Quantum Solution to the Hidden Subgroup Problem for Poly-Near-Hamiltonian Groups

Dmitry Gavinsky
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada, T2N 1N4
e-mail: gavinsky@cpsc.ucalgary.ca

Abstract

The *Hidden Subgroup Problem (HSP)* has been widely studied in the context of quantum computing and is known to be efficiently solvable for Abelian groups, yet appears to be difficult for many non-Abelian ones. An *efficient* algorithm for the HSP over a group G runs in time polynomial in $n \overset{\mathrm{def}}{=} \log |G|$.

For any subgroup H of G, let $N(H)$ denote the normalizer of H. Let M_G denote the intersection of all normalizers in G (i.e., $M_G = \cap_{H \leq G} N(H)$). M_G is always a subgroup of G and the index $[G : M_G]$ can be taken as a measure of "how non-Abelian" G is ($[G : M_G] = 1$ for Abelian groups). This measure was considered by Grigni, Schulman, Vazirani and Vazirani, who showed that whenever $[G : M_G] \in \exp(O(\log^{1/2} n))$ the corresponding HSP can be solved efficiently (under certain assumptions).

We show that whenever $[G : M_G] \in \mathrm{poly}(n)$ the corresponding HSP can be solved efficiently, under the same assumptions (actually, we solve a slightly more general case of the HSP and also show that some assumptions may be relaxed).

1 Introduction

The *Hidden Subgroup Problem (HSP)* has been studied by several authors in the context of Quantum Computing (e.g., see [B97], [EH99], [EHK99], [HRT00] and [GSVV01]).

The problem is defined as follows. Suppose we are given a (quantum) oracle for a function f, defined over a group G and satisfying the following: over the elements of each left coset of the *hidden subgroup* $H \leq G$, f is constant; on the other hand, f is distinct between elements of distinct cosets of H.

To solve the HSP means to construct a generating set for H (being provided with access to an oracle computing f). A number of efficient quantum algorithms are known which explicitly solve problems other than the HSP but at the same time may be viewed as algorithms for special cases of the HSP (e.g., see [S97], [S97a] and [K95]). On the other hand, Friedl, Ivanyos, Magniez, Santha and Sen [FIMSS03] have shown a quantum solution to the HSP for some groups.

The HSP is known to be solvable on a quantum machine in the case of Abelian G. The known quantum algorithm solving the HSP for such G is based on the Quantum Fourier Transform (QFT) over G.[1]

[1] An efficient quantum solution to the HSP for Hamiltonian groups has been given by Hallgren, Russell and Ta-Shma in [HRT00]; in particular, they show that the QFT for such groups is tractable. Efficient circuits performing the QFT for symmetric groups have been given by Beals [B97]. More recently, Moore, Rockmore and Russell [MRR03] have shown that the QFT may be performed efficiently for some other groups.

As a possible extension of the Abelian case, Grigni, Schulman, Vazirani and Vazirani [GSVV01] give a solution to the case of *almost Abelian* groups, defined as follows.[2] For any $H \leq G$ we denote the *normalizer* of H by:

$$N(H) \overset{\mathrm{def}}{=} \left\{ g \in G \,\middle|\, gHg^{-1} = H \right\}.$$

We define M_G to be the intersection of the normalizers:

$$M_G \overset{\mathrm{def}}{=} \bigcap_{H \leq G} N(H).$$

Set a parameter k_G:

$$k_G \overset{\mathrm{def}}{=} [G : M_G].$$

According to [GSVV01], G is called *almost Abelian* if $k_G \in \exp(O(\log^{1/2} n))$, for $n = \log |G|$.

We extend the notation and say that any group G is *k_G-near-Hamiltonian* (where k_G is the corresponding characteristic of G); in particular, G is *(poly-) near-Hamiltonian* if $k_G \in poly(n)$. [3]

1.1 Our results

We give an efficient quantum solution to the HSP for the case of near-Hamiltonian groups. Our final algorithm solves even a more general case, when the hidden subgroup H satisfies: $[G : HM_G] \in poly(n)$.

The paper is organized as follows. In Section 3 we construct an algorithm which efficiently solves the HSP for the case of near-Hamiltonian groups. Then in Section 4 we generalize our approach and construct an algorithm which efficiently solves the case when $[G : HM_G]$ is polynomially bounded.

2 Notation and Assumptions

For the constructions of this paper, all the subgroups are represented by (arbitrary) generating sets. We assume that under such representation subgroup membership can be efficiently tested for any group element (this assumption is common in the context of the HSP).[4]

Let A, B be groups; we will use the notation $A \leq B$ to express that A is a subgroup of B; by $A \trianglelefteq B$ we will mean that A is normal in B. If S is a subset of the elements of the group B, by $\langle S \rangle$ we mean the closure of S in B.

We denote by $\mathit{NHS}(A)$ the "standard" (e.g., see [GSVV01]) algorithm solving the normal HSP for the group A. The routine $\mathit{NHS}(A)$ has access to an oracle computing a function f, defined over the elements of A. If the function assigns values according to a normal hidden subgroup H of A then $\mathit{NHS}(A)$ produces a generating set for H (by $\mathit{NHS}(A)$ we will mean a generating set for $\langle \mathit{NHS}(A) \rangle$). For successful execution of $\mathit{NHS}(A)$ one has to be able to efficiently compute the QFT over A, as well as to find a generating set for the intersection of the kernels of given representations of A.

For simplicity we assume that when a solution is not found by $\mathit{NHS}(A)$ it returns a "failure" message. (If the oracle answers according to some non-normal H, the algorithm either finds generators of some subgroup of H or returns a "failure" message.) In fact, the known algorithm has exponentially low probability to fail; however, since all our algorithms use $\mathit{NHS}(A)$ as a subroutine at most polynomial

[2] Their algorithm requires efficient QFT over G, as well as some other conditions which we discuss later.

[3] A group is called *Hamiltonian* if it possesses only normal subgroups.

[4] In particular, Watrous in [W01] shows that subgroup membership for solvable groups can be efficiently tested on a quantum computer. On the other hand, membership testing for subgroups of symmetric groups is efficient on a classical computer (see Luks [L93]).

number of times, our assumption of $\mathit{NHS}(A)$ correctness is not held with only exponentially small probability.

For a quotient group G/N the elements are the left cosets of N in G. For convenience we will use the following shorthand notation: for any $x \subseteq G/N$,

$$\mathbf{U}_{G/N}(x) \overset{def}{=} \bigcup_{t \in \langle x \rangle} t$$

(note that $\mathbf{U}_{G/N}(x) \leq G$).

3 HSP for Near-Hamiltonian Groups

In this section we solve the HSP for near-Hamiltonian groups (i.e., $[G : M_G] \in \mathit{poly}(n)$).

It is shown in [GSVV01] that M_G (as defined above) is a normal subgroup of G. Let us consider the quotient group G/M_G.

Recall that our goal is to determine H. Since M_G is defined as the intersection of all the normalizers,

$$H \trianglelefteq HM_G. \tag{1}$$

At the same time it holds that

$$HM_G/M_G \leq G/M_G,$$

and if we could determine HM_G/M_G, that would give us a way to find HM_G :

$$HM_G = \cup_{t \in HM_G/M_G} t,$$

and then to find H, since (1) means that

$$H = \langle \mathit{NHS}(HM_G) \rangle .$$

We claim that given any $x \in G/M_G$, it is possible to check whether $x \in HM_G/M_G$, as follows. If $x \in HM_G/M_G$ then

$$\langle \{x\} \rangle \leq HM_G/M_G,$$

and

$$H \cap \mathbf{U}_{G/M_G}(\{x\}) \trianglelefteq \mathbf{U}_{G/M_G}(\{x\}) \leq HM_G,$$

which follows from (1). An execution of $\mathit{NHS}(\mathbf{U}_{G/M_G}(\{x\}))$ would return a generating set for

$$H_0 = H \cap \mathbf{U}_{G/M_G}(\{x\}).$$

Moreover, since $x = hM_G$ for some $h \in H$, it holds that $h \in H_0 \cap x$, and therefore

$$H_0 \cap x \neq \emptyset.$$

On the other hand, if $x \notin HM_G/M_G$ then $\mathit{NHS}(\mathbf{U}_{G/M_G}(\{x\}))$ would either fail or return a generating set for

$$H_0 = H \cap \mathbf{U}_{G/M_G}(\{x\}).$$

If the algorithm is successful, then

$$H_0 \cap x \subseteq H \cap x = \emptyset,$$

and this allows to distinguish the two cases.

Our assumption is $|G/M_G| \in poly(n)$, therefore we can efficiently check (probabilistically, with exponentially small probability to err) for any H_0 whether or not its intersection with a given coset x of M_G is empty. Note that the nonempty intersections of a subgroup H_0 with cosets of M_G are all of the same size (since M_G is normal). Therefore, a random element taken from H_0 has a probability not smaller than $1/|G/M_G|$ to belong to the coset x, if $H_0 \cap x \neq \emptyset$. As we assume that membership testing is efficient for subgroups of G, we can perform the whole test efficiently.

Based on this probabilistic check procedure, it is possible to efficiently construct HM_G/M_G and thus to solve the HSP. The algorithm $HS_1(G)$ for finding H follows in Figure 1.

```
HS_1(G)
1.  set: $T = \emptyset$
2.  for each $x \in G/M_G$ do
3.      set: $U = \mathbf{U}_{G/M_G}(\{x\})$
4.      if $NHS(U)$ is successful then
5.          if $\langle NHS(U) \rangle \cap x \neq \emptyset$ then
6.              set: $T = T \cup \{x\}$
7.  end-for
8.  return $NHS(\cup_{t \in T} t)$
```

Figure 1: Algorithm $HS_1(G)$ for solving the HSP for a near-Hamiltonian group G.

This algorithm uses the conditions in lines 4 and 5 in order to determine whether the given x belongs to HM_G/M_G, the correctness of such verification follows from the previous discussion.

3.1 Assumptions

In order to construct $HS_1(G)$ we have made some assumptions (similar to those made in [GSVV01]):

1. We assume that for any $U = \mathbf{U}_{G/M_G}(\{x\})$ (where $x \in G/M_G$), we can efficiently execute $NHS(U)$; in particular, this means to be able to efficiently perform the corresponding Quantum Fourier Transform $QFT(U)$ and to find a generating set for the intersection of the kernels of given representations of U.

2. We assume that we can efficiently iterate through all $x \in G/M_G$.

The latter assumption may be removed easily; the algorithm given in Section 4 solves a more general case of the HSP and does not iterate through G/M_G.

4 Finding the Hidden Subgroup H, When $[G : HM_G] \in poly(n)$

Denote by d an upper bound on $[G : HM_G]$ (which is polynomial in n).

Since $[G : HM_G] = [G/M_G : HM_G/M_G]$, by picking a random element from G/M_G we would get with probability at least $1/d$ an element from HM_G/M_G. We know that during the execution of HS_1 it holds that $T \subseteq G/M_G$ and

$$\langle T \rangle \leq HM_G/M_G;$$

as long as $\langle T\rangle \neq HM_G/M_G$, a random element from G/M_G comes from $(HM_G/M_G) \setminus \langle T\rangle$ with probability at least $1/2d$ (since $[HM_G/M_G : \langle T\rangle] \geq 2$).

In other words, as long as $\langle T\rangle \neq HM_G/M_G$, an element uniformly at random chosen from G/M_G belongs to HM_G/M_G and, being added to T, extends $\langle T\rangle$ with probability at least $1/2d$. Since every such extension of $\langle T\rangle$ at least doubles its size, it may happen at most n times before $\langle T\rangle = HM_G/M_G$.

Applying a Chernoff bound, we see that $4nd\ln(1/\delta)$ mutually independent uniform random samplings from G/M_G suffice in order to ensure the success probability of at least $1-\delta$. The algorithm (denoted by $HS_2(G, d, \delta)$) is given in Figure 2.

```
$HS_2(G, d, \delta)$
 1.  set: $T = \emptyset$
 2.  set: $n = \log|G|$
 3.  repeat $4nd\ln(1/\delta)$ times
 4.      draw uniformly: $x \in G/M_G$
 5.      set: $U = \mathbf{U}_{G/M_G}(\{x\})$
 6.      if $NHS(U)$ is successful then
 7.          if $\langle NHS(U)\rangle \cap x \neq \emptyset$ then
 8.              set: $T = T \cup \{x\}$
 9.  end-repeat
10.  return $NHS(\mathbf{U}_{G/M_G}(T))$
```

Figure 2: Algorithm $HS_2(G, d, \delta)$ for solving the HSP with probability at least $1-\delta$ when $[G : HM_G] \leq d$.

One technical difficulty arises in this case: how can we check for a given M_G's coset x, whether its intersection with

$$H_0 = \langle NHS(U)\rangle$$

is nonempty? (In HS_1 we used the fact that $|G/M_G| \in poly(n)$ in order to perform this check; this assumption is no longer valid.)

In general, the problem of *Coset Intersection* is often considered hard. However, in our case x is a coset of the normal subgroup M_G. Denoting $x = aM_G$, we establish the following:

$$\begin{aligned}
&H_0 \cap aM_G \neq \emptyset \iff \\
&\exists h \in H_0 :\ h \in aM_G \iff \\
&\exists h \in H_0 :\ a^{-1} \in M_G h^{-1} \iff \\
&a^{-1} \in M_G H_0 \iff \\
&a \in H_0 M_G.
\end{aligned}$$

The fact that M_G is normal means that $H_0 M_G$ is a subgroup, and therefore we may construct a generating set for $H_0 M_G$ by taking the union of the (known) generating sets for H_0 and for M_G.[5]

So, the condition check in line 7 of HS_2 may be performed efficiently and the whole algorithm is efficient.

[5] In general a "pointwise product" of two non-normal subgroups is not a subgroup; the union of two generating sets generates the *closure* of the subgroups' product.

4.1 Assumptions

In order to construct $HS_2(G,d,\delta)$ we have made only the assumption that for any $U = \mathbf{U}_{G/M_G}(\{x\})$ (where $x \in G/M_G$), we can efficiently execute $NHS(U)$.

5 Acknowledgments

I express my appreciation to Richard Cleve for valuable comments and suggestions.

References

[B97] R. Beals. Quantum computation of Fourier transforms over symmetric groups. *Proceedings of the 29th Symposium on Theory of Computing, pp. 48-53*, 1997.

[EH99] M. Ettinger and P. Hoyer. On quantum algorithms for noncommutative hidden subgroups. *Proceedings of the 16th Symposium on Theoretical Aspects of Computer, pp. 478-487*, 1999.

[EHK99] M. Ettinger, P. Hoyer and E. Knill. Hidden subgroup states are almost orthogonal. *http://arxiv.org, quant-ph/9901034*, 1999.

[FIMSS03] K. Friedl, G. Ivanyos, F. Magniez, M. Santha and P. Sen. Hidden Translation and Orbit Coset in Quantum Computing. *Proceedings of the 35th Symposium on Theory of Computing, pp. 1-9*, 2003.

[GSVV01] M. Grigni, L. J. Schulman, M. Vazirani and U. V. Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem. *Proceedings of the 33th Symposium on Theory of Computing, pp. 68-74*, 2001.

[HRT00] S. Hallgren, A. Russell and A. Ta-Shma. Normal subgroup reconstruction and quantum computation using group representations. *Proceedings of the 32nd Symposium on Theory of Computing, pp. 627-635*, 2000.

[K95] A. Yu. Kitaev. Quantum measurements and the Abelian stabilizer problem. *http://arxiv.org, quant-ph/9511026*, 1995.

[L93] E. M. Luks. Permutation groups and polynomial-time computation. *Groups and Computation, DIMACS series in Discrete Mathematics and Theoretical Computer Science 11, pp. 139-175*, 1993.

[MRR03] C. Moore, D. Rockmore and A. Russell. Generic Quantum Fourier Transforms. *http://arxiv.org, quant-ph/0304064*, 2003.

[S97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing 26(5), pp. 1484-1509*, 1997.

[S97a] D. R. Simon. On the power of quantum computation. *SIAM Journal on Computing 26(5), pp. 1474-1483*, 1997.

[W01] J. Watrous. Quantum algorithms for solvable groups. *Proceedings of the 33rd Symposium on Theory of Computing, pp. 60-67*, 2001.