COMPUTATIONAL COMPLEXITY OF ALGORITHMIC PROBLEMS AND CRYPTOGRAPHIC APPLICATIONS IN POLYCYCLIC GROUPS

JONATHAN GRYAK
DECEMBER 11, 2015

Committee
Professor Delaram Kahrobaei, Mentor, NYCCT, CUNY
Professor Benjamin Fine, Fairfield University
Professor Conchita Martinez-Perez, University of Zaragosa
Professor Vladimir Shpilrain, CCNY, CUNY
Professor Xiaowen Zhang, College of Staten Island, CUNY

ABSTRACT. In non-commutative cryptography, cryptographic systems are built upon groups and other algebraic structures whose underlying operations are non-commutative. These structures have been employed to construct protocols for public key exchange, digital signatures, authentications schemes, and secret sharing. The suitability of these structures for cryptography relies upon generalizations of cryptographic hardness assumptions of classical (commutative) systems. The conjugacy search problem is one such generalization, whose computational complexity is specific to the group under consideration. This has motivated a search for groups in which the conjugacy search problem is computationally difficult.

In this survey, we provide an overview of group-based cryptography. In particular, we explore the suitability of polycyclic and metabelian groups for use in cryptographic systems (which were proposed for cryptography in 2004 by Eick and Kahrobaei). We describe their construction and their associated algorithmic problems. We introduce a family of polycyclic and metabelian groups, and we analyze the computational complexity of the conjugacy search problem within it.

We also review the length-based attack, a heuristic algorithm originally designed for systems based upon braid groups. We adapt the length-based attack to solve the single conjugacy search problem, and provide experimental results of its performance on members of our family of groups.

CONTENTS

1. Introduction 2
2. Mathematical Preliminaries 3
2.1. Conjugacy 3
2.2. Torsion 4
2.3. Semidirect Products 4
2.4. Polycyclic Groups 4
2.5. Metabelian and Solvable Groups 5
2.6. Word and Conjugacy Problems 6
2.7. Growth Rates 7
1. Introduction

In cryptography, many of the most common key exchange protocols, including Rivest-Shamir-Adleman (RSA), Diffie-Hellman, and elliptic curve based schemes, reply upon hardness assumptions related to integer factorization and discrete logarithms. Currently there are no known efficient algorithms for computing discrete logarithms or factoring large integers on conventional computers. However, in 1994 Peter Shor devised a quantum algorithm [32] which solves both of these problems in polynomial time. This has motivated the search for alternative methods for constructing cryptosystems. One such methodology is non-commutative cryptography, which unlike the aforementioned commercial systems do not operate over the integers. Instead, non-commutative cryptographic systems are built upon groups and other algebraic structures whose underlying operations are non-commutative.

In 1999, Anshel, Anshel, and Goldfeld [1] and Ko, Lee, et al. [23] introduced key exchange protocols whose security was based in part on the conjugacy search problem: for a group $G$, given that $u, v \in G$ are conjugate, find $X$ so that $u^x = v$. Though both systems operated over braid groups, the same protocols can be performed using alternate groups, provided they possess the requisite cryptographic properties. These properties include having a solvable word problem, a computationally difficult conjugacy problem, a fast word growth rate, and exhibiting the namesake non-commutativity. As such, researchers are also interested in finding groups which meet these requirements.

In 2004 Eick and Kahrobaei [7] suggested polycyclic groups as a platform group for conjugacy-based cryptosystems. Moreover, they noticed that as the Hirsch length increases, the time it takes to solve conjugacy search problem becomes dramatically longer. Polycyclic groups are known to have a solvable conjugacy problem due to Fomanek [9] and Remeslennikov [31] independently. Subsequently, a number of cryptosystems were developed using polycyclic groups, including Kahrobaei and
Anshel [17], Kahrobaei and Khan [18], and Kahrobaei and Koupparis [19].

A heuristic algorithm known as the length-based attack (LBA) was introduced by Hughes and Tannenbaum [16] as a means to attack the AAG key exchange protocol over braid groups, as there was no known deterministic algorithm for solving the conjugacy search problem in these groups. In [27], Myasnikov and Ushakov made successive refinements to the algorithm to yield a high success rate. More recently, the authors of [10] and [20] analyzed the LBA on AAG over polycyclic groups.

In this paper, we explore a family of metabelian groups for which the conjugacy search problem has exponential time complexity. Section 2 provides some of the mathematical background on the algebraic constructions used in the remainder of the paper, along with a discussion of the computational complexity of the decision problems for finitely presented groups as introduced by Max Dehn [6]. Section 3 includes the security assumptions behind the discrete logarithm problem and a demonstration of the AAG key exchange protocol. We also introduce a modified version of the length-based attack, called the *length-based conjugacy search*, which provides a heuristic algorithm for solving the conjugacy search problem, rather than breaking the AAG protocol.

In Section 4 we introduce the family of metabelian groups $G = B \rtimes Q$, given by the general presentation below:

$$G = \langle q_1, \ldots, q_n, b_1, \ldots, b_s | [q_l, q_t] = 1, [b_i, b_j] = 1, b_i^{m_l(i, t)} = q_l b_i q_l^{-1} = b_i^{m_l(i, 1)} b_i^{m_l(i, 2)} \ldots b_i^{m_l(i, s)} \rangle$$

with $1 \leq l, t \leq n$, $1 \leq i, j \leq s$ and the $m_l(j, i)$ suitable integers so that the actions of the $q_l$ commute. We show that elements of these groups have linear representations as matrices over $\mathbb{Q}$, and that linear systems based on these groups can be solved in polynomial time.

In Section 5 we analyze the computation complexity of the conjugacy search problem in the above family of groups. We show that in general, the conjugacy search problem in these groups is exponential with respect to word length. We then show that for a particular subset of groups the conjugacy search problem is polynomial. We conclude the section by demonstrating that, for a subset of groups, the conjugacy search problem reduces to the discrete logarithm problem.

In Section 6, we provide experimental evidence that length-based conjugacy search is ineffective in solving the conjugacy search problem on generalized metabelian Baumslag-Solitar groups, which were constructed using a range of large prime numbers. We conclude the paper in Section 7 with a list of open problems related to these groups.

2. Mathematical Preliminaries

2.1. Conjugacy.

**Definition 2.1.** Given a group $G$, two elements $x, y \in G$ are *conjugate*, denoted $x \sim y$, if there exists an element $z \in G$ such that

$$x = zy z^{-1}.$$
Note that conjugation is often denoted as exponentiation:
\[ y^z = yzy^{-1} \]

2.2. Torsion.

**Definition 2.2.** Let \( G \) be an abelian group. An element \( g \in G \) has finite order \( n \) if \( \exists n \in \mathbb{Z} \) such that \( ng = 0 \). If no such \( n \) exists, the element is said to have infinite order. For each \( n \in \mathbb{Z} \), the elements which have the same order \( n \) form a subgroup of \( G \), denoted \( G[n] \). The set of all elements with finite order forms a subgroup \( T \) called the torsion subgroup of \( G \).

**Definition 2.3.** Let \( G \) be an abelian group. For \( p \in \mathbb{Z} \), with \( p \) a fixed prime, the subgroup \( G_p \) is called a \( p \)-primary component of \( G \).

**Definition 2.4.** The exponent of a torsion group \( T \), denoted \( \exp(T) \), is the smallest non-negative integer \( k \) such that \( kv = 0 \) for any \( v \in T \). If there is no such integer, then the exponent is infinite.

**Theorem 2.1** (Primary Decomposition Theorem). Let \( G \) be an abelian group with torsion subgroup \( T \), then
\[ T = \bigoplus_p G_p, \]
with \( G_p \) the \( p \)-primary components of \( G \).

2.3. Semidirect Products. One of the standard ways in which to construct non-commutative groups is the use of the semidirect product. The semidirect product of two groups is a generalization of the direct product, wherein only one of the groups is normal in the resultant group.

**Definition 2.5.** Given two groups \( H \) and \( K \), along with a homomorphism \( \phi : K \to \text{Aut}(H) \), we can construct a new group \( G \) called the semidirect product, denoted \( G = H \rtimes \phi K \).

Multiplication in the semidirect product \( G \) is defined as
\[ (h_1,k_1)(h_2,k_2) = (h_1\phi(k_1) \cdot h_2,k_1k_2) \]
with \((h_1,k_1),(h_2,k_2) \in G\) and \( \cdot \) denoting the action of \( \phi(k_1) \) on \( h_2 \). The \( \rtimes \) symbol is used to indicate that \( H \) is a normal subgroup of \( G \).

\( G \) contains subgroups \( H' = \{(h,1) \mid h \in H\} \) and \( K' = \{(1,k) \mid k \in K\} \), which are isomorphic to \( H \) and \( K \) respectively. From this it follows that the group action \( \phi(k) \cdot h \) is equivalent to \( h^k \), that is
\[ \phi(k) \cdot h = h^k = k^{-1}hk. \]

Note that \( G \) will be non-abelian provided that \( \phi \) is not equivalent to the trivial homomorphism.

2.4. Polycyclic Groups.

**Definition 2.6.** A group \( G \) is polycyclic if it has a subnormal series
\[ \{1\} = G_0 \triangleleft G_1 \triangleleft \cdots \triangleleft G_n = G \]
where \( G_{i+1}/G_i \) is cyclic. This series of subgroups is called a polycyclic series.
Polycyclic groups are a subset of solvable groups. In fact, polycyclic groups can be defined as the class of solvable groups for which every subgroup is finitely generated. The class of polycyclic groups is closed under many algebraic operations including factor groups, finite direct products, embeddings, and subgroups [14].

All polycyclic groups are finitely presented, and we have the following inclusion hierarchy of finitely presented groups:

\[
\text{cyclic} \subset \text{abelian} \subset \text{nilpotent} \subset \text{polycyclic} \subset \text{solvable}
\]

**Example 2.1.** The dihedral group of order 8 (symmetry group of a square) is a finite polycyclic group.

For a polycyclic group \( G \) and polycyclic series \( G_1, \ldots, G_n \), recall that each \( G_{i+1}/G_i \) is cyclic. Thus \( \exists x_i \ G \) such that \( \langle x_i G_{i+1} \rangle \) for all \( 1 \leq i \leq n \).

**Definition 2.7.** A **polycyclic sequence** for \( G \) consists of the elements \( X = [x_1, \ldots, x_n] \) such that \( \langle x_i G_{i+1} \rangle \) for all \( 1 \leq i \leq n \).

**Definition 2.8.** Let \( G \) be a polycyclic group with a polycyclic sequence \( X \) and corresponding polycyclic sequence \( \{G_i\} \). The entries of the sequence \( R(X) = (r_1, \ldots, r_n) \), defined by \( r_i = |G_i : G_{i+1}| \in \mathbb{N} \cup \{\infty\} \) are called the **relative orders** of \( X \).

\( R(X) \), as its notation implies, is dependent upon the particular polycyclic sequence \( X \). However, for any polycyclic group \( G \), the number of infinite terms in \( R(X) \) is invariant with respect to \( X \). This number is called the **Hirsch length** of \( G \).

Note that while that there can be multiple polycyclic series for a group, a polycyclic sequence uniquely determines a polycyclic series. In turn, a polycyclic sequence, together with its relative orders gives rise to a special presentation for a polycyclic group, called a **polycyclic presentation**. Trivial relations, specifically of the form \( x_i^2 = x_i \), are omitted from the list of relations in this presentations.

**Definition 2.9.** In a polycyclic group \( G \) with polycyclic sequence \( X \), any element \( g \) can be represented uniquely in **normal form** as a product of powers of the of \( G \), i.e.:

\[
g = x_1^{e_1} \cdots x_n^{e_n},
\]

with \( e_i \in \mathbb{Z} \). The sequence \( (e_1, \ldots, e_n) \) is called the **exponent vector** of \( g \) with respect to \( X \).

Any element in a polycyclic group can be converted to its normal form using an algorithm called **collection**, which was described theoretically by Hall [12] and implemented by Neubüser [28], with additional version by Havas [13] and Leedham-Green [24], which is the current standard method and is know as **collection from the left**.

Another important property of polycyclic groups is that they are isomorphic to subgroups of \( GL(n, \mathbb{Z}) \).[2]

2.5. **Metabelian and Solvable Groups.**
Definition 2.10. A group $G$ is metabelian if and only if it has an abelian normal subgroup $A$ such that $G/A$ is abelian. Alternatively, it is a group with a subnormal series of length 2:
\[ \{1\} = G_0 \triangleleft G_1 \triangleleft G_2 = G, \]
and $G_2/G_1$ is abelian.

The group of Example 2.1 is also metabelian.

Metabelian groups are a special case of solvable groups, which remove the restriction on the length of the subnormal series. Note that all polycyclic and metabelian groups are solvable, but not all metabelian groups are polycyclic.

Example 2.2. The Baumslag-Solitar group
\[ BS(1, 2) = \langle a, b \mid bab^{-1} = a^2 \rangle \]
is metabelian but not polycyclic.

2.6. Word and Conjugacy Problems.
In 1911, Max Dehn introduced three decision problems on finitely presented groups - the word problem, the conjugacy problem, and the isomorphism problem. We consider the first two:

Definition 2.11 (Word Problem). Let $G$ be a finitely presented group. The word problem asks if it is decidable, for any $g \in G$, if $g = 1_G$, where $1_G$ is the identity element in $G$.

The word problem is undecidable in general for finitely generated groups, as proved independently by Novikov[30] and Boone[4]. For polycyclic groups, the word problem is solvable in $O(n^2)$ with respect to the length of the word[8, Thm 2.3.10]. The word problem in metabelian groups is also solvable.

Definition 2.12 (Conjugacy Decision Problem). Let $G$ be a finitely presented group. The conjugacy decision problem asks if it is decidable, for any $u, v \in G$, if $u \sim v$.

Note that if the conjugacy decision problem is decidable, then the word problem is also, as the word problem is the special case of the conjugacy problem where $u \in G$ and $v = 1_G$. In non-commutative cryptography, we are interested in the following variant of the conjugacy problem:

Definition 2.13 (Conjugacy Search Problem). Let $G$ be a finitely presented group. The conjugacy search problem is to find, for any $u, v \in G$, a $z \in G$ such that $u^z = v$.

Polycyclic groups are known to have a decidable conjugacy problem due to Fomanek [9] and Remeslennikov [31] independently. The problem is decidable in metabelian groups due to Noskov [29]. Additionally, some protocols uses the following variant:

Definition 2.14 (Multiple Conjugacy Search Problem). Let $G$ be a finitely presented group. The multiple conjugacy search problem is to find, for any $u_i, v_i \in G$, $u_i \sim v_i$, a $z \in G$ such that $\forall i, u_i^z = v_i$.

Note that in polycyclic groups the multiple conjugacy problem reduces to the single conjugacy problem [7].
Definition 2.15 (Membership Search Problem). Let $G$ be a finitely presented group, and $(s_1, \ldots, s_n) \leq G$. The membership search problem is to find, for any $g \in G$, a word $g'$ such that $g' = g$ and

$$g' = s_1^{e_1} \cdots s_n^{e_n}.$$  

2.7. Growth Rates.

One desired component of cryptographic systems is a large set of candidate keys, called a *keyspace*. In classical system, there is a direct exponential relationship between the bit length of a key (e.g., an unsigned integer of length $n$) and the number of possible keys ($2^n$). In non-commutative cryptography, where a key is an element of a group $G$, we can utilize the concept of a growth function to capture this idea. In the definitions below, let $G$ be a finitely generated group with generating set $S$.

Definition 2.16 (Reduced Word). For $g \in G$, let $g$ be a reduced word if $\forall s \in S, ss^{-1}$ does not appear in $g$.

Definition 2.17 (Word Length). For a reduced word $g \in G$, let the word length of $g$, denoted $L_g$, be the sum of the number of occurrences in $g$ of all generators $s \in S$.

Definition 2.18 (Growth Function). Let the growth function $\gamma_S : \mathbb{Z} \to \mathbb{Z}$ be defined as $\gamma_S(n) = |\{g \in G | L(g) \leq n\}|$. $\gamma_s$ is called polynomial if $\gamma(n)_S = O(n^j)$ or exponential if $\gamma_S(n) = \Omega(k^n)$ for some constants $j, k \in \mathbb{N}$.

From the work of Milnor[26] and Wolf[34], we have the following results concerning the growth functions of various finitely generated groups:

Theorem 2.2 (Milnor-Wolf).

1. If $G$ is a polycyclic group that has a nilpotent subgroup (i.e., is virtually nilpotent), then the $\gamma_S$ is polynomial. Otherwise $\gamma_S$ is exponential.

2. If $G$ is non-polycyclic solvable groups, then $\gamma_S$ is exponential. In particular, non-polycyclic metabelian groups have exponential growth.

3. Cryptographic Background

3.1. The Discrete Logarithm Problem and Hardness Assumptions.

Many extant cryptosystems are considered secure despite the existence of instances in which the system can be broken. Absolute security is instead replaced by the idea that these instances should occur so rarely as to be considered irrelevant to the security of the system. Negligible functions [22, Def 3.4], are a convenient way of formalizing the notion:

Definition 3.1. A function $f$ is negligible if for every polynomial $p(x), \exists N$ such that $\forall n > N, n \in \mathbb{Z}, f(n) < \frac{1}{p(n)}$. A negligible function can be denoted $negl$.

Definition 3.2. Given a cyclic group $G$ and elements $g, y \in G$, with $y = \langle g \rangle$, the discrete logarithm problem is to find $x \in \mathbb{Z}$ such that $g^x = y$.

We can now define formally the hardness assumption of the discrete logarithm problem:
Definition 3.3. Let \( \mathcal{A}(G, g, y) \) be any probabilistic, polynomial time algorithm \( \mathcal{A} \) that, for a specified cyclic group \( G \) with elements \( g, y \), outputs 1 if an \( x \) is found such that \( g^x = y \) and 0 otherwise. The discrete logarithm assumption is that \( \exists G \) such that, for any probabilistic, polynomial time algorithm \( \mathcal{A} \), the following holds:

\[
Pr[\mathcal{A}(G, g, y) = 1] \leq \text{negl}(n),
\]

with \( n = |G| \) being the order of the cyclic group.

There is no known efficient algorithm for computing discrete logarithms for arbitrary groups on conventional computers. Exhaustive search for a discrete logarithm takes \( O(n) \). For scale, note that the order of a group \( \mathbb{Z}_p^* \), based on the currently standard size of 2048-bit primes would be approximately \( 10^{617} \). The baby-step, giant-step algorithm by Shanks is currently the most efficient for arbitrary groups, at \( O(\sqrt{n} \cdot \text{polylog}(n)) \) [22, pg. 306]. For groups of the form \( \mathbb{Z}_p^* \), \( p \) prime, the general number field sieve is the most efficient at \( 2^{O(n^{1/3} \cdot (\log n)^{2/3})} \) [22, pg. 307].

As mentioned previously, Peter Shor devised a quantum algorithm [32] which runs in bounded error quantum polynomial time (BQP), that is, the algorithm runs in polynomial time and the probability that the algorithm produces an incorrect answer can be bounded arbitrarily.


In their 1999 paper [1], Anshel, Anshel, and Goldfeld introduced the commutator key exchange protocol, which is also referred to as AAG key exchange or Arithmetic. The group-based version of the key exchange, in the style of [27], works as follows:

Let \( G \) be a non-abelian group. We shall utilize the indefatigable Alice and Bob to simulate the key exchange protocol.

1. Alice chooses a set \( \mathcal{A} = \{a_1, \ldots, a_{N_1}\} \), with Bob choosing \( \mathcal{B} = \{b_1, \ldots, b_{N_2}\} \), with \( a_i, b_i \in G \). Note that \( \mathcal{A} \) and \( \mathcal{B} \) both generate subgroups of \( G \). These sets are then exchanged publicly with each other.
2. Alice constructs her private key as \( A = a_{s_1}^{\epsilon_1} \cdots a_{s_L}^{\epsilon_L} \), with \( a_{s_i} \in \mathcal{A} \) and \( \epsilon_i \in \{-1, 1\} \). Similarly, Bob computes as his private key \( B = b_{t_1}^{\delta_1} \cdots b_{t_L}^{\delta_L} \), with \( b_{t_i} \in \mathcal{A} \) and \( \delta_i \in \{-1, 1\} \).
3. Alice then computes \( A^{-1}b_{t_1}A, \ldots, A^{-1}b_{t_L}A \) and send this collection to Bob, while Bob computes and sends Alice \( B^{-1}a_{s_1}B, \ldots, B^{-1}a_{s_L}B \).
4. Alice and Bob can now compute a shared key \( \kappa = A^{-1}B^{-1}AB \), which is the commutator of \( A \) and \( B \), denoted \([A, B]\). Alice computes:

\[
\kappa_A = A^{-1}(B^{-1}a_{s_1}B)^{\epsilon_1} \cdots (B^{-1}a_{s_L}B)^{\epsilon_L} = A^{-1}B^{-1}a_{s_1}B \cdots B^{-1}a_{s_L}B = A^{-1}B^{-1}a_{s_1}^{\epsilon_1}(BB^{-1})a_{s_2}^{\epsilon_2}B \cdots B^{-1}a_{s_{L-1}}^{\epsilon_{L-1}}(BB^{-1})a_{s_L}^{\epsilon_L}B = A^{-1}B^{-1}a_{s_1}^{\epsilon_1}a_{s_2}^{\epsilon_2} \cdots a_{s_{L-1}}^{\epsilon_{L-1}}a_{s_L}^{\epsilon_L}B = A^{-1}B^{-1}AB.
\]

Analogously, Bob computes \( \kappa_B = B^{-1}A^{-1}BA \), and derives \( \kappa = \kappa_B^{-1} \).

As noted in [33] the security of AAG is based on both the simultaneous conjugacy search problem and the membership search problem.
3.3. Length-Based Conjugacy Search.

Length-based conjugacy search (LBCS) is an incomplete, local search that attempts to solve the conjugacy search problem (or its generalized version) by using the length of a word as a heuristic. In cryptography it is known as the length-based attack (LBA). It was first introduced by Hughes and Tannenbaum [16] as a means to attack the AAG key exchange protocol over braid groups, as there was no known deterministic algorithm for solving the conjugacy search problem in braid groups with more than 6 strands. To perform LBCS, we associate to our group an effectively computable length function that has the property that conjugation generically increases the lengths of elements. Following that, we iteratively build a conjugating element by successively conjugating by generators of our group and then assuming that we are building a successful conjugator when there is a decrease in length.

As length-based conjugacy search is an iterative improvement search, it is susceptible to failing at peaks and plateaux in the search space. In [27], Myasnikov and Ushakov identified when these peaks occur and were able to make successive refinements to the algorithm to yield a high success rate. More recently, the authors of [10] and [20] analyzed the LBA on AAG over polycyclic groups.

The experimental results of section 6 utilize the algorithm below, which modifies “Algorithm with Memory 2” from [10], a local beam search version of the LBA, to solve the single conjugacy problem. In this variation, one maintains a set $S$ consisting of conjugates of our initial element, $y$. Each element of $S$ is conjugated by each generator and the results are stored in a set $S'$. After every element of $S$ has been conjugated by every generator, the user saves the $M$ elements with minimal length and sets that equal to $S$. The algorithm is terminated when the problem has been solved or after a user-specified time-out. It is also worth noting that any other variation of the LBCS can be adapted to a single conjugacy search problem in much the same way. We assume that our group $G$ has a length function, $|\cdot|$ such that $|g| < |xgx^{-1}|$ and also that our set $S$ generates $G$. Note that $S$ does not need to be a minimal generating set, namely it may have a strict subset that also generates $G$. As input we take $x, y \in G$ such that $|y| > |x|$ and $B$ such that $\langle B \rangle = G$. For convenience, we assume that $B$ is closed under inversion of elements. We also impose a user-specified time-out and a natural number $M$ specifying the number of elements we keep track of.
Algorithm 1 LBCS with Memory 2 (Single Conjugacy Problem)

Initialize $S = \{(|y|, y, \text{id}_G)\}$

while not time-out do
    for $(|z|, z, a) \in S$ do
        Remove $(|z|, z, a)$
        for $g \in G$ do
            if $gzg^{-1} = x$ then
                Return $ga$ as an element that conjugates $x$ to $y$
            else
                Save $(|gzg^{-1}|, gzg^{-1}, ga)$ in a set $S'$
            end if
        end for
    end for
    Copy the $M$ elements with minimal first coordinate into $S$ and delete $S'$
end while
return FAIL

4. The Family of Platform Groups $\mathcal{F} : B \rtimes Q$

4.1. Definition and Examples.

In this section we introduce a family of metabelian groups $G = B \rtimes Q$ which are defined by the following presentation:

$$G = \langle q_1, \ldots, q_n, b_1, \ldots, b_s \mid [q_l, q_t] = 1, [b_i, b_j] = 1, b_i^{q_l} = q_l b_i q_l^{-1} = b_1^{m_{l(1,i)}} b_2^{m_{l(2,i)}} \cdots b_s^{m_{l(s,i)}} \rangle$$

with $1 \leq l, t \leq n, 1 \leq i, j \leq s$ and the $m_{l(j,i)}$ suitable integers so that the actions of the $q_l$ commute.

These groups have a rich algebraic structure:

- $B, Q$ are free abelian groups - $B$ and $Q$ are abelian groups that have a basis, respectively $(b_1, \ldots, b_s)$ and $(q_1, \ldots, q_l)$. This also means that $B$ and $Q$ are free modules over $\mathbb{Z}$.
- $B, Q$ are torsion-free - The only elements which have finite order are $1 \in B$ and $1 \in Q$.
- $B \rightarrow Q^*$ - $B$ is a free module over $Q$ with basis $(b_1, \ldots, b_s)$
- Linear Representation of Actions - The action of each $q_l$ can be represented as a $s \times x$ integral matrix $M_l$:

$$M_l = \begin{bmatrix}
    m_{l(1,1)} & \cdots & m_{l(1,s)} \\
    \vdots & \ddots & \vdots \\
    m_{l(s,1)} & \cdots & m_{l(s,s)}
\end{bmatrix}$$

In the case that each $M_l^{-1}$ is also integral, the group $G$ is polycyclic.

- $G$ has finite Prüfer rank - As $B$ and $Q$ are finitely generated and by construction the subgroups of $G$, $G$ has finite Prüfer rank.

Elements of these groups exhibit the following normal form:

$q_1^{-\alpha_1} \cdots q_n^{-\alpha_n} b_1^{\beta_1} \cdots b_s^{\beta_s} q_1^{\gamma_1} \cdots q_n^{\gamma_n}$

1The groups in this section are from joint work in [11].
with $\alpha_1, \beta_1, \gamma_1 \in \mathbb{Z}$ and $\alpha_1, \ldots, \alpha_n \geq 0$. Collection from the left can again be employed to convert any word written in the generators of $G$ to is unique normal form.

**Example 4.1.** Let $m_1, \ldots, m_n$ be positive integers. We call the group given by the following presentation a general metabelian Baumslag-Solitar group:

$$G = \langle q_1, \ldots, q_n, b \mid b^m = b^n, i, j = 1 \ldots, n, [q_i, q_j] = 1 \rangle.$$  

It is a constructible metabelian group of finite Prüfer rank and $G \cong B \times Q$ with $Q = \langle q_1, \ldots, q_n \rangle \cong \mathbb{Z}^n$ and $B = \mathbb{Z}[m_1^{\pm 1}, \ldots, m_k^{\pm 1}]$ (as additive groups).

**Example 4.2.** Let $L : \mathbb{Q}$ be a Galois extension of degree $n$ and fix an integral basis $\{u_1, \ldots, u_s\}$ of $L$ over $\mathbb{Q}$. Then $\{u_1, \ldots, u_s\}$ freely generates the maximal order $\mathcal{O}_L$ as a $\mathbb{Z}$-module. Now, we choose integral elements, $q_1, \ldots, q_n$, generating a free abelian multiplicative subgroup of $L - \{0\}$. Each $q_i$ acts on $L$ by left multiplication and using the basis $\{u_1, \ldots, u_s\}$, we may represent this action by means of an integral matrix $M_i$. Let $B$ be the smallest sub $\mathbb{Z}$-module of $L$ closed under multiplication with the elements $q_i$ and $q_i^{-1}$ and such that $\mathcal{O}_L \subseteq B$, i.e.,

$$B = \mathcal{O}_L[q_1^{\pm 1}, \ldots, q_n^{\pm 1}].$$

We then may define $G = B \rtimes Q$, where the action of $Q$ on $B$ is given by multiplication by the $q_i$’s. The generalized Baumslag-Solitar groups of the previous example are a particular case of this situation when $L = \mathbb{Q}$. If the elements $q_i$ lie in $\mathcal{U}_L$, which is the group of units of $\mathcal{O}_L$, then the group $G$ is polycyclic.

4.2. **Linear Representation.**

As $G$ is a semidirect product its elements can be written as $bx$, with $b \in B$ and $x \in Q$ written as words in their respective generators. As $B \subseteq \mathbb{Q}^s$ and each $q_i$ can be represented linearly as before, elements of $G$ can be converted from word to a linear representation.

First assume that $g$ is given in normal form as a word in the generators:

$$q_1^{-\alpha_1} \cdots q_n^{-\alpha_n} b_1^{\beta_1} \cdots b_s^{\beta_s} q_1^{\gamma_1} \cdots q_n^{\gamma_n},$$

with $\alpha_1, \ldots, \alpha_n \geq 0$.

Then the following word also yields $g$:

$$q_1^{-\alpha_1} \cdots q_n^{-\alpha_n} b_1^{\beta_1} \cdots b_s^{\beta_s} q_1^{\alpha_1} \cdots q_n^{\alpha_n} q_1^{\gamma_1-\alpha_1} \cdots q_n^{\gamma_n-\alpha_n}.$$  

In the above expression $g = bx$ with $x = q_1^{\gamma_1-\alpha_1} \cdots q_n^{\gamma_n-\alpha_n}$ and additively

$$v = (q_1^{-\alpha_1} \cdots q_n^{-\alpha_n}) \cdot (\beta_1 b_1 + \cdots + \beta_s b_s).$$

To represent $b$ as a vector $v$ in $\mathbb{Q}^s$, we utilize the integral matrix $M_i$ form of each $q_i$:

$$v = M_1^{-\alpha_1} \cdots M_n^{-\alpha_n} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_s \end{pmatrix}.$$
Therefore, any \( bx \in G \) has the following linear representation \( vx \):

\[
vx = M_1^{-\alpha_1} \cdots M_n^{-\alpha_n} \left( \begin{array}{c} \beta_1 \\ \vdots \\ \beta_s \end{array} \right) M_1^{\gamma_1} \cdots M_n^{\gamma_n}.
\]

Now, consider the converse, in which \( vx \) is given as a vector in \( \mathbb{Q}^s \). We first observe that \( B \) is embedded in a particular subset of \( \mathbb{Q}^s \). For \( 1 \leq l \leq n \), let \( d_l \) be the smallest positive integer such that \( d_l M^{-1}_l \) is an integral matrix, i.e., \( d_l \) is the least common multiple of the \( m_{i_l(s,i)} \). Let \( d = \prod d_i \). Note that if \( G \) is polycyclic, \( d = 1 \). Observe that

\[
d^{\alpha_1 + \cdots + \alpha_n} \cdot v \in \mathbb{Z}^s
\]

thus \( v \in \mathbb{Z}^{\left[ \frac{1}{d} \right]} \), in other words, we have

\[
B \subseteq \mathbb{Z}^{\left[ \frac{1}{d} \right]} \subseteq \mathbb{Q}^s.
\]

Let

\[
B_\alpha = \{ b \in B \mid w \cdot b \in \mathbb{Z}^s \text{ with } w \in Q \text{ of length at most } \alpha \}.
\]

**Lemma 1.** There is some \( \alpha \) such that for any \( i \),

\[
B \cap \frac{1}{d} \mathbb{Z}^s \subseteq B_\alpha.
\]

**Proof.** Consider first the case when \( i = 1 \). We have

\[
\mathbb{Z}^s \subseteq B \cap \frac{1}{d} \mathbb{Z}^s \subseteq \frac{1}{d} \mathbb{Z}^s
\]

thus \( (B \cap \frac{1}{d} \mathbb{Z}^s)/\mathbb{Z}^s \) is a finite group. Take a (finite) set of representatives of the cosets of \( \mathbb{Z}^s \) in \( B \cap \frac{1}{d} \mathbb{Z}^s \) and for any representative \( a \) choose \( w_a \in Q \) of smallest possible length such that \( w_a \cdot a \in \mathbb{Z}^s \). We may assume that \( w_a \) consists of non-negative powers of the \( q_i \)'s only. Let \( w \in Q \) be the element with exponent of each \( q_i \) the largest common divisor of the exponents of \( q_i \) on each \( w_a \). Then \( w \cdot a \in \mathbb{Z}^s \) for any representative \( a \), thus \( w \cdot b \in \mathbb{Z}^s \) for any \( b \in B \cap \frac{1}{d} \mathbb{Z}^s \). Therefore taking \( \alpha \) the length of \( w \) we get the result in this case.

Now, we argue by induction. Take \( b \in B \cap \frac{1}{d} \mathbb{Z}^s \). Then \( db \in B \cap \frac{1}{d} \mathbb{Z}^s \) and by induction we may assume that \( db \in B_{(i-1)\alpha} \) thus there is some \( w' \in Q \) of length at most \( (i-1)\alpha \) such that \( w' \cdot db = v \in \mathbb{Z}^s \). Then

\[
\frac{1}{d} v = w' \cdot b \in B \cap \frac{1}{d} \mathbb{Z}^s.
\]

Therefore this element lies in \( B_\alpha \), in fact if \( w \) is the element above, we have

\[
wv' \cdot b = \frac{1}{d} w \cdot v \in \mathbb{Z}^s.
\]

As \( wv' \) has length at most \( \alpha + (1 - i)\alpha = i\alpha \) we are done.

Next, assume we are given a vector \( v \in \mathbb{Q}^s \). If \( v \) does not belong to \( \mathbb{Z}^{\left[ \frac{1}{d} \right]} \) then it cannot lie in \( B \). Assume that \( v \in \mathbb{Z}^{\left[ \frac{1}{d} \right]} \). If \( v \in \mathbb{Z}^s \), then \( v \in B \) and the coordinates of \( v \) are the coefficients of the \( b_i \)'s in the normal form expression for \( v \). Otherwise, there is some \( i > 0 \) such that \( v \in \frac{1}{d} \mathbb{Z}^s \). At this point, Lemma 1 implies that \( v \in B \).
if and only if \( v \in B_\alpha \). Therefore, to check whether \( v \in B \) we only have to check whether some element in the following finite set

\[
\{M_1^{\alpha_1} \ldots M_n^{\alpha_n} \cdot v \mid \sum_{j=1}^n \alpha_j = i\alpha, \alpha_j \geq 0\}
\]

belongs to \( \mathbb{Z}^s \). Alternatively, we only have to check whether the vector

\[(M_1 M_2 \ldots M_n)^{i\alpha} v\]

lies in \( \mathbb{Z}^s \). Observe that if the answer is yes then we get, as a by-product, a normal form expression for \( v \), and if the answer is no we can conclude \( v \not\in B \).

Regarding the time complexity of the above procedure, we have to compute the \((i\alpha)\)-th power of the matrix \((M_1 M_2 \ldots M_n)\). The complexity is polynomial, specifically \(O((n-1)s^3 \log i\alpha)\) using standard matrix multiplication and efficient exponentiation.

### 4.3. Solving Linear Systems in \( \mathcal{F} \)

To finish this section and for future reference, we are going to consider a problem related to the previous one. Assume that we have an integral \( s \times s \) matrix \( N \) and a column integer vector \( u \in \mathbb{Q}^s \) and we want to determine if the linear system

\[(1) \quad NX = u\]

has some solution \( v \in \mathbb{Q}^s \) that lies in \( B \). We assume that \( N \) commutes with all the matrices \( M_l \) and under this assumption we claim that a modification of the previous procedure can solve this problem. To demonstrate, take \( P \) and \( Q \) as invertible matrices in \( \text{SL}(s, \mathbb{Z}) \) such that \( D = QNP \) is the Smith normal form of \( N \). We may do this in such a way that if \( r \) is the rank of \( N \) then the first \((s-r) \times (s-r)\) diagonal block of \( D \) is zero. Let \( D_2 \) be the last \( r \times r \) diagonal block. Our system can then be transformed into

\[(2) \quad D \tilde{X} = \begin{pmatrix} 0 & 0 \\ 0 & D_2 \end{pmatrix} \tilde{X} = Qu\]

with \( \tilde{X} = P^{-1}X \). At this point, we see that the system has some solution if and only if the first \( s-r \) entries of \( Qu \) vanish. Assume that this is the case and let \( v_2 \) be the unique solution to the system

\[(3) \quad D_2 \tilde{X}_2 = (Qu)_2\]

where the subscript 2 in \( \tilde{X} \) and \( Qu \) means that we take the last \( r \) coordinates only.

The set of all the rational solutions to \((1)\) is

\[P \left\{ \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \mid v_1 \in \mathbb{Q}^{s-r} \right\}.
\]

Observe that the columns of \( P \) give a new basis of \( \mathbb{Z}^s \) that can be used to define \( B \) instead of \( b_1, \ldots, b_s \). In this new basis the action of each \( q_l \) is encoded by the matrix \( P^{-1}M_lP \). So checking whether some solution \( X \) of the system belongs to \( B \) is equivalent to checking whether there is some product \( M \) of non-negative powers of the matrices \( M_l \) and some solution \( \tilde{X} \) to \((2)\) such that \( P^{-1}MP \tilde{X} \) belongs to \( \mathbb{Z}^s \).

Recall that we are assuming that \( N \) commutes with each \( M_l \). This implies that \( M_l \) leaves \( \text{Ker}N \) invariant. By construction \( \text{Ker}N \) is generated by the first \( s-r \) columns of \( P \) and therefore each \( PM_lP \) has the following block upper triangular form

\[P^{-1}M_lP = \begin{pmatrix} A_l & B_l \\ 0 & C_l \end{pmatrix}.
\]
Moreover, $C_l$ is just the $r \times r$ matrix associated to the action of $q_l$ in the quotient $\mathbb{Q}^s/\text{Ker} N$ in the basis obtained from the rest of the columns in $P$. By the previous procedure we may determine whether there is a suitable product $C$ of positive powers of the matrices $C_l$ such that $Cv_2 \in \mathbb{Z}^r$. Observe that if there is no such $C$ then we conclude that no solution of the original system lies in $B$. Finally, if there is such a $C$ then we may find it and form the corresponding product $M$ of the matrices $M_l$. We have

$$P^{-1}MP = \begin{pmatrix} A & B \\ 0 & C \end{pmatrix}$$

for certain $s-r \times r$ matrix $B$ and certain $s-r \times s-r$ invertible matrix $A$. Therefore

$$P^{-1}MP\tilde{X} = \begin{pmatrix} A & B \\ 0 & C \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} Av_1 + Bv_2 \\ Cv_2 \end{pmatrix}.$$

This means that now we only have to find a $v_1 \in \mathbb{Q}^{s-r}$ such that $Av_1 + Bv_2 \in \mathbb{Z}^r$. To do it, observe that it is possible to find some $v'_1$ such that $v'_1 + Bv_2 \in \mathbb{Z}^r$. Then we only have to take $v_1 = A^{-1}v'_1$.

We finish with a few words about the complexity of this last procedure. Basically, we have to compute the Smith normal form of $N$, solve the system (3), check whether the solution $v_2 + \text{Ker} N$ lies in the suitable subgroup of $\mathbb{Q}^s/\text{Ker} N$ and compute $v_1$.

For an integral matrix $N$, the time complexity of computing the Smith normal form $D$ and invertible integral matrices $P$ and $Q$ such that $QN = N$ is polynomial, specifically $O(s^6 \log sa)$, where $a$ is the maximum absolute value of the entries of $N$.

For a proof of this fact see [21] in the non-singular case and [35] for the singular one. Once we have the Smith normal form, to compute $v_2$ we only have to perform the product of $D_2^{-1}$ and $(Qu)_2$, which bounded by the computation of $P^{-1}$: $O(s^3)$. Next, we have to compute the matrices $C_l$, of which there are $n$, and each requires time on the order of $O(s^3)$, thus $O(ns^3)$. We then apply the first procedure which as we have seen takes $O((n-1)s^3 \log i\alpha)$ time. Solving for $v_2$ and $v'_1$ via Gauss elimination take $O(r^3)$ and $O((s-r)^3)$, respectively, and calculating $v_1$ is $O((s-r)^3)$. As the lower order terms involving $s$ and $r$ are dominated by the complexity of calculating the Smith normal form, the complexity of the whole process is polynomial, specifically

$$O(s^6 \log sa) + O(ns^3) + O((n-1)s^3 \log i\alpha).$$

5. Complexity of the Conjugacy Search Problem in $\mathcal{F}$

5.1. The Conjugacy Search Problem.

Let $G \in \mathcal{F}$. Given $g, g_1 \in G$ and $g \sim g_1$, the conjugacy search problem is to find $h \in G$ such that $g^h = g_1$. Let $g = bx$, $g_1 = b_1x$ and $h = cy$, with $b, c \in B$ and $x, y \in Q$. Then:

$$g^h = hgh^{-1} = cybx c^{-1} = cybx^{-1}c^{-1} \text{ (Q is abelian)} = cb^x(c^{-1})^x.$$
The component $c b^y (c^{-1})^x$ belongs to the abelian group $B$. We can write it additively as:

$$cb^y (c^{-1})^x = c(c^{-1})^x b^y = cc^{-1} b^y = c - x \cdot c + y \cdot b = y \cdot b + (1 - x) \cdot c.$$ (linear representation of $B$ and $Q$)

This means that the conjugacy search problem above is equivalent to finding $c \in B$, $y \in Q$ such that

$$b_1 = y \cdot b + (1 - x) \cdot c.$$ 

The equation above can be further reduced. Notice that the subgroup $(1 - x) \cdot B$ is invariant under any action $y \in Q$:

$$y \cdot (1 - x) \cdot B = (y(1 - x)) \cdot B \quad \text{(compatibility)}$$
$$= ((1 - x)y) \cdot B \quad \text{($B$ is abelian)}$$
$$= (1 - x) \cdot b'.$$

with $b' = y \cdot b \in B$. Thus, $Q$ acts on the quotient group $\tilde{B} = B/(1 - x) \cdot B$. Let $\tilde{b}$ be the coset in $\tilde{B}$ associated with the element $b$. The conjugacy equation is then:

$$\tilde{b}_1 = y \cdot \tilde{b}.$$ 

Finally we consider the quotient group $\tilde{B}/T$, where $T$ is the torsion subgroup of $\tilde{B}$. As $\tilde{B}/T$ is torsion-free and of finite Prüfer rank it is embeddable in $Q$. This embedding can be made more precise by using the linear representation of our groups. Let $M$ be the linear representation of $x$ and let $N = I - M \cdot Q$ as a module is flat, thus tensor products over $Q$ preserve exact sequences. Therefore, our embedding of $\tilde{B}/T$ is:

$$\tilde{B}/T \hookrightarrow \tilde{B}/T \otimes Q = B/NB \otimes Q = B \otimes \mathbb{Q}/NB \otimes Q = \mathbb{Q}^s/N\mathbb{Q}.$$ 

In order to solve the conjugacy problem, we will consider the projections of $b$ and $b_1$ in $\tilde{B}/T$, $\tilde{b} + T$ and $\tilde{b}_1 + T$ respectively. We utilize the algorithm from [3] to solve the multiple torsion-free orbit problem:

$$y \cdot (\tilde{b} + T) = \tilde{b}_1 + T.$$ 

The algorithm provides a lattice of solutions, from which we can construct a quotient group that will contain a solution to the reduced problem $\tilde{b}_1 = y \cdot \tilde{b}$. This solution is found by checking the representatives of all cosets in this quotient, the number of which is bound by the size of the torsion group $T$. In the next section, we prove that $T$ is finite, thus this search will be successful. Moreover, we prove a relationship between $T$ and the word length of an element, from which we derive the computational complexity of the procedure.

5.2. Complexity of the Conjugacy Search Problem.

We begin with some technical lemmas needed to show that the group $T$ is finite. We want also to give a bound for $|T|$ in terms of the length $l$ of $x$ as a word in the generators $q_1, \ldots, q_s$.

**Lemma 2.** Let $T$ be a torsion abelian group of finite Prüfer rank $s$. Assume that $k = \exp(T) < \infty$. Then $T$ is finite and

$$|T| \leq k^s.$$ 

Proof. Observe that the \( p \)-primary component \( T_p \) vanishes for all primes \( p \) except of possibly those primes dividing \( k \). Moreover, \( T \) cannot contain quasicyclic groups \( C_{p^\infty} \). Then, using [25, 5.1.2] (see also item 3 in page 85), we see that for any prime \( p \) dividing \( k \), \( T_p \) is a sum of at most \( s \) copies of a cyclic group of order at most the \( p \)-part of \( k \). As \( T = \oplus_{p|k} T_p \) we deduce the result. \[ \square \]

Lemma 3. Let \( N \) be a square \( s \times s \) integer matrix and \( T \) the torsion subgroup of the group \( \mathbb{Z}^s/N\mathbb{Z}^s \). Then

\[
\exp(T) \leq \sqrt{s}a^s
\]

with \( a = \max\{|a_{ij}| \mid a_{ij} \text{ entry of } N\} \).

Proof. Let \( P \) and \( Q \) be matrices in \( \text{GL}_s(\mathbb{Z}) \) such that \( PNQ \) is the Smith normal form of \( N \), i.e., it is a diagonal matrix with diagonal entries \( k_1, \ldots, k_r, 0, \ldots, 0 \), such that \( 0 < k_i \) and each \( k_i \) divides the next \( k_{i+1} \), with \( r \) being the rank of \( N \). Then

\[
\exp(T) = k_r.
\]

It is well known that the product \( k_1 \ldots k_r \) is the greatest common divisor of the determinants of the nonsingular \( r \times r \) minors of the matrix \( N \). Let \( N_1 \) be one of those minors. Then

\[
k_r \leq k_1 \ldots k_r \leq |\det N_1|
\]

Now, the determinant of the matrix \( N_1 \) is bounded by the product of the norms of the columns \( c_1, \ldots, c_r \) of the matrix (this bound is due to Hadamard, see for example [15]) so we have

\[
|\det N_1| \leq \prod_{i=1}^{r} \|c_i\| \leq \sqrt{s}^r a^r.
\]

\[ \square \]

As before, for \( 1 \leq l \leq n \), let \( d_l \) be the smallest positive integer such that \( d_lM_i^{-1} \) is an integral matrix and let \( d \) be the product of all the integers \( d_1, \ldots, d_n \).

Theorem 5.1. Let \( T \) be the torsion subgroup of the abelian group \( \tilde{B} = B/(1-x) \cdot B \). Then \( T \) is finite and

\[
|T| \leq \sqrt{s}^d a^s (a + 1)^s^2
\]

where \( L \) is the length of the element \( x \) as a word in the generators of \( Q \), \( a \) is the maximum absolute value of an entry of \( M \) and \( M \) is the matrix associated to the action of \( x \) on \( B \).

Proof. Let \( N = I - M \). Assume that \( N \) is an integral matrix relating the exponent of \( T \) with the exponent of the torsion subgroup of \( \mathbb{Z}^n/N\mathbb{Z}^n \). Let \( k \) be this last exponent and choose \( b \in B \) such that \( b \) lies in \( T \). Denote by \( m \) the order of \( b \). Observe that \( mb = Nc \) for some \( c \in B \) and that \( m \) is smallest order possible under these conditions.

Next, we claim that there is some \( q \in Q \) such that \( q \cdot b \) and \( q \cdot c \) both lie in \( \mathbb{Z}^s \). It suffices to write \( b \) and \( c \) multiplicatively using their normal forms which, as both lie in \( B \), are expressions as

\[
q_1^{-\gamma_1} \cdots q_n^{-\gamma_n} b_1^{\beta_1} \cdots b_s^{\beta_s} q_1^{\gamma_1} \cdots q_n^{\gamma_n}
\]
with $\gamma_i \geq 0$ and $q$ the product of the $q_i$'s with the largest exponent $\gamma_i$ between the exponents $b$ and $c$. Then both $q \cdot b = qbq^{-1}$ and $q \cdot c = qcq^{-1}$ have normal forms with no $q_i$ which correspond to elements of $\mathbb{Z}^n$.

Then we have $mqb = qNc = Nqc \in NZ^*$ thus $qB + NZ^*$ lies in the torsion subgroup of $\mathbb{Z}^*/NZ^*$. Therefore, $kqB \in NZ^*$. Now, let $m_1$ be the greatest common divisor of $m$ and $k$ and observe that the previous equations imply $m_1qB \in NZ^*$.

This means that for some $c_1 \in \mathbb{Z}^*$ we have $m_1B = qNc_1 = Nqc_1$, thus

$$m_1b = q^{-1}Nc_1 = Nq^{-1}c_1 = Nc_2$$

with $c_2 = q^{-1}c_1 \in B$. By the minimality of $m$ we must have $m \leq m_1$. As $m_1$ divides both $k$ and $m$ we can conclude $m = m_1 \mid k$. This implies that $k$ is also the exponent of $T$.

Next, we consider the general case when $N$ could be non-integral. As $M$ is the product of $L$ matrices in the set $\{M_1^{\pm 1}, \ldots, M_n^{\pm 1}\}$ we see that the matrix $dL^2M$ is integral and therefore so is $dL^2N$. Obviously, the group $NB/dL^2NB$ is torsion thus

$$\exp(T) \leq \exp(\text{torsion subgroup of } B/dL^2NB).$$

The matrix $dL^2N$ also commutes with the $Q$-action so what we did above implies that this last exponent equals the exponent of the torsion subgroup of $\mathbb{Z}^*/dL^2NZ^*$. From all this together with Lemma 3 we get

$$\exp(T) \leq \sqrt{s}dL^2(a + 1)^s.$$

Finally, as the group $\bar{B}$ has finite Prüfer rank, so does $T$, therefore by Lemma 2 we get the result. \qed

The next result yields a bound on the order of $T$ which is exponential in the length $L$ of $x$:

**Proposition 1.** There is a constant $K$, depending on $G$ only such that

$$|T| \leq K^L.$$

**Proof.** Let $M_1$ and $M_2$ be $s \times s$ matrices. Observe that if $h$ is an upper bound for the absolute value of the entries of both $M_1$ and $M_2$, then the maximum absolute value of an entry in the product $M_1M_2$ is bounded by $sh^2$. Repeating this argument one sees that if $x$ has length $L$ as a word in $q_1, \ldots, q_n$ and $a$ is an upper bound for the absolute value of the entries of each $M_i$, then the maximum absolute value $a$ of an entry of $M$ is bounded by

$$sL^{-1}h^L$$

Recall that by Theorem 5.1

$$|T| \leq \sqrt{s}dL^2(a + 1)^s \leq \sqrt{s}dL^2(sL^{-1}h^L + 1)^s \leq (\sqrt{s}dsh + \sqrt{s}d)^sL$$

so we only have to take $K = (\sqrt{s}dsh + \sqrt{s}d)^s$. \qed

**Remark 1.** The maximum absolute value of an entry in the matrix $M$ is bounded exponentially on $L$. Therefore, its logarithm is bounded linearly on $L$. 
Before we sketch our algorithm, let us make a few observations about the relevance of the group \( T \) to the conjugacy search problem. Recall that the problem is to find a \( y \in Q \) such that \( y \cdot \overline{b} = \overline{b}_1 \), and, as this is the search variant, \( y \) exists. Moreover, the set of solutions lie in the set
\[
\Lambda = \{ q \in Q \mid q \cdot \overline{b} - \overline{b}_1 \in T \}.
\]

Choose some fixed \( h \in \Lambda \) and observe that \( \Lambda = h Q_1 \) where \( Q_1 \leq Q \) and
\[
Q_1 = C_Q(\overline{b} + T) = \{ q \in Q \mid q \cdot \overline{b} \in T \}.
\]

Thus, for any \( q \in Q_1 \), the element \(hq \cdot \overline{b} - \overline{b}_1 \) lies in \( T \) and as \( T \) is finite there are only finitely many possibilities for its value. Moreover, we know that eventually it takes the value 0.

Let also
\[
Q_2 = C_Q(\overline{b}) = \{ q \in Q \mid q \cdot \overline{b} = \overline{b} \} = \{ q \in Q \mid q \cdot b - b \in NB \}.
\]

We obviously have \( Q_2 \leq Q_1 \) and for \( q_1, q_2 \in Q_1 \),
\[
\begin{align*}
    hq_1 \cdot \overline{b} - \overline{b}_1 = & \quad hq_2 \cdot \overline{b} - \overline{b}_1 \\
\end{align*}
\]

if and only if \( q_1 Q_2 = q_2 Q_2 \). Taking into account that \( T \) is finite we conclude that the quotient \( Q_1/Q_2 \) is finite of order bounded by \( t = |T| \). If \( \{ y_1, \ldots, y_t \} \) is a set of representatives of the cosets of \( Q_2 \) in \( Q_1 \), then some element \( y \) in the finite set
\[
\{ hy_1, \ldots, hy_t \}
\]
is the \( y \in Q \) that satisfies \( y \cdot \overline{b} = \overline{b}_1 \).

In the next lemma we prove that by \( Q_1 \) being a lattice we can produce a full set of representatives as before, including our \( y \), by taking elements solely from \( Q_1 \), Moreover, the number of steps needed is bounded in terms of \( |T| \).

**Lemma 4.** Let \( Q_2 \leq Q_1 \) with \( Q_1 \) free abelian with generators \( x_1, \ldots, x_m \), and assume that the group \( Q_1/Q_2 \) is finite of order \( t \). Then the set
\[
\Omega = \{ x_1^{\alpha_1} \cdots x_m^{\alpha_m} \mid \sum_{i=1}^{m} |\alpha_i| < t \}
\]
has order bounded by \( (2t)^m \) and contains a full set of representatives of the cosets of \( Q_2 \) in \( Q_1 \).

**Proof.** Let \( v_1, \ldots, v_m \) be generators of the subgroup \( Q_2 \), which can be viewed as points in \( \mathbb{Z}^m \). Consider the parallelogram
\[
P = \{ t_1 v_1 + \cdots + t_m v_m \mid t_i \in \mathbb{R}, 0 \leq t_i < 1 \}.
\]

Then \( \mathbb{Z}^m \cap P \) is a set of representatives of the cosets of \( Q_2 \) in \( Q_1 \) and we claim that \( P \subseteq \Omega \). Observe that for any point \( p = (\alpha_1, \ldots, \alpha_m) \) in \( \mathbb{Z}^m \cap P \) there is a path in \( \mathbb{Z}^m \cap P \) from \( (0, \ldots, 0) \) to \( p \). We may assume that the path is simple and therefore its length is bounded by \( t \). On the other hand, the length of the path is greater than or equal to \( \sum_{i=1}^{m} |\alpha_i| \) thus
\[
\sum_{i=1}^{m} |\alpha_i| \leq t.
\]
\( \square \)
We now describe the algorithm:

**Step 1:** With \( M \) and \( N \) as before, form the quotient \( V = \mathbb{Q}^s/N\mathbb{Q}^s \) and the matrices encoding the action of each \( q_i \) on \( V \). Then use the algorithm in [3] to solve the multiple orbit problem

\[ y \cdot (\bar{b} + T) = \bar{b}_1 + T. \]

This algorithm determines the full lattice of solutions.

\[ \Lambda = \{ q \in Q \mid q \cdot \bar{b} - \bar{b}_1 \in T \}, \]

Furthermore, it allows one to compute a basis \( y_1, \ldots, y_m \) of \( Q_1 \) where for some fixed \( h \in \Lambda \),

\[ Q_1 = \{ h^{-1}q \mid q \in \Lambda \}. \]

**Step 2:** Choose a full order for the elements of \( Q_1 \) that respects the word length and check for each \( q \in Q_1 \) whether \( q \cdot b - b_1 \in NB \) with respect to that order. Each check consists of trying to solve a system of linear equations. More precisely, we have to check whether the system

\[ u = NX \]

with \( u = q \cdot b - b_1 \) has some solution \( c \) in \( B \). This can be done using the procedures of Subsection 4.3. As before, if the system has a unique solution \( c \), then the first procedure can be utilized to determine whether \( c \) lies in \( B \), otherwise the second procedure is used.

The previous discussion implies that after finitely many steps we will find a \( y \) such that \( y \cdot b - b_1 \in NB \). Moreover, the number of steps is bounded by the value \( |Q_1/Q_2| \) with \( |Q_1/Q_2| \leq |T| \). At this point, it is clear that smaller groups \( Q_1/Q_2 \) will reduce the running time of the algorithm. Observe that by construction, the element \( x \) belongs to the group \( Q_2 \). In the case when \( Q \) is cyclic this yields a dramatic improvement of our bound for \( |Q_1/Q_2| \): we only have one generator, say \( q_1 \) of \( Q \) thus \( x = q_1^L \) thus \( |Q_1/Q_2| \leq |Q/Q_2| = \mathcal{L} \). Moreover, in this case step 1 in our algorithm is not needed, so we only have to perform \( \mathcal{L} \) checkings as in step 2. In this case, our algorithm coincides with the one in [5].

Back to the general case, we can now prove Proposition 1:

**Theorem 5.2.** There is a constant \( K \), depending on \( G \) only such that

\[ |T| \leq K^\mathcal{L}. \]

**Proof.** We look at the complexity of the algorithm above. Observe that step 1 only requires polynomial time. As for step 2, we have to consider an exponential (in \( \mathcal{L} \)) number of systems of linear equations of the form

\[ u = NX \]

with \( u = q \cdot b - b_1 \). Moreover, we may find (by writing \( u \) in its normal form as a word in the generators) some \( z \in Q \) such that \( zu \) is in the group generated by \( b_1, \ldots, b_s \). If \( Z \) is the matrix representing the action of \( z \) this is equivalent to the vector \( Zu \) being integral. As \( Z \) and \( N \) commute our system can be transformed into

\[ NZX = Zu. \]
Obviously, $X$ lies in $B$ if and only if $ZX$ does, thus problem is equivalent to deciding whether
\[ d^E N X_1 = d^E Z u \]
has some solution $X_1$ in $B$.

Using the last procedure of section 4.3 this can be done in a time that is polynomial on the maximum absolute value of an entry in $d^E N$. As that entry is exponential on $L$, this time is polynomial on $L$. The exponential bound in the result then follows because we are doing this a number of times which is exponential on $L$. □

Next, we consider a particular case in which the running time of algorithm is reduced to polynomial time with respect to the length $L$ of $x$.

Let $s_1, s_2 \geq 0$ be integers with $s = s_1 + s_2$ and denote
\[ \Gamma_{s_1, s_2} := \{ \text{Matrices } \begin{pmatrix} I_{s_1} & M_1 \\ 0 & I_{s_2} \end{pmatrix} \} \leq SL(s, \mathbb{Z}). \]

**Proposition 2.** Assume that for $i = 1, \ldots, n$,
\[ q_i \in \Gamma_{s_1, s_2}. \]
Then there is some constant $K$, depending on $Q$ only such that
\[ |T| \leq K L^{s_2}. \]

**Proof.** We consider the bound of Theorem 5.1. As the matrices $M_i$ lie in $\text{Mat}(s_2 \times s_1, \mathbb{Z})$, we can choose $d = 1$, thus
\[ |T| \leq \sqrt{s}(a + 1)^{s_2}, \]
where $a$ is the maximum absolute value of an entry in $M$. Observe that $M$ is a product of matrices in $\Gamma_{s_1, s_2}$ and that
\[ \begin{pmatrix} I_{s_1} & M_1 \\ 0 & I_{s_2} \end{pmatrix} \begin{pmatrix} I_{s_1} & M_2 \\ 0 & I_{s_2} \end{pmatrix} = \begin{pmatrix} I_{s_1} & M_1 + M_2 \\ 0 & I_{s_2} \end{pmatrix}. \]
Therefore, if we let $h$ be the maximum absolute value of an entry in each of the matrices $M_1, \ldots, M_n$, then $a \leq L h$ and therefore
\[ |T| \leq \sqrt{s}(a + 1)^{s_2} \leq \sqrt{s}(L h + 1)^{s_2} \leq \sqrt{s}(2 L h)^{s_2} \]
so it suffices to take $K = \sqrt{s}(2h)^{s_2}$. □

This result together with the algorithm above (recall that $d = 1$ in this case) imply the following:

**Theorem 5.3.** If
\[ Q \leq \Gamma_{s_1, s_2} \]
then the complexity of the conjugacy problem in $G$ is at most polynomial.

We finish this section with a remark on conjugator lengths. Observe that our algorithm primarily consists of identifying a suitable subgroup $Q_1$ of $Q$ and showing that, for function dependent upon the length $L$ of $x$, there exists some $y \in Q_1$ whose length is bounded by that function and which is the $Q$-component of that conjugator. Essentially, we are providing an estimation for the $Q$-conjugator length function. We make this more precise in the next result.
Corollary 1. There exists a $K$ dependent upon $G$ only such that for some conjugator $yc$, the length of $y$ is bounded by $K^L$, where $L$ is the length of $x$. In the particular case when $Q \leq \Gamma_{s_1+s_2}$, the length of $y$ is bounded by $KL^{2^2}$.

5.3. A Reduction to the Discrete Logarithm Problem.

For this subsection, we restrict ourselves to the situation of Example 4.2 where $Q$ is a multiplicative subgroup of a field $L$ such that $L : Q$ is a Galois extension and $B$ is the additive group of the subring $O_L[\bar{q}_1^{\pm}, \ldots, \bar{q}_k^{\pm}]$ which is sandwiched between $Q$ and $L$. In particular, this means that the only element in $Q$ with an eigenvalue of 1 is the identity matrix: the eigenvalues of the matrix representing an element $h \in L$ are precisely $h$ itself and its Galois conjugates and thus cannot be 1 if $h \neq 1$. Recall also that Example 4.2 includes Example 4.1.

We will keep the notation of the previous section, with elements $bx, b_1x \in G$ such that there is some $cy \in G$ with (additively)

$$b_1 = y \cdot b + (1 - x) \cdot c.$$

We may consider $y$ and $1 - x$ as elements in the field $L$. From now on we omit the $\cdot$ from our notation and use juxtaposition to denote the action. Now, $B$ also has a ring structure and $(1 - x)B$ is an ideal in $B$. Moreover, in this case the quotient ring $\bar{B} = B/(1 - x)B$ is finite (because the matrix associated to $1 - x$ is regular.) In this finite quotient ring we wish to solve the equation

$$\bar{y}b = \bar{b}_1.$$

Let $y = q_1^{t_1} \cdots q_k^{t_k}$, then solving the discrete log problem in $B/(1 - x)B$ consists of finding $t_1, \ldots, t_k$ so that

$$\bar{q}_1^{t_1} \cdots \bar{q}_k^{t_k} \bar{b} = \bar{b}_1$$

in the finite ring $\bar{B}$.

This is a special type of discrete log problem as one can observe by recalling what happens when $Q$ is cyclic: $x = q_1^s$ for some $s$ thus we have to solve

$$q_1^{t_1} \bar{v} = \bar{w}$$

in $\bar{B} = B/(1 - q_1^s)B$. $s$ attempts are sufficient (see [5].) In general, as $\bar{h} = 1$ in $\bar{B}$, $\bar{q}_1^{t_1} \cdots \bar{q}_k^{t_k} = 1$. Assume that we choose $x = q_1$. Then $\bar{q}_1 = 1$ in $\bar{B}$ then the problem is to find $t_2, \ldots, t_k$ such that

$$\bar{q}_2^{t_2} \cdots \bar{q}_k^{t_k} \bar{b} = \bar{b}_1$$

in $\bar{B}$.

Let us restrict ourselves further to the case of generalized Baumslag-Solitar groups (i.e., the groups of Example 4.1.) We identify the elements $q_i$ with the integers $m_i$ encoding their action. Assume that each $m_i$ is coprime with $1 - m_1$. As before let $y = m_1^{i_1} \cdots m_k^{i_k}$ and choose $x = m_1$. Then as each $m_i$ is coprime with $1 - m_1$

$$B/(1 - x)B = \mathbb{Z}[m_1^{\pm}, \ldots, m_k^{\pm}]/(1 - x)\mathbb{Z}[m_1^{\pm}, \ldots, m_k^{\pm}] = \mathbb{Z}/(1 - x)\mathbb{Z} = \mathbb{Z}_{1-x}.$$}

We then have to find $t_2, \ldots, t_k$ such that

$$\bar{m}_2^{t_2} \cdots \bar{m}_k^{t_k} \bar{b} = \bar{b}_1$$

in the ring of integers module $1 - m_1$. If $k = 2$ this is an instance of the discrete logarithm problem.
6. Experimental Results

Using the notation of Example 4.1, the groups tested were of the form:

\[ G = \langle q_1, q_2, b | b^{q_1} = b^{m_1}, b^{q_2} = b^{m_2}, [q_1, q_2] = 1 \rangle, \]

where \( m_1 \) and \( m_2 \) are primes. Larger primes were chosen from the list of primes \( \text{Primes2} \) in GAP. The table below indicates the primes chosen for each group, together with their respective bit lengths:

<table>
<thead>
<tr>
<th>Group</th>
<th>( m_1 )</th>
<th>( m_2 )</th>
<th>Bit Lengths (( m_1, m_2 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>(2, 3)</td>
</tr>
<tr>
<td>4</td>
<td>\text{Primes2}[362]</td>
<td>\text{Primes2}[363]</td>
<td>(48, 48)</td>
</tr>
<tr>
<td>5</td>
<td>\text{Primes2}[559]</td>
<td>\text{Primes2}[560]</td>
<td>(96, 96)</td>
</tr>
<tr>
<td>6</td>
<td>\text{Primes2}[590]</td>
<td>\text{Primes2}[591]</td>
<td>(128, 130)</td>
</tr>
</tbody>
</table>

Table 1. Primes Used for Group Construction

Two different length functions were used as heuristics for LBCS. In the first three groups, a word’s length was calculated as

\[ \sum_i |e_i|, \]

whereas in the latter three groups the length was

\[ \sum_i |\log_{10}(e_i)|. \]

As the primes become larger it becomes difficult or sometimes impossible to create elements in a range which will work for all groups. Instead, a number \( l = \log_{10} p \) was used as an approximate unit size for each of the larger groups. Random elements were then selected from ranges in multiples of \( l \).

<table>
<thead>
<tr>
<th>Group</th>
<th>( l )</th>
<th>[10, 15]</th>
<th>[20, 23]</th>
<th>[40, 43]</th>
<th>[( l, 2l )]</th>
<th>[( 2l, 3l )]</th>
<th>[( 3l, 4l )]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N/A</td>
<td>20%</td>
<td>0%</td>
<td>0%</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>N/A</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>N/A</td>
<td>N/A</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>5</td>
<td>29</td>
<td>N/A</td>
<td>N/A</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>6</td>
<td>38</td>
<td>N/A</td>
<td>N/A</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 2. LBCS Results for GMBS Groups

7. Future Work

There is additional work to be performed with respect to the groups in \( \mathcal{F} \), including determining lower bounds on the conjugacy search problem and developing a more general understanding of the separation between polynomial and exponential cases. With further development, groups in the family \( \mathcal{F} \) may be used to construct a public key exchange protocol.
References


Jonathan Gryak, PhD Program in Computer Science, CUNY Graduate Center, City University of New York
E-mail address: jgryak@gradcenter.cuny.edu