



# Zero sum subsequences and hidden subgroups

Muhammad Imran<sup>1</sup> · Gábor Ivanyos<sup>2</sup>

Received: 17 April 2023 / Accepted: 17 December 2023 / Published online: 5 January 2024  
© The Author(s) 2024

## Abstract

We propose a method for solving the hidden subgroup problem in nilpotent groups. The main idea is iteratively transforming the hidden subgroup to its images in the quotient groups by the members of a central series, eventually to its image in the commutative quotient of the original group, and then using an abelian hidden subgroup algorithm to determine this image. Knowing this image allows one to descend to a proper subgroup unless the hidden subgroup is the full group. The transformation relies on finding zero sum subsequences of sufficiently large sequences of vectors over finite prime fields. We present a new deterministic polynomial time algorithm for the latter problem in the case when the size of the field is constant. The consequence is a polynomial time exact quantum algorithm for the hidden subgroup problem in nilpotent groups having constant nilpotency class and whose order only have prime factors also bounded by a constant.

**Keywords** Hidden subgroup problem · Nilpotent group · Zero sum subsequence · Exact quantum algorithm

## 1 Introduction

The standard version of the hidden subgroup problem (HSP for short) is the following. Given a function  $f$  on the group  $G \rightarrow \{0, 1\}^r$  with the property that there is a subgroup  $H$  such that  $f(x) = f(y)$  if and only if  $x$  and  $y$  are in the same left coset of  $H$ , find the subgroup  $H$ . Perhaps Kitaev was the first who observed that Shor's factoring and

---

✉ Muhammad Imran  
muh.imran716@gmail.com

Gábor Ivanyos  
Gabor.Ivanyos@sztaki.hun-ren.hu

<sup>1</sup> Department of Algebra and Geometry, Budapest University of Technology and Economics, Egrý József u. 1, Budapest 1111, Hungary

<sup>2</sup> Institute for Computer Science and Control, Hungarian Research Network, Kende u. 13-17, Budapest 1111, Hungary

discrete logarithm algorithms can be generalized to solve the HSP in finite abelian groups (and also in certain infinite commutative groups) in polynomial time. Much less is known about the complexity of the problem in non-commutative groups. The most general result is due to Ettinger, Hoyer and Knill. They showed in [1] that the *query complexity* of the problem in finite not necessarily abelian groups is polynomial. Regarding the time complexity, Kuperberg's subexponential time quantum algorithm [2] for the HSP in dihedral and very similar groups is perhaps the best known result. It has a remarkable extension by Alagic et al. [3] to a special HSP in a class including non-solvable groups. There are some classes of groups in which the HSP can be solved in polynomial time. See the survey papers by Lomont [4] and by Wang [5] for early results of this kind. The paper [6] by Lomonaco and Kauffman proposes interesting derivatives and generalizations of the Shor-Kitaev algorithm. The paper [7] by Horan and Kahrobaei discusses cryptographic aspects of the HSP and reports also on more recent results. The hidden shift problem in abelian groups (and hence the HSP in the related semidirect product groups) appears to be quite popular in post-quantum cryptography, see, e.g., [8] by Castryck and Vander Meer and [9] by Alagic and Russell. In [10], Bae and Lee propose a polynomial time solution to a continuous version of the hidden shift problem.

A quantum procedure is exact if it returns a correct output (after a final measurement) with probability one. Besides that exact quantum algorithms can be considered as counterparts of deterministic classical methods, their measurement-free versions can serve as ingredients of larger unitary procedures. A further possible practical advantage is that, as they do not introduce further errors, using them can help control the error of a procedure working an input with errors and also containing other, non-exact ingredients.

The method of [1] has an exact version, so it is natural to ask that in which classes of groups can the HSP be solved by an exact quantum algorithm in polynomial time. Brassard and Hoyer [11] presented a polynomial time exact method that works in  $\mathbb{Z}_2^n$ . In [12], Cai and Qiu proposed a simpler efficient exact method for Simon's problem (a special, though arguably the hardest instance of the HSP in  $\mathbb{Z}_2^n$ ). Efficient exact algorithms with optimal query complexity for the HSP in  $\mathbb{Z}_2^n$  appeared independently in [13] by Bonnetain and in [14] by Wu et al. Mosca and Zalka in [15] proposed an efficient exact solution of the discrete logarithm problem in cyclic groups of known order. An exact quantum algorithm for the HSP in  $\mathbb{Z}_{m^k}^n$  for general  $m$  was presented recently in [16], settling the case of abelian groups under the assumption that a multiple of the prime factors of the order of the group is known.

In this paper we present an approach to solving the hidden subgroup problem in nilpotent groups that have nilpotency class  $O(1)$ . Our main result is a polynomial time exact quantum algorithm for the HSP in such groups only having prime factors also of size  $O(1)$  in their order. An example for such groups is the group of the  $4n$  by  $4n$  block unitriangular matrices over the integers modulo  $10^d$  with block size  $n$ . These are the  $4n$  by  $4n$  matrices divided into 16 rectangular blocks of size  $n$  with integer entries, considered modulo  $10^d$ , whose 4 diagonal blocks are equal to the  $n$  by  $n$  identity matrix, whereas the 6 blocks below the diagonal have zero entries. These matrices can be represented by  $6dn^2$  decimal digits describing the entries of the blocks above the

diagonal. The order of the group is  $10^{6dn^2}$ , and it is nilpotent of class 3. To obtain sufficiently general results, we assume that the group  $G$  is given as a black-box group with unique encoding. The main strategy of our algorithm is essentially a reduction to instances of the hidden subgroup problem in quotient groups of subgroups of  $G$ . We choose an input model suitable for such a reduction.

In the standard version, the input is given by an oracle which is a unitary map computing  $|x\rangle|f(x)\rangle$  from  $|x\rangle|0\rangle$ . Many important applications belong to the case when the group  $G$  acts as a permutation group on a (large) set  $\Omega$ , and we are interested in computing the stabilizer of an element  $\omega \in \Omega$ . In that case, the stabilizer is the subgroup hidden by the function  $f : G \rightarrow \Omega$ , where  $x \mapsto x\omega$  the action of  $x$  on  $\omega$ . In fact, Kitaev used this framework to generalize Shor's algorithms.

The usual hidden subgroup algorithms start with computing the superposition  $\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle|f(x)\rangle$  using the oracle, and most of them ignore the second register that holds the value of  $f$  and work with the coset superpositions  $|xH\rangle = \frac{1}{\sqrt{|H|}} \sum_{y \in H} |xy\rangle$  in the sequel, see e.g., [6]. These methods, as noted in [2], remain applicable in the context where the oracle is assumed to generate copies of a mixture of the coset superpositions. This holds in particular in the case of the exact abelian hidden subgroup algorithm of [16].

Specifically, we consider the state  $\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle|f(x)\rangle$  as a purification of the mixed state  $\Xi_{G,H} = \frac{1}{|G:H|} \sum_{x \in X} |xH\rangle\langle xH|$ , where  $X$  is any left transversal of  $H$  in  $G$ , in order to have a unitary oracle. The state  $\Xi_{G,H}$  is referred to as a (hidden) subgroup state. We assume that our hidden subgroup  $H$  is given by a unitary map (referred as oracle) that, on zero input, returns a copy of an *arbitrary* (though fixed) purification of the subgroup state  $\Xi_{G,H}$ .

It will be convenient to introduce a subtask of the HSP, namely computing the hidden subgroup modulo the commutator subgroup of  $G$ , that is, the subgroup  $HG'$  where  $H$  is the hidden subgroup. We use the shorthand HSMC for this problem. To illustrate the power of HSMC in nilpotent groups note that it naturally includes the commutative case of the HSP and that having computed the subgroup  $HG'$  and it is a proper subgroup of  $G$ ; then, we can descend to it to compute  $H$ , while if  $HG' = G$  then  $H = G$  because in a nilpotent group every maximal subgroup contains the commutator.

We give a high-level description of a strategy for solving the problem HSMC in a class of nilpotent groups. We call a group  $G$  *semi-elementary* if  $G$  is a  $p$ -group for some prime  $p$  such that  $G/G'$  is elementary abelian. The Heisenberg group modulo  $p$ , that is, the group of  $3 \times 3$  upper triangular matrices having entries integers modulo  $p$  and 1s in the diagonal, is a semi-elementary  $p$ -group, as well as the groups of the unitriangular matrices of higher dimension groups modulo  $p$ , while the Heisenberg group modulo  $p^2$  is a  $p$ -group which is not semi-elementary. In a semi-elementary group  $G$ , our strategy for computing the hidden subgroup modulo the commutator is based on iterating the following procedure. Assume that  $L$  is an elementary abelian subgroup contained in the center of  $G$ . Then we create a copy of the subgroup state corresponding to  $HL/L$  in the quotient group  $G/L$  from sufficiently many copies of the subgroup state for  $H$  in  $G$ . We refer to this procedure (as well as some simpler ones) as *subgroup state conversion*. This conversion is based on finding zero sum

subsequences of sufficiently long sequences of elements of  $L$ . Eventually, in  $c - 1$  rounds of iteration, where  $c$  is the nilpotency class of  $G$ , we compute a copy of the subgroup state corresponding to  $HG'/G'$  in  $G$ . (The semi-elementary property ensures the existence of a standard central series of length  $c$  with elementary abelian factors.) Finally, from sufficiently many copies of such subgroup states we compute  $HG'/G'$  using the exact abelian hidden subgroup algorithm of [16]. Fortunately, semi-elementary groups occur as factor groups of subgroups of nilpotent groups frequently enough to make a reduction from the HSP to the special case of HSMC possible, see Proposition 1 for details. The main result we obtain is the following.

**Theorem 1** *Suppose that  $G$  is a nilpotent group of class bounded by a constant and that the prime factors of  $|G|$  are also bounded by a constant. We assume that  $G$  is a black-box group with unique encoding of elements by  $\ell$ -bit strings. Then there is an exact quantum algorithm that solves the hidden subgroup problem in  $G$  using  $\text{poly}(\ell)$  operations and  $\text{poly}(\log|G|)$  calls to the subgroup state creating oracle and its inverse.*

*Related results* In the exact setting, [16] efficiently solves the abelian case without the restriction on the prime factors of  $|G|$ . There are quite a few related non-exact polynomial-time algorithms. Among them, the result of [17], which solves the HSP in *solvable* groups that have derived series and exponent bounded by constants, is perhaps the closest to Theorem 1. This class of groups covers the groups for which our result is applicable, except those that have exponent divisible by large powers of small primes. Note however, that the case of these groups could be efficiently treated by a combination of the reduction of Proposition 1 with the algorithm of [17]. We remark that the semidirect product group in which the HSP is equivalent to the hidden shift problem over  $\mathbb{Z}_{2k}^n$  is a nilpotent group of class  $k$ . Bonnetain and Naya-Plasencia [18] propose a non-exact method whose main ingredient can be considered as a combination of Kuperberg's sieve with finding zero sum subsequences in  $\mathbb{Z}_2^n$  using linear algebra.

The case of nilpotency class at most two is efficiently treated by the non-exact method of [19], without any restriction on the size of the prime factors of  $|G|$ . It is worth mentioning that by technical content, [19] can be considered as the closest relative of the present paper. The idea of reducing the HSP to HSMC stems from there, and many ingredients of the reduction appeared in that paper. Also, the key tool of [19], using several coset superpositions and the quantum Fourier transform of a central subgroup can be considered as some (though less transparent) form of subgroup state conversion. In the class two case, however, there is a more powerful tool to cancel out characters of the subgroup: One can also apply twists with certain nice automorphisms of the group that do not change the hidden subgroup too much. Unfortunately, such automorphisms do not exist in general nilpotent groups of class greater than two.

The methods of [20, 21] offer efficient solution to the HSP in certain nilpotent groups of higher class, again with potentially large prime factors in their orders. These groups have a normal subgroup with an abelian factor group of restricted kind (e.g., cyclic). These methods as well as that of [17] are of highly non-exact nature. Probably, the technique of [20] can be made exact with some efforts.

The *Davenport constant*  $S(A)$  of a finite abelian group  $A$  is the smallest number  $s$  such that any sequence of  $s$  elements of  $A$  contains a non-empty subsequence adding up to the zero element of  $A$ . The name comes from that H. Davenport proposed

determining  $S(A)$  in the case when  $A$  is the ideal class group of a number field as a measure for non-uniqueness of factorization of the integers of the field. The general problem has become a famous question of additive combinatorics. Olson [22] determined the exact value of the Davenport constant of  $p$ -groups; in particular for  $\mathbb{Z}_p^n$  it is  $1 + n(p - 1)$ . What we are looking for is an “effective” Davenport constant: What is the smallest number  $S' = S^{\mathcal{B}}(A)$  such that from any sequence of  $S'$  elements of  $A$ , algorithm  $\mathcal{B}$  finds a non-empty zero sum subsequence in time polynomial in  $S' \log|A|$  (roughly this is the bit size of the input sequence). In this paper we give a deterministic algorithm  $\mathcal{B}$  running in time  $\text{poly}(n)$ , that, for  $p = O(1)$ , given a sequence of  $S^{\mathcal{B}}(\mathbb{Z}_p^n) = \text{poly}(n)$  vectors from  $\mathbb{Z}_p^n$ , returns a zero sum subsequence. Note that when we consider  $\mathbb{Z}_p$  as a field and  $\mathbb{Z}_p^n$  as an  $n$ -dimensional vector space then finding a zero sum subsequence is equivalent to represent zero as a non-trivial linear combination of the given vectors with coefficients 0 and 1 only. Alternatively, it is equivalent to finding a non-trivial solution of a system of  $n$  homogeneous linear congruences modulo  $p$  in  $S^{\mathcal{B}}(\mathbb{Z}_p^n)$  variables. Actually, our algorithm is based on elementary linear algebra: We find many linear dependencies in subsets of the given vectors and combine these relations in an iteration so that the large coefficients eventually cancel out.

The structure of the rest of the paper is the following. In Sect. 2, we give some background material on exact quantum procedures, on nilpotent black-box groups and on computations with them, on (hidden) subgroups states and their purifications and present methods to convert subgroup states in the entire group to those in subgroups and—in certain very easy cases—in factor groups. In Proposition 1, the existence of an exact polynomial time reduction from the HSP in general nilpotent groups to the problem HSMC in semi-elementary groups is proved in Sect. 3. Section 4 is devoted to converting several copies of a subgroup state in a semi-elementary group to a copy of a subgroup state in the abelian factor of the group. As an application of the technique, we prove Proposition 2 which tells us that we can solve by a polynomial time exact quantum algorithm the problem HSMC in a semi-elementary  $p$ -group of constant nilpotency class provided that we can find zero sum subsequences of sequences consisting of  $\text{poly}(n \log p)$  vectors from  $\mathbb{Z}_p^n$  in time  $\text{poly}(n \log p)$ . In Sect. 5, we prove Theorem 2 on efficiently solvability the latter task in the case when  $p$  is bounded by a constant. Propositions 1 and 2, together with Theorem 2, immediately imply Theorem 1. Section 6 is devoted to concluding remarks.

## 2 Preliminaries

### 2.1 On exact quantum computations

To obtain sufficiently general intermediate results, we use the model of uniform circuit families described by Nishimura and Ozawa [23]. This is because some of the exact methods of [16] as well as our main conversion technique work under the assumption that the quantum Fourier transforms and their inverses modulo the prime factors of  $|G|$  can be exactly implemented. As it is pointed out in [24], this task cannot be

accomplished using a fixed finite gate set. For the sake of transparency, we state our intermediate result using assumptions on availability of the quantum Fourier transforms rather than on gates required by the exact implementations of them. (See the implementation of the Fourier transform modulo general numbers proposed by Mosca and Zalka [15].) Note however that for the case of our Theorem 1, where these primes are assumed to be bounded by a constant, a constant number of gates are sufficient and hence, by [25], the theorem remains valid in the quantum Turing machine model of Bernstein and Vazirani [26].

## 2.2 Groups

For standard notations and concepts from group theory such as subgroups, normal subgroups, cosets, conjugates, commutators, commutator subgroup, center, etc., we refer the reader to the textbooks, e.g., to [27]. For subsets  $U$  and  $V$  of  $G$  we denote by  $UV$  the set  $\{uv : u \in U, v \in V\}$ . If both  $U$  and  $V$  are subgroups and either  $U$  or  $V$  is normal in  $G$  then  $UV$  is a subgroup. For subgroups  $U, V$ , by  $[U, V]$  we denote the *subgroup generated by the commutators*  $[u, v]$  ( $u \in U, v \in V$ ). Recall that the lower central series of a finite group  $G$  is the sequence  $G = G_0 > G_1 > \dots > G_c$  of normal subgroups  $G_i \triangleleft G$  recursively defined as  $G_i = [G, G_{i-1}]$ . Here we assume that  $c$  is the smallest index  $i$  such that  $G_i = [G, G_i]$ . The group  $G$  is nilpotent if  $G_c = \{1\}$  and then  $c$  is called the (nilpotency) class of  $G$ . A finite group is nilpotent if and only if it is the direct product of its Sylow subgroups.

To obtain sufficiently general results, we work over *black-box* groups with unique encoding of elements. The concept captures various “real” groups such as permutation groups and matrix groups over finite fields. Elements of a black-box group are represented by binary strings of a certain length  $\ell$  and the group operations are given by oracles and as input, a generating set for the group is given. Subgroups will also be given by sets of generators. One can use the exact polynomial time quantum membership test of [16] to reduce the size of generating sets to at most  $\log|G|$ .

During the rest of this part, we assume that  $G$  is a nilpotent black-box group of class  $c$  and the prime factors of  $|G|$  are known.

For a normal subgroup  $N$  of  $G$ , the subgroup  $[G, N]$  is a normal subgroup of  $G$  contained in  $N$ . If  $\Gamma$  and  $\Delta$  are sets of generators for  $G$  and  $N$ , respectively, a generating set for  $[G, N]$  can be obtained by taking the commutators  $[x, y]$  for  $x \in \Gamma, y \in \Delta$  and then adding iterated commutators with elements of  $\Gamma$  until the subgroup generated by the elements stabilizes. For testing stabilization, one can use the exact quantum subgroup membership algorithm of [16]. This gives a polynomial time exact method in particular to compute the lower central series.

Below we describe efficient solutions to some further group theoretic tasks that we use in our hidden subgroup algorithm. The  $p$ -Sylow subgroup of  $G$  can be computed as follows. Let  $\Gamma$  be a generating set for  $G$ . Then for each  $g \in \Gamma$  we compute the order  $o_g$  of  $g$  and decompose  $o_g$  as the product  $p^\alpha o_g'$  where  $o_g'$  is coprime with  $p$ . Then the  $g^{o_g'}$  ( $g \in \Gamma$ ) generate the (unique)  $p$ -Sylow subgroup of  $G$ . We shall compute hidden subgroups in  $G$  by computing the intersections with the Sylow subgroups. As an example, let us consider the Heisenberg group modulo 12. It consists of the

matrices of the form  $\begin{pmatrix} 1 & a & b \\ & 1 & c \\ & & 1 \end{pmatrix}$ , where  $a, b, c$  are integers taken modulo 12. (Here and throughout the paper entries of a matrix left empty stand for entries having value zero.) The 2-Sylow subgroup consists of the matrices whose off-diagonal entries are divisible by 3, while the matrices having off-diagonal entries divisible by 4 form the 3-Sylow subgroup. If the hidden subgroup  $H$  consists of matrices of the form  $\begin{pmatrix} 1 & d & 3e \\ & 1 & \\ & & 1 \end{pmatrix}$ , then its 2-Sylow subgroup is the set of matrices of the form  $\begin{pmatrix} 1 & 3g & 3h \\ & 1 & \\ & & 1 \end{pmatrix}$ , while the 3-Sylow consists of matrices of the form  $\begin{pmatrix} 1 & 4k \\ & 1 \\ & & 1 \end{pmatrix}$ .

The normalizer of a subgroup of  $G$  can be computed using the deterministic polynomial method of Kantor and Luks [28]. It was originally described for nilpotent permutation groups, but it also finds normalizers in any nilpotent black-box group of order having small prime factors only.

Assume that  $L$  is a subgroup of  $G$ . It will be useful to decompose elements  $x$  of  $G$  as products of the form  $\alpha_L(x)\beta_L(x)$  where  $\beta_L(x) \in L$  and  $\alpha_L(x)$  depend only on the coset  $xL$ . (Thus the range of  $\alpha_L$  is a transversal of  $L$  in  $G$ .) To this end, compute a chief series (a series of normal subgroups with cyclic factors of prime order)  $G = K_0 > K_1 > \dots > K_r = 1$ . Perhaps the easiest way to obtain such series is taking a refinement of the lower central series. By taking the subgroups  $K_i L$ , and removing repeated elements, we obtain a subnormal series  $G = M_0 > M_1 > \dots > M_s = L$  with cyclic factors of prime order. Also take elements  $a_i \in M_{i-1} \setminus M_i$  and denote by  $p_i$  the order of  $M_{i-1}/M_i$  ( $i = 1, \dots, s$ ). This generalizes the frequently used technique of linear algebra when, given a subspace  $U$  of a vector space  $V$ , we choose a basis of  $V$  consisting of a basis of  $U$  and further vectors. Then the elements  $a_1^{\gamma_1} a_2^{\gamma_2} \dots a_s^{\gamma_s}$  ( $(\gamma_1, \dots, \gamma_s) \in \prod_{i=1}^s \mathbb{Z}_{p_i}$ ) are a left transversal of  $L$  in  $G$ . For an element  $x \in G$ , the representative of the coset in this transversal can be computed as follows. First we find the smallest non-negative integer  $\gamma_1$  such that  $x a_1^{-\gamma_1} \in M_1$  by computing the base  $a_1$  discrete logarithm of  $x$  modulo  $M_1$ . This can be done by solving an instance of the hidden subgroup problem in  $\mathbb{Z}_{p_1}^2$ . Specifically, we define the function  $(\beta, \gamma) \mapsto |x^\gamma|_{a_1^{-\beta} M_1}$ . The function can be evaluated with the aid of computing the uniform superposition  $|M_1\rangle$  using the exact version [16] of Watrous’s method [29]. The values are  $p$  pairwise orthogonal states, and the hidden subgroup is  $\{(\delta, \gamma) : x^\delta a_1^{-\gamma} \in M_1\}$ . We use the exact hidden subgroup algorithm of [16] to find a generator of this group. From this,  $\gamma_1$  can be obtained in an obvious way. Now we proceed with  $x a_1^{-\gamma_1}$  to compute  $\gamma_2$ , and so on. We set  $\alpha_L(x) = a_1^{\gamma_1} \dots a_r^{\gamma_r}$  and  $\beta_L(x) = \alpha_L(x)^{-1} x$ .

If  $L$  is a normal subgroup of  $G$ , we can encode the coset  $xL$  by  $\alpha_L(x)$ . This makes the factor group  $G/L$  a black-box group: The elements are encoded by the elements of the

transversal  $\{\alpha(x); x \in G\}$ , and the multiplication oracle is obtained as a composition of the multiplication oracle for  $G$  with the computation of the function  $\alpha_L$ .

### 2.3 Subgroup states and purifications

Let  $G$  be a finite group and let  $H$  be a subgroup of  $G$ . We consider elements of the group algebra  $\mathbb{C}G$  as pure quantum states. (The “natural” scalar product  $(\sum_x \alpha_x |x\rangle, \sum_y \beta_y |y\rangle) = \sum \alpha \bar{\beta} xy^{-1}$  makes  $\mathbb{C}G$  a Hilbert space where the group elements form an orthonormal basis.)

A (left) coset superposition of  $H$  in  $G$  is the uniform superposition  $|aH\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |ah\rangle$  where  $a \in G$ . The (left) subgroup state of  $H$  in  $G$  is the mixed state with the density matrix

$$\Xi_{G,H} = \frac{1}{|G|} \sum_{a \in G} |aH\rangle \langle aH| = \frac{1}{|G : H|} \sum_{a \in X} |aH\rangle \langle aH|,$$

where  $X$  is any left transversal (a set of representatives of the left cosets) of  $H$  in  $G$ .

A purification of  $\Xi_{G,H}$  is any pure state  $|\psi\rangle \in \mathbb{C}G \otimes V$  for some Hilbert space  $V$  such that  $\Xi_{G,H}$  is the relative trace of  $|\psi\rangle \langle \psi|$  with respect to the second subsystem. For general facts about purification of mixed states, in particular for the connection with Schmidt decompositions, we refer the reader to Section 2.5 of [30]. The following lemma gives a characterization of purifications of subgroup states.

**Lemma 1** *The pure state  $|\psi\rangle \in \mathbb{C}G \otimes V$  is a purification of the subgroup state  $\Xi_{G,H}$  if and only if it can be written as*

$$|\psi\rangle = \frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |v(x)\rangle,$$

where the states  $|v(x)\rangle$  and  $|v(y)\rangle$  are equal if  $x$  and  $y$  are in the same left coset of  $H$  and orthogonal otherwise.

**Proof** The “if” part follows easily from that the conditions on  $|v(\cdot)\rangle$  imply  $|\psi\rangle = \frac{1}{\sqrt{k}} \sum_{a \in X} |aH\rangle |v(a)\rangle$ .

To see the “only if” part, recall that a Schmidt decomposition of a state  $|\psi\rangle \in \mathbb{C}G \otimes V$  is of the form  $|\psi\rangle = \sum_{i=1}^m \lambda_i |u_i\rangle |v_i\rangle$  where  $m = |G|$ ,  $|u_1\rangle, \dots, |u_m\rangle$  is an arbitrary orthonormal basis of  $\mathbb{C}G$  in which the relative trace of  $|\psi\rangle \langle \psi|$  w.r.t. to the second subsystem is diagonal (with entries  $\lambda_1, \dots, \lambda_m$ ) and the system of the vectors  $v_i$  corresponding to nonzero eigenvalues  $\lambda_i$  is an orthonormal system of vectors in  $V$ . The vectors  $v_i$  depend on the choice of the basis  $|u_i\rangle$  ( $i = 1, \dots, m$ ). Notice that the only nonzero eigenvalue of  $\Xi_{G,H}$  is  $\frac{1}{k}$  with multiplicity  $k$ , where  $k = |G : H|$ . The coset superpositions give an orthonormal basis of the corresponding eigenspace. Thus, if  $|\psi\rangle$  is a purification of  $\Xi_{G,H}$  then a Schmidt decomposition of  $|\psi\rangle$  which is a purification of  $\Xi_{G,H}$  is of the form  $|\psi\rangle = \frac{1}{\sqrt{k}} \sum_{i=1}^k |u_i\rangle |v_i\rangle$  where  $|u_1\rangle, \dots, |u_k\rangle$  is an arbitrary orthonormal basis of the  $\frac{1}{k}$ -eigenspace of  $\Xi_{G,H}$  and  $|v_1\rangle, \dots, |v_k\rangle$  is an orthonormal

system of  $V$ . In particular, if  $X = \{a_1, \dots, a_k\}$  then by taking  $|u_i\rangle = |a_i H\rangle$  and by defining  $|v(x)\rangle = |v_i\rangle$  for  $x \in a_i H$ , we obtain  $|\psi\rangle = \frac{1}{\sqrt{k}} \sum_{a \in X} |aH\rangle |v(a)\rangle = \frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |v(x)\rangle$ .  $\square$

### 2.4 Basic subgroup state conversions

Given a subgroup  $L$  of  $G$ , a copy of (a purification of) the subgroup state  $\Xi_{G,H}$  can be “converted” to a copy of (a purification of)  $\Xi_{L,H \cap L}$  by replacing  $|x\rangle$  with the decomposition  $|\beta_L(x)\rangle |\alpha_L(x)\rangle$  obtained by the method outlined in Sect. 2.2 for  $x \in G$  and “ignoring”  $|\alpha_L(x)\rangle$  (passing this part to the purifying subsystem). To see this, let  $\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |\psi(x)\rangle$  be a purification of  $\Xi_{G,H}$ , with  $|\psi(x)\rangle$  and  $|\psi(y)\rangle$  are equal if and only if  $y^{-1}x \in H$ , and orthogonal otherwise. Then, the substitution gives the state  $\frac{1}{\sqrt{|L|}} \sum_{x \in L} |x\rangle \frac{1}{\sqrt{|G:L|}} \sum_{y \in Y} |y\rangle |\psi(yx)\rangle$  where  $Y = \{\alpha_L(z) : z \in G\}$ . Now if  $x_1, x_2 \in L$  are from the same left coset of  $H \cap L$  then  $|\psi(yx_1)\rangle = |\psi(yx_2)\rangle$  for every  $y \in Y$  and hence the states  $\frac{1}{\sqrt{|G:L|}} \sum_{y \in Y} |y\rangle |\psi(yx_i)\rangle$  are equal ( $i = 1, 2$ ), while otherwise they do not overlap as for  $y_1, y_2 \in Y$  either  $|y_1\rangle$  and  $|y_2\rangle$  are orthogonal or (for  $y_1 = y_2$ )  $|\psi(y_1x_1)\rangle$  and  $|\psi(y_1x_2)\rangle$  are orthogonal. We shall refer to this procedure as *restriction*. The term is justified by that in the standard version of the HSP, one could obtain an instance of the HSP in the subgroup  $L$  by restricting the “hiding function” to  $L$ .

Similarly, assume that  $L$  is a normal subgroup of  $G$  contained in  $H$ . Then a copy of (a purification of) the subgroup state  $\Xi_{G,H}$  can be converted to a copy of (a purification of)  $\Xi_{G/L,H/L}$  by replacing  $x$  with  $|\alpha_L(x)\rangle |\beta_L(x)\rangle$  and passing  $|\beta_L(x)\rangle$  to the purifying subsystem. This corresponds to the technique called “pushing” in [6, 31].

## 3 A group-theoretic reduction

In this section we prove the following.

**Proposition 1** *Let  $G$  be a nilpotent black-box group of class at most  $c$  and assume that the prime factors of  $|G|$  are given as part of the input and that for each such prime  $p$  the quantum Fourier transform modulo a multiple of  $p$  and its inverse can be implemented by an efficient exact quantum procedure. Then, the HSP in  $G$  can be reduced by an exact procedure in time  $\text{poly}(\log \ell)$  to  $\text{poly}(\log |G|)$  instances of the problem HSMC in semi-elementary quotient groups of subgroups of  $G$ . (The elements of  $G$  are assumed to be uniquely encoded by strings of length  $\ell$ .)*

**Proof** A finite nilpotent group  $G$  is the direct product of its Sylow subgroups. Therefore any subgroup  $H$  is the product of its Sylow subgroups. The  $p$ -Sylow subgroup of  $H$  is  $P \cap H$  where  $P$  is the  $p$ -Sylow subgroup of  $G$ . The Sylow subgroups of  $G$  can be computed using the method outlined in Sect. 2.2. One can convert subgroup states in  $G$  to subgroup states in  $P$  using restriction, see Sect. 2.4.

In the rest of the description of the reduction we assume that  $G$  is a  $p$ -group. We maintain a subgroup  $H_0$  of  $H$ . Initially  $H_0 = \{1_G\}$ . In each round of an outer loop of the algorithm  $H_0$  will be increased if  $H_0 < H$ . If  $H_0$  is already  $G$  then we

can obviously stop. We will also maintain a subgroup  $K$  of  $G$  such that if  $H_0 < H$  then even  $H_0 < K \cap H$ . Initially  $K = N_G(H_0)$ . This is a good choice because in a nilpotent group every proper subgroup has a strictly larger normalizer; therefore, if  $H_0 < H$ , then  $H_0 < N_H(H_0) = H \cap N_G(H_0)$ . In an inner loop  $K$  will be decreased until either  $H_0$  is increased or  $K$  becomes identical with  $H_0$ . In the latter case we can conclude that  $H = H_0$  and stop the whole procedure. If the abelian factor  $K/(K'H_0)$  is not elementary, then we can replace  $K$  with a proper subgroup as follows. Let  $L > K'H_0$  be the subgroup of  $K$  such that  $L/(K'H_0)$  contains all the elements of order  $p$  of  $K/(K'H_0)$ . To compute  $L$ , first compute  $K'$  and  $K'H_0$ . Then take the set of generators  $\Gamma$  for  $K$  and for each element  $g \in \Gamma$ , compute the smallest positive integer  $\alpha_g$  such that  $g^{p^{\alpha_g}} \in K'H_0$ . The elements  $g^{p^{\alpha_g-1}}$  ( $g \in \Gamma$ ) generate  $L$ . If  $L$  is a proper subgroup of  $K$ , then we replace  $K$  with  $L$  and repeat the step above. (Correctness of this is justified by observing that  $L/H_0$  contains all the elements of order  $p$  of  $K/H_0$ , whence if  $H \cap K > H_0$  then also  $H \cap L > H_0$ .) Otherwise we have achieved that  $K/(K'H_0)$  is elementary abelian. Then using the HSMC in the quotient group  $S = K/H_0$  with the hidden subgroup  $(H \cap K)/H_0$ , we compute

$$\begin{aligned} ((H \cap K)/H_0)/S' &= ((H \cap K)/H_0)/((K'H_0)/H_0) = ((H \cap K)(K'H_0))/H_0 \\ &= ((H \cap K)K')/H_0, \end{aligned}$$

where the last equality follows from  $H_0 < H \cap K$ . If  $(H \cap K)K' = K$  then  $H \cap K = K$  because  $K'$  is contained in every maximal subgroup of  $K$ . Then we can increase  $H_0$  by replacing  $H_0$  with  $K$  and continue the outer loop. If  $(H \cap K)K' < K$  we can replace  $K$  with  $(H \cap K)K'$  and continue the inner loop.

Based on the descriptions above, we summarize the exact algorithm in the pseudocode below.

---

**Algorithm 1** Reduction to HSMC

---

```

1: Initialize:  $H_0 \leftarrow 1_G$ ;
2: while  $H_0 < G$  do
3:    $K \leftarrow N_G(H_0)$ ;
4:    $Found \leftarrow \text{False}$ ;
5:   while  $Found = \text{False}$  do
6:     if  $K/(K'H_0)$  is elementary then
7:       Use HSMC to compute  $(H \cap K)K'/H_0$ ;
8:       if  $(H \cap K)K' = K$  then
9:          $H_0 \leftarrow K$ ;
10:         $Found \leftarrow \text{True}$ ;
11:      else
12:         $K \leftarrow (H \cap K)K'$ ;
13:        if  $K = H_0$  then
14:          return  $H = K$ .
15:        end if
16:      end if
17:    else
18:      For each  $g \in \Gamma_K$  compute the smallest positive integer  $\alpha_g$  with  $g^{p^{\alpha_g}} \in K'H_0$ ;
19:      Compute  $L = \langle g^{p^{\alpha_g-1}} \mid g \in \Gamma_K \rangle$ ;
20:       $K \leftarrow L$ ;
21:    end if
22:  end while
23: end while

```

---

If  $|G| = p^n$  then the outer loop is executed at most  $n$  times, while within each round of the outer loop the inner loop has at most  $n$  rounds. Thus we need at most  $n^2$  calls to the HSMC procedure for factors of subgroups of  $G$  and further  $n^2 \text{poly}(\ell)$  group and other operations. Note that all the groups we need to apply the HSMC procedure are of class at most  $c$  because the family of nilpotent groups of class at most  $c$  is closed under taking subgroups and factor groups.  $\square$

We illustrate the algorithm by the following example. Assume that  $G$  is the Heisenberg group modulo 9 and that the hidden subgroup  $H$  is the group of the matrices of the form  $\begin{pmatrix} 1 & x \\ & 1 \\ & & 1 \end{pmatrix}$ . Initially  $H_0 = \{I\}$  and  $K = G$ . The commutator subgroup  $K' = G'$

consists of the matrices of the form  $\begin{pmatrix} 1 & a_{13} \\ & 1 \\ & & 1 \end{pmatrix}$  and  $K/K' \cong \mathbb{Z}_9^2$ , not elementary. Then

$K$  is replaced by the group of matrices  $\begin{pmatrix} 1 & 3a_{12} & a_{13} \\ & 1 & 3a_{23} \\ & & 1 \end{pmatrix}$ . (Here and in each step below,

the notation  $a_{ij}$  should be considered as independent of those used in the previous step(s).) It turns out that this  $K$  is abelian ( $K' = \{I\}$ ), but not elementary. It gets

replaced by the group of matrices of the form  $\begin{pmatrix} 1 & 3a_{12} & 3a_{13} \\ & 1 & 3a_{23} \\ & & 1 \end{pmatrix}$ , which is isomorphic

to  $\mathbb{Z}_3^3$  and  $H \cap K$  consists of the matrices of the form  $\begin{pmatrix} 1 & 3a_{12} \\ & 1 \\ & & 1 \end{pmatrix}$  and  $H_0$  becomes

this subgroup. The next  $K$  will be the normalizer of this  $H_0$ , which is the group of

matrices  $\begin{pmatrix} 1 & a_{12} & a_{13} \\ & 1 & 3a_{23} \\ & & 1 \end{pmatrix}$ , whose commutator subgroup  $K'$  consists of the matrices of

the form  $\begin{pmatrix} 1 & a_{13} \\ & 1 \\ & & 1 \end{pmatrix}$  and  $K/K'$  is isomorphic to  $\mathbb{Z}_3^3$ , so  $K$ , and hence also  $K/H_0$ ,

is semi-elementary. Therefore HSMC will find  $H/H_0$  in  $K/H_0$  and the procedure terminates.

### 4 The main conversion

Let  $L$  be a subgroup of the center of  $G$  isomorphic to  $\mathbb{Z}_p^n$  where  $p$  is a prime. Then  $L$  is a normal subgroup of  $G$ . Our aim is to convert a copy of the subgroup state  $\Xi_{G,H}$  to a copy of  $\Xi_{G/L,HL/L}$ . In the light of the second conversion (“pushing”) described in Sect. 2.4, one could do it by converting first to a copy of  $\Xi_{G,HL}$ .

To this end, it would be desirable to have a procedure that converts the coset superposition  $|aH\rangle$  to  $|aHL\rangle$ . A possible approach would be computing  $|L\rangle = \frac{1}{\sqrt{|L|}} \sum_{z \in L} |z\rangle$  in a new register, multiplying  $|aH\rangle$  with it to obtain  $\frac{1}{\sqrt{|HL|}} \sum_{z \in L} \sum_{x \in H} |azx\rangle|z\rangle$  and

then trying to “disentangle”  $|z\rangle$  from  $|azx\rangle$ . The quantum Fourier transform of  $L$  almost does this job: If we apply it to the second register, we obtain the state

$$\frac{1}{\sqrt{|L|}} \sum_{y \in L} \frac{\omega^{(y,z)}}{\sqrt{|HL|}} \sum_{z \in L} \sum_{x \in H} |azx\rangle |y\rangle,$$

where  $\omega = e^{\frac{2\pi i}{p}}$  and by  $(\cdot, \cdot)$  we denote the standard scalar product of  $L$  modulo  $p$ . For  $y \in L$ , let us denote by  $P_y$  the linear transformation of  $\mathbb{C}G$  mapping  $|x\rangle$  to  $\frac{1}{\sqrt{|L|}} \sum_{z \in L} \omega^{(y,z)} |xz\rangle$ . With this notation, the state we have can be rewritten as

$$\frac{1}{\sqrt{|L|}} \sum_{y \in L} |P_y(aH)\rangle |y\rangle.$$

Using the assumption that  $L$  is in the center of  $G$ , a direct calculation shows that for every  $x_1, x_2 \in G$ , we have

$$|P_y(x_1x_2)\rangle = |x_1P_y(x_2)\rangle = |(P_y(x_1)x_2)\rangle. \tag{1}$$

It is also straightforward to see that for every  $x \in G$  and for every  $w \in L$ , we have

$$|wP_y(x)\rangle = |P_y(x)w\rangle = \omega^{-(y,w)} |P_y(x)\rangle. \tag{2}$$

We define the support of an element  $|u\rangle$  of  $\mathbb{C}G$  as the set of elements appearing with nonzero coefficient in the decomposition of  $|u\rangle$  as a linear combination of group elements. Using equality (1), one can show that if  $x_1$  and  $x_2$  are not in the same left coset of  $LH$ , then the states  $|P_y(aH)x_1^{-1}\rangle$  and  $|P_y(aH)x_2^{-1}\rangle$  are orthogonal. This is because the support of  $|P_y(aH)x_i^{-1}\rangle$  is contained in  $LHx_i^{-1} = (x_iLH)^{-1}$  ( $i = 1, 2$ ). On the other hand, if  $x_1H = x_2H$ , then  $Hx_1^{-1} = Hx_2^{-1}$  and the two states are equal. By the characterization given in Lemma 1, it follows that for any left coset  $aH$ , the state  $\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |P_0(aHx^{-1})\rangle$  is a purification of  $\Xi_{G,LH}$ .

Of course it is hopeless to enforce  $y = 0$  in  $|P_y(aH)x_1^{-1}\rangle$ . However, we can compute a state with essentially the same effect using several copies of the subgroup state and by applying an algorithm that finds zero sum subsequences of sufficiently long sequences of elements of  $L$ . Assume that we have a procedure that, for some  $S = S(p, n)$ , given an element  $\underline{y} = (y_1, \dots, y_S) \in L^S$  computes a non-empty subset  $J(\underline{y})$  of  $\{1, \dots, S\}$  such that  $\sum_{j \in J(\underline{y})} y_j = 0$ .

Then, for a sequence  $|a_1H\rangle \dots |a_SH\rangle$  we compute first

$$|L|^{-S/2} \sum_{\underline{y} \in L^S} |\underline{y}\rangle |P_{y_1}(a_1H)\rangle \dots |P_{y_S}(a_SH)\rangle$$

by applying the Fourier method outlined above component-wise. We next compute  $\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle$  in a fresh register and multiply by  $x^{-1}$  the  $j$ th component of

$|P_{y_1} a_1 H\rangle \dots |P_{y_S} a_S H\rangle$  if  $j \in J(\underline{y})$ . Let  $\chi_{\underline{y}} : \{1, \dots, S\} \rightarrow \{0, 1\}$  denote the characteristic function of  $J(\underline{y})$ . Then the state we obtained is  $\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |\psi(x)\rangle$ , where

$$|\psi(x)\rangle = |L|^{-S/2} \sum_{\underline{y} \in L^S} |\underline{y}\rangle |P_{y_1} a_1 H x^{-\chi_{\underline{y}}(1)}\rangle \dots |P_{y_S} a_S H x^{-\chi_{\underline{y}}(S)}\rangle.$$

Consider the term of  $|\psi(x)\rangle$  corresponding to any  $\underline{y}$ . As  $J(\underline{y})$  is non-empty, we have that for  $x_1, x_2$  not in the same left coset of  $LH$ , the appropriate terms of  $|\psi(x_i)\rangle$  are orthogonal. As  $|\underline{y}\rangle$  also appears in the corresponding term, we have that  $|\psi(x_i)\rangle$  are also orthogonal. On the other hand, if  $x_1, x_2$  are in the same left coset of  $H$ , then these states are equal term by term. Finally, for  $x \in L$ , by (2), the term for  $\underline{y}$  only gets a phase change by  $\prod_{j \in J(\underline{y})} \omega^{-(y_j, x)} = \omega^{\sum_{j \in J(\underline{y})} (y_j, x)} = \omega^0 = 1$  by the choice of  $J(\underline{y})$ . It follows that if  $x_1$  and  $x_2$  are in the same left coset of  $LH$ ,  $|\psi(x_1)\rangle = |\psi(x_2)\rangle$ . Thus our state is a purification of  $\Xi_{G, LH}$ . As this holds for any fixed  $S$ -tuple of left cosets of  $H$ , by linearity we also obtain a purification of  $\Xi_{G, LH}$  if we apply the procedure to copies of a purification of  $\Xi_{G, H}$ . We obtained the following.

**Lemma 2** *Assume that we have an exact quantum procedure (e.g., a deterministic polynomial time algorithm) that, given any sequence  $y_1, \dots, y_S$  of  $S = S(p, n)$  elements of  $\mathbb{Z}_p^n$ , in time  $T(p, n) \geq S(p, n)$  finds a non-empty subset of  $\{1, \dots, S\}$  such that  $\sum_{j \in J} y_j = 0$ . Then we have an exact quantum procedure using  $n$  quantum Fourier transforms modulo  $p$  that converts  $S(p, n)$  copies of (a purification) of  $\Xi_{G, H}$  to a copy of (a purification of)  $\Xi_{G/L, HL/L}$  where  $L$  is subgroup of the center of  $G$  isomorphic to  $\mathbb{Z}_p^n$  in time  $T(p, n) \text{poly}(\log |G|)$ .*

In words, provided that we can efficiently solve the zero sum subsequence problem in the subgroup  $L \cong \mathbb{Z}_p^n$  contained in the center of  $G$ , we can convert copies of the subgroup state for  $H$  in  $G$  to a subgroup state for the subgroup  $HL/L$  (that is, “ $H$  modulo  $L$ ”) in the factor group  $G/L$ .

Using the lemma in iteration and applying the exact abelian hidden subgroup algorithm of [16], we can derive the following.

**Proposition 2** *Let  $G$  be a semi-elementary black-box group with unique encoding of order  $p^n$ . Assume that the quantum Fourier transform modulo  $p$  and its inverse can be implemented by an efficient exact algorithm and that, like in Lemma 2, we have an exact method to find zero sum subsequences of sequences of  $S(p, n)$  elements of  $\mathbb{Z}_p^n$  in time  $T(p, n) \geq S(p, n)$ . Then the problem HSMC can be solved by an exact quantum algorithm that uses  $\text{poly}(T(p, n)^{O(c)} \ell)$  elementary operations,  $\text{poly}(T(p, n)^{O(c)} \log |G|)$  applications of the group oracle, calls to the oracle computing the purification of the subgroup state; and the inverses of these. (The elements of  $G$  are assumed to be uniquely encoded by strings of length  $\ell$ .)*

**Proof** We compute the lower central series  $G = G_0 > G_1 > \dots > G_c = \{1\}$  using the method presented in Subsection 2.2. As  $G/G'$  is elementary abelian, so are the factors  $G_{i-1}/G_i$  ( $i = 1, \dots, c$ ). This is because the factor groups  $G_{i-1}/G_i$  are homomorphic images of tensor powers (as  $\mathbb{Z}$ -modules) of the  $G/G'$ , see Theorem 5.2.5

of [27]. Also, isomorphisms of  $G_{i-1}/G_i$  with  $\mathbb{Z}_p^{n_i}$  can be efficiently computed using the method of [16]. Iteration of Lemma 2 gives a procedure to convert  $\prod_{i=2}^c S(n_i)$  copies of a purification of  $\Xi_{G,H}$  to a copy of a purification of  $\Xi_{G/G',HG'/G'}$ . The composition of instances of the original subgroup state creating procedure (the calls to the oracle) with the conversion gives a procedure for creating a purification of  $\Xi_{G/G',HG'/G'}$ . We can use this as the oracle input for the exact hidden subgroup algorithm of [16] in  $\mathbb{Z}_p^{n_1}$ . For  $i = 1, \dots, c$ , we have  $S(p, n_i) \leq S(p, n)$  and  $T(p, n_i) \leq T(p, n)$  because  $\mathbb{Z}_p^{n_i}$  can be embedded in  $\mathbb{Z}_p^n$  as a subgroup.  $\square$

As an example, consider the group  $G$  of unitriangular 4 by 4 matrices modulo 3.

We can visualize  $G$  as  $\begin{pmatrix} 1 & * & * & * \\ & 1 & * & * \\ & & 1 & * \\ & & & 1 \end{pmatrix}$  Here and below in the example, a matrix with

asterisks denotes the set of matrices obtained by substituting the asterisks by integers modulo 3, independently at each place. Assume that  $H$  is the subgroup  $\begin{pmatrix} 1 & * & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}$ .

$G$  is a semi-elementary matrix group. The lower central series of  $G$  is  $G_0 = G$ ,  $G_1 = \begin{pmatrix} 1 & * & * \\ & 1 & * \\ & & 1 \end{pmatrix}$ ,  $G_2 = \begin{pmatrix} 1 & * \\ & 1 \end{pmatrix}$ , and  $G_3 = \{I\}$ , thus the nilpotency class is 3.

In the first round, we convert copies subgroup states for  $H$  to those for  $HG_2/G_2 = \begin{pmatrix} 1 & * & * \\ & 1 & * \\ & & 1 \end{pmatrix} / \begin{pmatrix} 1 & * \\ & 1 \end{pmatrix}$  and next to  $HG_1/G_1 = \begin{pmatrix} 1 & * & * \\ & 1 & * \\ & & 1 \end{pmatrix} / \begin{pmatrix} 1 & * \\ & 1 \end{pmatrix}$  in the elementary abelian group  $G/G_1$ . There we can solve the HSMC for  $G$  by using the exact abelian hidden subgroup algorithm from [16] in  $G/G_1$ .

In the non-exact setting, essentially the same proof gives the following.

**Proposition 3** *Let  $G$  be a semi-elementary black-box group with unique encoding of order  $p^n$ . Assume that there exists a quantum (or a randomized) algorithm that finds zero sum subsequences of sequences of  $S(p, n)$  elements of  $\mathbb{Z}_{p^n}$  in time  $T(p, n) \geq S(p, n)$  with high probability. Then the problem HSMC can be solved by a quantum algorithm that uses  $\text{poly}(T(p, n)^{O(c)}\ell)$  elementary operations,  $\text{poly}(T(p, n)^{O(c)} \log |G|)$  applications of the group oracle and calls to the oracle computing the purification of the subgroup state.*

### 5 Zero sum subsequences in $\mathbb{Z}_p^n$

In this section, we assume that our input is a sequence of vectors from  $\mathbb{Z}_p^n$ . We also assume that  $p$  is an odd prime as for  $p = 2$  a zero sum subsequence can be obtained from  $n + 1$  vectors in the form of a zero linear combination. As subsequences can be

represented as subsets of the index set, it will not be too misleading to use the term (sub)set for a (sub)sequence. Our strategy will be finding  $p$  pairwise disjoint subsets of input vectors having equal sums. We will achieve this goal by designing a method for finding a non-trivial pair of subsets having equal sum and then, like in [21], applying the algorithm recursively to obtain 4, 8, 16, etc. disjoint subsets with equal sum.

Note that a pair of disjoint subsets with equal sum can be interpreted as a representation of the zero vector by a linear combination of the input vectors with nonzero coefficients 1 or  $-1$  only. Based on this, it will be convenient to use the term *signed subsets* and signed subset sums. A signed subset of a set  $S$  of vectors is formally a function from  $S$  to the set  $\{0, 1, -1\}$ . The support of such a signed subset is the set of elements on which the function takes nonzero values. With some sloppiness, we use the term *signed subset sum* to refer both to the signed subset and to the value of the signed sum. (Technically, a signed subset sum could be a data structure consisting of the description of the signed subset and the value.) We call two or more subset sums disjoint if their supports are pairwise disjoint. Based on the observation that a signed subset sum of vectors that are results of pairwise disjoint subset sums is again a signed subset sum of the original vectors, one can build signed subset sums hierarchically from smaller disjoint signed subset sums. The trivial subset sum corresponds to the empty set with the zero vector as value.

A *linear relation* (or just relation for short) among a collection of vectors is formally an array of coefficients such that the corresponding linear combination is the zero vector. However, we continue to denote relations by equalities between linear combinations with the zero vector. It is often useful to omit the vectors to which coefficient zero is assigned. By taking the signed subsets of the vectors having the same or the opposite coefficient in a linear relation, we obtain a linear relation among pairwise disjoint signed subset sums in which the coefficients are form  $\{1, \dots, \frac{p-1}{2}\}$  and each coefficient appears at most once. We call such a relation of signed subset sums *standard*. As an example, let  $\{v_1, v_2, \dots, v_{10}\}$  be a fixed set of vectors satisfying the relation  $a_1 v_1 + a_2 v_2 + \dots + a_{10} v_{10} = 0$  for some constants  $a_i \in \mathbb{Z}_p$ . Moreover, suppose that the coefficients satisfy the following

$$\begin{aligned} a_1 = a_2, a_3 = -a_1, \text{ and } a_1 &\leq \frac{p-1}{2}, \\ a_4 = a_6, a_5 = a_7 = -a_4 \text{ and } a_4 &\leq \frac{p-1}{2}, \\ a_8 = a_9 = a_{10} \text{ and } a_8 &> \frac{p-1}{2}. \end{aligned}$$

Then we have the standard relation

$$a_1(v_1 + v_2 - v_3) + a_4(v_4 + v_6 - v_5 - v_7) + (p - a_8)(-v_8 - v_9 - v_{10}) = 0$$

among the signed subset sums  $v_1 + v_2 - v_3$ ,  $v_4 + v_6 - v_5 - v_7$  and  $-v_8 - v_9 - v_{10}$ . We shall build standard linear relations among signed subset sums with smaller and smaller coefficients (among increasingly larger subset sums). The key idea is constructing first  $\frac{(p-1)^2}{4}$  pairwise signed subset sums arranged in a square matrix having a relation

in each row as well as in each column and subtracting the sum of higher half of “horizontal” relations from the sum of the higher half of the “vertical” relations to obtain a relation with coefficients between 1 and  $\frac{p-1}{4}$  and iterating the construction.

As an example, let  $V = \{v_1, v_2, \dots, v_{40}\} \subset \mathbb{Z}_7^3$ . As the dimension is 3, any four vectors are linearly dependent. We form 10 disjoint groups consisting of 4 consecutive vectors and compute linear relations accordingly. Suppose that we find the following 10 relations, each having standard form  $u_{k1} + 2u_{k2} + 3u_{k3} = 0$  among the signed subset sums  $u_{k1}, u_{k2}$  and  $u_{k3}$  ( $k = 1, \dots, 10$ ):

1.  $v_1 + 5v_2 + 3v_3 + 6v_4 = 0$ , which gives the above standard form relation for the signed subset sums  $u_{11} = v_1 - v_4, u_{12} = -v_2$  and  $u_{13} = v_3$ ;
2.  $2v_5 + 6v_6 + 5v_7 + 3v_8 = 0$ , thus the signed subset sums are  $u_{21} = -v_6, u_{22} = v_5 - v_7$  and  $u_{23} = v_8$ ;
3.  $v_9 + 3v_{10} + 4v_{11} + 2v_{12} = 0$ , accordingly  $u_{31} = v_9, u_{32} = v_{12}$  and  $u_{33} = v_{10} - v_{11}$ ;
4.  $5v_{13} + 6v_{14} + 2v_{15} + 5v_{16} = 0$ , accordingly  $u_{41} = -v_{14}, u_{42} = v_{15} - v_{13} - v_{16}$  and  $u_{43} = 0$ ;
5.  $v_{17} + 4v_{18} + 4v_{19} + 5v_{20} = 0$ , accordingly  $u_{51} = v_{17}, u_{52} = -v_{20}$  and  $u_{53} = -v_{18} - v_{19}$ ;
6.  $5v_{21} + 6v_{22} + 5v_{23} + 2v_{24} = 0$ , accordingly  $u_{61} = -v_{22}, u_{62} = v_{24} - v_{21} - v_{23}$  and  $u_{63} = 0$ ;
7.  $6v_{25} + 4v_{26} + 3v_{27} + 3v_{28} = 0$ , accordingly  $u_{71} = -v_{25}, u_{72} = 0$  and  $u_{73} = v_{27} + v_{28} - v_{26}$ ;
8.  $3v_{29} + 2v_{30} + v_{31} + 2v_{32} = 0$ , accordingly  $u_{81} = v_{31}, u_{82} = v_{30} + v_{31}$  and  $u_{83} = v_{29}$ ;
9.  $2v_{33} + 5v_{34} + v_{35} + 3v_{36} = 0$ , accordingly  $u_{91} = v_{35}, u_{92} = v_{33} - v_{34}$  and  $u_{93} = v_{36}$ ;
10.  $4v_{37} + 6v_{38} + 2v_{39} + v_{40} = 0$ , accordingly  $u_{101} = v_{40} - v_{38}, u_{102} = v_{39}$  and  $u_{103} = -v_{37}$ .

(Here,  $u_{43} = 0, u_{63} = 0$  and  $u_{72} = 0$  mean that these signed subset sums are empty.) Note that for each  $k = 1, \dots, 10, u_{k1} + 2u_{k2} + 3u_{k3} = 0$ . Let  $u_k \in \mathbb{Z}_7^9$  be the concatenation of  $u_{k1}, u_{k2}$  and  $u_{k3}$  (considered now as vectors) for each  $k = 1, \dots, 10$ . Then one can obtain a linear relation for the vectors  $u_1, \dots, u_{10}$ . Suppose that we find  $2u_1 + 3u_2 + 6u_3 + 6u_4 + u_5 + 5u_6 + 5u_7 + 3u_8 + u_9 + 4u_{10} = 0$ . It has the standard relation  $u'_1 + 2u'_2 + 3u'_3 = 0$  again, this time among the signed subset sums  $u'_1 = u_5 + u_9 - u_3, u'_2 = u_1 - u_6 - u_7$  and  $u'_3 = u_2 + u_8 - u_4 - u_{10}$ . In terms of the vectors  $u_{kj}, 1 \leq k \leq 10, 1 \leq j \leq 3$ , we can represent each  $u'_i$  as the concatenation of  $w_{i1}, w_{i2}$  and  $w_{i3}$ , where

$$\begin{aligned}
 w_{11} &= u_{51} + u_{91} - u_{31}, & w_{12} &= u_{52} + u_{92} - u_{32}, & w_{13} &= u_{53} + u_{93} - u_{33}, \\
 w_{21} &= u_{11} - u_{61} - u_{71}, & w_{22} &= u_{12} - u_{62} - u_{72}, & w_{23} &= u_{13} - u_{63} - u_{73}, \\
 w_{31} &= u_{21} + u_{81} - u_{41} - u_{101}, & w_{32} &= u_{22} + u_{82} - u_{42} - u_{102}, & \text{and} \\
 w_{33} &= u_{23} + u_{83} - u_{43} - u_{103}.
 \end{aligned}$$

It follows that

$$\begin{array}{rcl}
 1w_{11} + 2w_{12} + 3w_{13} = 0 & & 1w_{11} \quad 1w_{12} \quad 1w_{13} \\
 & & + \quad + \quad + \\
 1w_{21} + 2w_{22} + 3w_{23} = 0 & & 2w_{21} \quad 2w_{22} \quad 2w_{23} \\
 & \text{and also} & + \quad + \quad + \\
 1w_{31} + 2w_{32} + 3w_{33} = 0 & & 3w_{31} \quad 3w_{32} \quad 3w_{33} \\
 & & \parallel \quad \parallel \quad \parallel \\
 & & 0 \quad 0 \quad 0
 \end{array}$$

Therefore, by subtracting the sum of the last two horizontal relations from the sum of the last two vertical relations, we obtain  $w_{12} + w_{13} + w_{32} - w_{21} - w_{23} - w_{31} = 0$ . Moreover, expanding each  $w_{ij}$  as signed subset sums of the original vectors from  $V$ , then obtain a representation of the zero vector as a signed sum of elements of  $V$ , or, equivalently, two subsets of  $V$  with equal sums (“colliding unsigned subset sums”).

We give the details in the following lemma and its proof. (We present a version that even saves up maintaining the first half of vertical relations.)

**Lemma 3** *Let  $d$  be a positive integer. Assume that there is a deterministic procedure  $\mathcal{A}$  that, given  $h(d, n)$  vectors from  $\mathbb{Z}_p^n$ , in time  $\text{poly}(h(d, n) \log p)$  finds  $d$  pairwise disjoint signed subset sums  $v_1, \dots, v_d$  of the input vectors, not all empty, such that  $\sum_{i=1}^d i v_i = 0$ . Then there also exists a deterministic procedure that, given  $h(d, n)h(d, \lceil d/2 \rceil n)$  vectors, in time  $\text{poly}(h(d, n)h(d, \lceil d/2 \rceil n) \log p)$  finds pairwise disjoint signed subset sums  $w'_1, \dots, w'_{\lceil d/2 \rceil}$ , not all empty, such that  $\sum_{i=1}^{\lceil d/2 \rceil} i w'_i = 0$ .*

**Proof** We divide the input set into  $h(d, \lceil d/2 \rceil n)$  pairwise disjoint parts of size  $h(d, n)$ . We apply procedure  $\mathcal{A}$  within each part. This way for each  $k = 1, \dots, h(d, \lceil d/2 \rceil n)$ , we get  $d$  pairwise disjoint subset sums  $u_{k1}, \dots, u_{kd}$ , not all empty, such that  $\sum_{j=1}^d j u_{kj} = 0$ . For each  $k$  we consider the concatenation  $u_k$  of the vectors  $u_{kj}$  ( $j = \lceil d/2 \rceil + 1, \dots, d$ ). These are vectors of dimension  $\lceil d/2 \rceil n$ . We apply procedure  $\mathcal{A}$  to find pairwise disjoint signed subsets  $M_1, \dots, M_d$  such that  $\sum_{i=1}^d i u'_i = 0$ , where  $u'_i$  is the signed sum of the  $u_k$ s corresponding to the signed subset  $M_i$ . Now for each  $1 \leq i \leq d$ ,  $u'_i$  is the concatenation of vectors  $w_{ij}$  ( $j = \lceil d/2 \rceil + 1, \dots, d$ ). Here, for  $1 \leq i, j \leq d$ ,  $w_{ij}$  stands for the signed subset sum obtained by joining the signed subset sums  $u_{kj}$  according to the signed subset  $M_i$ . The signed subset sums  $w_{ij}$  are pairwise disjoint, not all of them are empty and they satisfy the relations

$$\sum_{i=1}^d i w_{ij} = 0 \quad (j = \lceil d/2 \rceil + 1, \dots, d)$$

and

$$\sum_{j=1}^d j w_{ij} = 0 \quad (i = 1, \dots, d).$$

We subtract the sum of the last  $\lceil d/2 \rceil$  (“horizontal”) relations of the second kind from the sum the  $\lceil d/2 \rceil$  (“vertical”) relations of the first kind and obtain the relation

$$\sum_{i=1}^{\lceil d/2 \rceil} \sum_{j=\lceil d/2 \rceil+1}^d iw_{ij} - \sum_{i=\lceil d/2 \rceil+1}^d \sum_{j=1}^{\lceil d/2 \rceil} jw_{ij} + \sum_{i,j=\lceil d/2 \rceil+1}^d (i-j)w_{ij} = 0.$$

Notice that for  $\lceil d/2 \rceil + 1 \leq i, j \leq d$ , we have  $|i - j| \leq \lfloor d/2 \rfloor$ . Therefore, by flipping signs where appropriate and then joining the signed subsets with equal coefficients, we obtain pairwise disjoint subset sums  $w'_1, \dots, w'_{\lfloor d/2 \rfloor}$  with  $\sum_{i=1}^{\lfloor d/2 \rfloor} iw'_i = 0$ .

The subset sums  $w'_i$  can all be empty only if each  $w_{ij}$  is empty when  $i \neq j$  and at least one of  $i$  and  $j$  is greater than  $\lfloor d/2 \rfloor$ . Assume that this is the case. Then, if there is an index  $i > \lfloor d/2 \rfloor$  such that  $w_{ii}$  is non-empty then  $w_{ii}$  must be itself a non-trivial zero subset sum and gives a one-term solution. Otherwise not all  $w_{ij}$  are empty for  $i, j \leq \lfloor d/2 \rfloor$  and  $\sum_{i,j=1}^{\lfloor d/2 \rfloor} iw_{ij} = 0$ .  $\square$

Iterated application of the method of Lemma 3 gives the following result.

**Proposition 4** *Given  $S_{\pm}(p, n) = p^{O(p \log p)} n^{O(p)}$  vectors from  $\mathbb{Z}_p^n$ , a non-trivial signed subset sum representing the zero vector can be found in deterministic time  $\text{poly}(S_{\pm}(p, n))$ .*

**Proof** Put  $d_0 = \frac{p-1}{2}$ ,  $h_0(n) = n + 1$  and define  $d_i = \lfloor d_{i-1}/2 \rfloor$  and  $h_i(n) = h_{i-1}(n)h_{i-1}(\lfloor d_{i-1}/2 \rfloor n)$  recursively for  $i = 1, \dots, \lfloor \log d_0 \rfloor$ . As among any  $h_0(n) = n + 1$  vectors from  $\mathbb{Z}_p^n$  a non-trivial linear relation can be found in time  $\text{poly}(n \log p)$ , recursive applications of Lemma 3 give that among  $h_{\lfloor \log d_0 \rfloor}(n)$  vectors a single non-trivial signed subset sum (that is, a linear relation with nonzero coefficients  $\pm 1$  only) can be found in time  $\text{poly}(h_{\lfloor \log d_0 \rfloor}(n) \log p)$ . We show by induction that

$$h_i(n) \leq \left( \prod_{j=0}^{i-1} \lceil d_j/2 \rceil \right)^{2^{i-1}} (n + 1)^{2^i}. \tag{3}$$

For  $i = 0$ , both sides are equal to  $n + 1$ . Assume that the inequality holds for  $0 \leq i < \lfloor \log d_0 \rfloor$ . Then we also have

$$h_i(\lceil d_i/2 \rceil n) \leq \left( \prod_{j=0}^{i-1} \lceil d_j/2 \rceil \right)^{2^{i-1}} (\lceil d_i/2 \rceil n + 1)^{2^i}.$$

Using  $\lceil d_i/2 \rceil n + 1 \leq \lceil d_i/2 \rceil (n + 1)$ , we obtain

$$h_i(\lceil d_i/2 \rceil n) \leq \lceil d_i/2 \rceil^{2^{i-1}} \left( \prod_{j=0}^i \lceil d_j/2 \rceil \right)^{2^{i-1}} (n + 1)^{2^i}. \tag{4}$$

Multiplying inequalities (3) and (4) and using  $h_{i+1}(n) = h_i(n)h_i(\lceil d_i/2 \rceil n)$ , we obtain

$$h_{i+1}(n) \leq \left( \prod_{j=0}^i \lceil d_j/2 \rceil \right)^{2^i} (n+1)^{2^{i+1}},$$

which is inequality (3) for  $i + 1$  in place of  $i$ . Using  $d_j \leq d_0/2^j \leq d_0 = \frac{p-1}{2}$ , inequalities (3) for  $i = \lfloor \log d_0 \rfloor$  give

$$\begin{aligned} h_{\lfloor \log d_0 \rfloor} &\leq \left( \prod_{j=0}^{\lfloor \log \frac{p-1}{2} \rfloor - 1} \left\lceil \frac{p-1}{4} \right\rceil \right)^{2^{\lfloor \log \frac{p-1}{2} \rfloor - 1}} (n+1)^{2^{\lfloor \log \frac{p-1}{2} \rfloor}} \\ &= \left\lceil \frac{p-1}{4} \right\rceil^{\lfloor \frac{p-1}{4} \rfloor \lfloor \log \frac{p-1}{2} \rfloor} (n+1)^{\lfloor \frac{p-1}{2} \rfloor}. \end{aligned}$$

Therefore, we have

$$h_{\lfloor \log d_0 \rfloor}(n) = p^{O(p \log p)} n^{O(p)}.$$

□

We interpret a non-empty zero sum signed subset as a non-trivial collision between two disjoint subset sums. (Non-trivial means that at most one of the subsets can be empty.) We use the short-term *collision* for such a pair. We have the following.

**Proposition 5** *Suppose that there is an algorithm  $\mathcal{B}$  that given a set of vectors from  $\mathbb{Z}_p^n$  of size  $S_{\pm}(p, n)$  finds a collision. Then there is a deterministic procedure that, given  $S_{\pm}(p, n)^{\lceil \log p \rceil}$  vectors, finds a non-trivial zero sum subset using less than  $S_{\pm}^{\lceil \log p \rceil}$  applications of algorithm  $\mathcal{B}$  and  $\text{poly}((S_{\pm}(p, n))^{\lceil \log p \rceil})$  other operations.*

**Proof** Put  $S = S_{\pm}(p, n)$  and  $\ell = \lceil \log p \rceil$ . We start with finding a collision  $(H_1^+, H_1^-)$  among the first  $S$  vectors with common sum  $w_1$  using algorithm  $\mathcal{B}$ . We continue with the next  $S$  input vectors and find a collision  $(H_2^+, H_2^-)$  with sum  $w_2$ , and so on. We then take the first  $S$  subset sums  $w_1, \dots, w_S$  and find a pair of disjoint subsets  $(K^+, K^-)$  of  $\{1, \dots, S\}$ , not both empty, such that  $\sum_{i \in K^+} w_i = \sum_{i \in K^-} w_i = w$ . The four subsets  $L^{++} = \bigcup_{i \in K^+} H_i^+$ ,  $L^{+-} = \bigcup_{i \in K^+} H_i^-$ ,  $L^{-+} = \bigcup_{i \in K^-} H_i^+$  and  $L^{--} = \bigcup_{i \in K^-} H_i^-$  of input vectors are pairwise disjoint, not all empty and have common sum  $w$ . Iterating this we end up with at least  $p$  pairwise disjoint subsets (not all empty) with equal sum. If one of these sets is empty, then the common sum is zero, and we can take any of the non-empty subsets. Otherwise the union of the first  $p$  of the subsets has zero sum. The total number of applications of the collision finding algorithm  $\mathcal{B}$  is  $S^{\ell-1} + \dots + S + 1 < S^{\ell}$ . □

As an example, let us consider the following sequence of 3-dimensional vectors over  $\mathbb{Z}_3$ ,

$$\begin{aligned}
 v_1 &= \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}, v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, v_3 = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}, v_4 = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}, v_5 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, v_6 = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, v_7 = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \\
 v_8 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, v_9 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}, v_{10} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, v_{11} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}, v_{12} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}, v_{13} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \\
 v_{14} &= \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}, v_{15} = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}, v_{16} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}.
 \end{aligned}$$

Then for each  $\{v_1, v_2, v_3, v_4\}$ ,  $\{v_5, v_6, v_7, v_8\}$ ,  $\{v_9, v_{10}, v_{11}, v_{12}\}$ ,  $\{v_{13}, v_{14}, v_{15}, v_{16}\}$  we compute its linear relations and hence find the following collisions:

$$\begin{aligned}
 w_1 &= v_2 = v_1 + v_3 = [0, 1, 0]^T \\
 w_2 &= v_8 = v_6 + v_7 = [0, 0, 1]^T \\
 w_3 &= v_9 + v_{12} = v_{10} + v_{11} = [0, 1, 1]^T \\
 w_4 &= v_{13} + v_{15} + v_{16} = v_{14} = [2, 0, 0]^T.
 \end{aligned}$$

Next, we compute a linear relation for  $w_1, w_2, w_3, w_4$  and find a collision  $w = w_1 + w_2 = w_3$ . Therefore, we have 4 subsequences with equal sum  $w$ : The first two are  $V_1 = \{v_2, v_8\}$  and  $V_2 = \{v_1, v_3, v_6, v_7\}$ , and the last two are  $V_3 = \{v_9, v_{12}\}$  and  $V_4 = \{v_{10}, v_{11}\}$ . Hence, we obtain a zero sum subsequence by taking any three subsequences from  $V_1, V_2, V_3, V_4$ .

Propositions 4 and 5, together with the remark on the case  $p = 2$  immediately give the following.

**Theorem 2** *There is a deterministic algorithm that, given a sequence of  $S(p, n) = p^{O(p \log^2 p)} n^{O(p \log p)}$  vectors from  $\mathbb{Z}_p^n$ , finds a non-trivial zero sum subsequence in time  $\text{poly}(S(p, n))$ .*

We remark that in [21], an algorithm for a more general task is given. This task is finding a non-trivial representation of the zero vector as a linear combination of the input vectors with  $d$ th power coefficients. This includes our problem as the special case  $d = p - 1$ . The algorithm of [21] for  $d = p - 1$  would give  $S(p, n) = p^{O(p^2 \log p)} n^{O(p \log p)}$ , a parameter somewhat worse than that we have in Theorem 2, though would be still polynomial in  $n$  for  $p = O(1)$ . The method of [21] for finding a collision is more complicated than the present one: It is based on collecting relations organized in a  $d$ -dimensional hypercube rather than a square. (The method of doubling collisions is essentially identical with that described here in Proposition 5.)

## 6 Concluding remarks

We have shown that the hidden subgroup problem in a nilpotent group  $G$  of class bounded by a constant can be solved in polynomial time by an exact quantum algorithm provided that there is a polynomial time (that is, time  $\text{poly}(n \log p)$ ) exact method that finds zero sum subsequences in sequences consisting of polynomially many elements of  $\mathbb{Z}_p^n$  for prime divisors  $p$  of  $|G|$ . We have such a method for  $p = O(1)$ . By Olson's theorem [22], the shortest length for which a not necessarily polynomial time zero sum subsequence finding algorithm exists is around  $np$ .

We propose the question of existence of a  $\text{poly}(np)$ -time algorithm for finding zero sum subsequences from sequences of length  $(np)^d$  for a sufficiently large constant  $d$  as a problem for further research. A positive answer would imply existence of an exact polynomial time quantum algorithm for the case when  $|G|$  is smooth, that is, the prime factors of  $|G|$  are of size bounded by a polynomial in  $\log|G|$ . Even a non-exact method (e.g., a randomized algorithm) would be of great interest as, by Proposition 3, it would give a new result in the non-exact setting: existence of an efficient "probabilistic" quantum hidden subgroup algorithm for nilpotent groups of smooth order having  $O(1)$ -bounded nilpotency class. Even somewhat worse results would potentially lead to quantum hidden subgroup algorithms faster than the known ones.

For the purposes of "probabilistic" quantum hidden subgroup algorithms even a method that finds a zero sum subsequence "on average," that is for at least a  $1/\text{poly}(np)$  proportion of the possible sequences would be sufficient. However, as the following simple worst-case to average-case reduction shows, at least in the randomized setting, the gain cannot be better than polynomial. Assume that the classical randomized algorithm  $\mathcal{A}$  finds in time  $T = T(p, n)$  with probability at least  $\delta$  a subsequence of a random sequence of length  $S = S(p, n)$  of vectors from  $\mathbb{Z}_p^n$ . Here, probability is taken for the uniform distribution of the array of the vectors together with the random bits of  $\mathcal{A}$ . Then we can do the following. We start with an arbitrary sequence of  $\frac{1}{\delta} \cdot S^2$  input vectors; we draw  $\frac{1}{\delta} \cdot S^2$  uniformly random vectors, one for each input vector. Then we divide the input sequence into groups of length  $S$ , and to each input vector we add the corresponding random vector. Within each group, we apply procedure  $\mathcal{A}$ . As the sums are random vectors, in each group, procedure  $\mathcal{A}$  succeeds with probability at least  $\delta$  and, with probability at least  $\frac{1}{2}$ ,  $\mathcal{A}$  will succeed in at least  $S$  groups. If this is the case then we choose  $S$  "lucky" groups, in each group take the sum of the random vectors corresponding to the members of the zero sum subsequences. We apply algorithm  $\mathcal{A}$  for these  $S$  sums. It finds a non-trivial zero sum subsequence with probability at least  $\delta$ . Finally, we take the union of the corresponding subsequences. This way we obtain a procedure that finds a non-trivial zero sum subsequence of every sequence of length  $\frac{1}{\delta} \cdot S^2$  in time  $\text{poly}(T + \frac{1}{\delta} \cdot ST)$  with probability at least  $\delta/2$ .

**Acknowledgements** The research of the second author was supported by the Hungarian Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

**Funding** Open access funding provided by Budapest University of Technology and Economics.

**Data availability** Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

## Declarations

**Conflict of interest** The authors have no conflicts of interest to declare.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Ettinger, M., Høyer, P., Knill, E.: The quantum query complexity of the hidden subgroup problem is polynomial. *Inf. Process. Lett.* **91**, 43–48 (2004). [arXiv:quant-ph/0401083](#)
2. Kuperberg, G.: A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.* **35**(1), 170–188 (2005). [arXiv:quant-ph/0302112](#)
3. Alagic, G., Moore, C., Russell, A.: Quantum algorithms for Simon's problem over general groups. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1217–1224 (2007). [arXiv:quant-ph/0603251](#)
4. Lomont, C.: The hidden subgroup problem-review and open problems. Technical report (2004). [arXiv:quant-ph/0603251](#)
5. Wang, F.: The hidden subgroup problem. Technical report (2010). [arXiv:1101.1053](#)
6. Lomonaco, S.J., Kauffman, L.H.: Quantum hidden subgroup algorithms: an algorithmic toolkit. In: *Mathematics of Quantum Computation and Quantum Technology*, pp. 21–64. Chapman and Hall/CRC, New York (2007). [arXiv:quant-ph/0607046](#)
7. Horan, K., Kahrobaei, D.: The hidden subgroup problem and post-quantum group-based cryptography. In: *Mathematical Software—ICMS 2018*, pp. 218–226 (2018). Springer. [arXiv:1805.04179](#)
8. Castryck, W., Meeran, N.V.: Two remarks on the vectorization problem. In: *Progress in Cryptology—INDOCRYPT 2022*, pp. 658–678 (2023). Springer
9. Alagic, G., Russell, A.: Quantum-secure symmetric-key cryptography based on hidden shifts. In: *Advances in Cryptology—EUROCRYPT 2017*, pp. 65–93 (2017). Springer. [arXiv:1610.01187](#)
10. Bae, E., Lee, S.: Quantum algorithm based on the  $\epsilon$ -random linear disequations for the continuous hidden shift problem. *Quantum Inf. Process.* **20**(10), 347 (2021). [arXiv:1904.06225](#)
11. Brassard, G., Høyer, P.: An exact quantum polynomial-time algorithm for Simon's problem. In: *ISTCS 97*, pp. 12–23 (1997). [arXiv:quant-ph/9704027](#)
12. Cai, G., Qiu, D.: Optimal separation in exact query complexities for Simon's problem. *J. Comput. Syst. Sci.* **97**, 83–93 (2018). [arXiv:1610.01920](#)
13. Bonnetain, X.: Tight bounds for Simon's algorithm. In: *LATINCRYPT 2021*, pp. 2–23 (2021)
14. Wu, Z., Qiu, D., Tan, J., Li, H., Cai, G.: Quantum and classical query complexities for generalized Simon's problem. *Theor. Comput. Sci.* **924**, 171–186 (2022). [arXiv:1905.08549](#)
15. Mosca, M., Zalka, C.: Exact quantum Fourier transforms and discrete logarithm algorithms. *Int. J. Quantum Inf.* **02**, 91–100 (2003). [arXiv:quant-ph/0301093](#)
16. Imran, M., Ivanyos, G.: An exact quantum hidden subgroup algorithm and applications to solvable groups. *Quantum Inf. Comput.* **22**(9&10), 770–789 (2022). [arXiv:2202.04047](#)
17. Friedl, K., Ivanyos, G., Magniez, F., Santha, M., Sen, P.: Hidden translation and translating coset in quantum computing. *SIAM J. Comput.* **43**(1), 1–24 (2014) [arXiv:quant-ph/0211091](#). Preliminary version in STOC 2003
18. Bonnetain, X., Naya-Plasencia, M.: Hidden shift quantum cryptanalysis and implications. In: *Advances in Cryptology—ASIACRYPT 2018*, pp. 560–592 (2018). Springer

19. Ivanyos, G., Sanselme, L., Santha, M.: An efficient quantum algorithm for the hidden subgroup problem in nil-2 groups. *Algorithmica* **62**(1–2), 480–498 (2012). [arXiv:0707.1260](https://arxiv.org/abs/0707.1260)
20. Decker, T., Høyer, P., Ivanyos, G., Santha, M.: Polynomial time quantum algorithms for certain bivariate hidden polynomial problems. *Quantum Inf. Comput.* **14**(9–10), 790–806 (2014). [arXiv:1305.1543](https://arxiv.org/abs/1305.1543)
21. Ivanyos, G., Santha, M.: Solving systems of diagonal polynomial equations over finite fields. *Theor. Comput. Sci.* **657**, 73–85 (2017). [arXiv:1503.09016](https://arxiv.org/abs/1503.09016)
22. Olson, J.E.: A combinatorial problem on finite abelian groups. *I. J. Number Theory* **1**(1), 8–10 (1969). [https://doi.org/10.1016/0022-314X\(69\)90021-3](https://doi.org/10.1016/0022-314X(69)90021-3)
23. Nishimura, H., Ozawa, M.: Uniformity of quantum circuit families for error-free algorithms. *Theor. Comput. Sci.* **332**, 487–496 (2005). <https://doi.org/10.1016/j.tcs.2004.12.020>
24. Nishimura, H., Ozawa, M.: Computational complexity of uniform quantum circuit families and quantum Turing machines. *Theor. Comput. Sci.* **276**, 147–181 (2002). [arXiv:quant-ph/9906095](https://arxiv.org/abs/quant-ph/9906095)
25. Nishimura, H., Ozawa, M.: Perfect computational equivalence between quantum Turing machines and finitely generated uniform quantum circuit families. *Quantum Inf. Process.* **8**, 13–24 (2009). [arXiv:quant-ph/0511117](https://arxiv.org/abs/quant-ph/0511117)
26. Bernstein, E., Vazirani, U.: Quantum complexity theory. *SIAM J. Comput.* **26**, 1411–1473 (1997). <https://doi.org/10.1137/S0097539796300921>
27. Robinson, D.: *A Course in the Theory of Groups*, 2nd edn. Springer, Urbana (1995)
28. Kantor, W.M., Luks, E.M.: Computing in quotient groups. In: *STOC 1990*, pp. 524–534 (1990). <https://doi.org/10.1145/100216.100290>
29. Watrous, J.: Quantum algorithms for solvable groups. In: *STOC 2001*, pp. 60–67 (2001). [arXiv:quant-ph/0011023](https://arxiv.org/abs/quant-ph/0011023)
30. Nielsen, M.A., Chuang, I.K.: *Quantum Computation and Quantum Information*, 10th Anniversary Edition. Cambridge University Press, Cambridge (2010)
31. Lomonaco, S.J., Kauffman, L.H.: Is Grover’s algorithm a quantum hidden subgroup algorithm? *Quantum Inf. Process.* **6**, 461–476 (2007). [arXiv:quant-ph/0603140](https://arxiv.org/abs/quant-ph/0603140)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.