On the Complexity of the Hidden Subgroup Problem

Stephen Fenner[1,⋆] and Yong Zhang[2,⋆⋆]

[1] Department of Computer Science & Engineering, University of South Carolina
Columbia, SC 29208 USA
fenner@cse.sc.edu

[2] Department of Mathematical Sciences, Eastern Mennonite University
Harrisonburg, VA 22802, USA
yong.zhang@emu.edu

Abstract. We show that several problems that figure prominently in quantum computing, including Hidden Coset, Hidden Shift, and Orbit Coset, are equivalent or reducible to Hidden Subgroup. We also show that, over permutation groups, the decision version and search version of Hidden Subgroup are polynomial-time equivalent. For Hidden Subgroup over dihedral groups, such an equivalence can be obtained if the order of the group is smooth. Finally, we give nonadaptive program checkers for Hidden Subgroup and its decision version.

1 Introduction

The Hidden Subgroup problem generalizes many interesting problems that have efficient quantum algorithms but whose known classical algorithms are inefficient. While we can solve Hidden Subgroup over abelian groups satisfactorily on quantum computers, the nonabelian case is more challenging. Although there are many families of groups besides abelian ones for which Hidden Subgroup is known to be solvable in quantum polynomial time, the overall successes are considered limited. People are particularly interested in solving Hidden Subgroup over two families of nonabelian groups, permutation groups and dihedral groups, since solving them will immediately give solutions to Graph Isomorphism [Joz00] and Shortest Lattice Vector [Reg04], respectively.

To explore more fully the power of quantum computers, researchers have also introduced and studied several related problems. Van Dam, Hallgren, and Ip [vDHI03] introduced the Hidden Shift problem and gave efficient quantum algorithms for some instances. Their results provide evidence that quantum computers can help to recover shift structure as well as subgroup structure. They also introduced the Hidden Coset problem to generalize Hidden Shift and Hidden Subgroup. Recently, Childs and van Dam [CvD07] introduced the Generalized Hidden Shift problem, extending Hidden Shift from a

⋆ Partially supported by NSF grant CCF-0515269.
⋆⋆ Partially supported by an EMU Summer Research Grant 2006.

M. Agrawal et al. (Eds.): TAMC 2008, LNCS 4978, pp. 70–81, 2008.

different angle. In an attempt to attack HIDDEN SUBGROUP using a divide-and-conquer approach over subgroup chains, Friedl et al. [FIM+03] introduced the ORBIT COSET problem, which they claimed to be an even more general problem including HIDDEN SUBGROUP and HIDDEN SHIFT[1] as special instances. They called ORBIT COSET a *quantum* generalization of HIDDEN SUBGROUP and HIDDEN SHIFT, since the definition of ORBIT COSET involves *quantum functions*.

In Section 3, we show that all these related problems are equivalent or reducible to HIDDEN SUBGROUP with different underlying groups. In particular,

1. HIDDEN COSET is polynomial-time equivalent to HIDDEN SUBGROUP,
2. ORBIT COSET is equivalent to HIDDEN SUBGROUP if we allow functions in the latter to be quantum functions, and
3. HIDDEN SHIFT and GENERALIZED HIDDEN SHIFT reduce to instances of HIDDEN SUBGROUP over a family of wreath product groups.[2]

There are a few results in the literature about the complexity of HIDDEN SUBGROUP. HIDDEN SUBGROUP over abelian groups is in class **BQP** [Kit95, Mos99]. Ettinger, Hoyer, and Knill [EHK04] showed that HIDDEN SUBGROUP (over arbitrary finite groups) has polynomial quantum query complexity. Arvind and Kurur [AK02] showed that HIDDEN SUBGROUP over permutation groups is in the class $\mathbf{FP}^{\mathbf{SPP}}$ and is thus low for the counting complexity class **PP**. In Section 4 we study the relationship between the decision and search versions of HIDDEN SUBGROUP, denoted as HIDDEN SUBGROUP$_D$ and HIDDEN SUBGROUP$_S$, respectively. It is well known that **NP**-complete sets such as SAT are self-reducible, which implies that the decision and search versions of **NP**-complete problems are polynomial-time equivalent. We show this is also the case for HIDDEN SUBGROUP and HIDDEN SHIFT over permutation groups. There are evidences in the literature showing that HIDDEN SUBGROUP$_D$ over permutation groups is difficult [HRTS03, GSVV04, MRS05, KS05, HMR+06, MRS07]. In particular, Kempe and Shalev [KS05] showed that under general conditions, various forms of the Quantum Fourier Sampling method are of no help (over classical exhaustive search) in solving HIDDEN SUBGROUP$_D$ over permutation groups. Our results yield evidence of a different sort that this problem is difficult—namely, it is just as hard as the search version. For HIDDEN SUBGROUP over dihedral groups, our results are more modest. We show the search-decision equivalence for dihedral groups of smooth order, i.e., where the largest prime dividing the order of the group is small.

Combining our results in Sections 3 and 4, we obtain nonadaptive program checkers for HIDDEN SUBGROUP and HIDDEN SUBGROUP$_D$ over permutation groups. We give the details in Section 5.

[1] They actually called it the Hidden Translation problem.

[2] Friedl et al. [FIM+03] gave a reduction from HIDDEN SHIFT to instances of HIDDEN SUBGROUP over semi-direct product groups. However, their reduction only works when the group G is abelian.

2 Preliminaries

2.1 Group Theory

Background on general group theory and quantum computation can be found in textbooks [Sco87] and [NC00]. A special case of the wreath product groups plays an important role in several proof.

Definition 1. For any finite group G, the *wreath product* $G \wr \mathbb{Z}_n$ of G and $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ is the set $\{(g_1, g_2, \ldots, g_n, \tau) \mid g_1, g_2, \ldots, g_n \in G,\ \tau \in \mathbb{Z}_n\}$ equipped with the group operation $\circ$ such that

$$(g_1, g_2, \ldots, g_n, \tau) \circ (g_1', g_2', \ldots, g_n', \tau') = (g_{\tau'(1)}g_1', g_{\tau'(2)}g_2', \ldots, g_{\tau'(n)}g_n', \tau\tau').$$

We abuse notation here by identifying τ and τ' with permutations over the set $\{1, \ldots, n\}$ sending x to $x + \tau \bmod n$ and to $x + \tau' \bmod n$, respectively.

Let Z be a set and S_Z be the *symmetric group* of permutations of Z. We define the composition order to be from left to right, i.e., for $g_1, g_2 \in S_Z$, $g_1 g_2$ is the permutation obtained by applying g_1 first and then g_2. For $n \geq 1$, we abbreviate $S_{\{1,2,\ldots,n\}}$ by S_n. Subgroups of S_n are the *permutation groups* of degree n. For a permutation group $G \leq S_n$ and an element $i \in \{1, \ldots, n\}$, let $G^{(i)}$ denote the stabilizer subgroup of G that fixes the set $\{1, \ldots, i\}$ pointwise. The chain of the stabilizer subgroups of G is $\{id\} = G^{(n)} \leq G^{(n-1)} \leq \cdots \leq G^{(1)} \leq G^{(0)} = G$. Let C_i be a complete set of right coset representatives of $G^{(i)}$ in $G^{(i-1)}$, $1 \leq i \leq n$. Then the cardinality of C_i is at most $n-i$ and $\cup_{i=1}^{n} C_i$ forms a *strong generator set* for G [Sim70]. Any element $g \in G$ can be written uniquely as $g = g_n g_{n-1} \cdots g_1$ with $g_i \in C_i$. Furst, Hopcroft, and Luks [FHL80] showed that given any generator set for G, a strong generator set can be computed in polynomial time. For $X \subseteq Z$ and $G \leq S_Z$, we use G_X to denote the subgroup of G that stablizes X setwise. It is evident that G_X is the direct sum of S_X and $S_{Z \setminus X}$. We are particularly interested in the case when G is S_n. In this case, a generating set for G_X can be easily computed.

Let G be a finite group. Let Γ be a set of mutually orthogonal quantum states. Let $\alpha : G \times \Gamma \to \Gamma$ be a group action of G on Γ, i.e., for every $x \in G$ the function $\alpha_x : \Gamma \to \Gamma$ mapping $|\phi\rangle$ to $|\alpha(x, |\phi\rangle)\rangle$ is a permutation over Γ, and the map h from G to the symmetric group over Γ defined by $h(x) = \alpha_x$ is a homomorphism. We use the notation $|x \cdot \phi\rangle$ instead of $|\alpha(x, |\phi\rangle)\rangle$, when α is clear from the context. We let $G(|\phi\rangle)$ denote the orbit of $|\phi\rangle$ with respect to α, i.e., the set $\{|x \cdot \phi\rangle : x \in G\}$, and we let $G_{|\phi\rangle}$ denote the stabilizer subgroup of $|\phi\rangle$ in G, i.e., $\{x \in G : |x \cdot \phi\rangle = |\phi\rangle\}$. Given any positive integer t, let α^t denote the group action of G on $\Gamma^t = \{|\phi\rangle^{\otimes t} : |\phi\rangle \in \Gamma\}$ defined by $\alpha^t(x, |\phi\rangle^{\otimes t}) = |x \cdot \phi\rangle^{\otimes t}$. We need α^t because the input superpositions cannot be cloned in general.

Definition 2. Let G be a finite group.

- Given a generating set for G and a function f that maps G to some finite set S, where the values of f are constant on a subgroup H of G and distinct on each left (right) coset of H. The Hidden Subgroup problem is to find a generating set for H. The decision version of Hidden Subgroup, denoted as Hidden Subgroup$_D$, is to determine whether H is trivial. The search version, denoted as Hidden Subgroup$_S$, is to find a nontrivial element, if there exists one, in H.

- Given a generating set for G and n injective functions $f_1, f_2, \ldots, f_n$ defined on G, with the promise that there is a "shift" $u \in G$ such that for all $g \in G$, $f_1(g) = f_2(gu), f_2(g) = f_3(gu), \ldots, f_{n-1}(g) = f_n(gu)$, the Generalized Hidden Shift problem is to find u. If $n = 2$, this problem is called the Hidden Shift problem.
- Given a generating set for G and two functions f_1 and f_2 defined on G such that for some shift $u \in G$, $f_1(g) = f_2(gu)$ for all g in G, the Hidden Coset problem is to find the set of all such u.
- Given a generating set for G and two quantum states $|\phi_0\rangle, |\phi_1\rangle \in \Gamma$, the Orbit Coset problem is to either reject the input if $G(|\phi_0\rangle) \cap G(|\phi_1\rangle) = \emptyset$, or else output both a $u \in G$ such that $|u \cdot \phi_1\rangle = |\phi_0\rangle$ and also a generating set for $G_{|\phi_1\rangle}$.

2.2 Program Checkers

Let π be a computational decision or search problem. Let x be an input to π and $\pi(x)$ be the output of π. Let P be a deterministic program (supposedly) for π that halts on all inputs. We are interested in whether P has any bug, i.e., whether there is some x such that $P(x) \neq \pi(x)$. A efficient *program checker* C for P is a probabilistic expected-polynomial-time oracle Turing machine that uses P as an oracle and takes x and a positive integer k (presented in unary) as inputs. The running time of C does not include the time it takes for the oracle P to do its computations. C will output CORRECT with probability $\geq 1 - 1/2^k$ if P is correct on all inputs (no bugs), and output BUGGY with probability $\geq 1 - 1/2^k$ if $P(x) \neq \pi(x)$. This probability is over the sample space of all finite sequences of coin flips C could have tossed. However, if P has bugs but $P(x) = \pi(x)$, we allow C to behave arbitrarily. If C only queries the oracle nonadaptively, then we say C is a *nonadaptive checker*. See Blum and Kannan [BK95] for more details.

3 Several Reductions

The Hidden Coset problem is to find the set of all shifts of the two functions f_1 and f_2 defined on the group G. In fact, the set of all shifts is a coset of a subgroup H of G and f_1 is constant on H (see [vDHI03] Lemma 6.1). If we let f_1 and f_2 be the same function, this is exactly Hidden Subgroup. On the other hand, if f_1 and f_2 are injective functions, this is Hidden Shift.

Theorem 1. Hidden Coset *is polynomial-time equivalent to* Hidden Subgroup.

Proof. Let G and f_1, f_2 be the input of Hidden Coset. Let the set of shifts be Hu, where H is a subgroup of G and u is a coset representative. Define a function f that maps $G \wr \mathbb{Z}_2$ to $S \times S$ as follows: for any $(g_1, g_2, \tau) \in G \wr \mathbb{Z}_2$,

$$f(g_1, g_2, \tau) = \begin{cases} (f_1(g_1), f_2(g_2)) \text{ if } \tau = 0, \\ (f_2(g_2), f_1(g_1)) \text{ if } \tau = 1. \end{cases}$$

The values of f are constant on the set $K = (H \times u^{-1}Hu \times \{0\}) \cup (u^{-1}H \times Hu \times \{1\})$, which is a subgroup of $G \wr \mathbb{Z}_2$. Furthermore, the values of f are distinct on all left cosets of K. Given a generating set of K, there is at least one generator of the form $(k_1, k_2, 1)$. Pick k_2 to be the coset representative u of H. Form a generating set S of H as follows. S is initially empty. For each generator of K, if it is of the form $(k_1, k_2, 0)$, then add k_1 and uk_2u^{-1} to S; if it is of the form $(k_1, k_2, 1)$, then add uk_1 and k_2u^{-1} to S.

Corollary 1. Hidden Coset *has polynomial quantum query complexity.*

It was mentioned in Friedl et al. [FIM+03] that Hidden Coset in general is of exponential (classical) query complexity.

Recently Childs and van Dam [CvD07] proposed Generalized Hidden Shift where there are n injective functions (encoded in a single function). Using a similar approach, we show Generalized Hidden Shift essentially addresses Hidden Subgroup over a different family of groups.

Proposition 1. Generalized Hidden Shift *reduces to* Hidden Subgroup *in time polynomial in n.*

Proof. The input for Generalized Hidden Shift is a group G and n injective functions $f_1, f_2, \ldots, f_n$ defined on a group G such that for all $g \in G$, $f_1(g) = f_2(gu), \ldots, f_{n-1}(g) = f_n(gu)$. Consider the group $G \wr \mathbb{Z}_n$. Define a function f such that for any element in $(g_1, \ldots, g_n, \tau) \in G \wr \mathbb{Z}_n$, $f((g_1, \ldots, g_n, \tau)) = (f_{\tau(0)}(g_0), \ldots, f_{\tau(n)}(g_n))$. The function values of f are constant and distinct for left cosets of the n-element cyclic subgroup generated by $(u^{-(n-1)}, u, u, \cdots, u, 1)$.

Van Dam, Hallgren, and Ip [vDHI03] introduced the Shifted Legendre Symbol problem as a natural instance of Hidden Shift. They claimed that assuming a conjecture this problem can also be reduced to an instance of Hidden Subgroup over dihedral groups. By Proposition 1, this problem can be reduced to Hidden Subgroup over wreath product groups without any conjecture.

Next we show Orbit Coset is not a more general problem either, if we allow the function in Hidden Subgroup to be a quantum function. We need this generalization since the definition of Orbit Coset involves quantum functions, i.e., the ranges of the functions are sets of orthogonal quantum states. In Hidden Subgroup, the function is implicitly considered by most researchers to be a classical function, mapping group elements to a classical set. For the purposes of quantum computation, however, this generalization to quantum functions is natural and does not affect any existing quantum algorithms for Hidden Subgroup.

Proposition 2. Orbit Coset *is quantum polynomial-time equivalent to* Hidden Subgroup.

Proof. Let G and two orthogonal quantum states $|\phi_0\rangle, |\phi_1\rangle \in \Gamma$ be the inputs of Orbit Coset. Define the function $f : G \wr \mathbb{Z}_2 \to \Gamma \otimes \Gamma$ as follows:

$$f(g_1, g_2, \tau) = \begin{cases} |g_1 \cdot \phi_0\rangle \otimes |g_2 \cdot \phi_1\rangle \text{ if } \tau = 0, \\ |g_2 \cdot \phi_1\rangle \otimes |g_1 \cdot \phi_0\rangle \text{ if } \tau = 1. \end{cases}$$

The values of the function f are identical and orthogonal on each left coset of the following subgroup H of $G \wr \mathbb{Z}_2$: If there is no $u \in G$ such that $|u \cdot \phi_1\rangle = |\phi_0\rangle$, then $H = G_{|\phi_0\rangle} \times G_{|\phi_1\rangle} \times \{0\}$. If there is such a u, then $H = (G_{|\phi_0\rangle} \times G_{|\phi_1\rangle} \times \{0\}) \cup (G_{|\phi_1\rangle} u^{-1} \times uG_{|\phi_1\rangle} \times \{1\})$. For $i, j \in \{0, 1\}$, let $g_i \in G$ be the i'th coset representative of $G_{|\phi_0\rangle}$ (i.e., $|g_i \cdot \phi_0\rangle = |\phi_i\rangle$), and let $g_j \in G$ be the j'th coset representative of $G_{|\phi_1\rangle}$ (i.e., $|g_j \cdot \phi_1\rangle = |\phi_j\rangle$). Then elements of the left coset of H represented by $(g_i, g_j, 0)$ will all map to the same value $|\phi_i\rangle \otimes |\phi_j\rangle$ via f.

4 Decision Versus Search

For **NP**-complete problems, the decision and search version are polynomial-time equivalent. This equivalence was also obtained for problems which are not known to be NP-complete, such as Graph Isomorphism [Mat79] and Group Intersection [AT01]. Since both problems reduce to instances of Hidden Subgroup, we ask the question whether Hidden Subgroup has such property. In this section we show that over permutation groups S_n, this decision-search equivalence can actually be obtained for Hidden Subgroup and also Hidden Shift. On the other hand, over dihedral groups D_n, this equivalence can only be obtained for Hidden Subgroup when n has small prime factors.

Lemma 1. *Given (generating sets for) a group $G \leq S_n$, a function $f : G \to S$ that hides a subgroup $H \leq G$, and a sequence of subgroups $G_1, \ldots, G_k \leq S_n$, an instance of* Hidden Subgroup *can be constructed to hide the group $D = \{(g, g, \ldots, g) \mid g \in H \cap G_1 \cap \cdots \cap G_k\}$ inside $G \times G_1 \times \cdots \times G_k$.*

Proof. Define a function f' over the direct product group $G \times G_1 \times \cdots \times G_k$ so that for any element $(g, g_1, \ldots, g_k)$, $f'(g, g_1, \ldots, g_k) = (f(g), gg_1^{-1}, \ldots, gg_k^{-1})$. The values of f' are constant and distinct over left cosets of D.

In the following, we will use the tuple $\langle G, f\rangle$ to represent a standard Hidden Subgroup input instance, and $\langle G, f, G_1, \ldots, G_k\rangle$ to represent a Hidden Subgroup input instance constructed as in Lemma 1.

We define a natural isomorphism that identifies $S_n \wr \mathbb{Z}_2$ with a subgroup of S_Γ, where $\Gamma = \{(i, j) \mid i \in \{1, \ldots, n\},\ j \in \{1, 2\}\}$. This isomorphism can be viewed as a group action, where the group element (g_1, g_2, τ) maps (i, j) to $(g_j(i), \tau(j))$. Note that this isomorphism can be efficiently computed in both directions.

Theorem 2. *Over permutation groups,* Hidden Subgroup$_S$ *is truth-table reducible to* Hidden Subgroup$_D$ *in polynomial time.*

Proof. Suppose f hides a nontrivial subgroup H of G, first we compute a strong generating set for G, corresponding to the chain $\{id\} = G^{(n)} \leq G^{(n-1)} \leq \cdots \leq G^{(1)} \leq G^{(0)} = G$. Define f' over $G \wr \mathbb{Z}_2$ such that f' maps (g_1, g_2, τ) to $(f(g_1), f(g_2))$ if τ is 0, and $(f(g_2), f(g_1))$ otherwise. It is easy to check that for the group $G^{(i)} \wr \mathbb{Z}_2$, $f'|_{G^{(i)} \wr \mathbb{Z}_2}$ hides the subgroup $H^{(i)} \wr \mathbb{Z}_2$.

Query the Hidden Subgroup$_D$ oracle with inputs

$$\left\langle G^{(i)} \wr \mathbb{Z}_2, f'|_{G^{(i)} \wr \mathbb{Z}_2}, (S_\Gamma)_{\{(i,1),(j,2)\}}, (S_\Gamma)_{\{(i,2),(j',1)\}}, (S_\Gamma)_{\{(k,1),(\ell,2)\}} \right\rangle$$

for all $1 \leq i \leq n$, all $j, j' \in \{i+1, \ldots, n\}$, and all $k, \ell \in \{i, \ldots, n\}$.

Claim. Let i be such that $H^{(i)} = \{id\}$ and $H^{(i-1)} \neq \{id\}$. For all $i < j, j' \leq n$ and all $i \leq k, l \leq n$, there is a (necessarily unique) permutation $h \in H^{(i-1)}$ such that $h(i) = j$, $h(j') = i$ and $h(k) = \ell$ if and only if the query

$$\left\langle G^{(i-1)} \wr \mathbb{Z}_2, f'|_{G^{(i-1)} \wr \mathbb{Z}_2}, (S_\Gamma)_{\{(i,1),(j,2)\}}, (S_\Gamma)_{\{(i,2),(j',1)\}}, (S_\Gamma)_{\{(k,1),(\ell,2)\}} \right\rangle$$

to the Hidden Subgroup$_D$ oracle answers "nontrivial."

Proof of Claim. For any $j > i$, there is at most one permutation in $H^{(i-1)}$ that maps i to j. To see this, suppose there are two distinct $h, h' \in H^{(i-1)}$ both of which map i to j. Then $h'h^{-1} \in H^{(i)}$ is a nontrivial permutation, contradicting the assumption $H^{(i)} = \{id\}$. Let $h \in H^{(i-1)}$ be a permutation such that $h(i) = j$, $h(j') = i$, and $h(k) = \ell$. Then $(h, h^{-1}, 1)$ is a nontrivial element in the group $H^{(i-1)} \wr \mathbb{Z}_2 \cap (S_\Gamma)_{\{(i,1),(j,2)\}} \cap (S_\Gamma)_{\{(i,2),(j',1)\}} \cap (S_\Gamma)_{\{(k,1),(\ell,2)\}}$, and thus the oracle answers "nontrivial."

Conversely, if the oracle answers "nontrivial," then the nontrivial element must be of the form $(h, h', 1)$ where $h, h' \in H^{(i-1)}$, since the other form $(h, h', 0)$ will imply that h and h' both fix i and thus are in $H^{(i)} = \{id\}$. Therefore, h will be a nontrivial element of $H^{(i-1)}$ with $h(i) = j$, $h(j') = i$, and $h(k) = \ell$. This proves the Claim.

Find the largest i such that the query answers "nontrivial" for some $j, j' > i$ and some $k, \ell \geq i$. Clearly this is the smallest i such that $H^{(i)} = \{id\}$. A nontrivial permutation in $H^{(i-1)}$ can be constructed by looking at the query results that involve $G^{(i-1)} \wr \mathbb{Z}_2$.

Corollary 2. Hidden Subgroup$_D$ *and* Hidden Subgroup$_S$ *are polynomial-time equivalent over permutation groups,.*

Next we show that the search version of Hidden Shift, as a special case of Hidden Subgroup, also reduces to the corresponding decision problem.

Definition 3. Given a generating set for a group G and two injective functions f_1, f_2 defined on G, the problem Hidden Shift$_D$ is to determine whether there is a shift $u \in G$ such that $f_1(g) = f_2(gu)$ for all $g \in G$.

Theorem 3. *Over permutation groups,* Hidden Shift$_D$ *and* Hidden Shift$_S$ *are polynomial-time equivalent.*

Proof. We show that if there is a translation u for the two injective functions defined on G, we can find u with the help of an oracle that solves Hidden Shift$_D$. First compute the strong generator set $\cup_{i=1}^{n} C_i$ of G using the procedure in [FHL80]. Note that $\cup_{i=k}^{n} C_i$ generates $G^{(k-1)}$ for $1 \leq k \leq n$. We will proceed in steps along the stabilizer subgroup chain $G = G^{(0)} \geq G^{(1)} \geq \cdots \geq G^{(n)} = \{id\}$.

Claim. With the help of the Hidden Shift$_D$ oracle, finding the translation u_i for input $(G^{(i)}, f_1, f_2)$ reduces to finding another translation u_{i+1} for input $(G^{(i+1)}, f_1', f_2')$. In particular, we have $u_i = u_{i+1}\sigma_i$.

Proof of Claim. Ask the oracle whether there is a translation for the input instance $(G^{(i+1)}, f_1|_{G^{(i+1)}}, f_2|_{G^{(i+1)}})$. If the answer is yes, then we know $u_i \in G^{(i+1)}$ and therefore set $\sigma_i = id$ and $u_i = u_{i+1}\sigma_i$.

If the answer is no, then we know that u is in some right coset of $G^{(i+1)}$ in $G^{(i)}$. For every $\tau \in C_{i+1}$, define a function f_τ such that $f_\tau(x) = f_2(x\tau)$ for all $x \in G^{(i+1)}$. Ask the oracle whether there is a translation for input $(G^{(i+1)}, f_1|_{G^{(i+1)}}, f_\tau)$. The oracle will answer yes if and only if u and τ are in the same right coset of $G^{(i+1)}$ in $G^{(i)}$, since

$$\begin{aligned} & u \text{ and } \tau \text{ are in the same right coset of } G^{(i+1)} \text{ in } G^{(i)} \\ \iff & u = u'\tau \text{ for some } \tau' \in G^{(i+1)} \\ \iff & f_1(x) = f_2(xu) = f_2(xu'\tau) = f_\tau(xu') \text{ for all } x \in G^{(i)} \\ \iff & u' \text{ is the translation for } (G^{(i+1)}, f_1|_{G^{(i+1)}}, f_\tau). \end{aligned}$$

Then we set $\sigma_i = \tau$.

We apply the above procedure $n-1$ times until we reach the trivial subgroup $G^{(n)}$. The translation u will be equal to $\sigma_n\sigma_{n-1}\cdots\sigma_1$. Since the size of each C_i is at most $n-i$, the total reduction is in classical polynomial time.

For Hidden Subgroup over dihedral groups D_n, we can efficiently reduce search to decision when n has small prime factors. For a fixed integer B, we say an integer n is B-smooth if all the prime factors of n are less than or equal to B. For such an n, the prime factorization can be obtained in time polynomial in $B + \log n$. Without loss of generality, we assume that the hidden subgroup is an order-two subgroup of D_n [EH00].

Theorem 4. *Let n be a B-smooth number,* Hidden Subgroup *over the dihedral group D_n reduces to* Hidden Subgroup$_D$ *over dihedral groups in time polynomial in $B + \log n$.*

Proof. Without loss of generality, we assume the generator set for D_n is $\{r, \sigma\}$, where the order of r and σ are n and 2, respectively. Let $p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ be the prime factorization of n. Since n is B-smooth, $p_i \leq B$ for all $1 \leq i \leq k$. Let the hidden subgroup H be $\{id, r^a\sigma\}$ for some $a < n$.

First we find $a \bmod p_1^{e_1}$ as follows. Query the Hidden Subgroup$_D$ oracle with input groups (we will always use the original input function f) $\langle r^{p_1}, \sigma\rangle$, $\langle r^{p_1}, r\sigma\rangle, \ldots, \langle r^{p_1}, r^{p_1-1}\sigma\rangle$. It is not hard to see that the Hidden Subgroup$_D$ oracle will answer "nontrivial" only for the input group $\langle r^{p_1}, r^{m_1}\sigma\rangle$ where $m_1 = a \bmod p_1$. The next set of input groups to the Hidden Subgroup$_D$ oracle are $\langle r^{p_1^2}, r^{m_1}\sigma\rangle, \langle r^{p_1^2}, r^{p_1+m_1}\sigma\rangle, \ldots, \langle r^{p_1^2}, r^{(p_1-1)p_1+m_1}\sigma\rangle$. From the oracle answers we obtain $m_2 = a \bmod p_1^2$. Repeat the above procedure until we find $a \bmod p_1^{e_1}$.

Similarly, we can find $a \bmod p_2^{e_2}, \ldots, a \bmod p_k^{e_k}$. A simple usage of the Chinese Remainder Theorem will then recover a. The total number of queries is $e_1p_1 + e_2p_2 + \cdots + e_kp_k$, which is polynomial in $\log n + B$.

5 Nonadaptive Checkers

An important concept closely related to self-reducibility is that of a *program checker*, which was first introduced by Blum and Kannan [BK95]. They gave program checkers for some group-theoretic problems and selected problems in **P**. They also characterized the class of problems having polynomial-time checkers. Arvind and Torán [AT01] presented a nonadaptive **NC** checker for Group Intersection over permutation groups. In this section we show that Hidden Subgroup$_D$ and Hidden Subgroup over permutation groups have nonadaptive checkers.

For the sake of clarity, we give the checker for Hidden Subgroup$_D$ first. Let P be a program that solves Hidden Subgroup$_D$ over permutation groups. The input for P is a permutation group G given by its generating set and a function f that is defined over G and hides a subgroup H of G. If P is a correct program, then $P(G, f)$ outputs TRIVIAL if H is the trivial subgroup of G, and NONTRIVIAL otherwise. The checker $C^P(G, f, 0^k)$ checks the program P on the input G and f as follows:

Begin
Compute $P(G, f)$.
if $P(G, f) =$ NONTRIVIAL, **then**
Use Theorem 2 and P (as if it were bug-free) to search for a nontrivial element h of H.
if $f(h) = f(id)$, **then**
return CORRECT
else
return BUGGY
if $P(G, f) =$ TRIVIAL, **then**
Do k times (in parallel):
generate a random permutation $u \in G$.
define f' over G such that $f(g) = f'(gu)$ for all $g \in G$, use (G, f, f') to be an input instance of Hidden Shift
use Theorem 1 to convert (G, f, f') to an input instance $(G \wr \mathbb{Z}_2, f'')$ of Hidden Subgroup[3]

[3] Using the natural isomorphism we define in Section 4, the group $G \wr \mathbb{Z}_2$ is still considered as a permutation group.

use Theorem 2 and P to search for a nontrivial element h of the subgroup of $G \wr \mathbb{Z}_2$ that f'' hides.
if $h \neq (u^{-1}, u, 1)$, **then return** BUGGY
End-do
return CORRECT
End

Theorem 5. *If P is a correct program for* Hidden Subgroup$_D$, *$C^P(G, f, 0^k)$ always outputs CORRECT. If $P(G, f)$ is incorrect, the probability of $C^P(G, f, 0^k)$ outputting CORRECT is $\leq 2^{-k}$. Moreover, $C^P(G, f, 0^k)$ runs in polynomial time and queries P nonadaptively.*

Proof. If P is a correct program and $P(G, f)$ outputs NONTRIVIAL, then $C^P((G, f, 0^k)$ will find a nontrivial element of H and outputs CORRECT. If P is a correct program and $P(G, f)$ outputs TRIVIAL, the function f' constructed by $C^P(G, f, 0^k)$ will hide the two-element subgroup $\{(id, id, 0), (u, u^{-1}, 1)\}$. Therefore, $C^P(G, f, 0^k)$ will always recover the random permutation u correctly, and output CORRECT.

On the other hand, if $P(G, f)$ outputs NONTRIVIAL while H is actually trivial, then $C^P(G, f, 0^k)$ will fail to find a nontrivial element of H and thus output BUGGY. If $P(G, f)$ outputs TRIVIAL while H is actually nontrivial, then the function f'' constructed by $C^P(G, f, 0^k)$ will hide the subgroup $(H \times u^{-1}Hu \times \{0\}) \cup (u^{-1}H \times Hu \times \{1\})$. P correctly distinguishes u and other elements in the coset Hu only by chance. Since the order of H is at least 2, the probability that $C^P(G, f, 0^k)$ outputs CORRECT is at most 2^{-k}.

Clearly, $C^P(G, f, 0^k)$ runs in polynomial time. The nonadaptiveness follows from Theorem 2.

Similarly, we can construct a nonadaptive checker $C^P(G, f, 0^k)$ for a program $P(G, f)$ that solves Hidden Subgroup over permutation groups. The checker makes k nonadaptive queries.

Begin
Run $P(G, f)$, which outputs a generating sets S.
Verify that elements of S are indeed in H.
Do k times (in parallel):
generate a random element $u \in G$.
define f' over G such that $f(g) = f'(gu)$ for all $g \in G$, use (G, f, f') to be an input instance of Hidden Coset
use Theorem 1 to convert (G, f, f') to an input instance $(G \wr \mathbb{Z}_2, f'')$ of Hidden Subgroup
$P(G \wr \mathbb{Z}_2, f'')$ will output a set S' of generators and a coset representative u'
if S and S' don't generate the same group **or** u and u' are not in the same coset of S, **then**
return BUGGY
End-do
return CORRECT
End

The proof of correctness for the above checker is similar to the proof of Theorem 5.

6 Discussion

The possibility of achieving exponential quantum savings is closely related to the underlying algebraic structure of a problem. Some researchers argued that generalizing from abelian groups to nonabelian groups may be too much for quantum computers [qua], since nonabelian groups exhibit radically different characteristics comparing with abelian groups. The results in this paper seem to support this point of view. Given the rich set of nonabelian group families, such as wreath product groups, Hidden Subgroup is indeed a "robust" and difficult problem. Recently Childs, Schulman, and Vazirani [CSV07] suggested an alternative generalization of abelian Hidden Subgroup, namely to the problems of finding nonlinear structures over finite fields. They gave two such problems, Hidden Radius and Hidden Flat of Centers, which exhibit exponential quantum speedup. An interesting open problem is whether these two problems are instances of Hidden Subgroup over some nonabelian group families.

Acknowledgments

We thank Andrew Childs and Wim van Dam for valuable comments on a preliminary version of this paper. We also thank anonymous referees for helpful suggestions.

References

[AK02] Arvind, V., Kurur, P.P.: Graph Isomorphism is in SPP. In: Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science, IEEE, New York (2002)

[AT01] Arvind, V., Torán, J.: A nonadaptive NC checker for permutation group intersection. Theoretical Computer Science 259, 597–611 (2001)

[BK95] Blum, M., Kannan, S.: Designing programs that check their work. Journal of the ACM 42(1), 269–291 (1995)

[CSV07] Childs, A.M., Schulman, L.J., Vazirani, U.V.: Quantum algorithms for hidden nonlinear structures. In: FOCS 2007: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), pp. 395–404. IEEE Computer Society, Washington (2007)

[CvD07] Childs, A.M., van Dam, W.: Quantum algorithm for a generalized hidden shift problem. In: SODA 2007: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1225–1232 (2007)

[EH00] Ettinger, M., Høyer, P.: On quantum algorithms for noncommutative hidden subgroups. Advances in Applied Mathematics 25, 239–251 (2000)

[EHK04] Ettinger, M., Høyer, P., Knill, E.: The quantum query complexity of the hidden subgroup problem is polynomial. Information Processing Letters 91(1), 43–48 (2004)

[FHL80] Furst, M.L., Hopcroft, J.E., Luks, E.M.: Polynomial-time algorithms for permutation groups. In: Proceedings of the 21st IEEE Symposium on Foundations of Computer Science, pp. 36–41 (1980)

[FIM$^+$03] Friedl, K., Ivanyos, G., Magniez, F., Santha, M., Sen, P.: Hidden translation and orbit coset in quantum computing. In: Proceedings of the 35th ACM Symposium on the Theory of Computing, pp. 1–9 (2003)

[GSVV04] Grigni, M., Schulman, J., Vazirani, M., Vazirani, U.: Quantum mechanical algorithms for the nonabelian hidden subgroup problem. Combinatorica 24(1), 137–154 (2004)

[HMR$^+$06] Hallgren, S., Moore, C., Rötteler, M., Russell, A., Sen, P.: Limitations of quantum coset states for graph isomorphism. In: STOC 2006: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, pp. 604–617 (2006)

[HRTS03] Hallgren, S., Russell, A., Ta-Shma, A.: The hidden subgroup problem and quantum computation using group representations. SIAM Journal on Computing 32(4), 916–934 (2003)

[Joz00] Jozsa, R.: Quantum factoring, discrete algorithm and the hidden subgroup problem (manuscript, 2000)

[Kit95] Kitaev, A.Y.: Quantum measurements and the Abelian Stabilizer problem, quant-ph/9511026 (1995)

[KS05] Kempe, J., Shalev, A.: The hidden subgroup problem and permutation group theory. In: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms, Vancouver, British Columbia, January 2005, pp. 1118–1125 (2005)

[Mat79] Mathon, R.: A note on the graph isomorphism counting problem. Information Processing Letters 8, 131–132 (1979)

[Mos99] Mosca, M.: Quantum Computer Algorithms. PhD thesis, University of Oxford (1999)

[MRS05] Moore, C., Russell, A., Schulman, L.J.: The symmetric group defies strong fourier sampling. In: FOCS 2005: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, pp. 479–490 (2005)

[MRS07] Moore, C., Russell, A., Sniady, P.: On the impossibility of a quantum sieve algorithm for graph isomorphism. In: STOC 2007: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, pp. 536–545 (2007)

[NC00] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)

[qua] Quantum pontiff, `http://dabacon.org/pontiff/?p=899`

[Reg04] Regev, O.: Quantum computation and lattice problems. SIAM Journal on Computing 33(3), 738–760 (2004)

[Sco87] Scott, W.R.: Group Theory. Dover Publications, Inc. (1987)

[Sim70] Sims, C.C.: Computational methods in the study of permutation groups. In: Computational problems in abstract algebra, pp. 169–183 (1970)

[vDHI03] van Dam, W., Hallgren, S., Ip, L.: Quantum algorithms for some hidden shift problems. In: Proceedings of the 14th annual ACM-SIAM symposium on Discrete algorithms, pp. 489–498 (2003)