 95]` with `Rx[1 95]` and `Rx[2 31]` with `Rx[2 95]`. Nevertheless, the functionality of the circuit is not altered by these replacements; hence, the above genome represents an optimal solution.

The optimal quantum circuit for the $n = 2$ qubit parity problem is found with almost certainty after 500 generations (running time $= -1$ min). One in three runs of the GP system returned the optimal quantum circuit

Table 3. *Parameters of the GP system that evolved the circuits in Section 4.1 and 4.2.*

Population size	500
Max. no. of generations	500
Selection	Tournament, elitist
Tournament size	16
Crossover probability	0.05
Creation probability	0.05
Mutation probabilities	0.90
Swap	0.30×0.90
Grow	0.30×0.90
Shrink	0.20×0.90
Shrink2	0.20×0.90
No. of rotation angles	128
Max. no. of gates	100
Max. no. of oracle gates	8
Gate set	**CNOT**, $\mathbf{Rx}(2 \cdot \alpha_k)$, $\mathbf{Ry}(2 \cdot \alpha_k)$, **H**, **O**

Shrink2 mutation denotes a mutation operator that concatenates quantum gates. The rotation angle is specified by the integer $k \in \{0, 1, \ldots, 127\}$ via $\alpha_k = -\pi + (k+1) \cdot 2\pi/128$.

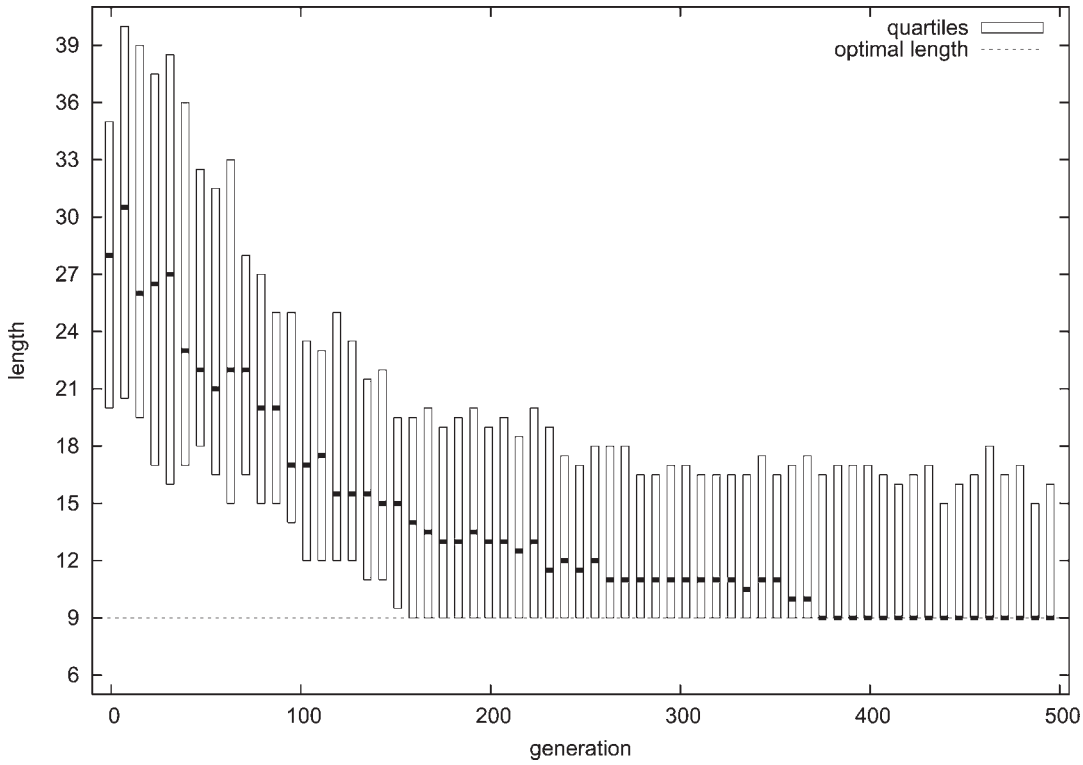

Fig. 3. This graph shows the length distribution (first, second, and third quartiles) of the best individual in each generation for 60 independent runs of the GP system for the $n = 3$ qubit parity problem. The optimal algorithm is presented in the lower part of Fig. 4. A horizontal bar denotes the median (second quartile). The lower, upper end of each box indicates the first, third quartile, respectively.

for the special case of the hidden subgroup problem ($n = 3$ qubits) when the population contains 2500 individuals (running time = ~7 min). For $n = 4$ qubits, the optimal quantum circuit was found in 3 of 100 runs (running time = ~9 h).

4.1. A quantum algorithm that solves the parity problem

When the parity problem is formulated as a blackbox problem, the desired property of parity can be written as the

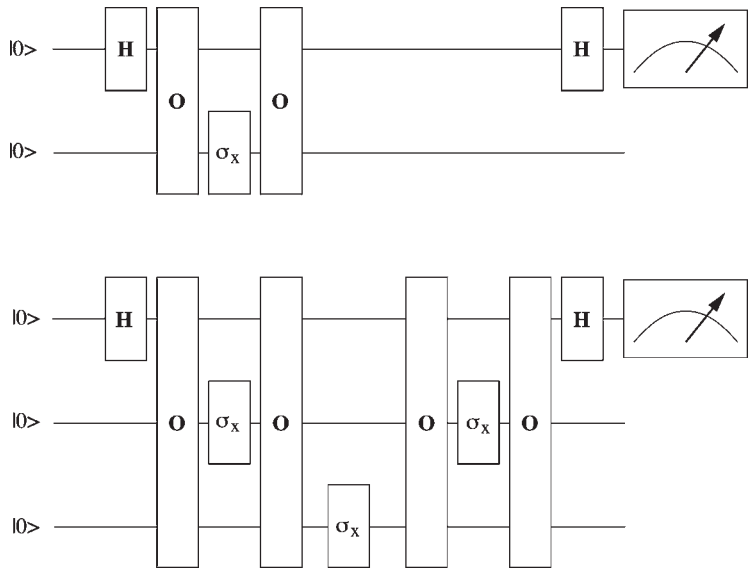

Fig. 4. These two exact parity circuits for $n = 2$ qubits (top) and $n = 3$ (bottom) qubits found by the GP system were sufficient to extract the general scaling mechanism for an arbitrary number of qubits (see Stadelhofer et al., 2005).

Boolean function:

$$p(X) = x_0 \oplus x_1 \cdots \oplus x_{N-1}. \tag{9}$$

Here, $\oplus$ denotes the XOR operation (addition Modulo 2). A classical computer has to call the oracle with each of the N possible inputs i to determine $p(X)$, whereas Beals et al. (2001) and Farhi et al. (1998) showed that a QC requires only $N/2$ oracle calls.

We found an optimal QA in the sense of Beals et al. (2001) and Farhi et al. (1998). This algorithm can be applied to pure and mixed states. Variants also found by the GP system could be optimized for application to ensemble QCs like liquid-state NMR-QCs such that the number of oracle calls decreases exponentially relative to a single-issue QC provided the signal/noise ratio is sufficiently high (Stadelhofer et al., 2005). The gates used by the QA are Hadamard gate **H**, **NOT** gate $\sigma_x = {}_1\mathbf{R}_x(\pi)$ and oracle gate **O**. Queried by the basis state $|i\rangle$ the oracle **O** defined in Eq. (5) returns the binary value $x_i \in X$ encoded into a phase shift of the querying state.

By means of the circuit that solves the parity problem for $n = 2$ query qubits in the upper part of Figure 4 we demonstrate the functionality of our parity algorithm. With two query qubits four different blackbox elements can be addressed; therefore, one only has to check the circuit for the 16 different blackboxes $X = (x_0, x_1, x_2, x_3)$ with $x_i \in \{0, 1\}$. Because we use oracle gates that encode blackbox elements into phase shifts it is sufficient to check the circuit for eight different blackboxes (see also Section 3.2).

The computation starts with the initial state $|\psi_{\text{init}}\rangle = |00\rangle$ that is transformed into the superposition $|\psi\rangle = 1/\sqrt{2}\ (|00\rangle + |01\rangle)$ by the first Hadamard gate **H**. Thus, the oracle call queries two blackbox elements in parallel. To query the remaining two elements a **NOT** operation is applied to the remaining qubit, then the oracle is called again. After a final Hadamard operation the final state is given by

$$|\psi_{\text{final}}\rangle = \frac{(-1)^{x_0 \oplus x_2} + (-1)^{x_1 \oplus x_3}}{2}|10\rangle + \frac{(-1)^{x_0 \oplus x_2} - (-1)^{x_1 \oplus x_3}}{2}|11\rangle, \tag{10}$$

which is equivalent to

$$|\psi_{\text{final}}\rangle = (-1)^{x_0 \oplus x_2}\left(\frac{1 + (-1)^{p(X)}}{2}|10\rangle + \frac{1 - (-1)^{p(X)}}{2}|11\rangle\right). \tag{11}$$

Thus, one finds the rightmost qubit to be in state $|0\rangle$ if $p(X) = 0$, and one measures this qubit in state $|1\rangle$ if $p(X) = 1$. It follows that the parity of the four elements of the blackbox can be computed by only two oracle calls because, due to the superposition over two states, each single call returns two blackbox elements.

As shown in Stadelhofer et al. (2005), this circuit can be scaled up to an arbitrary number of qubits.

With our algorithm, parity is obtained by a single one-qubit measurement compared to the 2^{n-1} measurements necessary by the parity algorithm proposed by Beals et al. (2001) or to the n measurements used by the parity algorithm in Farhi et al. (1998). The latter parity algorithm is somewhat similar to ours though. Instead of a single **NOT** operation between every two oracle calls an n-qubit gate is used, which itself has to be decomposed into elementary quantum gates. Therefore, our algorithm is more efficient with respect to the number of necessary gate operations other than oracle calls.

We also used our GP system to search QAs for ensemble QCs (liquid-state NMR-QCs) and, indeed, were able to find the quantum circuits presented in Figure 5 that need fewer oracle calls than the circuits for single issue QCs presented above.

These ensemble quantum circuits appear to violate the lower bounds proven in (Farhi et al., 1998; Beals et al., 2001). However, as illustrated in Appendix A, these proofs seem not to apply to ensemble QCs.

4.2. A quantum algorithm that solves a special case of the hidden subgroup problem

The first instance of the problem we consider in this section is to distinguish between blackboxes $X \in A$ with $A = \{(a, a, a, a)\}$ and blackboxes $X \in B$ with

$$B = \{(a, a, b, b), (a, b, a, b), (a, b, b, a)\},$$

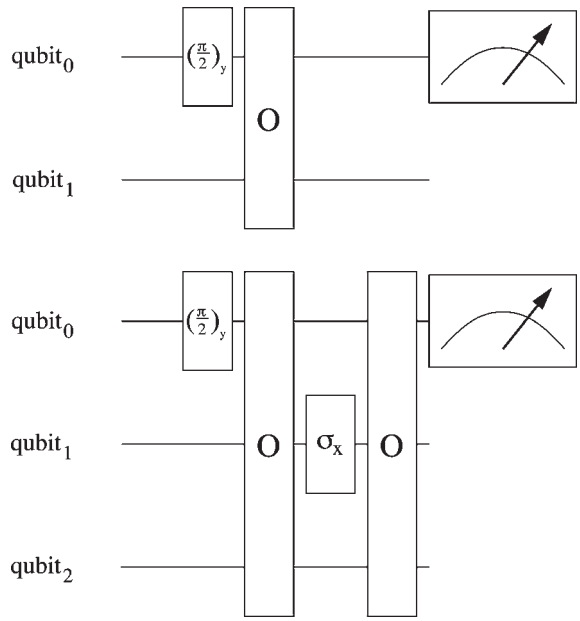

Fig. 5. Mixed-state parity circuits found by the GP system for $n = 2$ (top) and $n = 3$ (bottom) qubits. The $(\pi/2)_y$-gate corresponds to the rotation $\mathbf{R}_y(\pi/2)$. The measurement-gate symbolizes a measurement of the magnetization along the x axis. In both cases one measures $\langle \mathbf{I}_x^{(0)} \rangle = 0$ if $p(X) = 1$ and $\langle \mathbf{I}_x^{(0)} \rangle = \pm\omega_0/4$ if $p(X) = 0$.

where $a \neq b$ and $a, b \in \{0, 1, 2, 3\}$. To solve this problem we first tried to encode the blackbox entries entirely into phase shifts via the oracle gate $\mathbf{O}|k\rangle = (\imath)^{x_k}|k\rangle$. This approach did not return satisfactory results. Therefore, we tried the combined approach defined in Eq. (6). As a blackbox call can return four different values $x_k \in \{0, 1, 2, 3\}$ for $k \in \{0, 1, 2, 3\}$ there is only the following possibility to use a combined encoding into a phase shift and an output register:

$$\mathbf{O}|k\rangle|b\rangle = (-1)^{x_k^{(\text{phase})}}|k\rangle|b \oplus x_k^{(\text{XOR})}\rangle \tag{12}$$

with $b, x_k^{(\text{phase})}, x_k^{(\text{XOR})} \in \{0, 1\}$ and $k \in \{0, 1, 2, 3\}$. Here, the blackbox entries are decomposed into the two binary values $x_k^{(\text{phase})}$ and $x_k^{(\text{XOR})}$. Consider, for example, the blackbox $X = (0, 3, 0, 3)$: decomposing the blackbox entries into two binary values returns the alternative representation $X' = (00, 11, 00, 11)$. One gets

$$x_0^{(\text{phase})}x_0^{(\text{XOR})} = 00, \quad x^{(\text{phase})}x_1^{(\text{XOR})} = 11,$$
$$x_2^{(\text{phase})}x_2^{(\text{XOR})} = 00, \quad x_3^{(\text{phase})}x_3^{(\text{XOR})} = 11.$$

The oracle gate $\mathbf{O}_{(0,3,0,3)}$, therefore, looks like

$$\mathbf{O}_{(0,3,0,3)} = \begin{pmatrix} 1 & 0 & & & & & & \\ 0 & 1 & & & & & & \\ & & 0 & -1 & & & & \\ & & -1 & 0 & & & & \\ & & & & 1 & 0 & & \\ & & & & 0 & 1 & & \\ & & & & & & 0 & -1 \\ & & & & & & -1 & 0 \end{pmatrix}$$

The weakness of this approach is that it requires an additional output qubit. Nevertheless, this still is an advantage over the two additional qubits necessary for a traditional encoding of the blackbox entries $x_k \in \{0, 1, 2, 3\}$ into an output register. In addition, it is still possible to reduce the number of blackboxes that are to be tested: if one uses the oracle defined in Eq. (12), the blackbox $X_1 = (0, 0, 1, 1)$ and the blackbox $X_{31} = (2, 2, 3, 3)$ have the same oracle gate up to a global phase shift. Therefore, it is sufficient to only test X_1. It follows that one only has to test two instead of the four blackboxes $X \in A$ and 18 instead of the 36 blackboxes $X \in B$. Thus, the evaluation of a quantum circuit by the GP system is reduced by a factor of 2×2 in comparison to the traditional encoding. The quantum circuit returned by the GP system is shown in Figure 6.

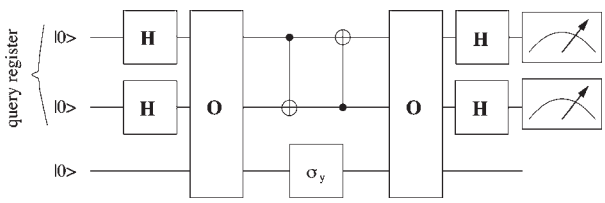

Fig. 6. This circuit distinguishes between blackboxes $X \in A$ and blackboxes $X \in B$ by mapping the initial state $|\psi_{init}\rangle = |00\rangle$ of the query register to the final state $|\psi_{fin}\rangle = |00\rangle$ if $X \in A$, otherwise, the state $|00\rangle$ of the query register has a probability amplitude of zero. The gate σ_y is realized by $\mathbf{R}_y(\pi)$: $\sigma_y = \imath\mathbf{R}_y(\pi)$.

To see if this quantum circuit can be scaled up to an arbitrary number of qubits we tried to find a related problem for blackboxes with eight elements (three query qubits). After several attempts we found a problem for which the GP system returned the quantum circuit shown in Figure 7. The problem solved by this circuit is to distinguish between blackboxes $X \in A$ with $A = (a, a, a, a, a, a, a, a)$ and blackboxes $X \in B$ with

$$B = \left\{ \begin{array}{ll} (a, a, a, a, b, b, b, b), & (a, b, a, b, a, b, a, b), \\ (a, a, b, b, a, a, b, b), & (a, a, b, b, b, b, a, a), \\ (a, b, a, b, b, a, b, a), & (a, b, b, a, a, b, b, a), \\ (a, b, b, a, b, a, a, b) & \end{array} \right\},$$

where $a \neq b$ and $a, b \in \{0, 1, 2, 3\}$.

The two instances of the problem presented above were sufficient to find the general structure of the problem that can be stated as the following special case of the hidden subgroup problem ($\underline{i}$ denotes the binary decomposition of the integer i).

One is given the finite Abelian group $G = \{0, 1\}^n$ of the size $|G| = N = 2^n$ and a subgroup $H \subset G$. The group multiplication operation is the bitwise XOR operation ($\oplus$). In addition, one is given a blackbox $X = (x_0, x_1, \ldots, x_{N-1})$ with $x_i \in \{0, 1, 2, 3\}$ and $\underline{i} \in G$. One is promised that either $X \in A$

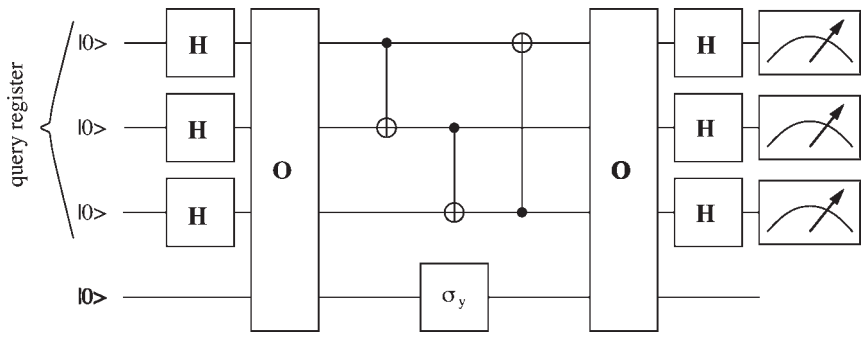

Fig. 7. This circuit distinguishes between blackboxes $X \in A$ and blackboxes $X \in B$ by mapping the state $|\psi_{init}\rangle = |000\rangle$ of the query register to the final state $|\psi_{fin}\rangle = |000\rangle$ if $X \in A$, otherwise, the state $|000\rangle$ of the query register has a probability amplitude of zero.

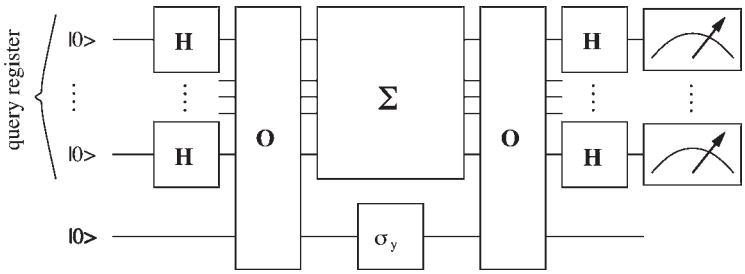

Fig. 8. Generalized circuit on $(n+1)$ qubits. The upper n qubits encode $i \in \{0, 1, \ldots, 2^n - 1\}$; Σ abbreviates the gate sequence $\mathbf{CNOT}_{0,1} - \mathbf{CNOT}_{1,2} - \cdots - \mathrm{CNOT}_{n-2,n-1} - \mathbf{CNOT}_{n-1,0}$. $\mathrm{CNOT}_{l,m}$ denotes a **CNOT** gate where qubit l is the control qubit and qubit m the target qubit. This circuit distinguishes between blackboxes $X \in A$ and blackboxes $X \in B$ by mapping the initial n-qubit state $|\psi_{init}\rangle = |0 \cdots 0\rangle$ of the query register to the final state $|\psi_{fin}\rangle = |0 \cdots 0\rangle$ if $X \in A$, otherwise, the state $|0 \cdots 0\rangle$ of the n-qubit query register has a probability amplitude of zero.

or $X \in B$, where $X \in A$ denotes that only for $H = G$ there exists an element $g \in G$ such that $x_i = x_j \Leftrightarrow \underline{i}, \underline{j} \in g \oplus H$, and $X \in B$ denotes that only for $H \subset G$ with $|H| = |G|/2$ there exists an element $g \in G$ with $x_i = x_j \Leftrightarrow \underline{i}, \underline{j} \in g \oplus H$. Here, we used the notation $g \oplus H$ to denote the coset of H.[7] In addition to the usual definition of the hidden subgroup problem we required that $x_i \in \{0, 1, 2, 3\}$ and that the subgroups H to be distinguished contain N or $N/2$ elements, respectively. In the hidden subgroup problem (see Nielsen & Chuang, 2000) the group members need not be binary strings, and thus the group multiplication operation can be of a different type then the one used here. There, one is interested in the generating set of a subgroup $H \in G$, where $|H|$ can be of any size. Knowing this generating set makes it possible to distinguish between subgroups H. Therefore, any algorithm solving the hidden subgroup problem solves the special case presented here as well.

Note that the problem this algorithm solves is not to distinguish between constant and balanced blackboxes, as in general not every balanced blackbox belongs to set B.

The general structure of quantum circuits that solve our special case of the hidden subgroup problem is shown in Figure 8.

This QA solves this problem exactly with two oracle calls only. An exact classical algorithm that solves this problem needs up to $n + 1$ blackbox calls (here: $|G| = 2^n$), where n denotes the number of query qubits.[8] A probabilistic classical algorithm can solve this problem using $k > 1$ calls with an error probability of 2^{-k+1} (for $|G| \gg 1$).

As shown in Brassard and Høyer (1998), Simon's QA can be generalized to a broader class of problems called Simon's hidden subgroup problems that are a subset of hidden subgroup problems. This generalized algorithm is probabilistic, and can also be used to solve the problem presented here. The error probability solving the problem decreases with 2^{-k+1} in the number k of repetitions of this algorithm. Because this generalized algorithm calls the oracle only once, k repetitions of the algorithm correspond to k oracle calls. In Brassard and Høyer (1998) the authors also present an exact algorithm capable of solving Simon's hidden subgroup problem, which in our case has to call the oracle three times.

It follows that the QA found with the help of the GP system is faster than any classical algorithm as well as the QAs known to us.

5. CONCLUSION

The successful development of two formerly unknown QAs presented in Section 4 demonstrates the usefulness of evolutionary methods for generating new QAs. Despite the fact that the fitness functions used did not fulfill the requirements we discussed in the beginning of Section 3.3 in every aspect, the results obtained are very promising.

We restricted our investigations to the blackbox model of computation where the Boolean property $p(X)$ of a blackbox X is to be computed with a minimal number of blackbox calls. Our GP system only considers quantum decision trees that, after a sequence of oracles and quantum gates, reveal the property of the blackbox by a final measurement (see Section 3.1). This is similar to the approach used by Beals et al. (2001) in proving lower bounds for the parity problem. Surprisingly, the GP system returned quantum circuits that beat this lower bound on ensemble QCs. The error probability of these quantum circuits on a single issue QC is 1/2 (see Appendix A); nevertheless, running such a circuit several times on the same initial state makes it possible to solve the parity problem with an error probability that decreases exponentially in the number of repetitions.

Indeed, QAs like Simon's algorithm also use several independent runs of a QA to collect measurement results for a further classical treatment. At the current stage our GP system would not be able to evolve such hybrid algorithms for single-issue QCs. To our knowledge, such hybrid algorithms

[7] Let $H \subset G$ denote a subgroup of G. To each $g \in G$ one can define a set $g \oplus H = \{g \oplus h: h \in H\}$, which is called a coset of H.

[8] The definition of the problem shows that a deterministic classical algorithm has to choose a subgroup $H \in G$ with $|H| = |G|/2$. Then the algorithm has to calculate a coset $g \oplus \langle H \rangle$ of the generators $\langle H \rangle$ of H. The elements $k \in g \oplus \langle H \rangle$ of this coset plus the neutral element 0 are used to query the blackbox for the elements x_k. If all elements are equal, the blackbox can still belong to set B; thus, an additional element of a different coset has to be tested. If this query also returns the same answer the blackbox belongs to set A, otherwise to B. With $|G| = 2^n$ one has $k = \log_2(|H|) = \log_2(2^{n-1}) = n - 1$ generators, one thus has to call the blackbox $(n + 1)$ times in the worst case.

were not investigated by other authors (Spector et al., 1999; Leier & Banzhaf, 2003*a*; Spector, 2004) who use GP to design QAs.

ACKNOWLEDGMENTS

The authors thank Dr. André Leier for helpful discussions. Financial assistance was provided by the Deutsche Forschungsgesellschaft through Graduiertenkolleg 726.

REFERENCES

Beals, R., Buhrman, H., Cleve, R., Mosca, M., & de Wolf, R. (2001). Quantum lower bounds by polynomials. *Journal of the Association for Computing Machinery 48*, 778.

Bennett, C.H., Bernstein, E., Brassard, G., & Vazirani, U. (1997). Strengths and weaknesses of quantum computing. *SIAM Journal on Computing 26*(5), 1510–1523.

Bernstein, E., & Vazirani, U. (1997). Quantum complexity theory. *SIAM Journal on Computing 26*, 1411.

Brassard, G., & Høyer, P. (1997). An exact quantum polynomial-time algorithm for Simon's problem. *Proc. Fifth Israeli Symp. Theory of Computing and Systems (ISTCS)*, pp. 12–23.

Burhman, H., & de Wolf, R. (2002). Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science 288*, 21.

Collins, D., Kim, K.W., & Holton, W.C. (1998). Deutsch–Jozsa algorithm as a test of quantum computation. *Physical Review A 58*, R1633.

Deutsch, D., & Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London A 439*, 553–558.

Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (1998). Limit on the speed of quantum computation in determining parity. *Physical Review Letters 81*, 5442.

Leier, A., & Banzhaf, W. (2003a). Exploring the search space of quantum programs. *Proc. 2003 Congr. Evolutionary Computation*, Vol. I, pp. 170–177.

Leier, A., & Banzhaf, W. (2003b). Evolving Hogg's quantum algorithm using linear-tree GP. *Proc. Genetic and Evolutionary Computation Conf. GECCO-03*, pp. 390–400.

Lloyd, S. (2000). Quantum search without entanglement. *Physical Review A*, *61*, 010301.

Massey, P., Clark, J., & Stepney, S. (2004). Evolving quantum circuits and programs through genetic programming. *Proc. Genetic and Evolutionary Computation Conf. GECCO-2004.*

Nielsen, M., & Chuang, I. (2000). *Quantum Computation and Quantum Information.* New York: Cambridge University Press.

Rieffel, E., & Polak, W. (2000). An introduction to quantum computing for non-physicists. *ACM Computing Surveys 32*, 300.

Shor, P.W. (1994). Algorithms for quantum computation: discrete logarithm and factoring. *IEEE Symp. Foundations of Computer Science*, pp. 124–134.

Simon, D.R. (1994). On the power of quantum computation. *Proc. 35th Annual Symp. Foundations of Computer Science*, pp. 116–123.

Spector, L. (2004). *Automatic Quantum Computer Programming: A Genetic Programming Approach.* New York: Kluwer Academic.

Spector, L., Barnum, H., Bernstein, H.J., & Swamy, N. (1999). Quantum computing applications of genetic programming. In *Advances in Genetic Programming* (Spector, L., Langdon, W.B., O'Reilly, U., & Angeline, P.J., Eds.), Vol. 3, p. 135. Cambridge, MA: MIT Press.

Stadelhofer, R., Suter, D., & Banzhaf, W. (2005). Quantum and classical parallelism in parity algorithms for ensemble quantum computers. *Physical Review A 71*, 032345.

Williams, C.P., & Gray, A.G. (1998). Automated design of quantum circuits. *Proc. First NASA Int. Conf. Quantum Computing and Quantum Communications (QCQC)*, pp. 113–125.

Ralf Stadelhofer studied physics from 1995 to 2001 at the University of Konstanz. From 2001 to 2004 he was a scholarship holder within the Materials and Concepts of Quantum Information Processing PhD program at the University of Dormund, Germany. Since then he has worked as the Chair of Bioinformatics at the University of Konstanz. Dr. Stadelhofer's interests focus mainly on inverse problems like the protein folding problem and the phase problem in X-ray crystallography.

Wolfgang Banzhaf was educated as a theoretical physicist and is now a Professor of computer science and Head of the Department of Computer Science at Memorial University of Newfoundland. In the last 20 years he has published more than 150 technical contributions, was lead author of a textbook on genetic programming, edited conference proceedings on topics of evolutionary computation and artificial life, and served as Editor in Chief of *Genetic Programming and Evolvable Machines*. His interests span a wide range of technical topics, from basic physics and computing to biology.

Dieter Suter works in experimental condensed matter physics in the Department of Physics at the Technical University of Dortmund. He obtained a PhD in physical chemistry from ETH Zürich in 1985, working on nuclear magnetic resonance (NMR). His postdoctoral work at Berkeley and ETH Zürich was in the fields of magnetic resonance and quantum optics. In 1995 he joined the faculty of the Technical University of Dortmund. In addition to quantum information processing, Dr. Suter is developing spectroscopic techniques in the fields of magnetic resonance and laser spectroscopy.

APPENDIX A: A PROBABILISTIC PARITY QUANTUM ALGORITHM

Usually (see, e.g., Farhi et al., 1998; Beals et al., 2001) one only considers probabilistic algorithms that have an error probability of $1/2 - \varepsilon$ with $0 < \varepsilon \leq 1/2$ because their error probability can be reduced further by running them several times (Nielsen & Chuang, 2000). Here we exemplify how a quantum decision tree with an error rate of 1/2 can be used to solve the parity problem. To do so we investigate the behavior of a modified instance of the ensemble quantum circuit shown in the upper part of Figure A.1 on a single issue

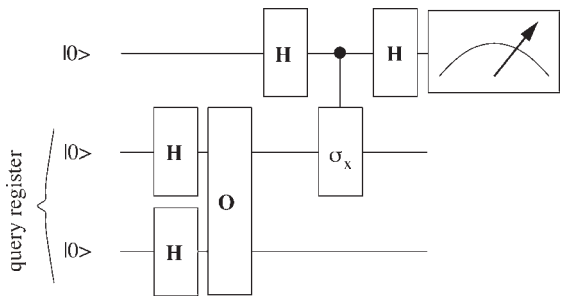

Fig. A.1. Modified instance of the ensemble quantum circuit, shown in the upper part of Fig. 5, for a single issue QC. Measurement of the magnetization is replaced by a measurement in the computational basis. As shown in [2, 17] any exact or probabilistic QA would need $N/2$ oracle gates with $N = 2^n$ where n denotes the number of query qubits. In our example this corresponds to 2 oracle gates instead of the single oracle gate used by our circuit.

QC. For convenience, we added an additional qubit, a controlled gate and two Hadamard gates to this instance, which makes it possible to replace the measurement of the magnetization by a projective measurement in the computational basis, as usual for single issue QCs (see Fig. A.1). The original ensemble circuit returns $\langle \mathbf{I}_x^{(0)} \rangle = 0$ for blackboxes X with $p(X) = 1$ and $\langle \mathbf{I}_x^{(0)} \rangle = \pm\, \omega_0/4$ if $p(X) = 0$. For the single-issue QC this corresponds to the additional qubit's final state $|\psi\rangle = |0\rangle$ or $|\psi\rangle = |1\rangle$ if $p(X) = 0$. If $p(X) = 1$, the additional qubit's final state is described by the superposition $|\psi\rangle = 1/\sqrt{2}(|0\rangle \pm |1\rangle)$. It follows that for $p(X) = 1$ a measurement will return the state $|0\rangle$ or $|1\rangle$ with equal probability, whereas for $p(X) = 0$, only one of these two states will be measured with a nonzero probability. Thus, a single run of the quantum circuit on a single issue QC does not reveal any useful information. Nevertheless, if several runs return different measurement results one knows that $p(X) = 1$. If, in contrast, $l \in \mathbb{N}$ runs return equal results, one has an error probability of 2^{-l+1} in claiming that $p(X) = 0$. Thus, the quantum circuit in Figure A.1 provides a useful probabilistic quantum circuit despite the fact that a single run does no better than guessing.