The hidden subgroup problem for infinite groups

Greg Kuperberg, UC Davis

We consider the hidden subgroup problem for infinite groups, beyond the celebrated original cases established by Shor and Kitaev.

1 Motivation and main results

Some of the most important quantum algorithms solve rigorously stated computational problems and are superpolynomially faster than classical alternatives. This includes Shor's algorithm for period finding [13]. The hidden subgroup problem is a popular framework for many such algorithms [10]. If G is a discrete group and X is an unstructured set, a function $f : G \to X$ *hides* a subgroup $H \le G$ means that $f(x) = f(y)$ if and only if $y = xh$ for some $h \in H$. In other words, f is a *hiding function* for the *hidden subgroup* H when it is right H-periodic and otherwise injective, as summarized in this commutative diagram:

$$\begin{tikzcd} G \arrow[rr, "f"] \arrow[dr] & & X \\ & G/H \arrow[ur, hook] & \end{tikzcd}$$

The *hidden subgroup problem* (HSP) is then the problem of calculating H from arithmetic in G and efficient access to f.

The computational complexity of HSP depends greatly on the ambient group G, as well as other criteria such as that H is normal or lies in a specific conjugacy class of subgroups. The happiest cases of HSP are those that both have an efficient quantum algorithm, and an unconditional proof that HSP is classically intractable when f is given by an oracle.

Shor's algorithm solves HSP when $G = \mathbb{Z}$, the group of integers under addition. Even though $\mathbb{Z}$ is infinite and Shor's algorithm is a major motivation for HSP, many of the results since then have been about HSP for finite groups [2, Ch. VII]. Here we examine HSP in some key cases when G is a discrete, infinite group, with the rule that the hidden subgroup H can be confirmed with polynomial query complexity. We obtain five main results for different types of G. In three cases, we obtain hardness results; in two others, we obtain algorithms. To state the hardness results properly, we define the hidden subgroup *existence* problem (HSEP) to be the decision problem of whether H is non-trivial.

Theorem 1. *Consider HSP in $\mathbb{Q}$, the rational numbers viewed as a discrete group under addition. Then it is NP-complete, with reduction in BQP, to determine whether the hidden subgroup $H \le \mathbb{Q}$ is larger than $\mathbb{Z}$. Equivalently, HSEP in the quotient $\mathbb{Q}/\mathbb{Z}$ is NP-complete.*

Theorem 1 is in contrast to both Shor's algorithm for $G = \mathbb{Z}$, as well as Hallgren's algorithm when $G = \mathbb{R}$ and the hiding function f is Lipschitz and takes values in a Hilbert space [3, 5]. Assuming the conjecture that $\mathsf{NP} \not\subseteq \mathsf{BQP}$, the theorem implies that HSP in $\mathbb{Q}$ has no efficient quantum algorithm.

Theorem 2. *HSEP in a finitely generated, non-abelian free group F_k is NP-complete, even assuming that the hidden subgroup is normal.*

Theorem 2 is in contrast to the case of finite groups. If G is finite and H is normal, then HSP is in BQP whenever the quantum Fourier transform on G has a polynomial time quantum algorithm [6].

The proofs of Theorems 1 and 2 both relativize. In relative form, we can show that HSP in these two cases has exponential query complexity.

Theorem 3. *Consider HSEP for the group $G = \mathbb{Z}^k$ where the hiding function f has pseudo-polynomial query cost. If this problem is in* BQP^f *uniformly in f and in the dimension k, then the unique short vector problem* uSVP *for integer lattices $L \leq \mathbb{Z}^k$ and polynomial parameters is also in* BQP.

Theorem 4. *HSP in $\mathbb{Z}^k$ with binary encoding of vectors can be solved in* BQP*, uniformly in the dimension k and in the bit complexity of the answer.*

Theorem 4 is in contrast to the Shor-Kitaev algorithm [7], which achieves the same thing with the crucial extra hypothesis that the hidden subgroup $H \leq \mathbb{Z}^k$ has full rank $\ell = k$, equivalently when H is a finite-index subgroup. The case when H has lower rank $\ell < k$ is a new result as far as we know.

Theorem 5. *Let G be a fixed, finitely generated group with a finite-index abelian subgroup $K \leq G$. Then HSP in G can be solved in time $2^{O(\sqrt{n})}$, where n is the bit complexity of the answer.*

The special case of HSP Theorem 5 is equivalent to the hidden shift problem in the abelian group K. In contrast to our other main results, Theorem 5 can be proven using existing algorithms. The result follows from the author's prior work on the dihedral hidden subgroup problem [8, 9]. The proof is much nicer using a refinement of the author's second algorithm recently obtained by Peikert [12].

2 Elements of some of the proofs

2.1 In the proof of Theorem 1

Given a decision problem $d(x)$ in NP, we assume a predicate $z(x,y)$ that accepts prime numbers $y = p$ as certificates. We let the hidden subgroup $H \leq \mathbb{Q}$ be generated by 1, and by $1/p$ for every accepted p. Our technique is to construct an H-periodic hiding function $f(a/b)$ that takes a rational number a/b as input, and that can also be efficiently computed using access to the predicate z. To do this, we use the fact that rational numbers have a canonical partial fraction form analogous to partial fractions for rational functions in calculus. For example,

$$\frac{1}{60} = -2 + \frac{1}{2} + \frac{1}{4} + \frac{2}{3} + \frac{3}{5}.$$

This partial fraction form requires factoring the denominator b, which is why the reduction is in BQP. (There is also a workaround using a classical algorithm for partial factorization, only it is conditional on conjectures in number theory.) The value of $f(a/b)$ is now given by striking the integer term, and striking every term whose denominator p is prime and accepted by z. H contains $\mathbb{Z}$ regardless, and $H \neq \mathbb{Z}$ if and only if there is an accepted certificate. Thus, the decision problem d reduces to HSP in $\mathbb{Q}$.

2.2 In the proof of Theorem 2

Since the hidden subgroup $H = N \leq F_k$ is normal, it is *normally generated* by a set of words R, meaning that N is generated by elements of R and their conjugates. The quotient group F_k/N is thus realized as the presented group $\langle A|R\rangle$, where $|A| = k$ is an alphabet. As in the proof of Theorem 1, we have a decision problem $d(x)$ in NP with a predicate $z(x,y)$, and we want to generate N by some encoding of the accepted certificates. We want an N-periodic hiding function $f(w)$ on words $w \in F_k$ that can be computed efficiently from z. We construct $f(w)$ as a canonical word for the element $[w] = wN$ in the quotient group F_k/N. In other words, we need an efficient solution to the word problem in the group $K = F_k/N$. Even though the word problem for finitely presented groups is RE-complete [1, 11], we rely on a restricted version which does have an efficient algorithm, even with only guess-and-check access to the relators.

We assume that $k = 14$ (strictly for simplicity), and we let F_{14} be generated by the alphabet

$$A = \{a_1, b_1, a_2, b_2, \ldots, a_7, b_7\}.$$

Taking each possible certificate y initially as a binary string, we interpret it as a word $y(a,b)$ in two letters a and b, and we assume the corresponding relator

$$r_y = y(a_1,b_1)y(a_2,b_2)\cdots y(a_7,b_7).$$

The resulting group presentation $\langle A|R\rangle$ satisfies the crucial $C'(1/6)$ hypothesis of Greendlinger [4], that any common substring of any two relators (as cyclic words) is less than $1/6$ of the length of each relator.

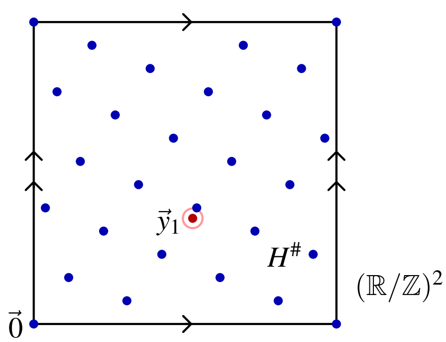
Greendlinger showed that if a word w represents the identity in a $C'(1/6)$ presented group, then it can be reduced to the trivial word with the greedy algorithm. If a relator r can be used to greedily shorten w, then it must share more than half of its length with w. Using the special form of our relators $r = r_y$, r_y can thus be calculated with a guess-and-check procedure using the predicate z. Using a refinement of Greendlinger's algorithm, we can also calculate the shortlex equivalent $v \sim w$ of any word w in polynomial time. In the refined algorithm, each relator r is only invoked when at least $1/6$ of its length has already been computed. If $r = r_y$, then we can again confirm y with the predicate z.

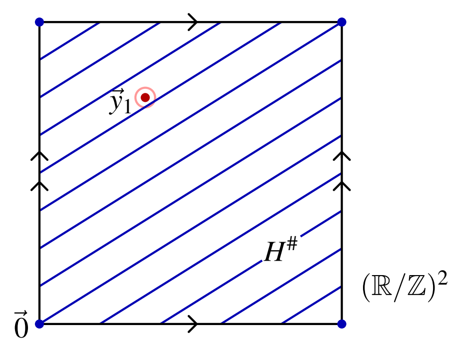
2.3 In the proof of Theorem 4

Our algorithm for HSP in $\mathbb{Z}^k$ has a standard quantum stage:

1. Prepare an approximate Gaussian state $|\psi_G\rangle$ on a finite box in $\mathbb{Z}^k$.
2. Evaluate f to make $U_f|\psi_G\rangle$, and measure the output register to obtain a coset state $|\psi_{H+\vec{v}}\rangle$.
3. Apply the QFT operator $F_{(\mathbb{Z}/Q)^k}$ to $|\psi_{H+\vec{v}}\rangle$ and measure a Fourier mode $\vec{y}_0 \in (\mathbb{Z}/Q)^k$.

The rescaled Fourier mode $\vec{y}_1 = \vec{y}_0/Q \in (\mathbb{R}/\mathbb{Z})^k$ approximates a random element of the dual group $H^\# \leq (\mathbb{R}/\mathbb{Z})^k$, which consists of all $\vec{y}$ such that $\vec{x}\cdot\vec{y} \in \mathbb{Z}$ for all $\vec{x} \in H$. If H has full rank, then $H^\#$ is a finite group, and the Shor-Kitaev algorithm denoises each coordinate of $\vec{y}_1$ using continued fractions. If H has lower rank, then $H^\#$ is a pattern of stripes in $(\mathbb{R}/\mathbb{Z})^k$, and this denoising step is not directly possible.





In our remedy, we calculate the connected subgroup $H_1^\#$ from a single sample $\vec{y}_1$ with high probability, when there is a little enough noise. In this case, $\vec{y}_1$ has multiples that land close enough to $\vec{0}$ to reveal the tangent directions to $H^\#$. We can find useful multiples of $\vec{y}_1$ this type using the LLL algorithm. Given an approximate basis of tangent directions to $H^\#$, we can put its matrix B_1 in RREF form using some careful linear algebra to bound the noise in the matrix entries. We can then apply the continued fraction algorithm to remove the noise and learn $H_1^\#$. Finally the dual $H_1 \leq \mathbb{Z}^k$ of $H_1^\#$ itself contains H as a full-rank subgroup, and we can finish the algorithm using Shor-Kitaev.

References

[1] William W. Boone, *The word problem*, Ann. of Math. (2) **70** (1959), 207–265.

[2] Andrew M. Childs and Wim van Dam, *Quantum algorithms for algebraic problems*, Rev. Modern Phys. **82** (2010), no. 1, 1–52, `arXiv:0812.0380`.

[3] Kirsten Eisenträger, Sean Hallgren, Alexei Kitaev, and Fang Song, *A quantum algorithm for computing the unit group of an arbitrary degree number field*, STOC '14— Proceedings of the 2014 ACM Symposium on Theory of Computing, ACM, New York, 2014, pp. 293–302.

[4] Martin Greendlinger, *Dehn's algorithm for the word problem*, Comm. Pure Appl. Math. **13** (1960), 67–83.

[5] Sean Hallgren, *Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem*, J. ACM **54** (2007), no. 1, Art. 4, 19 pp.

[6] Sean Hallgren, Alexander Russell, and Amnon Ta-Shma, *Normal subgroup reconstruction and quantum computation using group representations*, ACM Symposium on Theory of Computing, 2000, pp. 627–635.

[7] Alexei Kitaev, *Quantum measurements and the Abelian stabilizer problem*, 1995, `arXiv:quant-ph/9511026`.

[8] Greg Kuperberg, *A subexponential-time quantum algorithm for the dihedral hidden subgroup problem*, SIAM J. Comput. **35** (2005), no. 1, 170–188, `arXiv:quant-ph/0302112`.

[9] ———, *Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem*, 8th Conference on the Theory of Quantum Computation, Communication and Cryptography **22** (2013), 20–34, `arXiv:1112.3333`.

[10] Michele Mosca and Artur Ekert, *The hidden subgroup problem and eigenvalue estimation on a quantum computer*, Quantum computing and quantum communications (Palm Springs, CA, 1998), Lecture Notes in Comput. Sci., vol. 1509, Springer, Berlin, 1999, `arXiv:quant-ph/9903071`, pp. 174–188.

[11] Pyotr S. Novikov, *On the algorithmic unsolvability of the word problem in group theory*, Trudy Mat. Inst. im. Steklov., vol. 44, Izdat. Akad. Nauk SSSR, Moscow, 1955.

[12] Chris Peikert, *He gives c-sieves on the CISDH*, 2019, `CryptologyePrintArchive,Report2019/725`.

[13] Peter W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput. **26** (1997), no. 5, 1484–1509, `arXiv:quant-ph/9508027`.