# ON THE CONJUGACY PROBLEM IN CERTAIN METABELIAN GROUPS

JONATHAN GRYAK, DELARAM KAHROBAEI, AND CONCHITA MARTINEZ-PEREZ

Abstract. We analyze the computational complexity of the conjugacy search problem in a certain family of metabelian groups. We prove that in general the time complexity of the conjugacy search problem for these groups is at most exponential. For a subfamily of groups we prove that the conjugacy search problem is polynomial. We also show that for some of these groups the conjugacy search problem reduces to the discrete logarithm problem. We provide some experimental evidence which illustrates our results probabilistically.

Contents



## 1. Introduction

In a finitely presented group $G$, the conjugacy decision problem asks if it is decidable, for any $g, g_1 \in G$, whether or not they are conjugate. Along with the word and isomorphism problems, it was one of the original group-theoretic decision problems introduced by Max Dehn in 1911. There is a variation called the conjugacy search problem, in which we assume that the two elements $g$ and $g_1$ are conjugate and are asked to find a conjugating element in $G$. There are groups for which the conjugacy decision problem is not solvable, whereas the search variant is always solvable.

In this paper we consider the conjugacy search problem for a certain family $\mathcal{F}$ of finitely presented metabelian groups. Recall that the conjugacy decision problem for finitely generated metabelian groups is solvable ([10, 4.5.6], [12]). A group $G \in \mathcal{F}$ is given by a presentation of the form

$$G = \langle q_1, \ldots, q_n, b_1, \ldots, b_s \mid [q_l, q_t] = 1, [b_i, b_j] = 1, \mathcal{R} \rangle \text{ with}$$

$$\mathcal{R} = \{q_l b_i q_l^{-1} = b_1^{m_{l(1,i)}} b_2^{m_{l(2,i)}} \ldots b_s^{m_{l(s,i)}}\}$$

where $1 \leq l, t \leq n$, $1 \leq i, j \leq s$ and the $m_{l(j,i)}$ are suitable integers so that the actions of the $q_l$ commute. Observe that $q_1, \ldots, q_n$ generate a free abelian group which we denote by $Q$ and that $b_1, \ldots, b_s$ and their $Q$-conjugated elements generate a torsion-free abelian group $B$ such that $G = B \rtimes Q$, with $B$ a normal subgroup of $G$. Throughout the paper, we will consider $B$ as a $Q$-module with left action and will denote conjugation as $b_i^{q_l} = q_l b_i q_l^{-1}$.

Under these conditions one can show that there is an embedding $B \hookrightarrow \mathbb{Q}^s$ mapping $b_1, \ldots, b_s$ to a free basis of $\mathbb{Q}^s$. This means that the group $G$ has finite Prüfer rank $n + s$. Recall that a group has finite Prüfer rank if the number of generators needed to generate any finitely generated subgroup is bounded. Observe that the action of $Q$ on $B$ can be described using integral matrices: the action of $q_l$ is encoded by the $(s \times s)$-matrix $M_l$ with entries $m_{l(j,i)}$. These matrices commute pairwise, thus $Q$ maps onto an abelian subgroup of $\mathrm{GL}(s, \mathbb{Q})$. Our group $G$ need not be polycyclic: in fact, it is polycyclic if and only if the matrices $M_l$ have integral inverses [2].

The groups in $\mathcal{F}$ enjoy strong finiteness properties, for example they are of cohomological type $\mathrm{FP}_\infty$ [4, Proposition 1] (see also the proof of Theorem 8 in the same paper) and constructible, meaning that can be constructed in finitely many steps from the trivial group using finite index extensions and ascending HNN-extensions. In fact, our groups are iterated, strictly ascending HNN-extensions of the group $\mathbb{Z}^s$. Moreover, any constructible torsion-free split metabelian group of finite Prüfer rank has this form and any metabelian group of finite Prüfer rank can be embedded in a metabelian constructible group [4].

In Section 3, we analyze the computational complexity of an algorithm to solve the conjugacy search problem for groups $G \in \mathcal{F}$. Particularly, we prove the following two theorems:

**Theorem 1.1.** *For any $G \in \mathcal{F}$, the time complexity of the conjugacy search problem for conjugate elements $g, g_1 \in G$ is at most exponential in the length of $g$ and $g_1$.*

**Theorem 1.2.** *Fix $s_1, s_2 \geq 0$ with $s = s_1 + s_2$ and assume that for $1 \leq i \leq n$,*

$$M_i \in \left\{ \textit{Matrices } \begin{pmatrix} I_{s_1} & A \\ 0 & I_{s_2} \end{pmatrix} \textit{ with } A \in \mathit{Mat}(s_1 \times s_2, \mathbb{Z}) \right\}.$$

*Let $G \in \mathcal{F}$ be defined using the matrices $M_i$. Then the time complexity of the conjugacy search problem in $G$ is polynomial.*

As a corollary, we also deduce some consequences about conjugator lengths (Corollary 3.9).

There are some particular cases in which one can show that the conjugacy search problem for our groups reduces to a type of discrete logarithm problem, which is discussed in Subsection 3.4. In particular, this applies to generalized metabelian Baumslag-Solitar groups of the form:

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

Finally in the last section we perform experiments on the generalized metabelian Baumslag-Solitar groups as above. Such experiments utilize a heuristic algorithm called length-based conjugacy search, which is adapted from an attack of the same name originating in group-based cryptography. Our experiments indicate that these generalized metabelian Baumslag-Solitar groups are resistant to such search algorithms, i.e., probabilistically the conjugator cannot be found given sufficient time.

## 2. Split Metabelian Groups of Finite Prüfer Rank

Let $G$ be a split extension $G = B \rtimes Q$ with both groups $B$ and $Q$ abelian. We use multiplicative notation for the whole group $G$ but additive notation for $B$. So if $c \in B$, $x \in Q$, the action of the element $x$ maps $c$ to

$$x \cdot c \text{ with additive notation or,}$$
$$c^x = xcx^{-1} \text{ with multiplicative notation.}$$

Assume that we have conjugate elements $g, g_1 \in G$ and we want to solve the conjugacy search problem for $g$, $g_1$, i.e., we want to find $h \in G$ such that

$$g^h = g_1.$$

Let $g = bx$, $g_1 = b_1x_1$ and $h = cy$ with $b, b_1, c \in B$, $x, x_1, y \in Q$, then

$$b_1x = g_1 = g^h = hgh^{-1} = cybxy^{-1}c^{-1} = cb^y(c^{-1})^x x.$$

Therefore, we conclude that $x = x_1$, and from now on we denote this element solely by $x$. The element $cb^y(c^{-1})^x$ belongs to the abelian group $B$. We write it additively

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

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

$$b_1 = y \cdot b + (1 - x) \cdot c \tag{1}$$

when $b, b_1 \in B$ and $x \in Q$ are given.

As stated in the introduction, the groups we are considering admit a presentation of the form

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

Recall also that we are denoting by $Q$ the group generated by $q_1, \ldots, q_n$, and by $B$ the group generated as a normal subgroup of $G$ by $b_1, \ldots, b_s$. One of the main advantages of these groups is that they admit a set of normal forms:

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

with $\alpha_1, \ldots, \alpha_n \geq 0$ and such that whenever $\alpha_i \neq 0$, the element $q_i^{-1} b_1^{\beta_1} \ldots b_s^{\beta_s} q_i$ does not belong to the subgroup generated by $b_1, \ldots, b_s$. There is an efficient algorithm (collection) to transform any word in the generators to the corresponding normal form: given an arbitrary word in the generating system, use the relators to move all of the instances of $q_i$ with negative exponents to the left and all the instances of $q_i$ with positive exponents to the right (see example 2.1).

**Example 2.1.** *Generalized Metabelian Baumslag-Solitar Groups.* Let $m_1, \dots, m_n$ be positive integers. We call the group given by the following presentation a *generalized metabelian Baumslag-Solitar group*

$$G = \langle q_1, \dots, q_n, b \mid b^{q_i} = b^{m_i},\, 1 \leq i, j \leq n, [q_i, q_j] = 1 \rangle.$$

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

Let us examine how collection works for these groups. Consider the group

$$G = \langle q_1, q_2, b \mid b^{q_1} = b^2, b^{q_2} = b^3, [q_1, q_2] = 1 \rangle,$$

with $G \cong \mathbb{Z}\left[\frac{1}{2}, \frac{1}{3}\right] \rtimes \mathbb{Z}^2$, and an uncollected word in $G$:

$$w = q_1^{-1} q_2 b^{-1} q_1 q_2^{-1}.$$

As the $q_l$'s commute we have

$$w = q_1^{-1} q_2 b^{-1} q_2^{-1} q_1.$$

We then apply the negated form of the relation $b^{q_2} = b^3$ to yield the reduced word in normal form:

$$w = q_1^{-1} q_2 q_2^{-1} b^{-3} q_1 = q_1^{-1} b^{-3} q_1.$$

**Example 2.2.** Let $L : \mathbb{Q}$ be a Galois extension of degree $n$ and fix an integral basis $\{u_1, \dots, u_s\}$ of $L$ over $\mathbb{Q}$. Then $\{u_1, \dots, u_s\}$ freely generates the maximal order $\mathcal{O}_L$ as a $\mathbb{Z}$-module. Now, we choose integral elements, $q_1, \dots, q_n$, generating a free abelian multiplicative subgroup of $L - \{0\}$. Each $q_l$ acts on $L$ by left multiplication and using the basis $\{u_1, \dots, u_s\}$, we may represent this action by means of an integral matrix $M_l$. Let $B$ be the smallest sub $\mathbb{Z}$-module of $L$ closed under multiplication with the elements $q_l$ and $q_l^{-1}$ and such that $\mathcal{O}_L \subseteq B$, i.e.,

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

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

2.1. **Linear Representations.**

As noted previously, $B$ embeds in $\mathbb{Q}^s$, therefore any element $g \in G$ can be represented by a pair $(v, x)$ where $x \in Q$ and $v \in \mathbb{Q}^s$ is a vector. We will omit brackets and simply write $vx$. It will be useful in the next section to use this representation of our elements since this will allow us to use some linear algebra. Here we consider the problem of swapping between this linear representation and the usual representation of group elements as words in the generators of $G$.

Assume first that $g$ is given as a word in the generators. We may assume that $g$ is in normal form:

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

then the following word also yields $g$:

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

In the semidirect representation we have $g = bx$ with $x = q_1^{\gamma_1-\alpha_1} \dots q_n^{\gamma_n-\alpha_n}$ and additively

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

To represent $b$ as a vector $v \in \mathbb{Q}^s$, recall that the action of each $q_l$ is encoded by the integral matrix $M_l$, then

$$v = M_1^{-\alpha_1} \cdots M_n^{-\alpha_n} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_s \end{pmatrix}.$$

The complexity of the above procedure using Gaussian elimination for inverses, standard matrix multiplication, and efficient exponentiation is:

$$O((n-1)[s^3 + s^3 \log \max_l(\alpha_l) + s^3 \log \max_l(\gamma_l - \alpha_l)] + s^2 + s^3).$$

Now, consider the converse, in which we have $vx$ with $v$ given as a vector in $\mathbb{Q}^s$. In order to convert $v$ into its normal form, we first show that $B$ is embedded in a particular subset of $\mathbb{Q}^s$. Testing for membership in this subset will then yield an element $b \in B$ in normal form as desired. In the following discussion, we identify $B$ with its image in $\mathbb{Q}^s$ and the group generated by $b_1 \dots, b_s$ with $\mathbb{Z}^s$.

For $1 \le l \le n$, let $d_l$ be the smallest positive integer such that $d_l M_l^{-1}$is an integral matrix, i.e., $d_l$ is the lowest common denominator of the matrix entries $m_{l(s,i)}$. Let $d = \prod_l d_l$. Note that if $G$ is polycyclic, $d = 1$. Observe that for any $v \in B$,

$$d^{\alpha_1+\dots+\alpha_n} v \in \mathbb{Z}^s$$

thus $v \in \mathbb{Z}[\frac{1}{d}]^s$, in other words, we have

$$B \subseteq \mathbb{Z}[\tfrac{1}{d}]^s \subset \mathbb{Q}^s.$$

**Remark 2.3.** This implies that for any $v \in B$, if $i$ is be the smallest positive integer such that $d^i v$ lies in $\mathbb{Z}^s$, then $i$ is bounded by twice the length of $v$ as a word in normal form.

$B$ can also be constructed from $\mathbb{Z}^s$ and $M = \prod_l M_l$. Observe that

$$\mathbb{Z}^s \subseteq M^{-1}\mathbb{Z}^s \subseteq \dots \subseteq M^{-j}\mathbb{Z}^s \subseteq M^{-j-1}\mathbb{Z}^s \subseteq \dots \subseteq B$$

and in fact $B = \cup_{j=0}^{\infty} M^{-j}\mathbb{Z}^s$. To check this, note that any vector in $B$ has the form $M_1^{-\beta_1} \dots M_n^{-\beta_n} u$ for some $u \in \mathbb{Z}^s$ and certain $\beta_1, \dots, \beta_n \ge 0$. Let $\beta = \max\{\beta_1, \dots, \beta_n\}$, then

$$M_1^{-\beta_1} \dots M_n^{-\beta_n} u = M^{-\beta} M_1^{\beta-\beta_1} \dots M_n^{\beta-\beta_n} u = M^{-\beta} w$$

where $w = M_1^{\beta-\beta_1} \dots M_n^{\beta-\beta_n} v$ lies in $\mathbb{Z}^s$. Consequently, if $q = q_1 \dots q_n$, then the group $B \rtimes \langle q \rangle$ is a strictly ascending HNN extension of $\mathbb{Z}^s$.

**Lemma 2.4.** *There is some $\alpha$ depending on $G$ only such that for any $i$,*

$$B \cap \frac{1}{d^i}\mathbb{Z}^s \subseteq M^{-i\alpha}\mathbb{Z}^s.$$

*Moreover $\alpha \le s \log d$.*

*Proof.* Consider first the case when $i = 1$. We have $\mathbb{Z}^s \subseteq \frac{1}{d}\mathbb{Z}^s$ and

$$\mathbb{Z}^s \subseteq M^{-1}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s \subseteq \ldots \subseteq M^{-j}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s \subseteq M^{-j-1}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s \subseteq \ldots \subseteq \frac{1}{d}\mathbb{Z}^s.$$

As the quotient of $\frac{1}{d}\mathbb{Z}^s$ over $\mathbb{Z}^s$ is the finite group $\mathbb{Z}_d \times \ldots \times \mathbb{Z}_d$ of order $d^s$, this sequence stabilizes at some degree, say $\alpha$. Then $B \cap \frac{1}{d}\mathbb{Z}^s = M^{-\alpha}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s$ and

$$B \cap \frac{1}{d}\mathbb{Z}^s \subseteq M^{-\alpha}\mathbb{Z}^s$$

as desired. Moreover, we claim that it stabilizes precisely at the first $\alpha$ such that

$$M^{-\alpha}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s = M^{-\alpha-1}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s.$$

To demonstrate, let $b \in M^{-\alpha-2}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s$. Then $Mb \in M^{-\alpha-1}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s = M^{-\alpha}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s$ thus $b \in M^{-\alpha-1}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s = M^{-\alpha}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s$. Repeating the argument implies that for all $\beta > \alpha$,

$$M^{-\alpha}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s = M^{-\beta}\mathbb{Z}^s \cap \frac{1}{d}\mathbb{Z}^s.$$

As a consequence, $\alpha$ is bounded by the length of the longest chain of proper subgroups in $\mathbb{Z}_d \times \ldots \times \mathbb{Z}_d$, i.e., $\alpha \leq \log(d^s) = s \log d$.

Now we argue by induction. Let $b \in B \cap \frac{1}{d^i}\mathbb{Z}^s$, then $db \in B \cap \frac{1}{d^{i-1}}\mathbb{Z}^s$ and by induction we may assume that $db \in M^{-(i-1)\alpha}\mathbb{Z}^s$, thus $M^{(i-1)\alpha}db = v \in \mathbb{Z}^s$. Then

$$\frac{1}{d}v \in B \cap \frac{1}{d}\mathbb{Z}^s \subseteq M^{-\alpha}\mathbb{Z}^s.$$

Therefore

$$M^{\alpha}M^{(i-1)\alpha}b = \frac{1}{d}M^{i\alpha}v \in \mathbb{Z}^s$$

and $b \in M^{-i\alpha}\mathbb{Z}^s$. □

It is easy to construct examples with $\alpha \neq 1$:

**Example 2.5.** Consider the group $G \in \mathcal{F}$ given by the following presentation:

$$G = \langle b_i, q_i \mid b_1^{q_1} = b_1^2, b_2^{q_2} = b_2^4, b_3^{q_3} = b_3^{16}, b_i^{q_j} = b_i \text{ for } i \neq j, [b_i, b_j] = 1, [q_i, q_j] = 1\rangle,$$

with $1 \leq i, j \leq 3$.

From the presentation above $s = 3$. The linear representations of the $q_l$'s (and their product $M$) are then:

$$M_1 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix} M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 16 \end{bmatrix}; M = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 16 \end{bmatrix}.$$

From visual inspection of $M$ it is clear that $d = 16$. Moreover, it is easy to check that $\frac{1}{16}\mathbb{Z}^s \subseteq B$ and that in fact

$$\frac{1}{16}\mathbb{Z}^s = \frac{1}{16}\mathbb{Z}^s \cap B \subseteq M^{-4}\mathbb{Z}^s$$

and 4 is smallest possible in these conditions thus $\alpha = 4$.

In determining whether a vector $v \in \mathbb{Q}^s$ lies in $B$, it is clear from the previous discussion that a necessary condition is that $v$ belongs to $\mathbb{Z}[\frac{1}{d}]^s$, and therefore there exists an $i > 0$ such that $v \in \frac{1}{d^i}\mathbb{Z}^s$. In the particular case when $v$ is integral, then $v \in B$ and the coordinates of $v$ are the exponents of the $b_j$'s in the normal form expression for $v$.

If $v$ is strictly rational, we can perform the following procedure to check whether $d \in \frac{1}{d^i}\mathbb{Z}^s$ for some $i$ and to find the smallest possible such $i$. First, compute the least common multiple of the denominators of the entries of $v$. By reducing if necessary, we may assume that

$$v = \frac{1}{m}(v_1, \ldots, v_s)$$

with the $v_j$ integers so that no prime divides all of $m, v_1, \ldots, v_s$ simultaneously. We then claim that $d^i v$ is integral if and only if $d^i = 0$ modulo $m$. For assume that $d^i v$ is integral (the other direction is obvious). This implies that $m$ divides $d^i v_j$ for $j = 1, \ldots, s$ and the assumption on $m$ and the $v_j$'s implies that $m$ divides $d^i$ as we wanted.

This claim implies that we only have to check whether some $d^i = 0$ modulo $m$. If explicit factorizations of $m$ and $d$ are not available, we need only compute $d^i$ for $1 \leq i \leq m$. If there is no such $i$, then $v$ does not belong to $\mathbb{Z}[\frac{1}{d}]^s$. Otherwise observe that $i \leq m$.

**Lemma 2.6.** *Let $v \in \mathbb{Z}[\frac{1}{d}]^s$ and $i$ the smallest possible integer such that $d^i v$ is integral. Then $v \in B$ if and only if*

$$M^{is\lfloor \log d \rfloor} v \in \mathbb{Z}^s$$

*where $M = M_1 M_2 \ldots M_n$. The complexity of this computation is polynomial, specifically $O((n-1)s^3 \log is\lfloor \log d \rfloor)$. (Alternatively, the same result holds true but with $\alpha$ instead of $s\lfloor \log d \rfloor$).*

*Proof.* Lemma 2.4 implies that $v \in B$ if and only if $M^{i\alpha} v$ is integral. Thus if $v \in B$,

$$M^{is\lfloor \log d \rfloor} v = M^{(is\lfloor \log d \rfloor - i\alpha)} M^{i\alpha} v$$

is integral because $is\lfloor \log d \rfloor - i\alpha \geq 0$. The converse is obvious.

Regarding the time complexity, we have to compute the $(is\lfloor \log d \rfloor v)$-th power of the matrix $M$. The complexity estimation is obtained using standard matrix multiplication and efficient exponentiation. $\square$

**Remark 2.7.** Note that the exponent $is\lfloor \log d \rfloor$ is just an upper bound and often a much smaller value suffices to obtain an expression of a given $v \in B$ as product of conjugated of the $b_i$'s. Consider for example the group of Example 2.5 and the vector $v \in \mathbb{Q}^3$:

$$v' = \left[\frac{1}{32}, \frac{3}{64}, \frac{5}{16}\right].$$

Here, $i = 2$, $s = 3$, $d = 16$ and $d =$ thus $is\lfloor \log d \rfloor = 24$ but note that already $M^5 v$ is integral.

2.2. **Solving Linear Systems.**

To finish this section and for future reference, we are going to consider the following problem. Assume that we have a square $s \times s$ integral matrix $N$ that commutes with all the matrices $M_l$ and a column rational vector $u \in \mathbb{Q}^s$, and we want to determine if the linear system

$$NX = u \tag{2}$$

has some solution $v \in \mathbb{Q}^s$ that lies in $B$. To solve this problem, we will use a standard technique to solve these kind of systems in $\mathbb{Z}$. The Smith normal form for $N$ is a diagonal matrix $D$ with diagonal entries $k_1, \dots, k_r, 0, \dots, 0$, such that $0 < k_j$ and each $k_j$ divides the next $k_{j+1}$, with $r$ being the rank of $N$. Moreover, there are invertible matrices $P$ and $Q$ in $\mathrm{SL}(s, \mathbb{Z})$ such that $D = QNP$.

We set

$$a = \max\{|a_{lj}| \mid a_{lj} \text{ entry of } N\}.$$

**Lemma 2.8.** *Let $N$ be any integral $s \times s$ matrix and let $D = diag(k_1, \dots, k_r, 0, \dots, 0)$ be its Smith normal form, then*

$$k_1 \dots k_r \leq \sqrt{s} a^s$$

*Proof.* It is well known that the product $k_1 \dots k_r$ 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

$$k_r \leq k_1 \dots k_r \leq |\det N_1|.$$

Now, the determinant of the matrix $N_1$ is bounded by the product of the norms of the columns $c_1, \dots, c_r$ of the matrix (this bound is due to Hadamard, see for example [8]) so we have

$$|\det N_1| \leq \prod_{j=1}^{r} \|c_j\| \leq \sqrt{r}^r a^r.$$

$\square$

Recall that we are assuming that $N$ commutes with all the matrices $M_l$. Under this assumption we claim that we can solve the problem above by using Lemma 2.6. To demonstrate, let $P$ and $Q$ be invertible matrices in $\mathrm{SL}(s, \mathbb{Z})$ such that $D = QNP = \mathrm{diag}(k_1, \dots, k_r, 0, \dots, 0)$ is the Smith normal form of $N$. Our system can then be transformed into

$$D\tilde{X} = \begin{pmatrix} 0 & 0 \\ 0 & D_2 \end{pmatrix} \tilde{X} = Qu \tag{3}$$

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

$$D_2 \tilde{X}_2 = (Qu)_2 \tag{4}$$

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

$$v_2 = D_2^{-1}(Qu)_2.$$

The set of all the rational solutions to (2) is

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

Equivalently, this set can be written as

$$v + \mathrm{Ker}N \text{ where } v = P\begin{pmatrix} 0 \\ v_2 \end{pmatrix}.$$

Observe that the columns of $P$ give a new basis of $\mathbb{Z}^s$ that can be used to define $B$ instead of $b_1, \dots, b_s$. In this new basis the action of each $q_l$ is encoded by the matrix $P^{-1}M_lP$. The fact that $N$ commutes with each $M_l$ implies that $M_l$ leaves $\mathrm{Ker}N$ (setwise) invariant. By construction, $\mathrm{Ker}N$ is generated by the first $s-r$ columns of $P$ and therefore each $P^{-1}M_lP$ 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 with the action of $q_l$ in the quotient $\mathbb{Q}^s/\mathrm{Ker}N$, written in the basis obtained from the last $r$ columns of $P$.

**Proposition 2.9.** *A solution to the system (3) exists in $B$ if and only if $v_2 \in \mathbb{Z}[\frac{1}{d}]^r$ and*

$$C^{ir\lfloor \log d \rfloor} v_2 \in \mathbb{Z}^r,$$

*with $C = \prod_l C_l$ and $i$ the smallest possible integer such that $d^i v_2$ is integral. (We can use $s$ instead of $r$).*

*Proof.* Assume first that $C^{ir\lfloor \log d \rfloor} v_2 \in \mathbb{Z}^r$, with $i$ as above. We have

$$P^{-1}M^{i\alpha}P = \begin{pmatrix} A & S \\ 0 & C^{ir\lfloor \log d \rfloor} \end{pmatrix}$$

for certain $(s-r) \times r$ matrix $S$ and certain $(s-r) \times (s-r)$ invertible matrix $A$, with $M = \prod_l M_l$ as before. Therefore

$$P^{-1}M^{ir\lfloor \log d \rfloor}P\tilde{X} = \begin{pmatrix} A & S \\ 0 & C^{ir\lfloor \log d \rfloor} \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} Av_1 + Sv_2 \\ C^{ir\lfloor \log d \rfloor} v_2 \end{pmatrix}.$$

This means that now we only have to find a $v_1 \in \mathbb{Q}^{s-r}$ such that $Av_1 + Sv_2 \in \mathbb{Z}^s$. To do it, observe that it suffices to take $v_1 = -A^{-1}Sv_2'$.

Conversely, assume that some $P\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ lies in $B$. Then some product of positive powers of the $M_l$'s transforms $P\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ into an integral vector, thus there is a product of the $C_l$'s that transforms $v_2$ into an integral vector. We may use now Lemma 2.6 applied to $\mathbb{Q}^r = \mathbb{Q}^s/\mathrm{Ker}N$ with respect to the action of the matrices $C_l$ to conclude that $v_2 \in \mathbb{Z}[\frac{1}{d}]^r$ and

$$C^{ir\lfloor \log d \rfloor} v_2 \in \mathbb{Z}^r,$$

with $i$ the smallest possible integer such that $d^i v_2$ is integral. (Note that $dC_l^{-1}$ is integral so we can use the same $d$ for this quotient as for the original group.)

□

**Remark 2.10.** Observe that, as $N$ is integral, a necessary condition for (2) to have some solution in $B$ is that $u$ lie in $\mathbb{Z}[\frac{1}{s}]$. Let $i_0$ be such that $d^{i_0}u$ is integral. Then $d^{i_0}\det(D_2)v_2$ is also integral. If this lies in $\mathbb{Z}[\frac{1}{d}]^s$, it means that for some $i_1$ such that $d^{i_1} \leq \det(D_2)$, we have that $d^{i_0+i_1}v_2$ is integral. By Lemma 2.8 $\det(D_2) \leq \sqrt{s}a^s$, thus $i_1 \leq \sqrt{s}a^s$. As a consequence, if $i$ is as in Proposition 2.9, we have

$$i \leq i_0 + \sqrt{s}a^s.$$

Now we are ready to show:

**Proposition 2.11.** *There is an algorithm to decide whether the system (2) has some solution in $B$ and to compute that solution. The complexity of this algorithm is polynomial, specifically*

$$O(s^6 \log sa + (s-r)^5 + (s-r)^3 + (n-1)[s^3 \log is \log d + 1] + r^3).$$

*where $i \leq i_0 + \sqrt{s}a^s$ and $i_0$ is such that $d^{i_0}u$ is integral. (If there is no such $i_0$ then the system has no solution in $B$).*

*Proof.* The algorithm has been described above. In summary, we have to transform the original system using the Smith normal form for $N$, compute $v_2$ and the matrices $C_l$ and $C = C_1 \dots C_n$, and then check whether $v_2$ lies in $\mathbb{Z}[\frac{1}{d}]$. If it does, we may either compute $i$ such that $d^i v_2$ is integral or estimate $i$ as $i_0 + i_1$ (see Remark 2.10). Then we compute

$$C^{ir\lfloor \log d \rfloor} v_2$$

and check whether it is integral or not. To estimate the complexity of this procedure observe that for an integral matrix $N$, the time complexity of computing the Smith normal form $D$ and invertible integral matrices $P$ and $Q$ such that $QNP = D$ 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 [9] in the non-singular case and [13] 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 $(Qu)_2$: $O(r^3)$. Next, we have to compute the matrices $C_l$, which requires $n-1$ matrix multiplications, thus $O((n-1)s^3)$. We then check whether $C^{ir\lfloor \log d \rfloor} v_2$ is integral which takes at most $O((n-1)s^3 \log is \log d)$ time. Solving for $v_2$ and $v_1'$ via Gaussians elimination take $O(r^3)$ and $O((s-r)^3)$, respectively, and calculating $v_1$ is $O((s-r)^5$. The overall time complexity is then the sum of of the above operations, which is denoted in the proposition. Note that the lower order terms involving $s$ and $r$ are dominated by the complexity of calculating the Smith normal form. □

## 3. On the Complexity of the Conjugacy Problem

### 3.1. An Algorithm for Split Metabelian Groups of Finite Prüfer Rank.

In this section, we describe and analyze the complexity of an algorithm to solve the conjugacy search problem in the groups under consideration, i.e., the groups admitting a presentation as in Section 2. As we have seen above, the problem is equivalent to the problem of finding $c \in B$, $y \in Q$ such that

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

where $b_1, b \in B$, $x \in Q$ are given. Throughout this section we use additive notation for elements in $B$. When useful, elements in $Q$ will be identified with the matrices encoding their action. Elements in $B$ will be represented either as words in the generators of $G$ or as vectors in $\mathbb{Q}^s$, recalling that we may switch from one representation to the other in polynomial time.

Observe that $(1-x) \cdot B$ is a $Q$-invariant subgroup of $B$. Therefore, $Q$ acts on the quotient group $\bar{B} = B/((1-x) \cdot B)$. We use $^-$ to denote the coset in $\bar{B}$ associated with a given element. From the equation above we get

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

in $\bar{B}$. We let $M_x$ be the rational matrix associated with the action of $x$ on $B$ (with respect to the set $b_1, \ldots, b_s$), $N = I - M_x$, and use $NB$ to denote $(1-x) \cdot B$ thus $\bar{B} = B/NB$.

Let $T$ be the torsion subgroup of $\bar{B}$. Obviously, it is invariant under the $Q$ action, thus $Q$ factors through $\bar{B} \twoheadrightarrow \bar{B}/T$ and acts on the torsion-free group $\bar{B}/T$. As $\bar{B}/T$ is torsion-free and of finite Prüfer rank, it can be embedded in $\mathbb{Q}^{s_1}$ for some $s_1$. In fact, as $\mathbb{Q}$ is flat we have

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

So one can perform this embedding and find the matrices associated with the action of each of the elements $q_l$ in this quotient.

The idea of the algorithm is to decompose the problem of finding the conjugator $h$ into two problems - one of them is a multiple orbit problem in a vector space and the other is a type of discrete log problem. For the first we take advantage of the polynomial time solution in [3] and for the latter we provide an upper bound for its complexity, which is essentially dependent upon the size of the subgroup $T$.

*Description of the algorithm.*

**Step 1**: With $M_x$ and $N$ as before, form the quotient $V = \mathbb{Q}^s/N\mathbb{Q}^s$ and the matrices encoding the action of each $q_l$ on $V$. Consider the projections $\bar{b} + T$ and $\bar{b}_1 + T$ of $b$ and $b_1$ in $\bar{B}/T$ and see them as elements in $V$ (via the embedding $\bar{B}/T \hookrightarrow V$). 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**: Order the elements of $Q_1$ according to word length. For each $q \in Q_1$ check whether $q \cdot b - b_1 \in NB$. 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 Proposition 2.9.

Of course, a priori this procedure may never halt. But we will show that is not the case: the number of iterations of Step 2 is bounded by the size of the group $T$, which will be shown to be finite. We can now be more explicit. Recall that the problem is to find a $y \in Q$ such that $y \cdot \bar{b} = \bar{b}_1$, and, as this is the search variant of the conjugacy, $y$ exists. Moreover, all the solutions lie in the set

$$\Lambda = \{q \in Q \mid q \cdot \bar{b} - \bar{b}_1 \in T\}.$$

Choose some fixed $h \in \Lambda$ and observe that $\Lambda = hQ_1$ where $Q_1 \leq Q$ and

$$Q_1 = C_Q(\bar{b} + T) = \{q \in Q \mid q \cdot \bar{b} - \bar{b} \in T\}.$$

Thus, for any $q \in Q_1$, the element $hq \cdot \bar{b} - \bar{b}_1$ 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

$$Q_2 = C_Q(\bar{b}) = \{q \in Q \mid q \cdot \bar{b} = \bar{b}\} = \{q \in Q \mid q \cdot b - b \in NB\}.$$

We obviously have $Q_2 \leq Q_1$ and for $q_1, q_2 \in Q_1$,

$$hq_1 \cdot \bar{b} - \bar{b}_1 = hq_2 \cdot \bar{b} - \bar{b}_1$$

if and only if $q_1 Q_2 = q_2 Q_2$. As $T$ will be shown to be finite we conclude that the quotient $Q_1/Q_2$ is of finite 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 $y \cdot \bar{b} = \bar{b}_1$.

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 3.1.** *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*

$$\Omega = \{x_1^{\alpha_1} \ldots x_m^{\alpha_m} \mid \sum_{j=1}^{m} |\alpha_j| < t\}$$

*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

$$P = \{t_1 v_1 + \ldots + t_m v_m \mid t_j \in \mathbb{R}, 0 \leq t_j < 1\}.$$

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 $\sum_{j=1}^{m} |\alpha_j|$ thus

$$\sum_{j=1}^{m} |\alpha_j| \leq t.$$

□

The number of iterations of Step 2 is bounded by the value $|Q_1/Q_2|$. 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, if $x = q_1^{\mathcal{L}}$, $|Q_1/Q_2| \leq |Q/Q_2| = \mathcal{L}$. Moreover, in this case Step 1 in our algorithm is not needed, so we only have to perform $\mathcal{L}$ iterations of Step 2, and our algorithm coincides with the one in [5].

3.2. **On the Subgroup $T$.**
. We proceed to showing that $T$ is indeed finite, and to bound its size by the length of $x$ as a word in the generators $q_1, \dots, q_s$.

Recall that 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). The following lemma is well known, but we include it here for completeness:

**Lemma 3.2.** *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 as $T$ has finite exponent, its $p$-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 [10, 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. □

**Lemma 3.3.** *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 $D = \operatorname{diag}(k_1, \dots, k_r, 0, \dots, 0)$ be the Smith normal form of $N$. Then

$$\exp(T) = k_r \leq k_1, \dots, k_r$$

so it suffices to apply Lemma 2.8.

□

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

**Theorem 3.4.** *Let $T$ be the torsion subgroup of the abelian group $\bar{B} = B/(1-x)\cdot B$. Then $T$ is finite and*

$$|T| \leq \sqrt{s}^s d^{\mathcal{L}s^2} (a+1)^{s^2}$$

*where $\mathcal{L}$ is the length of the element $x$ as a word in the generators of $Q$, $a$ is the maximum absolute value of an entry in $M_x$, the matrix associated with the action of $x$ on $B$.*

*Proof.* Let $N = I - M_x$. Assume first that $M_x$ is an integral matrix, so the same happens with $N$. We want to relate the exponent of $T$ with the exponent of the torsion subgroup of $\mathbb{Z}^s/N\mathbb{Z}^s$. Let $k$ be this last exponent and choose $b \in B$ such that $1 \neq \bar{b}$ lies in $T$. Denote by $m > 0$ the order of $\bar{b}$. Observe that $mb = Nc$ for some $c \in B$ and that $m$ is the smallest possible under these conditions.

Next, choose $q \in Q$ such that $q \cdot b$ and $q \cdot c$ both lie in $\mathbb{Z}^s$. To find such a $q$ it suffices to write $b$ and $c$ multiplicatively using their normal forms and take as $q$ a product of the $q_l$'s with big enough exponents.

Then we have $m(q \cdot b) = q \cdot Nc = N(q \cdot c) \in N\mathbb{Z}^s$ thus $q \cdot b + N\mathbb{Z}^s$ lies in the torsion subgroup of $\mathbb{Z}^s/N\mathbb{Z}^s$. Therefore, $k(q \cdot b) \in N\mathbb{Z}^s$. Now, let $m_1$ be the greatest common divisor of $m$ and $k$ and observe that the previous equations imply $m_1(q\cdot b)\in N\mathbb{Z}^s$. This means that for some $c_1\in\mathbb{Z}^s$ we have $m_1(q\cdot b)=Nc_1$, thus

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

with $c_2=q^{-1}\cdot c_1\in B$. By the minimality of $m$ we must have $m\le m_1$. 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 $M_x$ is the product of $\mathcal{L}$ matrices in the set $\{M_1^{\pm 1},\dots,M_n^{\pm}\}$ we see that the matrix $d^{\mathcal{L}}M_x$ is integral and therefore so is $d^{\mathcal{L}}N$. Obviously, the group $NB/d^{\mathcal{L}}NB$ is torsion thus

$$\exp(T)\le\exp(\text{torsion subgroup of } B/d^{\mathcal{L}}NB).$$

The matrix $d^{\mathcal{L}}N$ also commutes with the $Q$-action so what we did above implies that this last exponent equals the exponent of the torsion subgroup of $\mathbb{Z}^s/d^{\mathcal{L}}N\mathbb{Z}^s$. From all this together with Lemma 3.3 and using that the biggest absolute value of an entry of $d^{\mathcal{L}}$ is bounded by $d^{\mathcal{L}}N$ we get

$$\exp(T)\le\sqrt{s}d^{\mathcal{L}s}(a+1)^s.$$

Finally, as the group $\bar{B}$ has finite Prüfer rank, so does $T$, therefore by Lemma 3.2 we get the result. □

**Remark 3.5.** The maximum absolute value of an entry in the matrix $M_x$ is bounded exponentially on $\mathcal{L}$. Therefore, its logarithm is bounded linearly on $\mathcal{L}$. To see it, observe first that if $M_1$ and $M_2$ are $s\times s$ matrices and $h$ 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 $\mathcal{L}$ as a word in $q_1,\dots,q_n$ and $h$ is an upper bound for the absolute value of the entries of each $M_l$, then the maximum absolute value $a$ of an entry of $M_x$ is bounded by

$$s^{\mathcal{L}-1}h^{\mathcal{L}}$$

The next result yields a bound on the order of $T$ which is exponential in the length $\mathcal{L}$ of $x$.

**Proposition 3.6.** *With the previous notation, there is a constant $K$, depending on $G$ only such that for $T$ the torsion subgroup of $B/NB=(1-M_x)B$,*

$$|T|\le K^{\mathcal{L}}$$

*where $\mathcal{L}$ is the length of $x$.*

*Proof.* By Theorem 3.4 and the observation above

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

so we only have to take $K=(\sqrt{s}dsh+\sqrt{s}d)^{s^2}$. □

### 3.3. **Complexity Analysis and Consequences.**

. We can now prove Theorem 1.1:

*Proof.* We consider the complexity of the algorithm 3.1. We assume that $g$ and $g_1$ are given as words in normal form. 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) some $z \in Q$ such that $z \cdot u$ is in the group generated by $b_1 \dots, b_s$. 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 the problem is equivalent to deciding whether

$$d^{\mathcal{L}} N X_1 = d^{\mathcal{L}} Zu$$

has some solution $X_1$ in $B$.

Using Proposition 2.9 and the complexity computation of Proposition 2.11 we see that this can be done in a time that is polynomial on log of the maximum absolute value of an entry in $d^{\mathcal{L}}N$. Observe that our integrality assumption on $Zu$ implies that the integer denoted $i_0$ in Proposition 2.11 can be taken to be 0. As the maximum absolute value of an entry in $d^{\mathcal{L}}N$ is exponential on $\mathcal{L}$, this time is polynomial on $\mathcal{L}$. The exponential bound in the result then follows because we are doing this a number of times which is exponential on $\mathcal{L}$. $\square$

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

Let $s_1, s_2 \geq 0$ be integers with $s = s_1 + s_2$ and denote

$$\Gamma_{s_1,s_2} := \left\{ \text{Matrices } \left(\begin{smallmatrix} I_{s_1} & A \\ 0 & I_{s_2} \end{smallmatrix}\right) \right\} \leq SL(s, \mathbb{Z}).$$

As these matrices are invertible in $SL(s, \mathbb{Z})$, we can choose $d = 1$.

**Proposition 3.7.** *With the previous notation, assume that for $l = 1, \dots, n$,*

$$M_l \in \Gamma_{s_1,s_2}.$$

*Then there is some constant $K$ depending on $G$ only such that for $T$, the torsion subgroup of $B/NB = (1 - M_x)B$,*

$$|T| \leq K\mathcal{L}^{s^2}$$

*where $\mathcal{L}$ is the length of $x$.*

*Proof.* We consider the bound of Theorem 3.4 for $d = 1$ (see above)

$$|T| \leq \sqrt{s}(a+1)^{s^2},$$

where $a$ is the maximum absolute value of an entry in $A$. Observe that $A$ is a product of matrices in $\Gamma_{s_1,s_2}$ and that

$$\begin{pmatrix} I_{s_1} & A_1 \\ 0 & I_{s_2} \end{pmatrix} \begin{pmatrix} I_{s_1} & A_2 \\ 0 & I_{s_2} \end{pmatrix} = \begin{pmatrix} I_{s_1} & A_1 + A_2 \\ 0 & I_{s_2} \end{pmatrix}.$$

Therefore, if we let $h$ be the maximum absolute value of an entry in each of the matrices $A_1, \dots, A_n$, then $a \leq \mathcal{L}h$ and therefore

$$|T| \leq \sqrt{s}(a+1)^{s^2} \leq \sqrt{s}(\mathcal{L}h+1)^{s^2} \leq \sqrt{s}(2\mathcal{L}h)^{s^2}$$

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 3.8.** *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. Let $g$ and $g_1$ be conjugate elements in $G$. Our algorithm primarily consists of identifying a suitable subgroup $Q_1$ of $Q$ and showing that, for a function dependent upon the length $\mathcal{L}$ of $x$, there exists some $y \in Q_1$ whose length is bounded by that function and which is the $Q$-component of an element $h$ such that $g^h = g_1$. Essentially, we are providing an estimation for the $Q$-conjugator length function. We make this more precise in the next result.

**Corollary 3.9.** *There exists a $K$ dependent upon $G$ only such that for any conjugate elements $g, g_1 \in G$, with $g = bx$, $g_1 = b_1x$ for $x \in Q$ and $b, b_1 \in B$, there is some $h = cy$ for $c \in B$, $y \in Q$ and $g^h = g_1$ such that the length of $y$ is bounded by $K^{\mathcal{L}}$, where $\mathcal{L}$ is the length of $x$. In the particular case when $Q \leq \Gamma_{s_1+s_2}$, the length of $y$ is bounded by $K\mathcal{L}^{s^2}$.*

3.4. **Reduction to the Discrete Logarithm Problem.**
For this subsection, we restrict ourselves to the situation of Example 2.2 where $Q$ is a multiplicative subgroup of a field $L$ such that $L : \mathbb{Q}$ is a Galois extension and $B$ is the additive group of the subring $\mathcal{O}_L[q_1^{\pm}, \dots, q_n^{\pm}]$ which is sandwiched between $\mathbb{Q}$ and $L$. In particular, this means that the only element in $Q$ with an associated matrix having 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 2.2 includes Example 2.1.

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 with $1-x$ is regular.) In this finite quotient ring we wish to solve the equation

$$y\bar{b} = \bar{b_1}.$$

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

$$q_1^{t_1} \dots q_k^{t_k}\bar{b} = \bar{b_1}$$

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 $\bar{B} = B/(1-q_1^s)B$. To solve it $s$ trials are sufficient (see [5]). In general, as $\bar{h} = 1$ in $\bar{B}$, $q_1^{l_1} \dots q_k^{l_k} = 1$. Assume that we choose $x = q_1$. Then $\bar{q}_1 = 1$ in $\bar{B}$ thus the problem is to find $t_2, \dots, t_k$ such that

$$q_2^{t_2} \dots q_k^{t_k}\bar{b} = \bar{b_1}$$

in $\bar{B}$.

Let us restrict ourselves further to the case of generalized Baumslag-Solitar groups (i.e., the groups of Example 2.1.) We identify the elements $q_l$ with the integers $m_l$ encoding their action. Assume that each $m_l$ is coprime with $1 - m_1$. As before let $y = m_1^{t_1} \dots m_k^{t_k}$ and choose $x = m_1$. Then as each $m_l$ is coprime with $1 - m_1$

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

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

$$m_2^{t_2} \dots m_k^{t_k}\bar{b} = \bar{b_1}$$

in the ring of integers modulo $1 - m_1$. If $k = 2$ this is an instance of the ordinary discrete logarithm problem.

## 4. Length Based Conjugacy Search

Length based conjugacy search is a heuristic method that attempts to solve the conjugacy search problem or the generalized conjugacy search problem (multiple instances of the conjugacy search problem where there is a common conjugating element in a specified subgroup). The latter problem is well known since it is related to the security of the Arithmetica protocol. To perform the 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.

Most previous work such as [11] and [7] study the LBCS in the context of braid groups while the authors of [6] perform the LBCS on polycyclic groups. Both groups have the advantage of having certain length functions that satisfy the properties of the previous paragraph. It is worth noting that the LBCS can be performed on an arbitrary finitely presented group as long as it admits a length function that is generically monotone increasing under conjugacy. The algorithm will work in the same way: starting with an arbitrary presentation, assign the group a length function, conjugate by successive elements in the group, and attempt to build a conjugator by investigating which elements shorten your word.

It is important to note that for length based conjugacy search to work, there needs to be an effective way to to apply the relations of the group. As such, it is best tailored towards groups that have a normal form that is easily computable. Another difference with using LBCS to solve the general conjugacy problem versus using it to break Arithmetica, is that the elements we conjugate by would need to generate the group as we are not searching within a specific subgroup. As such, we can assume that our set contains the standard generators as are given by the presentation. For a given instance of the conjugacy problem, another set of generators may be more effective, but such knowledge of effective generators is something we cannot assume in general.

In what follows we provide the pseudocode for the LBCS with memory 2 from [6], the most effective algorithm from their paper, applied to a single instance of the conjugacy problem. In this variation, one maintains a set $S$ full 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 seen in this paper (or elsewhere) 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, id_G)}
while not time-out do
    for (|z|, z, a) ∈ S do
        Remove (|z|, z, a)
        for g ∈ 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
```

## 5. Experimental Results

Tests were run on an Intel Core i7-4770K computer, running Ubuntu 14.04 LTS and using GAP version 4.7.5 [1] with 6 GB of memory allowance.

### 5.1. **LBCS in Generalized Metabelian BS Groups.**

Using the notation of 2.1, 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 `Primes2` in GAP. The table below indicates the primes chosen for each group, together with their respective bit lengths:

| Group | $m_1$ | $m_2$ | Bit Lengths $(m_1, m_2)$ |
|---|---|---|---|
| 1 | 2 | 3 | $(2, 2)$ |
| 2 | 2 | 4 | $(2, 3)$ |
| 3 | `Primes2[20]` | `Primes2[25]` | $(24, 25)$ |
| 4 | `Primes2[362]` | `Primes2[363]` | $(48, 48)$ |
| 5 | `Primes2[559]` | `Primes2[560]` | $(96, 96)$ |
| 6 | `Primes2[590]` | `Primes2[591]` | $(128, 130)$ |

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 $l$.

| Group | $l$ | $[10, 15]$ | $[20, 23]$ | $[40, 43]$ | $[l, 2l]$ | $[2l, 3l]$ | $[3l, 4l]$ |
|---|---|---|---|---|---|---|---|
| 1 | N/A | 20% | 0% | 0% | N/A | N/A | N/A |
| 2 | N/A | 0% | 0% | 0% | N/A | N/A | N/A |
| 3 | N/A | 0% | 0% | 0% | N/A | N/A | N/A |
| 4 | 14 | N/A | N/A | N/A | 0% | 0% | 0% |
| 5 | 29 | N/A | N/A | N/A | 0% | 0% | 0% |
| 6 | 38 | N/A | N/A | N/A | 0% | 0% | 0% |

Table 2. LBCS Results for GMBS Groups

## Acknowledgements

We thank Bren Cavallo who helped us in the beginning stage of this paper. Delaram Kahrobaei is partially supported by a PSC-CUNY grant from the CUNY Research Foundation, the City Tech Foundation, and ONR (Office of Naval Research) grants N000141210758 and N00014-15-1-2164. Conchita Martínez-Pérez was supported by Gobierno de Aragón, European Regional Development Funds and partially supported by MTM2010-19938-C03-03

## References


[1] GAP – Groups, Algorithms, and Programming, Version 4.7.5, May 2014.

[2] Louis Auslander. On a problem of philip hall. *Annals of Mathematics*, 86(1):pp. 112–116, 1967.

[3] Laszlo Babai, Robert Beals, Jin-yi Cai, G abor Ivanyos, and Eugene M Luks. Multiplicative equations over commuting matrices. In *Proc. 3rd ACM-SIAM SODA (Symp. on Discrete Algorithms)*. Citeseer, 1996.

[4] Gilbert Baumslag and Robert Bieri. Constructable solvable groups. *Mathematische Zeitschrift*, 151(3):249–257, 1976.

[5] Bren Cavallo and Delaram Kahrobaei. A polynomial time algorithm for the conjugacy problem in $\mathbb{Z}^n \rtimes \mathbb{Z}$. *Reports@ SCM*, 1(1), 2014.

[6] David Garber, Delaram Kahrobaei, and Ha T Lam. Length based attack for polycyclic groups. *Journal of Mathematical Cryptology, De Gruyter*, pages 33–44, 2015.

[7] David Garber, Shmuel Kaplan, Mina Teicher, Boaz Tsaban, and Uzi Vishne. Length-based conjugacy search in the braid group. *Contemporary Mathematics*, 418:75, 2006.

[8] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1985.

[9] R. Kannan and A. Bachem. Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. *SIAM J. Compt. 8*, 8(4):499–507, 1979.

[10] J. C. Lennox and D. J. S. Robinson. *The Theory of Infinite Soluble Groups*. Oxford Mathematical Monographs. The Clarendon Press, Oxford University Press, Oxford, 2004.

[11] Alex D Myasnikov and Alexander Ushakov. Length based attack and braid groups: cryptanalysis of anshel-anshel-goldfeld key exchange protocol. In *Public Key Cryptography–PKC 2007*, pages 76–88. Springer, 2007.

[12] G.A. Noskov. Conjugacy problem in metabelian groups. *Mathematical notes of the Academy of Sciences of the USSR*, 31(4):252–258, 1982.

[13] Chee K. Yap. Lecture notes on sylvester identity, lecture x, linear systems. 1999.



Jonathan Gryak, PhD Program in Computer Science, CUNY Graduate Center, City University of New York

*E-mail address*: jgryak@gradcenter.cuny.edu

Delaram Kahrobaei, CUNY Graduate Center, PhD Program in Computer Science and NYCCT, Mathematics Department, City University of New York

*E-mail address*: dkahrobaei@gc.cuny.edu

Conchita Martinez-Perez, Departamento de Matemáticas, University of Zaragoza, Spain

*E-mail address*: conmar@unizar.es