Improved linear time inversion heuristic for the Burau representation

Arkadius G. Kalka

Abstract

Though not explicitly stated, Krammer's faithfulness proof for the Lawrence-Krammer representation (rep) of B_4 [**Kr00**] contains an algorithm that computes the unique preimage braid of a given LK matrix directly in dual (or Birman-Ko-Lee (BKL)) Garside normal form. We use ideas from [**Kr00**] to develop an inversion heuristic for the Burau rep, which also computes a preimage braid of a given Burau matrix directly in dual (BKL) Garside normal form. The succes rates of this Burau inversion heuristic are significantly better (but, depending on parameters, only about 10 % better) than the success rates of the Hughes heuristic (introduced in [**Hu02**]), the best known linear time (linear in word or canonical length) inversion heuristic for the Burau rep.
Nevertheless, the success rates are far away from the self-correcting Lee-Park algorithm, for which we obtain in our computer experiments even better results than stated in [**LP03**]. But no bounds for the complexity of this algorithm are known so far, and, e.g., it seems to be to slow to be applicable for cryptanalytic purposes in braid-based cryptography.

1 Preliminaries and Notation

1.1 Braid and Garside groups

Here we refer to the well-known notion of Garside monoids and groups. For details see e.g. [**DP99**]. The presentations of the braid group B_n on n strands connected to Garside structures are the Artin presentation [**Ar26**, **Ar47**]

$$B_n = \langle \sigma_1, \ldots, \sigma_{n-1} \left| \begin{array}{ll} \sigma_i\sigma_j = \sigma_j\sigma_i & \forall |i-j| > 1, \\ \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} & \forall i = 1, \ldots, n-2 \end{array} \right. \rangle$$

and the Birman-Ko-Lee or dual presentation [**BKL98**]

$$\langle a_{ts} : 1 \leq s < t \leq n \left| \begin{array}{ll} a_{ts}a_{sr} = a_{sr}a_{tr} = a_{tr}a_{ts}, & 1 \leq r < s < t \leq n, \\ a_{ts}a_{rq} = a_{rq}a_{ts}, & (t-r)(t-q)(s-r)(s-q) > 1 \end{array} \right. \rangle.$$

The submonoids of all positive words (including the identity) according to these presentations are denoted by B_n^+ and BKL_n^+, respectively. (B_n^+, Δ_n) and (BKL_n^+, δ_n) are the only known Garside structures in B_n [**Th92**, **BKL98**], where $\Delta_n = \sigma_1(\sigma_2\sigma_1)\cdots(\sigma_{n-1}\cdots\sigma_1)$ and $\delta_n = a_{n,n-1}\cdots a_{21}$ are the Garside

elements (or fundamental braids) in question. The simple elements for the Artin presentation, i.e. the left (and right) divisors of Δ_n, are permutation braids, while the simple elements for the dual presentation, i.e. the left divisors of δ_n, are characterized by non-crossing partitions or products of parallel descending cycles [**BKL98**]. We denote the set of simple elements for the Garside structure in question by Ω and Q, respectively, and their cardinalities are $|\Omega| = n!$ and $|Q| = C_n$, where $C_n = \frac{1}{n+1}\binom{2n}{n}$ is the n-th Catalan number.
In Garside groups there exist natural normal forms, the left and right (greedy or Garside) normal form, which provide a solution to the word problem, whose time complexity is quadratic in the length of the instance braid. In particular, for every $a \in G$, G Garside group, there exists an unique decomposition (Left normal form) $a = \Delta^p a_1 \cdots a_l$, where Δ denotes the Garside element, p, the infimum of a, is the largest integer r such that $\Delta^{-r}a$ lies in the Garside monoid, and the simple elements $a_1, \ldots, a_l$ satisfy the following (left) greedy condition: For $i = 1, \ldots, l$ a_i equals the greatest common left divisor of $a_i \cdots a_l$ and the Garside element Δ.

1.2 Iterative construction of braid representations

There exists a method for constructing new representations from known linear representations of B_n. See, e.g. [**Lo94**].
Consider a representation $\rho : B_{n+1} \to Aut(V)$, where V denotes the representation space. Then we may construct a representation $\rho^+ : B_n \to Aut(V^{\oplus n})$ defined by [**Lo94**]

$$\rho^+(\sigma_i) = [\mathrm{Id}_V^{\oplus(i-1)} \oplus \begin{pmatrix} 0 & \rho(x_{i+1}) \\ \mathrm{Id}_V & \mathrm{Id}_V - \rho(x_{i+1}^{-1}x_i x_{i+1}) \end{pmatrix} \oplus \mathrm{Id}_V^{\oplus(n-i-1)}]\rho(\sigma_i),$$

where $x_i = a_{i,n+1}^2$. Note that the subgroup generated by $x_1, \ldots, x_n$ is free in B_{n+1}. Further, we may construct a representation $\rho^\ddagger : B_n \to \mathrm{Aut}(V^n)$ by a slightly different, simpler blockmatrix formula:

$$\rho^\ddagger(\sigma_i) = \rho(\sigma_i)^{\oplus(i-1)} \oplus \mathcal{R}_i^\ddagger \oplus \rho(\sigma_i)^{\oplus(n-i-1)}, \quad \mathcal{R}_i^\ddagger = \begin{pmatrix} 0 & \rho x_{i+1}\sigma_i \\ \rho\sigma_i & \rho\sigma_i - \rho x_{i+1}\sigma_i \end{pmatrix}.$$

Summarizing, if V is an R-module of rank m, i.e., $V \cong R^m$, we obtain from an m-dimensional representation $\rho : B_{n+1} \to GL(m, R)$ the mn-dimensional representation ρ^+ (or $\rho^\ddagger$).

Examples:

1. Let $R = \mathbb{Z}$. Starting with the (one-dim.) trivial representation τ (defined by $\sigma_i \mapsto 1$), we obtain the standard (or "defining") representation of S_n, i.e., the representation of B_n which factors through S_n:

$$\begin{aligned} \tau^+ : B_n &\longrightarrow GL(n, \mathbb{Z}) \\ \sigma_i &\longmapsto \mathrm{Id}_{i-1} \oplus \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \oplus \mathrm{Id}_{n-i-1}. \end{aligned}$$

2. Let R be the Laurent polynomial ring $\mathbb{Z}[s^{\pm 1}]$ in one variabe s and $\rho : B_{n+1} \to GL(m, \mathbb{Z})$ a given family of representations. Then we can define

a one-parameter representation by

$$\rho_s : B_{n+1} \longrightarrow GL(m, \mathbb{Z}[s^{\pm 1}]), \quad \sigma_i \mapsto s \cdot \rho(\sigma_i).$$

Applying the augmenting construction to ρ_s yields a one-parameter representation ρ_s^+ of B_n which contains more information than ρ^+, i.e. there exist elements $b \in \ker \rho^+$ with $b \notin \ker \rho_s^+$.

(a) $s^{-1} \cdot \tau_s^+ : B_n \to GL(n, \mathbb{Z}[s^{\pm 1}])$ defined by $\sigma_i \mapsto \mathrm{Id}_{i-1} \oplus \left(\begin{smallmatrix} 0 & s^2 \\ 1 & 1-s^2 \end{smallmatrix}\right) \oplus \mathrm{Id}_{n-i-1}$ is a Burau-type representation.

(b) Starting with the reduced Burau representation β^{red} of B_{n+1}, we get an n^2-dimensional representation $(\beta_s^{\mathrm{red}})^+ : B_n \to GL(n^2, \mathbb{Z}[s^{\pm 1}, t^{\pm 1}])$. This representation can be reduced to the $\binom{n}{2}$-dimensional Lawrence-Krammer representation [**Lo94**]. Indeed, according to Corollary 2.10 in [**Lo94**], iteration of the augmenting construction, beginning with the trivial representation, yields all summands of the Jones representation.

Note that the 2×2 Burau blockmatrix $\left(\begin{smallmatrix} 1-q & q \\ 1 & 0 \end{smallmatrix}\right)$ fulfilles the equation

$$(*) \ (\mathcal{R}^+ \oplus \mathrm{Id}_V)(\mathrm{Id}_V \oplus \mathcal{R}^+)(\mathcal{R}^+ \oplus \mathrm{Id}_V) = (\mathrm{Id}_V \oplus \mathcal{R}^+)(\mathcal{R}^+ \oplus \mathrm{Id}_V)(\mathrm{Id}_V \oplus \mathcal{R}^+).$$

Further, every $\mathcal{R}^+ \in \mathrm{Aut}(V^{\oplus 2}) = GL(2m, R)$ which satisfies this equation gives rise to a linear representation of the braid groups defined by

$$\begin{aligned} \rho(\mathcal{R}^+) : B_n &\longrightarrow \mathrm{Aut}(V^{\oplus n}) = GL(mn, R) \\ \sigma_i &\longmapsto \mathrm{Id}_V^{\oplus(i-1)} \oplus \mathcal{R}^+ \oplus \mathrm{Id}_V^{\oplus(n-i-1)}. \end{aligned}$$

Equation $(*)$ expresses the Artin relations $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$ $(i = 1, \ldots, n-1)$, while the far commutativity relations $\sigma_i \sigma_j = \sigma_j \sigma_i$ $\forall |i-j| \geq 2$ are satisfied in the $\rho(\mathcal{R}^+)$-image by construction.
This can be viewed as a direct sum analogue of the well known $\mathcal{R}$-matrix method [**Ji86**], where, given a $\mathcal{R}$-matrix $\mathcal{R} \in \mathrm{Aut}(V^{\otimes 2}) = GL(m^2, R)$, we can introduce the representation

$$\begin{aligned} \rho(\mathcal{R}) : B_n &\longrightarrow \mathrm{Aut}(V^{\otimes n}) = GL(m^n, R) \\ \sigma_i &\longmapsto \mathrm{Id}_V^{\otimes(i-1)} \otimes \mathcal{R} \otimes \mathrm{Id}_V^{\otimes(n-i-1)} \end{aligned}$$

if $\mathcal{R}$ satisfies the famous quantum Yang-Baxter equation

$$(**) \quad (\mathcal{R} \otimes \mathrm{Id}_V)(\mathrm{Id}_V \otimes \mathcal{R})(\mathcal{R} \otimes \mathrm{Id}_V) = (\mathrm{Id}_V \otimes \mathcal{R})(\mathcal{R} \otimes \mathrm{Id}_V)(\mathrm{Id}_V \otimes \mathcal{R}).$$

The theory of quantum groups provides a classification of the solutions to equation $(**)$. In so far all possible $\mathcal{R}$-matrix representations of B_n are known [**Tu88**], while a corresponding classification of the $\mathcal{R}^+$-matrix representations of B_n, i.e. of the solutions of $(*)$, remains open. Here we view just the simple case $m = \dim V = 1$, i.e., $\mathcal{R}^+ = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ with $ad - bc \neq 0$. In this case equation $(*)$ specifies to

$$\begin{pmatrix} a^2 + bac & ab + bad & b^2 \\ ca + dac & cb + dad & db \\ c^2 & cd & d \end{pmatrix} = \begin{pmatrix} a & ba & b^2 \\ ac & ada + bc & adb + bd \\ c^2 & cda + dc & cdb + d^2 \end{pmatrix}.$$

Thus we have $bad = 0 = dac$. If $b = c = 0$, then $\mathcal{R}^+ = \mathrm{Id}_2$. If $a = 0, d \neq 0$ or $a \neq 0, d = 0$ we obtain with $\mathcal{R}^+ = \left(\begin{smallmatrix} 0 & b \\ c & 1-bc \end{smallmatrix}\right)$ or $\mathcal{R}^+ = \left(\begin{smallmatrix} 1-bc & b \\ c & 0 \end{smallmatrix}\right)$ (generalized) Burau-type representations. And the case $a = d = 0$ yields $\mathcal{R}^+ = \left(\begin{smallmatrix} 0 & b \\ c & 0 \end{smallmatrix}\right)$, a generalized version of the Tong-Yang-Ma (TYM) [**TYM96**] or standard representation of B_n [**Fo96**].

2 Inverting algorithms for the Lawrence-Krammer representation

Let V be a free $\mathbb{Z}[t^{\pm 1}, q^{\pm 1}]$-module with rank $\frac{n(n-1)}{2}$. Its basis elements are denoted by v_{jk}, $1 \leq j < k \leq n$. The Lawrence-Krammer representation [**La90**] $\rho : B_n \to Aut(V) = GL(\binom{n}{2}, \mathbb{Z}[t^{\pm 1}, q^{\pm 1}])$ is defined by the following action of an Artin generator σ_k $(1 \leq i < k < k+1 < j \leq n)$

$$\begin{aligned}
(\rho\sigma_k)v_{k,k+1} &= tq^2 v_{k,k+1}, \\
(\rho\sigma_k)v_{k+1,j} &= v_{kj}, \\
(\rho\sigma_k)v_{kj} &= tq(q-1)v_{k,k+1} + (1-q)v_{kj} + qv_{k+1,j}, \\
(\rho\sigma_k)v_{i,k+1} &= v_{ik}, \\
(\rho\sigma_k)v_{ik} &= q(q-1)v_{k,k+1} + (1-q)v_{ik} + qv_{i,k+1}, \text{ and} \\
(\rho\sigma_k)v_{i_1 i_2} &= v_{i_1 i_2} \qquad \text{for} \quad \{i_1, i_2\} \cap \{k, k+1\} = \emptyset.
\end{aligned}$$

The Lawrence-Krammer representation was proved to be faithful for $n = 4$ and $q \in (0,1)$ by D. Krammer in 2000 [**Kr00**]. S. Bigelow developed a deep topological proof for the faithfulness of the Lawrence-Krammer representation for all $n \in \mathbb{N}$ [**Bi01**], implying that braid groups are linear. Bigelow's proof can be seen as a converse to the construction of Burau kernel elements given in [**Mo91**, **LP93**, **Bi99**]. In 2002 Krammer published a combinatorial proof for the faithfulness of the Lawrence-Krammer representation for all $n \in \mathbb{N}$ and $t \in (0,1)$ [**Kr02**].
According to Krammers faithfulness proof of the LK representation [**Kr02**] it is possible to compute the preimage braid $x \in B_n$ of a given LK matrix ρx directly in the Garside normal form (of the Artin presentation). An explicit algorithm for inverting the LK representation was first published by Cheon and Jun in [**CJ03a**, **CJ03b**]. Krammer's faithfulness proof [**Kr02**] has been generalised by Cohen and Wales [**CW02**], and Digne [**Di03**] to a proof for the linearity of all Artin groups.
Krammers faithfulness proof of the LK representation of B_4 [**Kr00**] given in standard fork basis can also be used to develop an inversion algorithm. Though this inversion algorithm was not explicitly published so far, its idea is contained in the definition of the cones C_x used by Krammer in [**Kr00**].
Recall that, according to corollary 3.7 in [**BKL98**], the starting (and the finishing) set of a descending cycle $\delta_\pi = a_{t_m,t_{m-1}} \cdots a_{t_2,t_1}$ $(\pi = (t_m, t_{m-1}, \ldots, t_1)$ with $1 \leq t_1 < \cdots < t_m \leq n)$ is given by

$$S(\delta_\pi) = F(\delta_\pi) = \{a_{t_j,t_i} \mid 1 \leq i < j \leq m\}.$$

A simple element $s \in Q$ is given by a product of parallel descending cycles $\pi_1, \ldots, \pi_k$, i.e., $s = \delta_{\pi_1} \cdots \delta_{\pi_k}$. And the starting (and finishing) sets of s are

$$S(s) = F(s) = S(\delta_{\pi_1}) \cup \cdots \cup S(\delta_{\pi_k}).$$

Obviously, there exists a simply computable bijection between the set of simples $Q \subset BKL_n^+$ and the set of starting sets of simples. We will use this fact in algorithm 1.

Algorithm 1: Invert the Lawrence Krammer representation

Input: A LK matrix $\rho' x := \rho x|_{t=1/2} \in GL(\binom{n}{2}, \mathbb{Q}[q^{\pm 1}])$ according to basis $\{v_{jk}\}_{1 \leq j < k \leq n}$.

Output: The unique preimage braid $x \in B_n$ in left normal form.

Transpose the instance matrix to obtain $\rho'^* x$.
Compute the smallest $p \in \mathbb{Z}$ such that $M = (\rho'^* \delta_n)^p \rho'^* x \in GL(\binom{n}{2}, \mathbb{Q}[q])$.
Initialize $k := 0$;
while $M \neq \mathrm{Id}_{V^*}$ **do**
  $k := k + 1$;
  Compute the zero rows of $M|_{q=0}$.
  They determine a starting set $S(s)$ for some $s \in S$.
  Compute the canonical factor s from $S(s)$, and set $x[k] := s$.
  $M := (\rho'^* x[k])^{-1} \cdot M$;
return *LNF* $x = \delta_n^{-p} x[1] \cdots x[k]$;

Due to the lack of a faithfulness proof of the LK representation if we set $t \in (0, 1)$, algorithm 1 is just a heuristic for $n \geq 5$. Nevertheless, we have implemented this algorithm using MAGMA 2.10 [**BCP97**], and we have shown in thousands of computer experiments with different parameter values that the preimage braids can be recovered by algorithm 1 for $n \geq 5$, too. This confirms Krammers main conjecture in [**Kr00**].

3 Computing preimage braids for the Burau representation

3.1 Burau representation

Recall the Burau representation [**Bu36**] $\beta : B_n \to GL(n, \mathbb{Z}[q^{\pm 1}])$ defined by

$$\beta(\sigma_i) = \mathrm{Id}_{i-1} \oplus \begin{pmatrix} 1-q & q \\ 1 & 0 \end{pmatrix} \oplus \mathrm{Id}_{n-i-1} \quad \forall i = 1, \ldots, n-1.$$

It can be viewed as a deformation of the standard representation of S_n, i.e., substituting $q = 1$ gives back the representation of B_n which factors through S_n. Like the standard representation of S_n, which is known to be reducible, the Burau representation β splits into the trivial 1-dimensional representation and an $(n-1)$-dimensional irreducible representation, the reduced Burau representation. Let W be a free $\mathbb{Z}[q^{\pm 1}]$-module of rank $n-1$ and denote its basis elements by $w_{k,k+1}$, $1, \ldots, n-1$. Then the reduced Burau representation is defined by the following action of Artin generators ($j \neq k-1, k, k+1$):

$$(\beta^{\mathrm{red}} \sigma_k) w_{k-1,k} = w_{k-1,k} + q w_{k,k+1}, \qquad (\beta^{\mathrm{red}} \sigma_k) w_{k,k+1} = -q w_{k,k+1},$$
$$(\beta^{\mathrm{red}} \sigma_k) w_{k+1,k+2} = w_{k,k+1} + w_{k+1,k+2}, \qquad (\beta^{\mathrm{red}} \sigma_k) w_{j,j+1} = w_{j,j+1}.$$

Now, let the coefficient ring of the reduced Burau module W be a commutative ring R where q and 2 are invertible in R. The symmetric square S^2W of the reduced Burau module W is a free R-module with basis $\{w_{ij}^2 \mid 1 \leq i < j \leq n\}$, where we define $w_{ij} = \sum_{k=i}^{j-1} w_{k,k+1}$ $(i < j)$. The reduced Burau representation $\beta^{\mathrm{red}} : B_n \to Aut(W)$ induces a representation $\beta^2 : B_n \to Aut(S^2W)$ defined by

$$(\beta^2\sigma_k)w_{ij}^2 := [(\beta^{\mathrm{red}}\sigma_k)w_{ij}]^2 \quad \text{for all} \quad 1 \leq i < j \leq n,\ 1 \leq k < n.$$

Therefore S^2W is also a B_n-module. Note that we can express mixed products as linear combinations of squares $(1 \leq i < j < k < l \leq n)$:

$$\begin{aligned} w_{ij}w_{jk} &= \frac{1}{2}(w_{ik}^2 - w_{ij}^2 - w_{jk}^2), & w_{ij}w_{ik} &= \frac{1}{2}(w_{ij}^2 + w_{ik}^2 - w_{jk}^2), \\ w_{ik}w_{jk} &= \frac{1}{2}(w_{ik}^2 + w_{jk}^2 - w_{ij}^2), & w_{ij}w_{kl} &= \frac{1}{2}(w_{il}^2 + w_{jk}^2 - w_{ik}^2 - w_{jl}^2), \\ w_{ik}w_{jl} &= \frac{1}{2}(w_{il}^2 + w_{jk}^2 - w_{ij}^2 - w_{kl}^2), & w_{il}w_{jk} &= \frac{1}{2}(w_{ik}^2 + w_{jl}^2 - w_{ij}^2 - w_{kl}^2). \end{aligned}$$

Using the equations above, we can compute the σ_k-action (induced by the representation β^2) on the basis elements of S^2W. This leads to the following observation.

Proposition 3.1 (Proposition 3.2 in [**Kr00**]) *Let R be a commutative ring where q and 2 are invertible elements. Consider the Lawrence-Krammer module V and the symmetric square S^2W of the reduced Burau module over the coefficient ring R. The map $\phi : V \to S^2W$ given by $v_{ij} \mapsto w_{ij}^2$ is a B_n-isomorphism for $t = 1$.*

It is known for a long time that the Burau representation is faithful for $n \leq 3$ [**MP69**] (see also Theorem 3.15 in [**Bi74**]). Further, it was regarded as a candidate for a faithful representation of B_n for all n until Moody proved in 1991 [**Mo91**, **Mo93**] that it is not faithful for $n \geq 9$. This result was improved by Long and Paton to all $n \geq 6$ [**LP93**]. A further improvement is due to Bigelow, who found a Burau kernel element in B_5 [**Bi99**]. The case $n = 4$ remains open. Since the structure of the kernel of the Burau representation is not understood so far, there exists no (feasible) deterministic inversion algorithm for the Burau representation as for the Lawrence-Krammer representation. Only heuristic algorithms for computing preimage braids for the Burau representation have been developed so far.
Since, for $x \in B_n$, $x' := \Delta_n^u x \in B_n^+$ for some sufficiently great $u \in \mathbb{Z}$, we may deal only with positive instance braids for the inversion heuristics. Note that u is an upper bound for $-\inf(x)$.

3.2 Hughes' algorithm

The first heuristic inversion algorithm for the Burau representation was proposed in [**Hu02**] by J. Hughes. The goal is to compute a braid $x \in B_n^+$ with $\beta(x) = X$ for a given Burau matrix $X \in \beta(B_n^+)$. Hughes' algorithm reconstructs x from $\beta(x)$ generator by generator from right to left. It uses the observation that, if $c_H(\beta(x))$ denotes the first column with highest q-degree entry in $\beta(x)$, then $c_H(x)$ is with high probability an element in the finishing set

Algorithm 2: Hughes' Algorithm

Input: $X \in \beta(B_n^+)$.
Output: $z \in B_n^+$.

```
Compute l such that det(X) = (-q)^l.
z := e;
for i := l to 1 by -1 do
    Compute j_c := c_H(X);
    if j_c = n then break; z := σ_{j_c} · z;
    X := X · β(σ_{j_c})^{-1};
return z;
```

$F(x) := \{i \in \mathbb{Z} \mid \exists w \in B_n^+ : x = w\sigma_i\}$, at least for sufficiently short $x \in B_n^+$.

Note that, since the row sum of every Burau matrix equals 1, $c_H(X) = n$ implies $X \notin \beta(B_n^+)$.
E. Lee and Park introduced a slight variation of Hughes' algorithm, which uses the fact that, if $\sigma_j \in F(x)$, then every entry in the $(j+1)$-th column of $\beta(x)$ is always in $q\mathbb{Z}[q]$ [**LP03**]. Now, let $c_{LP}(X)$ denote the integer indicating the first column containing a highest-degree entry in X among the columns whose next column's entries are all in $q\mathbb{Z}[q]$, if such a $c_{LP}(X)$ exists.

Algorithm 3: Lee-Park's Algorithm without self-correction

Input: $X \in \beta(B_n^+)$.
Output: $z \in B_n^+$.

```
Compute l such that det(X) = (-q)^l.
z := e;
for i := l to 1 by -1 do
    if there does not exist such a c_LP(X) then
        break;
    else
        Compute j_c := c_LP(X);
        z := σ_{j_c} · z;
        X := X · β(σ_{j_c})^{-1};
return z;
```

We tried to reproduce the results of the computer experiments in [**LP03**]: The experiment was performed on a standard computer using the computer-algebra system MAGMA [**BCP97**].
The table 1 shows the experimental results for the algorithms 2 and 3. On input (n, l), the program chooses at random 10000 x's from B_n^+ with $|x| = l$, computes $\rho_B(x)$ from x, computes z from $\rho_B(x)$ by each algorithm, and then checks whether or not z is equal to x by comparing their normal forms.
The lucid analysis in section 4.2 of [**LP03**] explains why the Hughes heuristic works so surprisingly good. Note that in the case $n = 3$ the success rate of the Hughes algorithm is 100% and the just mentioned analysis [**LP03**] contains a

fatihfulness proof of the Burau representation.

Table 1: Success rate of recovering x from $\beta(x)$ (unit: %)

n	5			7			10		
$\|x\|$	30	40	50	40	55	70	60	80	100
Alg. 2	90.89	81.36	70.61	88.71	73.72	56.71	84.74	67.74	49.94
Alg. 3	91.57	81.52	71.12	89.08	74.06	56.56	84.16	67.37	50.22
[**LP03**]:	96	83	76	91	76	64	87	67	42

Observation: We can reproduce similar results for the success rates of algorithm 3 (and algorithm 2) as in [**LP03**].

3.3 Self-correcting algorithm

In [**LP03**] E. Lee and Park introduced an upgraded, self-correcting version of algorithm 3. For $x \in B_n^+$, we introduce the abbreviations $x_1 = \sigma_{c_{LP}(x)}$, $x_2 = \sigma_{c_{LP}(xx_1^{-1})}$, ..., $x_k = \sigma_{c_{LP}(xx_1^{-1}\cdots x_{k-1}^{-1})}$, and $x' = xx_1^{-1}\cdots x_k^{-1}$ for $1 \leq k < |x|$. E. Lee and Park made the observation that, if $c_H(x') \neq c_{LP}(x')$, then $c_{LP}(y) \notin F(y)$ ($y := x'x_k \cdots x_i$) for some $1 \leq i \leq k$. Further they observed that, if $c_{LP}(y) > 1$ and if every entry in the $c_{LP}(y)$-th column of $\beta(y)$ is in $q\mathbb{Z}[q]$, then it is probable that $c_{LP}(y) - 1 \in F(y)$. This is the main idea of the selfcorrection in algorithm 4.
Here we reprint (Alg. 4) a corrected version of algorithm 2 in [**LP03**], published as algorithm 4 in [**Le06**]. M_j denotes the j-th column of the matrix M.

We tried to reproduce the results for the success rates of the selfcorrecting algorithm 4 given in [**LP03**, **Le06**] using a better platform and a more optimized software:
We implemented algorithm 4 using the computer algebra system MAGMA [**BCP97**]. Now, our program only chooses at random 1000 x's from B_n^+.

Table 2: Success rate of recovering x from $\beta(x)$ using algorithm 4

n	5			7			10		
$\|x\|$	30	40	50	40	55	70	60	80	100
Alg. 4	99.9	99.2	99.4	99.7	98.8	98.2	99.2	98.7	96.9*
[**LP03**, **Le06**]	100	99	97	99	97	82	99	90	69

*We used a conditional statement (if elapsed time greater than 1 hour then break;) to reduce the expenditure of time. In so far 969 is just a lower bound for the number of successfully recoverable preimage braids in this computer experiment.

Observation: For the parameter values $(n, l) = (7, 70), (10, 80)$ and (10,100) there is a significant gap between the results reported in [**LP03**, **Le06**] and our results. Indeed, the success rates of our implementation of algorithm 4 are actually much higher than those published in [**LP03**, **Le06**].

Algorithm 4: Lee-Park's Algorithm with self-correction

```
Input: X ∈ β(B_n^+).
Output: z ∈ B_n^+.
if X = Id_n then
    z := e;
else
    Compute l such that det(X) = (−q)^l.
    M[l] := X;
    for i := l to 1 by -1 do
        Compute j_a := c_H(M[i]) and j_c := c_LP(M[i]).
        if there exists such j_c and j_c = j_a then
            A[i] := j_c; M[i − 1] := M[i] · β(σ_{j_{A[i]}})^{−1};
        else
            if i = l then
                break;
            if there exists k (> i) such that j_c = j_a > 1 for M[k],
            A[k] = j_a and every entry of M[k]_{j_c} is in tZ[t] then
                reset i to be the smallest value among such k's.
                i := k; A[i] := A[i] − 1; M[i − 1] := M[i] · β(σ_{A[i]})^{−1};
            else
                break;
        z := e;
    else
        z := σ_{A[i]} ··· σ_{A[l]};
return z;
```

3.4 A simple linear inversion heuristic for the Burau representation

There exist a very simple connection between the Burau representation and the BKL presentation. Though BKL-positive braids are in general not Artin-positive, we still have

$$\beta(b) \in Mat(n, \mathbb{Z}[q]) \quad \forall q \in BKL_n^+,$$

i.e. there are no negative powers of q in the entries of $\beta(b)$. This can be proved easily be induction over the BKL word length. Indeed, let $\{w_i | i = 1, \ldots, n\}$ be the standard basis of the unreduced Burau module, then the action of BKL generators on these basis elements is described by

$$(\beta a_{ts})w_s = w_t + (1-q)\sum_{i=s}^{t-1} w_i, \quad (\beta a_{ts})w_t = qw_s + (q-1)\sum_{i=s+1}^{t-1} w_i,$$

and $(\beta a_{ts})w_i = w_i \ \forall i \neq s, t$, i.e., we have $\beta(a_{ts}) \in Mat(n, \mathbb{Z}[q])$ for all $1 \leq s < t \leq n$. We can use this fact to develop a very simple inversion heuristic for the Burau representation. This heuristic is based on the false assumption that for

all $b \in BKL_n^+$ we have

$$a_{ts} \in S(b) \quad \Leftrightarrow \quad \beta(a_{ts}^{-1} \cdot b) \in Mat(n, \mathbb{Z}[q]).$$

However, it is a theorem for $n \leq 4$ and conjectured for all n that for the LK representation (in standard fork basis) the analogous equivalence

$$a_{ts} \in S(b) \quad \Leftrightarrow \quad \rho(a_{ts}^{-1} \cdot b) \in Mat(\binom{n}{2}, \mathbb{Z}[q, t^{\pm 1}])$$

holds $\forall b \in BKL_n^+$, even if you evaluate $t \in (0,1)$ [**Kr00**].

Algorithm 5: Simple Inversion Heuristic for Burau representation β

Input: $X \in \beta(B_n^+)$.
Output: $z \in B_n^+$.

```
found := true;
while found do
    for s := n − 1 to 1 by -1 do
        for t := n to s + 1 by -1 do
            if β(a_ts^{-1}) · X is Laurent-positive then
                X := β(a_ts^{-1}) · X;
                z := z · a_ts;
                found := true;
                break s;
return z;
```

The success rates of recovering $x \in B_n^+$ from $\beta(x)$ (unit: %) using algorithm 5 are given in table 3. As for algorithms 2 and 3 ,on input (n, l), we have choosen at random 10000 x's from B_n^+ with $|x| = l$. We restricted to B_n^+ rather than BKL_n^+ in order to compare success rates with the Hughes algorithm.

Table 3: Success rate of Algorithm 5.

n	5			7			10		
$\lvert x\rvert$	30	40	50	40	55	70	60	80	100
Alg. 5	88.29	78.11	67.83	85.75	65.60	47.99	79.65	58.52	38.33

Observation: The success rates of algorithm 5 are lower than the success rates of the Hughes algorithm. But for such a simple heuristic they are still surprisingly good.

3.5 An improved linear complexity algorithm

First, for the purpose of motivation, we describe an inversion algorithm for the representations $s^{-1}(\beta_s^{\mathrm{red}})^+$, $s^{-1}(\beta_s^{\mathrm{red}})^\ddagger : B_n \to GL(n^2, \mathbb{Z}[s^{\pm 1}, q^{\pm 1}])$. (See the notation used in section 1.2.) Here the rows (and columns) of an n^2-dimensional matrix are indexed by $(i,j) \in \{1, \ldots, n\}^2$. Explicitly, the (i,j)-th row (column) of the matrix M is denoted by ${}_{(i,j)}M$ ($M_{(i,j)}$).

Algorithm 6: Inverting algorithm for $s^{-1}(\beta_s^{\mathrm{red}})^+$, $s^{-1}(\beta_s^{\mathrm{red}})^{\ddagger}$

Input: A matrix $X = \rho x$ with $\rho = s^{-1}(\beta_s^{\mathrm{red}})^+$ or $\rho = s^{-1}(\beta_s^{\mathrm{red}})^{\ddagger}$ for some unknown $x \in B_n$.

Output: $z \in B_n$ in right normal form.

```
Compute the smallest p ∈ ℤ such that M = ρx(ρδ_n)^p ∈ GL(n^2, ℤ[s, q^{±1}]).
Initialize k := 0;
while M ≠ Id_{n^2} do
    k := k + 1;
    St := ∅;
    for i := 1 to n − 1 do
        for j := i + 1 to n do
            if M_{(i,k)}|_{s=0} = M_{(j,k)}|_{s=0} for all k = 1, ..., n then
                Include a_{ji} in St.
    if St is the starting set of some s̄ ∈ Q then
        Compute x[k] = s̄ ∈ Q such that St = S(s̄).
    else
        break;
    M := M · (ρs̄)^{−1};
return z = x[k] ··· x[1]δ_n^{−p};
```

This algorithm is reminiscent of algorithm 1 based on the ideas of D. Krammer, used in his faithfulness proof of B_4 [**Kr00**]. Obviously it has linear time complexity in the dual canonical length.

Though we are not able to prove that algorithm 6 computes the unique preimage braid $x \in B_n$ such that $X = \rho x$ ($\rho = s^{-1}(\beta_s^{\mathrm{red}})^+$ or $\rho = s^{-1}(\beta_s^{\mathrm{red}})^{\ddagger}$), hundreds of computer experiments with different parameter values, where the input braids were always reconstructed successfully, supports this assumption.

Analogously, we can apply this algorithm to the $n(n+1)$-dimensional representations $s^{-1}\beta_s^+$, $s^{-1}(\rho_s^{TYM})^+$ where ρ^{TYM} denotes the Tong-Yang-Ma representation defined by $\sigma_i \mapsto \mathrm{Id}_{i-1} \oplus \left(\begin{smallmatrix} 0 & q \\ 1 & 0 \end{smallmatrix}\right) \oplus \mathrm{Id}_{n-i-1}$. Note that in the case of $s^{-1}(\rho_s^{TYM})^+$ an analogue of algorithm 6 fails to recover a preimage braid with increasing probabilty for increasing length of the input braid. Indeed, we conjecture that the augmented representation $s^{-1}(\rho_s^{TYM})^+$ is not faithful for $n \geq 4$.

Further, we can apply an analogue of algorithm 6 to the n-dimensional representation $s^{-1}\tau_s^+$ (see example 2(a) in section 1.2), i.e., to Burau-type representations. However, if we want to apply it to the Burau matrices, explicitly defined in section 3.1, we set $s^2 = q$ and we build a somehow reverse (or transposed) algorithm:

For the purpose of comparability with the success rates of the Lee-Park algorithm (without self-correction), we constrained the inputs to Artin positive braids. The table 4 shows the experimental results for the algorithms 3 and 7. On input (n, l), the program chooses at random 10000 x's from B_n^+ with $|x| = l_{\Omega_1}(x) = l$, computes $\rho_B(x)$ from x, computes z from $\rho_B(x)$ by each algorithm, and then checks whether or not z is equal to x by comparing their normal forms. Further, we compare algorithm 7 with algorithm 1 where we

Algorithm 7: Linear inversion heuristic for the Burau represention

Input: A Burau matrix $X \in \beta(B_n)$.
Output: $z \in B_n$ in left normal form.

```
Compute the smallest p ∈ ℤ such that M = βx(βδ_n)^p ∈ GL(n, ℤ[q]).
Initialize k := 0;
while M ≠ Id_n do
    k := k + 1;
    St := ∅;
    for i := 1 to n − 1 do
        for j := i + 1 to n do
            if ₁M|_{q=0} =_j M|_{q=0} then
                Include a_{ji} in St.
    if St is the starting set of some s ∈ Q then
        Compute x[k] = s ∈ Q such that St = S(s).
    else
        break;
    M := (βs)^{−1} · M;
return z = δ_n^{−p} x[1] ··· x[k];
```

have set $t = 1$. Since, in this case, we deal with $\binom{n}{2}$-dimensional matrices, we performed just 1000 experiments per (n, l)-value.

Table 4: Success rate of recovering x from $\beta(x)$ (unit: %)

n	5			7			10		
$\lvert x\rvert$	30	40	50	40	55	70	60	80	100
Alg. 3	91.57	81.52	71.12	89.08	74.06	56.56	84.16	67.37	50.22
Alg. 7	95.49	90.16	83.87	92.67	81.72	68.76	88.49	72.86	55.75
Alg. 1 ($t = 1$)	95.5	91.2	84.8	93.0	82.3	67.4	88.3	73.9	52.8

Observation: The success rates of our linear complexity inverting heuristic for the Burau representation are slightly, but significantly, better than the corresponding success rates of the Hughes or the Lee-Park algorithm (without self-correction) for all investigated parameter values. Further, the success rates of algorithm 7 and algorithm 1 with $t = 1$ are roughly equal. Since the Lawrence-Krammer module for $t = 1$ is the symmetric square of the (reduced) Burau module, this is far from being a surprising effect.
But the success rates of algorithm 4 are not within reach for our algorithm 7. This is due to a lack of self-correction in this algorithm. The development of a self-correcting version of algorithm 7 remains as a task for future research. Nevertheless, since algorithm 7 has linear time complexity (as Hughes' algorithm), it can be used as a cryptanalytic tool in representation attacks against braid-based cryptosystems.

Note that success rates of all inversion algorithms for the Burau represen-

tation are 100% for the 3-strand braid group. But they are lower than 100% in the case $n = 4$. This also holds for algorithm 1 setting $t = 1$. This confirms the conjecture that the Burau representation is not faithful for $n = 4$. But non-trivial Burau kernel elements in B_4 have not been found so far. According to an exhaustive search, using the topological characterization of Burau kernel elements reported in [**Bi99**], such elements, if they really exist, must be quite long elements.

Acknowledgements. Parts of this paper are part of the author's Ph.D. Thesis [**Ka07**] at Ruhr-Universität Bochum. This research was partially supported by the Graduate College "Mathematische und ingenieurwissenschaftliche Methoden für sichere Datenübertragung und Informationsvermittlung", the Emmy Noether Research Institute for Mathematics and the Minerva Foundation. I thank my PhD supervisor Lothar Gerritzen, my host Mina Teicher at Bar-Ilan University, and Boaz Tsaban for fruitful discussions.

References

[**Ar26**] Emil Artin, *Theorie der Zöpfe*, Abhandlungen aus dem Mathematischen Seminar der Hamburgischen Universität **4** (1926), 47-72.

[**Ar47**] Emil Artin, *Theory of braids*, Annals of Math. (2) **48** (1947), 101-126.

[**BCP97**] Wieb Bosma, John Cannon, and Catherine Playoust. *The Magma algebra system. I. The user language.* J. Symbolic Comput., 24(3-4):235-265, 1997.

[**Bi74**] Joan S. Birman, *Braids, links and mapping class groups*, Annals of Mathematics Studies **82**, Princeton University Press (1974).

[**Bi99**] Stephen J. Bigelow, *The Burau representation is not faithful for* $n = 5$, Geom. Topol. **3** (1999), 397-404.

[**Bi01**] Stephen J. Bigelow, *Braid groups are linear*, J. Amer. Math. Soc. **14**, no. 2 (2001), 471-486.

[**BKL98**] Joan S. Birman, Ki Hyoung Ko, Sang Jin Lee, *New approaches to the word and conjugacy problem in the braid groups*, Advances in Math. **139** (1998), 322-353. Eprint archive: `arXiv:math.GT/9712211`

[**BLM92**] Joan S. Birman, Darren D. Long and John A. Moody, *Finite-Dimensional Representations of Artin's Braid Group*, in: Abikoff, Birman and Kuiken (eds.): *The mathematical legacy of Wilhelm Magnus: Groups, Geometry and special functions*, Contemporary Mathematics **169**, AMS (1992).

[**Bu36**] Werner Burau, *Über Zopfgruppen und gleichsinnig verdrillte Verkettungen*, Abhandlungen aus dem Mathematischen Seminar der Hansischen Universität **11** (1936), 179-186.

[**CJ03a**] Jung Hee Cheon and Byungheup Jun, *Diffie-Hellman Conjugacy Problem on Braids*, Preprint, electronically available from: `http://citeseer.ist.psu.edu/557008.html`

[**CJ03b**] Jung Hee Cheon and Byungheup Jun, *A Polynomial Time Algorithm for the Braid Diffie-Hellman Conjugacy Problem*, Advances in Cryptology - CRYPTO 2003, LNCS **2729**, Springer (2003).

[**CK+01**] Jae Choon Cha, Ki Hyoung Ko, Sang Jin Lee, Jae Woo Han and Jung Hee Cheon, *An efficient implementation of braid groups*, Advances in Cryptology - ASIA-CRYPT 2001, LNCS **2248**, Springer (2001).

[**CW02**] Arjeh M. Cohen and David B. Wales, *Linearity of Artin groups of finite type*, Israel J. Math. **131** (2002), 101-123.

[**Di03**] Francois Digne, *On the linearity of Artin braid groups*, Journal of Algebra **268**, no.1 (2003), 39-57.

[**DP99**] Patrick Dehornoy and Luis Paris, *Gaussian groups and Garside groups, two generalizations of Artin groups*, Proc. London Math. Soc. **79-3** (1999), 569-604.

[**EC+92**] David B. A. Epstein, James W. Cannon, Derek F. Holt, Silvio V. F. Levy, Michael S. Paterson and William P. Thurston, *Word processing in groups*, Jones and Bartlett (1992).

[**Fo96**] Ed Formanek, *Braid group representations of low degree*, Proc. London Math. Soc. **73** (1996), 279-322.

[**Hu02**] James Hughes, *A linear algebraic attack on the AAFG1 braid group cryptosystem*, Information Security and Privacy, LNCS **2384**, Springer (2002).

[**Ji86**] Michio Jimbo, *Quantum R-matrix for the generalized Toda system*, Comm. Math. Physics **102** (1986), 537-547.

[**Ka07**] Arkadius G. Kalka, *Linear representations of braid groups and braid-based cryptography*, Ph.D. Thesis Ruhr-Universität Bochum (2007).

[**Kr00**] Daan Krammer, *The braid group B_4 is linear*, Invent. Math. **142**, no. 3 (2000), 451-486.

[**Kr02**] Daan Krammer, *Braid groups are linear*, Ann. of Math. (2) **155**, no. 1 (2002), 131-156.

[**La90**] Ruth J. Lawrence, *Homological representations of the Hecke algebra*, Comm. Math. Phys. **135**, no. 1 (1990), 141-191.

[**Le06**] Eonkyung Lee, *Inverting the Burau and Lawrence-Krammer Representations*, In: L. Gerritzen, D. Goldfeld, M. Kreuzer, G. Rosenberger and V. Shpilrain (Eds.), *Algebraic Methods in Cryptography*, Contemporary Mathematics **418**, AMS (2006), 153-160.

[**Lo94**] Darren D. Long, *Constructing representations of braid groups*, Communications in Analysis and Geometry, **2**, no. 2 (1994), 217-238.

[**LP93**] D. D. Long and M. Paton, *The Burau representation is not faithful for $n \geq 6$*, Topology **32** (1993), 439-447.

[**LP03**] Eonkyung Lee and Je Hong Park, *Cryptanalysis of the Public-key Encryption based on Braid Groups*, Advances in cryptology - EUROCRYPT 2003, LNCS **2656**, Springer (2003).

[**Mo91**] John A. Moody, *The Burau representation of the braid group is not faithful for large n*, Bull. Amer. Math. Soc. **25** (1991), 379-384.

[**Mo93**] John A. Moody, *The faithfulnees question for the Burau representation*, Proc. Amer. Math. Soc. **119** (1993), 439-447.

[**MP69**] Wilhelm Magnus and A. Peluso, *On a theorem of V. I. Arnold*, Com. on Pure and Appl. Math. **XXII** (1969), 683-692.

[**Th92**] William Thurston, *braid groups*, chapter 9 in [**EC$^+$92**].

[**Tu88**] Vladimir Turaev, *The Yang-Baxter equation and invariants of links*, Invent. Math. **92** (1988), 527-553.

[**TYM96**] Dian-Min Tong, Shan-De Yang and Zhong-Qi Ma, *A new class of representations of braid groups*, Comm. Theoret. Phys. **26**, no. 4 (1996), 483-486.

[**Zi01**] Matthew G. Zinno, *On Krammer's representations of the braid groups*, Math. Ann. **321**, no. 1 (2001), 197-211. Eprint archive: `arXiv:math.RT/0002136`