Quantum Algorithms for Solvable Groups

John Watrous*
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
jwatrous@cpsc.ucalgary.ca

ABSTRACT

In this paper we give a polynomial-time quantum algorithm for computing orders of solvable groups. Several other problems, such as testing membership in solvable groups, testing equality of subgroups in a given solvable group, and testing normality of a subgroup in a given solvable group, reduce to computing orders of solvable groups and therefore admit polynomial-time quantum algorithms as well. Our algorithm works in the setting of black-box groups, wherein none of these problems have polynomial-time classical algorithms. As an important byproduct, our algorithm is able to produce a pure quantum state that is uniform over the elements in any chosen subgroup of a solvable group, which yields a natural way to apply existing quantum algorithms to factor groups of solvable groups.

1. INTRODUCTION

The focus of this paper is on quantum algorithms for group-theoretic problems. Specifically we consider finite solvable groups, and give a polynomial-time quantum algorithm for computing orders of solvable groups. Naturally this algorithm yields polynomial-time quantum algorithms for testing membership in solvable groups and several other related problems that reduce to computing orders of solvable groups. Our algorithm is also able to produce a uniform pure quantum state over the elements in any chosen subgroup of a solvable groups, which yields a natural way of applying certain quantum algorithms to factor groups of solvable groups. For instance, we describe a method by which existing quantum algorithms for abelian groups may be applied to abelian factor groups of solvable groups, despite the fact that the factor groups generally do not satisfy an important requirement of the existing quantum algorithms—namely, that elements have unique, succinct classical representations.

We will be working within the context of *black-box groups*, wherein elements are uniquely encoded by strings of some given length n and the group operations are performed by a black-box (or *group oracle*) at unit cost. Black-box groups were introduced by Babai and Szemerédi [7] in 1984 and have since been studied extensively [1, 2, 3, 4, 5, 6, 9]. Any efficient algorithm that works in the context of black-box groups of course remains efficient whenever the group oracle can be replaced by an efficient procedure for computing the group operations. In the black-box group setting it is provably impossible to compute order classically in polynomial time, even for abelian groups [7].

Essentially all previously identified problems for which quantum algorithms offer exponential speed-up over the best known classical algorithms can be stated as problems regarding abelian groups. In 1994, Shor [36] presented polynomial time quantum algorithms for integer factoring and computing discrete logarithms, and these algorithms generalize in a natural way to the setting of finite groups. Specifically, given elements g and h in some finite group G it is possible, in quantum polynomial time, to find the smallest positive integer k such that $h = g^k$, provided there exists such a k. In case h is the identity one obtains the order of g, to which there is a randomized polynomial-time reduction from factoring when the group is the multiplicative group of integers modulo the integer n to be factored. It should be noted that while the group G need not necessarily be abelian for these algorithms to work, we may view the algorithms as taking place in the abelian group generated by g.

Shor's algorithms for integer factoring and discrete logarithms were subsequently cast in a different group-theoretic framework by Kitaev [29, 30]. This framework involves a problem called the Abelian Stabilizer Problem, which may be informally stated as follows. Let k and n be positive integers, and consider some group action of the additive abelian group $\mathbb{Z}^k$ on a set $X \subseteq \Sigma^n$, where the group action can be computed efficiently. The problem, which can be solved in quantum polynomial time, is to compute a basis (in $\mathbb{Z}^k$) of the stabilizer $(\mathbb{Z}^k)_x$ of a given $x \in X$. Appropriate choice of the group action allows one to solve order finding and discrete logarithms for any finite group as above. In this case, the group G in question corresponds to the set X and the group action of $\mathbb{Z}^k$ on X is determined by the group structure of G.

Kitaev's approach was further generalized by Brassard and Høyer [12], Jozsa [27], and Mosca and Ekert [33], who formulated the Hidden Subgroup Problem. (See also Høyer [24].) The Hidden Subgroup Problem may be informally stated as follows. Given a finitely generated group G and an efficiently computable function f from G to some finite

*Research partially supported by Canada's NSERC.

STOC'01, July 6-8, 2001, Hersonissos, Crete, Greece.

set X such that f is constant and distinct on left-cosets of a subgroup H of finite index, find a generating set for H. Mosca and Ekert showed that Deutsch's Problem [16], Simon's Problem [37], order finding and computing discrete logarithms [36], finding hidden linear functions [11], testing self-shift-equivalence of polynomials [21], and the Abelian Stabilizer Problem [29, 30] can all be solved in polynomial time within the framework of the Hidden Subgroup Problem. In the black-box group setting, the Hidden Subgroup Problem can be solved in quantum polynomial time whenever G is abelian, as demonstrated by Mosca [32]. Mosca also proved that several other group-theoretic problems regarding abelian black-box groups can be reduced to the Hidden Subgroup Problem, and thus can be computed in quantum polynomial time as well. For instance, given a collection of generators for a finite abelian black-box group, one can find the order of the group, and in fact one can decompose the group into a direct product of cyclic subgroups of prime power order, in polynomial time.[1] (See also Cheung and Mosca [13] for further details.)

The Hidden Subgroup Problem has been considered in the non-abelian case, although with limited success (see Ettinger and Høyer [17], Ettinger, Høyer, and Knill [18], Grigni, Schulman, Vazirani, and Vazirani [20], Hallgren, Russell, and Ta-Shma [23], Ivanyos, Magniez, and Santha [26], and Rötteler and Beth [35]). Quantum polynomial-time algorithms for finding non-abelian hidden subgroups are known for only limited classes of finite groups—most notably, the recent paper of Ivanyos, Magniez, and Santha [26] gives quantum polynomial-time algorithms based on an algorithm of Beals and Babai [9] for solving certain special cases of this problem. The Non-abelian Hidden Subgroup Problem is of particular interest as it relates to the Graph Isomorphism Problem; Graph Isomorphism reduces to a special case of the Hidden Subgroup Problem in which the groups in question are the symmetric groups. Beals [8] has shown that quantum analogues of Fourier transforms over symmetric groups can be performed in polynomial time, although thus far this has not proven to be helpful for solving the Graph Isomorphism Problem.

In this paper we move away from the Hidden Subgroup Problem and consider other group-theoretic problems for non-abelian groups—in particular we consider solvable groups. Our main algorithm finds the order of a given solvable group and, as an important byproduct, produces a quantum state that approximates a uniform superposition over the elements of the given group.

Theorem 1. *There exists a quantum algorithm operating as follows (relative to an arbitrary group oracle). Given generators $g_1, \ldots, g_m$ such that $G = \langle g_1, \ldots, g_m \rangle$ is solvable, the algorithm outputs the order of G with probability of error bounded by ε in time polynomial in $mn + \log(1/\varepsilon)$ (where n is the length of the strings representing the generators). Moreover, the algorithm produces a quantum state ρ that approximates the state $|G\rangle = |G|^{-1/2} \sum_{g \in G} |g\rangle$ with accuracy ε (in the trace norm metric).*

[1] This is interesting from the standpoint of computational number theory since, assuming the Generalized Riemann Hypothesis, it follows that there is a polynomial-time quantum algorithm for computing class numbers of imaginary quadratic number fields—a problem often considered as a candidate for a problem harder than integer factoring. See Cohen [15] for information about computing class numbers.

Several other problems reduce to the problem of computing orders of solvable groups, including membership testing in solvable groups, testing equality of subgroups in a given solvable group, and testing that a given subgroup of some solvable group is normal. Thus, these problems can be solved in quantum polynomial time as well.

Since any subgroup of a solvable group is solvable, our algorithm can be applied to any subgroup H of a solvable group G in order to obtain a close approximation to the state $|H\rangle$. One application of being able to efficiently prepare uniform superpositions over subgroups of solvable groups is that it gives us a simple way to apply existing quantum algorithms for abelian groups to abelian factor groups of solvable groups, despite the fact that we do not have unique classical representations for elements in these factor groups. This method is discussed further in Section 4.

While black-box group algorithms are appealing because of their generality, it is natural to ask if the algorithms described in this paper give polynomial-time algorithms for any problems in the standard (no oracles) model that are not known to have polynomial-time classical algorithms. Example of such problems can be obtained by considering groups of invertible matrices over finite fields. For instance, let $\mathbb{F}$ be a finite field of characteristic p, and consider the problem of finding the order of $G = \langle g_1, \ldots, g_k \rangle$ for given elements $g_1, \ldots, g_k \in \mathrm{GL}(n, \mathbb{F})$ under the assumption that G is solvable. The most efficient classical algorithm known for this problem is due to Luks [31], and runs in time polynomial in the input size plus the largest prime other than p dividing $|G|$. Our quantum algorithm solves this problem in polynomial time without dependence on the primes dividing $|G|$. Moreover, our algorithm gives polynomial-time solutions to related problems regarding factor groups of such solvable matrix groups as described in Section 4.

Arvind and Vinodchandran [1] have shown that several problems regarding solvable groups, including membership testing and order verification, are low for the complexity class PP, which means that an oracle for these problems is useless for PP computations. Fortnow and Rogers [19] proved that any problem in BQP is low for PP, and thus we have obtained an alternate proof that membership testing and order verification for solvable groups are both low for PP. It is left open whether some of the other problems proved low for PP by Arvind and Vinodchandran have polynomial-time quantum algorithms. An interesting example of such a problem is testing whether two solvable groups have a nontrivial intersection.

Very recently, after a preliminary version of the present paper appeared, Ivanyos, Magniez, and Santha [26] showed how an algorithm of Beals and Babai [9] can be combined with a quantum algorithm for the (Abelian) Hidden Subgroup Problem to obtain polynomial-time quantum algorithms for several group theoretic problems, including computing orders of solvable groups as well as several problems not considered in the present paper. In addition, they combine their method with our technique for preparing uniform superpositions over solvable subgroups to obtain polynomial-time quantum algorithms for other problems regarding factor groups. Unlike the approach of Ivanyos, Magniez, and Santha, our algorithm is self-contained and completely elementary (not relying on a statistical analysis of the classification of finite simple groups, in particular).

The remainder of this paper has the following organization. In Section 2 we review necessary background information for this paper, including a discussion of black-box groups in the context of quantum circuits and other information regarding computational group theory. Section 3 describes our quantum algorithm for finding the order of a solvable group as stated in Theorem 1, and Section 4 discusses other problems that can be solved by adapting this algorithm. We conclude with Section 5, which mentions some open problems.

2. PRELIMINARIES

In this section we review information regarding computational group theory that is required for the remainder of the paper. We assume the reader is familiar with the theory of quantum computation, and specifically with the quantum circuit model, so we will not review this model further except to discuss black-box groups in the context of quantum circuits. The reader not familiar with quantum circuits is referred to Nielsen and Chuang [34]. We also assume the reader is familiar with the basic concepts of group theory (see, for example, Isaacs [25]).

Given a group G and elements $g, h \in G$ we define the *commutator* of g and h, denoted $[g, h]$, as $[g, h] = g^{-1}h^{-1}gh$, and for any two subgroups $H, K \leq G$ we write $[H, K]$ to denote the subgroup of G generated by all commutators $[h, k]$ with $h \in H$ and $k \in K$. The *derived subgroup* of G is $G' = [G, G]$, and we write

$$G^{(0)} = G,$$
$$G^{(j)} = (G^{(j-1)})', \quad \text{for } j \geq 1.$$

A group G is said to be *solvable* if $G^{(m)} = \{1\}$ (the group consisting of just one element) for some value of m. An equivalent way to define what it means for a (finite) group to be solvable is as follows. A finite group G is solvable if there exist elements $g_1, \ldots, g_m \in G$ such that if we define $H_j = \langle g_1, \ldots, g_j \rangle$ for each j, then

$$\{1\} = H_0 \triangleleft H_1 \triangleleft \cdots \triangleleft H_m = G.$$

Note that H_{j+1}/H_j is necessarily cyclic in this case for each j. Given an arbitrary collection of generators for a solvable group G, a polynomial-length sequence $g_1, \ldots, g_m$ as above can be found via a (classical) Monte Carlo algorithm in polynomial time [6] (discussed in more detail below). It is important to note that we allow the possibility that $H_j = H_{j+1}$ for some values of j in reference to this claim.

We will be working in the general context of black-box groups, which we now discuss. In a black-box group, each elements is uniquely encoded by some binary string, and we have at our disposal a black-box (or group oracle) that performs the group operations on these encodings at unit cost. For a given black-box group, all of the encodings are of a fixed length n, which is the *encoding length*. Thus, a black-box group with encoding length n has order bounded above by 2^n. Note that not every binary string of length n necessarily corresponds to a group element, and we may imagine that our group oracle has some arbitrary behavior given invalid encodings. (Our algorithms will never query the oracle for invalid group element encodings given valid input elements). When we say that a particular group or subgroup is given (to some algorithm), we mean that a set of strings that generate the group or subgroup is given.

Since we will be working with quantum circuits, we must describe black-box groups in this setting. Corresponding to a given black-box group G with encoding length n is a quantum gate U_G acting on $2n$ qubits as follows:

$$U_G|g\rangle|h\rangle = |g\rangle|gh\rangle.$$

Here we assume g and h are valid group elements—in case any invalid encoding is given, U_G may act in any arbitrary way so long as is remains reversible. The inverse of U_G acts as follows:

$$U_G^{-1}|g\rangle|h\rangle = |g\rangle|g^{-1}h\rangle.$$

When we say that a quantum circuit has access to a group oracle for G, we mean that the circuit may include the gates U_G and U_G^{-1} for some U_G as just described. More generally, when we are discussing uniformly generated families of quantum circuits, a group oracle corresponds to an infinite sequence of black-box groups $G_1, G_2, \ldots$ (one for each encoding length), and we allow each circuit in the uniformly generated family to include gates of the form U_{G_n} and $U_{G_n}^{-1}$ for the appropriate value of n.

As noted by Mosca [32], the gates U_G and U_G^{-1} above can be approximated efficiently if we have a single gate V_G acting as follows on $3n$ qubits:

$$V_G|g\rangle|h\rangle|x\rangle = |g\rangle|h\rangle|x \oplus gh\rangle,$$

again where we assume g and h are valid group elements (and x is arbitrary). Here, $x \oplus gh$ denotes the bitwise XOR of the string x and the string encoding the group element gh. This claim follows from the fact that given the gate V_G, we may find the order of any element g using Shor's algorithm, from which we may find the inverse of g. Once we have this, techniques in reversible computation due to Bennett [10] allow for straightforward simulation of U_G and U_G^{-1}. Since it is simpler to work directly with the gates U_G and U_G^{-1}, however, we will assume that these are the gates made available for a given black-box group.

Now we return to the topic of solvable groups, and review some known facts about solvable groups in the context of black-box groups. First, with respect to any given group oracle, if we are given generators $g_1, \ldots, g_m$ of encoding length n, it is possible to test whether $G = \langle g_1, \ldots, g_m \rangle$ is solvable via a polynomial time (in nm) Monte Carlo algorithm [6]. Moreover, the same algorithm can be used to construct (with high probability) generators $g_1^{(j)}, \ldots, g_k^{(j)}$, for $j = 0, \ldots, n-1$ and where $k = O(n)$, such that $G^{(j)} = \langle g_1^{(j)}, \ldots, g_k^{(j)} \rangle$. At this point we notice (under the assumption that G is solvable) that by relabeling the elements

$$g_1^{(n-1)}, \ldots, g_k^{(n-1)}, g_1^{(n-2)}, \ldots, g_k^{(n-2)}, \ldots, g_1^{(0)}, \ldots, g_k^{(0)},$$

as $h_1, \ldots, h_{kn}$ (in the order given) we have the following. If $H_j = \langle h_1, \ldots, h_j \rangle$ for $j = 0, \ldots, kn$, then

$$\{1\} = H_0 \triangleleft H_1 \triangleleft \cdots \triangleleft H_{kn} = G.$$

This follows from the fact that $G^{(j)} \triangleleft G^{(j-1)}$ for each j, and further that $G^{(j-1)}/G^{(j)}$ is necessarily abelian. The fact that each factor group H_j/H_{j-1} is cyclic will be important for our quantum algorithm in the next section.

The problem of computing the order of a group cannot be solved classically in polynomial time in the black-box setting even for abelian (and therefore for solvable) groups [7].

3. COMPUTING ORDERS OF SOLVABLE GROUPS

In this section we describe our quantum algorithm for finding the order of a given solvable black-box group G and preparing a uniform superposition over the elements of G.

We assume we have elements $g_1, \ldots, g_m \in G$ such that if we define $H_j = \langle g_1, \ldots, g_j \rangle$ for each j, then

$$\{1\} = H_0 \triangleleft H_1 \triangleleft \cdots \triangleleft H_m = G. \tag{1}$$

We allow the possibility that $H_j = H_{j+1}$ for some values of j. The existence of such a chain is equivalent to the solvability of G, and given an arbitrary collection of generators of G such a sequence can be found via a Monte Carlo algorithm in polynomial time as discussed in the previous section. Calculation of the orders of the factor groups in this chain reveals the order of G; if

$$r_1 = |H_1/H_0|, \ \ldots, \ r_m = |H_m/H_{m-1}|,$$

then $|G| = \prod_{j=1}^{m} r_j$.

The calculation of the orders of the factor groups is based on the following idea. Suppose we have several copies of the state $|H\rangle$ for some subgroup H of G, where $|H\rangle$ denotes the uniform superposition over the elements of H:

$$|H\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |h\rangle.$$

Then using a simple modification of Shor's order finding algorithm we may find the *order of g with respect to H*, which is the smallest positive integer r such that $g^r \in H$, for any $g \in G$. In case $H = H_{j-1} = \langle g_1, \ldots, g_{j-1} \rangle$ and $g = g_j$ for some j, this order is precisely $r_j = |H_j/H_{j-1}|$.

Since this requires that we have several copies of $|H_{j-1}\rangle$ in order to compute each r_j, we must demonstrate how the state $|H_{j-1}\rangle$ may be efficiently constructed. In fact, the construction of the states $|H_0\rangle, |H_1\rangle, \ldots$ is done in conjunction with the computation of $r_1, r_2, \ldots$; in order to prepare several copies of $|H_j\rangle$ it will be necessary to compute r_j, and in turn these copies of $|H_j\rangle$ are used to compute r_{j+1}. This continues up the chain until r_m has been computed and $|H_m\rangle$ has been prepared. More specifically, we will begin with a large (polynomial) number of copies of $|H_0\rangle$ (which are of course trivial to prepare), use some relatively small number of these states to compute r_1, then convert the rest of the copies of $|H_0\rangle$ to copies of $|H_1\rangle$ using a procedure described below. We continue up the chain in this fashion, for each j using a relatively small number of copies of $|H_{j-1}\rangle$ to compute r_j, then converting the remaining copies of $|H_{j-1}\rangle$ to copies of $|H_j\rangle$.

In subsections 3.1 and 3.2 we discuss the two components (computing the r_j values and converting copies of $|H_{j-1}\rangle$ to copies of $|H_j\rangle$) individually, and in subsection 3.3 we describe the main algorithm that combines the two components. The following notation and conventions will be used in these subsections. Given a finite group G and a subgroup H of G, for each element $g \in G$ define $r_H(g)$ to be the smallest positive integer r such that $g^r \in H$ (which we have referred to as the order of g with respect to H). For any positive integer m and $k \in \mathbb{Z}_m$ we write $e_m(k)$ to denote $e^{2\pi i k/m}$. For any finite set S we write $|S\rangle = |S|^{-1/2} \sum_{g \in S} |g\rangle$. Finally, whenever we refer to an observation of some quantum register, it is assumed that the observation takes place in the standard (computational) basis.

3.1 Finding orders with respect to a subgroup

Our method for computing the order of an element g with respect to a subgroup H (i.e., computing the r_j values) is essentially Shor's order finding algorithm, except that we begin with one of the registers initialized to $|H\rangle$, and during the algorithm this register is reversibly multiplied by an appropriate power of g. In short, initializing one of the registers to $|H\rangle$ gives us an easy way to work over the cosets of H, the key properties being (i) that the states $|g^i H\rangle$ and $|g^j H\rangle$ are orthogonal whenever g^i and g^j are elements in different cosets of H (and of course $|g^i H\rangle = |g^j H\rangle$ otherwise), and (ii) we will not need to be able to recognize which coset we are in (or even look at the corresponding register at all) to be able to compute the order of g with respect to H correctly.

Now we describe the method in more detail. However, since the analysis is almost identical to the analysis of Shor's algorithm, we will not discuss the analysis in detail and instead refer the reader to Shor [36] and to other sources in which analyses of closely related techniques are given in detail [14, 29].

We assume we are working over a black-box group G with encoding length n, and that a quantum register $\mathbf{R}$ has been initialized to state $|H\rangle$ for H some subgroup of G. For given g we are trying to find $r = r_H(g)$, which is the smallest positive integer such that $g^r \in H$. Let $\mathbf{A}$ be a quantum register whose basis states correspond to $\mathbb{Z}_N$ for N to be chosen later, and assume $\mathbf{A}$ is initialized to state $|0\rangle$.

Similar to Shor's algorithm, we (i) perform the quantum Fourier transform modulo N (QFT_N) on $\mathbf{A}$, (ii) reversibly left-multiply the contents of $\mathbf{R}$ by g^a, for a the number contained in $\mathbf{A}$, and (iii) perform $\mathrm{QFT}_N^\dagger$ on $\mathbf{A}$. Multiplication by g^a can easily be done reversibly in polynomial time using the group oracle along with repeated squaring. The state of the pair $(\mathbf{A}, \mathbf{R})$ is now

$$\frac{1}{N} \sum_{a \in \mathbb{Z}_N} \sum_{b \in \mathbb{Z}_N} e_N(-ab) |b\rangle |g^a H\rangle.$$

Observation of $\mathbf{A}$ yields some value $b \in \mathbb{Z}_N$; we will have with high probability that b/N is a good approximation to k/r (with respect to "modulo 1" distance), where k is randomly distributed in $\mathbb{Z}_r$. Assuming N is sufficiently large, we may find relatively prime integers u and v such that $u/v = k/r$ with high probability via the continued fraction method—choosing $N = 2^{2n+O(\log(1/\varepsilon))}$ allows us to determine u and v with probability $1-\varepsilon$. Now, to find r, we repeat this process $O(\log(1/\varepsilon))$ times and compute the least common multiple of the v values, which yields r with probability at least $1-\varepsilon$.

3.2 Creating uniform superpositions over subgroups

Next we describe how several copies of the state $|H\rangle$ may be converted to several copies of the state $|\langle g\rangle H\rangle$. We assume g normalizes H (i.e., $gH = Hg$, implying that $\langle g\rangle H$ is a group and that $H \triangleleft \langle g\rangle H$) and further that

$$r = r_H(g) = |\langle g\rangle H/H|$$

is known. For the main algorithm this corresponds to converting the copies of $|H_{j-1}\rangle$ to copies of $|H_j\rangle$. We note that this is the portion of the algorithm that apparently requires the normal subgroup relations in (1), as the assumption that g normalizes H is essential for the method.

Specifically, for sufficiently large l, l copies of $|H\rangle$ are converted to $l-1$ copies of $|\langle g\rangle H\rangle$ with high probability; the procedure fails to convert just one of the copies. We assume that we have registers $\mathbf{R}_1,\dots,\mathbf{R}_l$, each in state $|H\rangle$. Let $\mathbf{A}_1,\dots,\mathbf{A}_l$ be registers whose basis states correspond to $\mathbb{Z}_r$, and assume $\mathbf{A}_1,\dots,\mathbf{A}_l$ are each initialized to $|0\rangle$. For each $i=1,\dots,l$ do the following: (i) perform QFT_r on register $\mathbf{A}_i$, (ii) reversibly left-multiply the contents of $\mathbf{R}_i$ by g^{a_i}, where a_i denotes the contents of $\mathbf{A}_i$, and (iii) again perform QFT_r on $\mathbf{A}_i$. Each pair $(\mathbf{A}_i,\mathbf{R}_i)$ is now in the state

$$\frac{1}{r}\sum_{a_i\in\mathbb{Z}_r}\sum_{b_i\in\mathbb{Z}_r} e_r(a_ib_i)|b_i\rangle|g^{a_i}H\rangle.$$

Now, measure $\mathbf{A}_1,\dots,\mathbf{A}_l$, denoting the results by $b_1,\dots,b_l$. Let $|\psi_i\rangle$ denote the resulting (normalized) state of $\mathbf{R}_i$ for each i, i.e.,

$$|\psi_i\rangle=\frac{1}{\sqrt{r}}\sum_{a_i\in\mathbb{Z}_r} e_r(a_ib_i)|g^{a_i}H\rangle.$$

Now we hope that at least one of the values b_i is relatively prime to r; this fails to happen with probability at most ε whenever $l\in\Omega((\log\log r)(\log 1/\varepsilon))$. Assuming we are in this case, choose k such that b_k is relatively prime to r. We will use $|\psi_k\rangle$ to "correct" the state in each of the remaining registers $\mathbf{R}_i$, $i\neq k$, by doing the following: reversibly multiply the contents of $\mathbf{R}_k$ by f^c, where f denotes the group element contained in $\mathbf{R}_i$ and c is any integer satisfying $c\equiv b_ib_k^{-1}\ (\mathrm{mod}\, r)$. We claim at this point that $\mathbf{R}_i$ contains the state $|\langle g\rangle H\rangle$ and $\mathbf{R}_k$ is unchanged (i.e., still contains $|\psi_k\rangle$). To see this, consider an operator M_{g^jh} that multiplies the contents of $\mathbf{R}_k$ by g^jh (for arbitrary $h\in H$). As g normalizes H we have

$$\begin{aligned} M_{g^jh}|\psi_k\rangle &= \frac{1}{\sqrt{r}}\sum_{a_k\in\mathbb{Z}_r} e_r(a_kb_k)|g^{j+a_k}H\rangle\\ &= \frac{1}{\sqrt{r}}\sum_{a_k\in\mathbb{Z}_r} e_r((a_k-j)b_k)|g^{a_k}H\rangle\\ &= e_r(-jb_k)|\psi_k\rangle, \end{aligned}$$

which shows that the state $|\psi_k\rangle$ is an eigenvector of M_{g^jh} with associated eigenvalue $e_r(-jb_k)$. Thus, after performing the above multiplication, the state of the pair $(\mathbf{R}_i,\mathbf{R}_k)$ is

$$\begin{aligned} &\frac{1}{\sqrt{r|H|}}\sum_{a_i\in\mathbb{Z}_r}\sum_{h\in H} e_r(a_ib_i)|g^{a_i}h\rangle M_{(g^{a_i}h)^c}|\psi_k\rangle\\ &= \frac{1}{\sqrt{r|H|}}\sum_{a_i\in\mathbb{Z}_r}\sum_{h\in H} e_r(a_ib_i-a_ib_ib_k^{-1}b_k)|g^{a_i}h\rangle|\psi_k\rangle\\ &= \frac{1}{\sqrt{r|H|}}\sum_{a_i\in\mathbb{Z}_r}\sum_{h\in H} |g^{a_i}h\rangle|\psi_k\rangle\\ &= |\langle g\rangle H\rangle\,|\psi_k\rangle. \end{aligned}$$

This procedure is repeated for each value of $i\neq k$ and then $\mathbf{R}_k$ is discarded; this results in $l-1$ copies of $|\langle g\rangle H\rangle$ as desired.

It should be noted that it is not really necessary that one of the b_i values is relatively prime to r, but a more complicated procedure is necessary in this case. Since we already have a polynomial-time algorithm without the more complicated procedure, we will not discuss it further.

3.3 The main algorithm

As above, we assume we have elements $g_1,\dots,g_m\in G$ such that for $H_j=\langle g_1,\dots,g_j\rangle$ for each j, we have

$$\{1\}=H_0\lhd H_1\lhd\cdots\lhd H_m=G.$$

Defining $r_j=r_{H_{j-1}}(g_j)=|H_j/H_{j-1}|$ for each j we have $|G|=\prod_{j=1}^m r_j$. Consider the algorithm in Figure 1. Here, k is a parameter to be chosen later.

Prepare $k(m+1)$ copies of the state $|H_0\rangle$, where $H_0=\{1\}$.

Do the following for $j=1,\dots,m$:

> Using $k-1$ of the copies of $|H_{j-1}\rangle$, compute $r_j=r_{H_{j-1}}(g_j)$ (and discard these $k-1$ states).
>
> Use one of the copies of $|H_{j-1}\rangle$ to convert the remaining copies of $|H_{j-1}\rangle$ to copies of $|H_j\rangle$.

End of for loop.

Output $\prod_{j=1}^m r_j$.

Figure 1: Algorithm to compute the order of a solvable group G

It is clear that the algorithm operates correctly assuming that each evaluation of r_j is done without error, and that the copies of $|H_{j-1}\rangle$ are converted to copies of $|H_j\rangle$ without error on each iteration of the loop. To have that the algorithm works correctly with high probability in general, we must simply choose parameters so that the error in all of these steps is small. If we want the entire process to work with probability of error less than ε, we may perform the computations of each of the r_j values such that each computation errs with probability at most $\varepsilon/(2m)$, and for each j the copies of $|H_{j-1}\rangle$ are converted to copies of $|H_j\rangle$ with error at most $\varepsilon/(2m)$. Thus, choosing $k=O((\log n)(\log m/\varepsilon))$ suffices. In time polynomial in $mn+\log(1/\varepsilon)$ we may therefore achieve probability of error ε by choosing k polynomial in $mn+\log(1/\varepsilon)$ and computing the r_j values with sufficient accuracy.

A similar choice of parameters allows $|G\rangle$ to be approximated with accuracy ε in the trace-norm metric in time polynomial in mn and $\log(1/\varepsilon)$, as claimed in Theorem 1.

4. OTHER PROBLEMS

In this section we discuss other problems regarding solvable groups that can be solved in quantum polynomial time with the help of our main algorithm. First we discuss membership testing and other problems that easily reduce to computing order. We then we discuss the general technique for computing over factor groups of solvable groups.

4.1 Membership testing and simple reductions to order finding

Suppose we are given elements $g_1,\dots,g_k$ and h in some black-box group. Clearly $h\in\langle g_1,\dots,g_k\rangle$ if and only if $|\langle g_1,\dots,g_k\rangle|=|\langle g_1,\dots,g_k,h\rangle|$. Thus, if $\langle g_1,\dots,g_k,h\rangle$ is solvable, then the question of whether $h\in\langle g_1,\dots,g_k\rangle$ can be computed in quantum polynomial time. Since there is a classical algorithm for testing solvability, it is really only necessary that $\langle g_1, \dots, g_k\rangle$ is solvable; if $\langle g_1, \dots, g_k\rangle$ is solvable but $\langle g_1, \dots, g_k, h\rangle$ is not, then clearly $h \notin \langle g_1, \dots, g_k\rangle$.

Several other problems reduce to order computation or membership testing in solvable groups. A few examples are testing whether a given solvable group is a subgroup of another (given $g_1, \dots, g_k$ and $h_1, \dots, h_l$, is it the case that $\langle h_1, \dots, h_l\rangle \le \langle g_1, \dots, g_k\rangle$?), testing equality of two solvable groups (given $g_1, \dots, g_k$ and $h_1, \dots, h_l$, is it the case that $\langle g_1, \dots, g_k\rangle = \langle h_1, \dots, h_l\rangle$?), and testing whether a given group is a normal subgroup of a given solvable group (given $g_1, \dots, g_k$ and $h_1, \dots, h_l$, is $\langle h_1, \dots, h_l\rangle$ normal in $\langle g_1, \dots, g_k\rangle$?). To determine if $\langle h_1, \dots, h_l\rangle$ is a subgroup of $\langle g_1, \dots, g_k\rangle$, we may test that $|\langle h_1, \dots, h_l, g_1, \dots, g_k\rangle| = |\langle g_1, \dots, g_k\rangle|$ (or we may test that each h_j is an element of $\langle g_1, \dots, g_k\rangle$ separately), to test equality we verify that $\langle g_1, \dots, g_k\rangle \le \langle h_1, \dots, h_l\rangle$ and $\langle h_1, \dots, h_l\rangle \le \langle g_1, \dots, g_k\rangle$, and to test normality we verify that $g_i^{-1} h_j g_i \in \langle h_1, \dots, h_l\rangle$ for each i and j (as well as $\langle h_1, \dots, h_l\rangle \le \langle g_1, \dots, g_k\rangle$). See Babai [3] for more examples of problems reducing to order computation.

In another paper [38] we have shown that there exist succinct *quantum* certificates for various group-theoretic properties, including the property that a given integer divides the order of a group (i.e., given an integer d and generators $g_1, \dots, g_k$ in some black-box group, where $G = \langle g_1, \dots, g_k\rangle$ is not necessarily solvable, verify that d divides $|G|$). We note here that our quantum algorithm for calculating orders of solvable groups can be used to prove the existence of succinct *classical* certificates for this property. Suppose we are given d and $g_1, \dots, g_k$ as above. Then a classical certificate for the property that d divides $|G|$ may consist of descriptions of p-subgroups of G for the primes p dividing d. More specifically, suppose $d = p_1^{a_1} \cdots p_m^{a_m}$ for distinct primes $p_1, \dots, p_m$. Then for each prime power $p_j^{a_j}$, the certificate will include a description of some subgroup of G having order $p_j^{a_j}$. If $p_j^{a_j}$ indeed divides $|G|$ there will exist such a subgroup, which is necessarily solvable since all groups of prime power order are solvable. Thus, the order of each given p-subgroup can be found using the order calculation algorithm. Since G is not necessarily solvable, however, testing that each of the given p-subgroups is really a subgroup of G might not be possible with our algorithm. However, the certificate may also include proofs of membership for each of the generators of the p-subgroups in G. (See Babai and Szemerédi [7] for details on proofs of membership.)

4.2 Computing over abelian factor groups

For abelian black-box groups, many group-theoretic problems can be solved in polynomial time on a quantum computer. For instance, given generators for an abelian black-box group G with encoding length n, we may compute prime powers $q_1, \dots, q_m$ such that $G \cong \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_m}$ in quantum polynomial time. Furthermore, there exists an isomorphism $\theta : G \to \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_m}$ such that for any $h \in G$, $\theta(h)$ may be computed in time polynomial in n. Consequently, computing the order of an abelian group, testing isomorphism of abelian groups, and several other problems can be performed in quantum polynomial time [13, 24, 32].

We may apply these techniques for problems about abelian groups to problems about solvable groups by working over factor groups. To illustrate how this may be done, consider the following problem. Suppose we have a solvable group G given by generators $g_1, \dots, g_k$, and furthermore that we have generators $h_1, \dots, h_l$ for a normal subgroup H of G such that G/H is abelian. We may hope to determine the structure of G/H using the technique for abelian groups mentioned above, i.e., we wish to compute prime powers $q_1, \dots, q_m$ such that $G/H \cong \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_m}$. However, a complication arises since we do not have unique classical representations for elements of G/H, and so we cannot apply the technique directly. Instead, we will rely on the fact that we may efficiently construct copies of the state $|H\rangle$ in polynomial time in order to work over the factor group G/H. Assume that $r_1 = \mathrm{order}(g_1), \dots, r_k = \mathrm{order}(g_k)$ have already been computed using Shor's algorithm, and let $N = \mathrm{lcm}(r_1, \dots, r_k)$. The algorithm described in Figure 2 will allow us to solve the problem.

Prepare register $\mathbf{R}$ in state $|H\rangle$ using the algorithm from Section 3.

Initialize registers $\mathbf{A}_1, \dots, \mathbf{A}_k$ each in state $\frac{1}{\sqrt{N}} \sum_{a=0}^{N-1} |a\rangle$.

Reversibly (left-)multiply the contents of register $\mathbf{R}$ by $g_1^{a_1} \cdots g_k^{a_k}$, where each a_j denotes the contents of register $\mathbf{A}_j$.

For $j = 1, \dots, k$, perform the quantum Fourier transform modulo N on register $\mathbf{A}_j$.

Observe $\mathbf{A}_1, \dots, \mathbf{A}_k$.

Figure 2: Quantum subroutine used for determining the structure of G/H.

To analyze this algorithm, let us define $f : \mathbb{Z}_N^k \to G/H$ as $f(a_1, \dots, a_k) = g_1^{a_1} \cdots g_k^{a_k} H$. The mapping f is a homomorphism with

$$\ker(f) = \{(a_1, \dots, a_k) \in \mathbb{Z}_N^k \,|\, g_1^{a_1} \cdots g_k^{a_k} \in H\}.$$

Let $\ker(f)^\perp$ denote the set of all $(b_1, \dots, b_k) \in \mathbb{Z}_N^k$ such that $\sum_{j=1}^k a_j b_j \equiv 0 \pmod{N}$ for all $(a_1, \dots, a_k) \in \ker(f)$. We have that $\ker(f)^\perp \cong G/H$, and in fact f is an isomorphism when restricted to $\ker(f)^\perp$. A straightforward analysis reveals that observation of $\mathbf{A}_1, \dots, \mathbf{A}_k$ will give a random element in $\ker(f)^\perp$.

Thus, running the algorithm in Figure 2 $O(k)$ times results in a generating set for $\ker(f)^\perp$ with high probability. Letting B be a matrix whose columns are the randomly generated elements of $\ker(f)^\perp$, we may determine the numbers $q_1, \dots, q_m$ in polynomial time by computing the Smith normal form of B (see Kannan and Bachem [28] and Hafner and McCurley [22] for polynomial-time algorithms for computing Smith normal forms).

This method for working over factor groups can be applied to other problems. In general, we may represent elements in a factor group G/H by quantum states of the form $|gH\rangle$. Two states $|gH\rangle$ and $|g'H\rangle$ are of course identical whenever $gH = g'H$, and are orthogonal otherwise. Multiplication and inverses work as expected—for U_G as in Section 2 we have $U_G|gH\rangle|g'H\rangle = |gH\rangle|gg'H\rangle$ and $U_G^{-1}|gH\rangle|g'H\rangle = |gH\rangle|g^{-1}g'H\rangle$. (This requires $H \triangleleft G$.) Hence this gives us a natural way to represent elements of factor groups by quantum states.

5. CONCLUSION

We have given a polynomial-time quantum algorithm for calculating the order and preparing a uniform superposition over a given solvable group, and shown how this algorithm may be used to solve other group-theoretic problems regarding solvable groups in polynomial time.

There are several other problems about solvable black-box groups for which we do not have polynomial-time algorithms. Examples include Group Intersection (given generating sets for two subgroups of a solvable black-box group, do the subgroups have a nontrivial intersection?) and Coset Intersection (defined similarly). See Babai [3] for more examples of group-theoretic problems we may hope to solve in quantum polynomial time in the solvable black-box group setting.

Another interesting question is whether there exist polynomial-time quantum algorithms for similar problems for arbitrary (not necessarily solvable) finite groups. The recent work of Ivanyos, Magniez, and Santha [26] represents progress in this direction.

Acknowledgments

I thank Eric Bach, Richard Cleve, Alexei Kitaev, Michele Mosca, and Miklos Santha for helpful comments and suggestions.

6. REFERENCES

[1] V. Arvind and N. V. Vinodchandran. Solvable black-box group problems are low for PP. *Theoretical Computer Science*, 180:17–45, 1997.

[2] L. Babai. Local expansion of vertex-transitive graphs and random generation in finite groups. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, pages 164–174, 1991.

[3] L. Babai. Bounded round interactive proofs in finite groups. *SIAM Journal on Discrete Math*, 5(1):88–111, 1992.

[4] L. Babai. Randomization in group algorithms: conceptual questions. In *Groups and Computation, II*, volume 28 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 1–17. American Mathematical Society, 1997.

[5] L. Babai and R. Beals. A polynomial-time theory of black box groups I. In *Groups St. Andrews 1997 in Bath*, volume 260 of *London Math. Soc. Lecture Note Ser.* Cambridge University Press, 1999.

[6] L. Babai, G. Cooperman, L. Finkelstein, E. Luks, and Á. Seress. Fast Monte Carlo algorithms for permutation groups. *Journal of Computer and System Sciences*, 50:296–307, 1995.

[7] L. Babai and E. Szemerédi. On the complexity of matrix group problems I. In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science*, pages 229–240, 1984.

[8] R. Beals. Quantum computation of Fourier transforms over symmetric groups. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 48–53, 1997.

[9] R. Beals and L. Babai. Las Vegas algorithms for matrix groups. In *Proceedings of the 34th Annual IEEE Conference on Foundations of Computer Science*, pages 427–436, 1993.

[10] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.

[11] D. Boneh and R. Lipton. Quantum cryptanalysis of hidden linear functions. In *Advances in Cryptology – Crypto'95*, volume 963 of *Lecture Notes in Computer Science*, pages 242–437. Springer-Verlag, 1995.

[12] G. Brassard and P. Høyer. An exact quantum polynomial-time algorithm for Simon's problem. In *Fifth Israeli Symposium on Theory of Computing and Systems*, pages 12–23, 1997.

[13] K. Cheung and M. Mosca. Decomposing finite Abelian groups. Los Alamos Preprint Archive, quant-ph/0101004, 2001.

[14] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society, London*, A454:339–354, 1998.

[15] H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1993.

[16] D. Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London*, A400:97–117, 1985.

[17] M. Ettinger and P. Høyer. On quantum algorithms for noncommutative hidden subgroups. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1563 of *Lecture Notes in Computer Science*, pages 478–487, 1999.

[18] M. Ettinger, P. Høyer, and E. Knill. Hidden subgroup states are almost orthogonal. Los Alamos Preprint Archive, quant-ph/9901034, 1999.

[19] L. Fortnow and J. Rogers. Complexity limitations on quantum computation. *Journal of Computer and System Sciences*, 59(2):240–252, 1999.

[20] M. Grigni, L. Schulman, M. Vazirani, and U. Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, 2001.

[21] D. Grigoriev. Testing shift-equivalence of polynomials using quantum machines. In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, pages 49–54, 1996.

[22] J. Hafner and K. McCurley. Asymptotically fast triangularization of matrices over rings. *SIAM Journal on Computing*, 20(6):1068–1083, 1991.

[23] S. Hallgren, A. Russell, and A. Ta-Shma. Normal subgroup reconstruction and quantum computation using group representations. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 627–635, 2000.

[24] P. Høyer. *Quantum Algorithms*. PhD thesis, Odense University, Denmark, 2000.

[25] I. M. Isaacs. *Algebra: a Graduate Course*. Brooks/Cole, 1994.

[26] G. Ivanyos, F. Magniez, and M. Santha. Efficient algorithms for some instances of the non-Abelian hidden subgroup problem. Los Alamos Preprint Archive, quant-ph/0102014, 2001.

[27] R. Jozsa. Quantum algorithms and the Fourier transform. *Proceedings of the Royal Society of London A*, pages 323–337, 1998.

[28] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, 1979.

[29] A. Kitaev. Quantum measurements and the Abelian stabilizer problem. Los Alamos Preprint Archive, quant-ph/9511026, 1995.

[30] A. Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, 1997.

[31] E. Luks. Computing in solvable matrix groups. In *Proceedings of the 33rd Symposium on the Foundations of Computer Science*, pages 111–120, 1992.

[32] M. Mosca. *Quantum Computer Algorithms*. PhD thesis, University of Oxford, 1999.

[33] M. Mosca and A. Ekert. The hidden subgroup problem and eigenvalue estimation on a quantum computer. In *Proceedings of the 1st NASA International Conference on Quantum Computing and Quantum Communication*, volume 1509 of *Lecture Notes in Computer Science*, 1999.

[34] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[35] M. Rötteler and T. Beth. Polynomial-time solution to the hidden subgroup problem for a class of non-abelian groups. Los Alamos Preprint Archive, quant-ph/9812070, 1999.

[36] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[37] D. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.

[38] J. Watrous. Succinct quantum proofs for properties of finite groups. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 537–546, 2000.