J. Group Theory **21** (2018), 973–1000
DOI 10.1515/jgth-2018-0027

On the centralizer of generic braids

Juan González-Meneses and Dolores Valladares

Communicated by Pierre-Emmanuel Caprace

Abstract. We study the centralizer of a braid from the point of view of Garside theory, showing that generically a minimal set of generators can be computed very efficiently, as the ultra summit set of a generic braid has a very particular structure. We present an algorithm to compute the centralizer of a braid whose generic-case complexity is quadratic on the length of the input, and which outputs a minimal set of generators in the generic case.

1 Introduction

Braid groups, introduced by Artin [1, 2], and more generally Artin–Tits groups of spherical type [6, 7], have a very particular algebraic structure, called a Garside structure, which can be used to deduce many properties, make computations and solve decision problems in these groups. Groups admitting such a Garside structure are called Garside groups [11, 12].

Among the basic results concerning braid groups that are obtained using its Garside structure, one can find the solutions to the word problem [3, 14, 17] and the conjugacy problem [3, 4, 13, 16–20]. Closely related to the latter is the problem of determining the centralizer of a given braid [15, 21, 23, 25]. In 1971, Makanin [25] gave an algorithm to compute a generating set of the centralizer $Z(x)$ of any given braid x. This method can be generalized easily to all Garside groups, but it has a huge complexity and highly redundant generating sets. A much better algorithm was given in [15], although the actual complexity of the algorithm and the number of generators given as output is not clear. A different approach uses the geometric properties of braids [21], since the geometric classification of a braid as periodic, reducible or pseudo-Anosov determines the structure of its centralizer. For instance, if x is pseudo-Anosov, one can use the well-known result by McCarthy [22, 26] applied to braids [21], to see that $Z(x)$ is isomorphic to $\mathbb{Z} \times \mathbb{Z}$. One generator is pseudo-Anosov, usually a root of x, and the other generator is periodic, and can be chosen to be a root of the *Garside element* Δ^2. But in order to compute the generators of $Z(x)$, the procedure in [15] is usually applied.

Both authors partially supported by Spanish Projects MTM2013-44233-P and FEDER. First author partially supported by MTM2016-76453-C2-1-P and FEDER.

The main idea of the algorithm in [15] is to compute, given a braid x, a finite set of elements called its *ultra summit set* ($\mathrm{USS}(x)$), which is an invariant subset of the conjugacy class of x. The elements of this set are computed along with conjugating elements connecting them, so the ultra summit set is stored as a directed graph, Γ_x. A generating set of the fundamental group of Γ_x yields a generating set of the centralizer of x, in a natural way.

The algorithm in [15] is very efficient in practice. So the question arises of whether, generically, this algorithm will produce a small number of generators in a short time or not. One can see from computations that, in most cases, the graph Γ_x is quite small and satisfies a very particular property: conjugating elements correspond to factors of the normal form of x. In this case we will say that $\mathrm{USS}(x)$ is *minimal* (cf. Definition 4.1). We will show in this paper that, indeed, this case is generic.

Before stating the result, we need to clarify what we understand by *generic*. We will consider the braid group B_n with some natural generating set (the set of *simple elements*). In the Cayley graph of B_n with respect to this set of generators, we will consider the ball $\mathbf{B}(l)$ of radius l centered at the neutral element. We will say that a property $\mathcal{P}$ is *generic* in B_n, or that the braids satisfying $\mathcal{P}$ are generic, if the proportion of braids which satisfy $\mathcal{P}$ in $\mathbf{B}(l)$ tends to 1 as l tends to infinity.

For instance, Wiest and Caruso [9] showed that braids which admit a "short" conjugation to a *rigid* pseudo-Anosov braid are generic (rigidity is a Garside theoretical property that will be explained later). In particular, pseudo-Anosov braids are generic. In [9] it is also proved that braids which admit a given factor in its normal form are generic. Studying the properties of these generic braids in more detail, we show the following:

Theorem 4.8. *The proportion of braids in* $\mathbf{B}(l)$ *whose ultra summit set is minimal tends to 1 exponentially fast, as* l *tends to infinity.*

In this generic case in which the ultra summit set of a braid x is minimal, we will be able to describe an explicit minimal set of generators for its centralizer, as follows. In general $\mathrm{USS}(x)$ consists of a set of orbits under a special kind of conjugation called *cycling*. The conjugating elements for iterated cycling, starting from x, will be denoted by a_1, a_2, etc. As this orbit is finite, there is some $k > 0$ such that cycling k times x one gets back to x. In other words, if we define $\mathrm{PC}(x) = a_1 \cdots a_k$, then $\mathrm{PC}(x)$ commutes with x. We will show that if $\mathrm{USS}(x)$ is minimal, then it has at most two orbits, and we can describe all possible cases as follows:

Theorem 5.3. *Let* x *be a rigid braid such that* $\mathrm{USS}(x)$ *is minimal and has two orbits under cycling. Then* $Z(x) = \langle \mathrm{PC}(x), \Delta^2 \rangle$.

Theorem 5.5. *Let x be a rigid braid such that* $\mathrm{USS}(x)$ *is minimal and has only one orbit under cycling of length k. One has:*

(i) *If $\Delta^{-1}x\Delta \neq x$, then $Z(x) = \langle a_1 \cdots a_{\frac{k}{2}} \Delta^{-1}, \Delta^2 \rangle$.*

(ii) *If $\Delta^{-1}x\Delta = x$, then $Z(x) = \langle \mathrm{PC}(x), \Delta \rangle$.*

From Theorem 4.8 it follows that Theorem 5.3 and Theorem 5.5 are describing the centralizers of generic braids. This yields an algorithm to compute a minimal set of generators for the centralizer of a braid, which runs generically in polynomial time.

Corollary 6.2. *There exists an algorithm to compute a generating set for the centralizer of a braid $y \in B_n$, whose generic-case complexity is $O(l^2 n^4 \log n)$, where $l = \ell(y)$.*

The plan of the paper is as follows: in Section 2.1 we describe basic facts about Garside groups. In Sections 2.2 and 2.3 we give the basic definitions and results concerning the left normal form and the conjugacy problem, respectively. In Section 2.4 we describe the structure of ultra summit sets, defining the minimal simple elements introduced in [15] and two special kinds of conjugations from [5]. We finish this section with basic notions about the transport map, which will be used several times in this paper. In Section 3, thanks to the work by Caruso and Wiest [9] about genericity of rigid pseudo-Anosov braids, we prove some needed results about generic braids. In Section 4, we define minimal ultra summit sets and we show that the ultra summit set of a generic braid is minimal. Section 5 is devoted to determine the centralizer of generic braids. Finally, in Section 6 we present our algorithm and study its complexity.

2 Background

2.1 Garside groups

A *Garside group* is a group G which admits a submonoid P and a special element $\Delta \in P$ satisfying some suitable properties [12]. We will apply Garside properties to the particular case of the braid group B_n, so we will just enumerate the necessary results in this setting, starting with the classical presentation of B_n:

$$\begin{aligned} B_n = \langle \sigma_1, \ldots, \sigma_{n-1} \mid \sigma_i\sigma_j &= \sigma_j\sigma_i \text{ if } |j-i| > 1, \\ \sigma_i\sigma_j\sigma_i &= \sigma_j\sigma_i\sigma_j \text{ if } |j-i| = 1 \rangle. \end{aligned}$$

The elements of B_n which can be written as a product of positive powers of generators are called *positive elements*, and they form the monoid P, usually denoted by B_n^+. The special element Δ, called the *Garside element*, is

$$\Delta = (\sigma_1\sigma_2\cdots\sigma_{n-1})(\sigma_1\sigma_2\cdots\sigma_{n-2})\cdots(\sigma_1\sigma_2)\sigma_1.$$

Given two braids $a, b \in B_n$, we say that a is a prefix of b, and we write $a \preccurlyeq b$, if $a^{-1}b \in B_n^+$. This determines a partial order in B_n which is invariant under left-multiplication, and is a lattice order. We denote by $a \wedge b$ (resp. $a \vee b$) the meet (resp. the join) of the braids a and b with respect to $\preccurlyeq$. We will also say that $a \wedge b$ is the greatest common prefix of a and b, and that $a \vee b$ is their least common multiple.

The set $[1, \Delta] = \{s \in B_n \mid 1 \preccurlyeq s \preccurlyeq \Delta\} \subset B_n^+$, called the set of *simple elements* of B_n, is a very particular finite set of generators of B_n, which are the building blocks of the normal forms that we will see later. The generators $\sigma_1, \dots, \sigma_n$ are called *atoms*. Conjugation by the Garside element Δ determines an inner automorphism of B_n that we denote by τ (that is, $\tau(x) = \Delta^{-1}x\Delta$). It is easy to check that $\tau(\sigma_i) = \sigma_{n-i}$ for every $i = 1, \dots, n-1$, so τ has order 2 and Δ^2 is central. Actually, the center of B_n is cyclic, generated by Δ^2.

Throughout this paper we will focus on the braid group B_n, although some results could be extended or adapted to other Garside groups, as their proofs basically involve techniques from Garside theory.

2.2 Left normal form

Let us see how the Garside structure of B_n allows us to define a particular unique decomposition of each braid, called its *left normal form*.

Definition 2.1. Given a simple element $s \in [1, \Delta]$, the *right complement* of s is defined by $\partial(s) = s^{-1}\Delta$, and the *left complement* of s is $\partial^{-1}(s) = \Delta s^{-1}$.

Notice that the map ∂ is a bijection of $[1, \Delta]$ and that $\partial^2 = \tau$ (see [4]).

Definition 2.2 ([14]). Let s, t be two simple elements in B_n. We say that the decomposition st is *left-weighted* if $\partial(s) \wedge t = 1$ or, equivalently, $\Delta \wedge st = s$.

Definition 2.3 ([14]). Given an element $x \in B_n$, we say that a decomposition $x = \Delta^p x_1 \cdots x_l$ ($p \in \mathbb{Z}, l \geq 0$) is the *left normal form* of x if $x_i \in [1, \Delta] \setminus \{1, \Delta\}$ for $i = 1, \dots, l$ and $x_i x_{i+1}$ is a left-weighted decomposition for $i = 1, \dots, l-1$.

Definition 2.4 ([14]). Given $x \in B_n$ whose left normal form is $\Delta^p x_1 \cdots x_l$, we define the *infimum*, *supremum* and *canonical length* of the element x, respectively, by $\inf(x) = p$, $\sup(x) = p + l$ and $\ell(x) = l$.

It is well known that left normal forms of elements in B_n exist and are unique [13, 14]. The right complement plays an important role when comparing the left normal forms of x and x^{-1}.

Theorem 2.5 ([4]). *If $x = \Delta^p x_1 \cdots x_l$ is in left normal form, then the left normal form of x^{-1} is equal to*

$$x^{-1} = \Delta^{-p-l} x'_l \cdots x'_1,$$

where $x'_i = \tau^{-p-i}(\partial(x_i))$ for $i = 1, \dots, l$.

As we saw in Definition 2.2, the decomposition st is left-weighted if $\partial(s)$ and t have no prefixes in common, except the trivial one. As $\partial(s)$ is the simple element such that $s\partial(s) = \Delta$, the prefixes of $\partial(s)$ are precisely those simple elements s' for which ss' is simple. Hence, the decomposition st is left-weighted if and only if the only prefix s' of t such that ss' is simple is the trivial one.

Given two simple elements s and t, it is easy to find the left-weighted decomposition of the product st using the description above. If the decomposition st is not left-weighted, this means that there is a nontrivial prefix $s' \preccurlyeq t$ such that ss' is simple. Since $\preccurlyeq$ is a lattice order, there is a maximal element satisfying this property, namely $s' = \partial(s) \wedge t$. Hence, transforming the decomposition st into a left-weighted one means sliding the prefix s' from the second factor to the first one. That is, write $t = s't'$ and then consider the decomposition $st = (ss')t'$, with ss' as the first factor and t' as the second one. Since s' is maximal, the decomposition $(ss')t'$ is left-weighted.

The action of obtaining the left-weighted decomposition of the product of two simple elements s and t, by sliding the simple element s' from the second factor to the first factor, is known as a *local sliding* applied to the decomposition st. Using local slidings, one can compute the left normal form of any element of a Garside group.

2.3 The conjugacy problem

The known algorithms to solve the conjugacy problem in braid groups share a common strategy. Given an element $x \in B_n$, the idea is to compute a finite subset of the conjugacy class x^{B_n} of x, which consists of those conjugates satisfying some suitable conditions, and which only depends on x^{B_n}, not on x itself. There are two special types of conjugations ever since the work of ElRifai and Morton in [13]:

Definition 2.6. Given $x = \Delta^p x_1 \cdots x_l$ written in left normal form, where $l > 0$, the *cycling* of x is

$$\mathbf{c}(x) = \Delta^p x_2 \cdots x_l \tau^{-p}(x_1),$$

and the *decycling* of x is

$$\mathbf{d}(x) = x_l \Delta^p x_1 \cdots x_{l-1}.$$

The conjugating elements involved in a cycling or a decycling will play a crucial role later, so to avoid repeated references to the automorphism τ we introduce the following definition.

Definition 2.7 ([4]). Let x be a braid with left normal form $x = \Delta^p x_1 \cdots x_l$, where $l > 0$. We define the *initial factor* of x as $\iota(x) = \tau^{-p}(x_1)$, and the *final factor* of x as $\varphi(x) = x_l$. If $l = 0$, we define $\iota(\Delta^p) = 1$ and $\varphi(\Delta^p) = \Delta$.

Notice that, up to conjugation by Δ^p, the simple element $\iota(x)$ (resp. $\varphi(x)$) corresponds to the first (resp. last) non-Δ factor in the left normal form of x. From Theorem 2.5, the initial and final factors of x and x^{-1} are closely related.

Lemma 2.8 ([4]). *For every $x \in B_n$ one has*

$$\iota(x^{-1}) = \partial(\varphi(x)) \quad \textit{and} \quad \varphi(x^{-1}) = \partial^{-1}(\iota(x)).$$

Lemma 2.9 ([5]). *Given $x \in B_n$ with $\ell(x) > 0$, one has*

$$x^{\iota(x)} = \mathbf{c}(x), \quad x^{\varphi(x)^{-1}} = \mathbf{d}(x), \quad x^{\iota(x^{-1})} = x^{\partial(\varphi(x))} = \tau(\mathbf{d}(x)).$$

We define the *twisted decycling* of x as $\tau(\mathbf{d}(x))$.

Going back to the conjugacy problem, the finite invariant subset of x^{B_n} defined in [13] is called the *super summit set* of x, denoted by $\mathrm{SSS}(x)$. It consists of the conjugates of x having minimal canonical length. This minimal canonical length among the conjugates of x is called the *summit length* of x, and denoted by $\ell_s(x)$. In [13] it was shown that one obtains an element in $\mathrm{SSS}(x)$, starting with x, by iterated application of cyclings and decyclings.

We will not consider $\mathrm{SSS}(x)$ in this paper, but the following invariant subset of $\mathrm{SSS}(x)$ introduced by Gebhardt:

Definition 2.10 ([18]). The *ultra summit set* $\mathrm{USS}(x)$ of $x \in B_n$ is the set of elements $y \in \mathrm{SSS}(x)$ such that $\mathbf{c}^m(y) = y$, for some $m > 0$.

We remark that one obtains an element in $\mathrm{USS}(x)$ by iterated application of cycling to an element in $\mathrm{SSS}(x)$. Thus, the ultra summit set $\mathrm{USS}(x)$ consists of a finite set of disjoint, closed orbits under cycling. Once one obtains an element in $\mathrm{USS}(x)$, it is explained in [18] how to compute all elements in $\mathrm{USS}(x)$, together with conjugating elements connecting them.

This was improved in [20] by introducing a new kind of conjugation, called cyclic sliding, to replace cycling and decycling. Cyclic sliding simplifies the algorithms concerning conjugacy in Garside groups, and in particular in braid groups.

Definition 2.11 ([18]). Given $x \in B_n$, its *preferred prefix* is

$$\mathfrak{p}(x) = \iota(x) \wedge \iota(x^{-1}) = \iota(x) \wedge \partial(\varphi(x)).$$

Definition 2.12 ([20]). For $x \in B_n$, the *cyclic sliding* of x is

$$\mathfrak{s}(x) = x^{\mathfrak{p}(x)} = \mathfrak{p}(x)^{-1} x \mathfrak{p}(x).$$

Using cyclic sliding, one can define a new subset of the conjugacy class of a braid x, which is contained in $\mathrm{USS}(x)$. If one applies iterated cyclic sliding to an element $x \in B_n$, one eventually obtains a repeated element, say y. The orbit of y under cyclic sliding will be called a sliding circuit. The set of all (disjoint) sliding circuits in the conjugacy class of x is the mentioned invariant subset:

Definition 2.13 ([20]). The *set of sliding circuits* $\mathrm{SC}(x)$ of $x \in B_n$ is the set of conjugates $y \in x^{B_n}$ such that $\mathfrak{s}^m(y) = y$, for some $m > 0$.

In this paper we will study the centralizer of a special kind of braids, called rigid braids. The idea of studying rigidity came from the study of elements whose left normal form changes only in the obvious way under cyclings, decyclings and powers, so their ultra summit sets are easier to study. Rigid elements in Garside groups were studied in [4].

Definition 2.14 ([4]). Let $x = \Delta^p x_1 \cdots x_l$ be in left normal form, with $l > 0$. Then x is *rigid* if $\varphi(x)\iota(x)$ is left-weighted as written, that is, if $\mathfrak{p}(x) = 1$.

2.4 Minimal simple elements

Let us focus on the ultra summit set $\mathrm{USS}(x)$ of a braid $x \in B_n$. As we said earlier, the elements of $\mathrm{USS}(x)$ are computed along with conjugating elements connecting them. We will describe those conjugating elements in detail. We shall use the notation $y^s = s^{-1} y s$.

Definition 2.15 ([18]). Given $x \in B_n$ and $y \in \mathrm{USS}(x)$, we say that a simple element $1 \neq s \in [1, \Delta]$ is a *minimal simple element* for y (with respect to $\mathrm{USS}(x)$) if $y^s \in \mathrm{USS}(x)$, and $y^t \notin \mathrm{USS}(x)$ for every $1 \precneqq t \precneqq s$.

Definition 2.16. Given $x \in B_n$, the directed graph Γ_x is defined as follows:

- The vertices are the elements of $\mathrm{USS}(x)$.
- There is an arrow with label s going from y to y^s, for every minimal simple element s for y.

In [18] Gebhardt described how to compute the minimal simple elements for a given $y \in \mathrm{USS}(x)$. Moreover, he shows the following:

Theorem 2.17 ([18, Theorem 1.18, Corollary 1.19]). *Let $x \in B_n$ and $y \in \mathrm{USS}(x)$.*

(i) *If $s, t \in B_n$ are such that $y^s \in \mathrm{USS}(x)$, $y^t \in \mathrm{USS}(x)$, then $y^{s \wedge t} \in \mathrm{USS}(x)$.*

(ii) *For every $u \in B_n^+$ there is a unique element $c_y(u)$ which is minimal with respect to $\preccurlyeq$ among the elements υ satisfying $u \preccurlyeq \upsilon$ and $y^\upsilon \in \mathrm{USS}(x)$.*

(iii) *The graph Γ_x described in Definition* 2.16 *is finite and connected. Its transitive closure is a complete graph, i.e., every vertex is reachable from every other vertex by an oriented path.*

By the above result, given one element $\widetilde{x} \in \mathrm{USS}(x)$, one can obtain any other element in $\mathrm{USS}(x)$ just through conjugations by minimal simple elements. In this way one can compute the whole graph Γ_x, starting with a single element $\widetilde{x} \in \mathrm{USS}(x)$.

From now on, to simplify the notation, we will assume that x belongs to its own ultra summit set. That is, $x \in \mathrm{USS}(x)$. It is known that conjugations by minimal simple elements are quite special:

Corollary 2.18 ([5, Theorem 2.5]). *Let $x \in \mathrm{USS}(x)$ with $\ell(x) > 0$ and let s be a minimal simple element for x. Then s is a prefix of either $\iota(x)$ or $\iota(x^{-1})$, or both.*

Definition 2.19. Let $x \in B_n$. A *partial cycling* of x is a conjugation of x by a prefix of $\iota(x)$. A *partial twisted decycling* of x is a conjugation of x by a prefix of $\iota(x^{-1}) = \partial(\varphi(x))$.

Consequently, by Theorem 2.17 and Corollary 2.18, given $x, y \in \mathrm{USS}(x)$, there exists a sequence of partial cyclings and partial twisted decyclings joining x to y.

From Corollary 2.18, we see that there are two kinds of minimal simple elements, hence there are two kinds of arrows in Γ_x. Following [5], we say that an arrow s starting at a vertex $y \in \mathrm{USS}(x)$ is *black* if s is a prefix of $\iota(y)$, and it is *grey* if s is a prefix of $\iota(y^{-1})$. In other words, an arrow starting at y is black if it corresponds to a partial cycling of y, and it is grey if it corresponds to a partial twisted decycling of y. Notice that an arrow can, a priori, be black and grey at the same time. In that case, we say it is a bi-colored arrow.

Definition 2.20. A *path* in Γ_x is a (possibly empty) sequence $(s_1^{e_1}, \dots, s_k^{e_k})$, where s_i is an arrow in Γ_x and $e_i = \pm 1$, such that the endpoint of $s_i^{e_i}$ is equal to the starting point of $s_{i+1}^{e_{i+1}}$ for every $i = 1, \dots, k-1$. We say that a path $(s_1^{e_1}, \dots, s_k^{e_k})$ is *oriented* if $e_i = 1$ for $i = 1, \dots, k$.

Remark. Every path $(s_1^{e_1}, \dots, s_k^{e_k})$ determines an element $\alpha = s_1^{e_1} \cdots s_k^{e_k}$. Distinct paths may determine the same element. Since the labels of arrows are simple elements, it follows that if the path is oriented then $\alpha \in B_n^+$.

We end this subsection with the case of rigid braids, since the structure of Γ_x we described above will be simpler in this case. Concerning the arrows of Γ_x, the main difference between the case of a rigid braid and the general case is the following:

Lemma 2.21 ([5]). *Let $x \in \mathrm{USS}(x)$ and $\ell(x) > 0$. Then x is rigid if and only if there are no bi-colored arrows starting at x.*

Refer to [5, Section 2.1] to see several examples illustrating the notions we have seen in this subsection.

2.5 The transport map

We finish this introductory section explaining a tool that will be used several times in this paper: The transport map. Given two conjugate braids x and $x^\alpha = \alpha^{-1}x\alpha$, the images of x and x^α under cycling are also conjugate, and we can relate α to a conjugating element for the images $\mathbf{c}(x)$ and $\mathbf{c}(x^\alpha)$.

Definition 2.22 ([18]). Given $x, \alpha \in B_n$, we define the *transport* of α at x under cycling as

$$\alpha^{(1)} = \iota(x)^{-1}\alpha\iota(x^\alpha).$$

That is, $\alpha^{(1)}$ is the conjugating element that makes the following diagram commutative, in the sense that the conjugating element along any closed path is trivial:

$$\begin{tikzcd}
x \arrow[r, "\iota(x)"] \arrow[d, "\alpha"'] & \mathbf{c}(x) \arrow[d, "\alpha^{(1)}"] \\
x^\alpha \arrow[r, "\iota(x^\alpha)"] & \mathbf{c}(x^\alpha).
\end{tikzcd}$$

Note that the horizontal rows in this diagram correspond to applications of cycling. For an integer $i > 1$ we recursively define $\alpha^{(i)} = (\alpha^{(i-1)})^{(1)}$, which is the transport of $\alpha^{(i-1)}$ at $\mathbf{c}^{i-1}(x)$. We also define $\alpha^{(0)} = \alpha$.

Under certain conditions, the transport under cycling respects many aspects of the Garside structure. In particular, if x and x^α as above are super summit elements, that is, x and x^α have minimal canonical length in their conjugacy class, the transport map preserves products, left divisibility, greatest common prefixes, and powers of Δ. For more details, refer to [18].

Suppose that a given braid x is rigid. Then x belongs to its ultra summit set (it trivially belongs to $\mathrm{SC}(x)$, which is contained in $\mathrm{USS}(x)$), and the minimal simple elements for x have some particular properties that we will show to end this section. These properties can be deduced from the results in [5], but we will provide proofs.

Lemma 2.23. *Let x be a rigid braid with normal form $\Delta^p x_1 \cdots x_l$ $(l > 0)$, and let $a = \iota(x)$. Then $a^{(kl+r)} = \tau^{-(k+1)p}(x_{r+1})$ for all $k \geq 0$ and all $r = 0, \dots, l-1$.*

Proof. For every $i > 0$, we have the following diagram:

$$\begin{tikzcd}
x \arrow[r, "\iota(x)"] \arrow[d, "a=\iota(x)"] & \mathbf{c}(x) \arrow[r, "\iota(\mathbf{c}(x))"] \arrow[d, "a^{(1)}"] & \mathbf{c}^2(x) \arrow[r] \arrow[d, "a^{(2)}"] & \cdots \arrow[r] & \mathbf{c}^{i-1}(x) \arrow[r, "\iota(\mathbf{c}^{i-1}(x))"] \arrow[d, "a^{(i-1)}"] & \mathbf{c}^i(x) \arrow[d, "a^{(i)}"] \\
\mathbf{c}(x) \arrow[r, "\iota(\mathbf{c}(x))"] & \mathbf{c}^2(x) \arrow[r, "\iota(\mathbf{c}^2(x))"] & \mathbf{c}^3(x) \arrow[r] & \cdots \arrow[r] & \mathbf{c}^i(x) \arrow[r, "\iota(\mathbf{c}^i(x))"] & \mathbf{c}^{i+1}(x).
\end{tikzcd}$$

By definition of the transport of a at x,

$$\begin{aligned} a^{(i)} &= \iota(\mathbf{c}^{i-1}(x))^{-1} \cdots \iota(\mathbf{c}(x))^{-1} \iota(x)^{-1} \iota(x) \iota(\mathbf{c}(x)) \cdots \iota(\mathbf{c}^{i-1}(x)) \iota(\mathbf{c}^i(x)) \\ &= \iota(\mathbf{c}^i(x)). \end{aligned}$$

By hypothesis, $a = \iota(x) = \tau^{-p}(x_1)$, so the result is true for $k, r = 0$. Since x is rigid, $\mathbf{c}(x) = \Delta^p x_2 \cdots x_l a$ is in left normal form as written. Then $\mathbf{c}(x)$ is also rigid. By applying the same argument to each cycling of x, it follows that $\mathbf{c}^i(x)$ is rigid for every $i > 0$, and that cycling just corresponds to cyclic permutation of the factors of x (up to conjugation by some power of Δ). More precisely, every l cyclings, the factors of the normal form of x go back to their original position, but each one is conjugated by Δ^{-p}. Therefore

$$a^{(kp+r)} = \iota(\mathbf{c}^{kp+r}(x)) = \tau^{-(k+1)p}(x_{r+1}),$$

as we wanted to show. □

We will use several times the following important property, which shows that, in $\mathrm{USS}(x)$, iterated transport of an element always comes back to the original element.

Lemma 2.24 ([18, Lemma 2.6]). *Let x belong to Garside group, and let u be a positive element. Assume that $x, x^u \in \mathrm{USS}(x)$. Let m be a positive integer such that $\mathbf{c}^m(x) = x$ and $\mathbf{c}^m(x^u) = x^u$. Then $u^{(km)} = u$ for some $k > 0$.*

We can deduce the following property for minimal simple elements.

Lemma 2.25. *Let $x \in \mathrm{USS}(x)$ and let u be a minimal simple element for x with respect to* $\mathrm{USS}(x)$. *Then $u^{(i)}$ is a minimal simple element for $\mathbf{c}^i(x)$, for all $i \geq 1$.*

Proof. Suppose that $u^{(i)}$ is not a minimal simple element for some i. So there exist nontrivial positive elements a, b such that $u^{(i)} = ab$ and $(\mathbf{c}^i(x))^a \in \mathrm{USS}(x)$. By Lemma 2.24, iterated transport of a comes back to a, hence $a^{(j)} \neq 1$ for every $j > 0$ (as if a trivial element is obtained, all the forthcoming transports would be trivial). In the same way, $b^{(j)} \neq 1$ for every $j > 0$. But we know, by Lemma 2.24 again, that $u^{(i+t)} = u$ for some $t > 0$. As transport preserves products, we have

$$u = u^{(i+t)} = (u^{(i)})^{(t)} = (ab)^{(t)} = a^{(t)}b^{(t)},$$

where $a^{(t)} \neq 1$, $b^{(t)} \neq 1$ and, by construction, $x^{(a^{(t)})} \in \mathrm{USS}(x)$. This contradicts the minimality of u. □

Definition 2.26. Let x be a rigid braid (so $x \in \mathrm{USS}(x)$) with left normal form $\Delta^p x_1 \cdots x_l$. We say that a simple factor of the normal form of x, say x_r, is *minimal* if $\tau^{-p}(x_r)$ is a minimal simple element for $\mathbf{c}^{r-1}(x)$ with respect to $\mathrm{USS}(x)$.

As a direct consequence of Lemmas 2.25 and 2.23, we have the following.

Corollary 2.27. *Let x be a rigid braid whose left normal form is $\Delta^p x_1 \cdots x_l$. If a simple factor of its normal form is minimal, then all factors are minimal.*

Proof. Suppose that x_r is minimal for some $r = 1, \dots, l$. This means that $\tau^{-p}(x_r)$ is a minimal simple element for $\mathbf{c}^{r-1}(x)$ with respect to $\mathrm{USS}(x)$. Since x is rigid, $\mathbf{c}^{r-1}(x)$ is also rigid. Hence, by Lemma 2.25, $(\tau^{-p}(x_r))^{(i)}$ is a minimal simple element for every $i > 0$.

On the other hand, by Lemma 2.23, every simple factor of the normal form of x will eventually appear as a transport of $\tau^{-p}(x_r)$, up to conjugation by some power of Δ. Since conjugation by Δ preserves minimal simple elements, the result follows. □

3 Generic braids

We will now present some results by Caruso and Wiest [8, 9] on generic braids, and adapt them to our purposes.

Let $\mathbf{B}(l)$ be the ball of radius l centered at 1 in the Cayley graph of the braid group B_n, with generators the simple braids. We are interested in the proportion of braids in $\mathbf{B}(l)$ which have a very particular ultra summit set, when l tends to infinity. In order to find this proportion, we will follow the arguments in [9].

Let

$$B_n^{p,l} = \{x \in B_n \mid \inf(x) = p,\ \ell(x) = l\}.$$

The sets $B_n^{p,l}$ are disjoint as left normal forms are unique. Since left normal forms are closely related to the so-called mixed normal forms [10], and the latter are geodesics in the aforementioned Cayley graph of B_n, it follows that

$$\mathbf{B}(l) = \bigsqcup_{k=0}^{l} \bigsqcup_{\eta=-l}^{l-k} B_n^{\eta,k}. \tag{3.1}$$

So $\mathbf{B}(l)$ is the disjoint union of a finite number of sets of the form $B_n^{\eta,k}$. Also, it is shown in [8] that

$$|B_n^{\eta,k}| = \Theta(\lambda^k)$$

for some $\lambda > 1$, meaning that the sequences $\frac{|B_n^{\eta,k}|}{\lambda^k}$ and $\frac{\lambda^k}{|B_n^{\eta,k}|}$ stay bounded as k tends to infinity.

Definition 3.1 ([9]). Let x be a braid with left normal form $x = \Delta^p x_1 \cdots x_l$. Let us write

$$P(x) = x_{2\cdot\lceil\frac{l}{5}\rceil+1} \cdots x_{l-2\cdot\lceil\frac{l}{5}\rceil}$$

(the middle fifth of the left normal form). A conjugation of x is *non-intrusive* if the normal form of the conjugated braid contains $P(x)$ as a subword.

Proposition 3.2 ([9]). *There exists a constant* $0 < \mu_R < 1$ *(which depends only on n) such that, among the braids in $B_n^{p,l}$, the proportion of those that can be non-intrusively conjugated to a rigid pseudo-Anosov braid is at least* $1 - \mu_R^l$*, for sufficiently large l.*

Lemma 3.3 ([8]). *Let w be a fixed braid of infimum* 0*. Then there exists a constant* $0 < \mu_M < 1$ *such that the proportion of braids $x \in B_n^{p,l}$ for which w appears as a set of consecutive factors of the normal form of $P(x)$ is at least* $1 - \mu_M^l$*, for sufficiently large l.*

We are interested in the following kind of braids:

Definition 3.4. Let x be a braid with left normal form $x = \Delta^p x_1 \cdots x_l$. We say that x is a σ_1*-non-intrusive braid* if x admits a non-intrusive conjugation to a rigid pseudo-Anosov braid and σ_1 and $\partial(\sigma_1)$ are simple factors of $P(x)$.

Corollary 3.5. *There exists a positive constant* $0 < \mu < 1$ *such that the proportion of σ_1-non-intrusive braids in $B_n^{p,l}$ is at least* $1 - \mu^l$*, for sufficiently large l.*

Proof. We just need to take $\mu > \max(\mu_R, \mu_{M_1}, \mu_{M_2})$, where μ_R is the constant appearing in Proposition 3.2, μ_{M_1} is the constant appearing in Lemma 3.3 for $w = \sigma_1$, and μ_{M_1} is the constant appearing in Lemma 3.3 for $w = \partial(\sigma_1)$. □

Corollary 3.6. *The proportion of σ_1-non-intrusive braids in $\mathbf{B}(l)$ tends to* 1 *as l tends to infinity. Moreover, this convergence happens exponentially fast.*

Proof. This is similar to [9, proof of Theorem 5.1]. The ball $\mathbf{B}(l)$ can be decomposed as in (3.1), where the size of each $B_n^{\eta,k}$ is $\Theta(\lambda^k)$ for some $\lambda > 1$. In each $B_n^{\eta,k}$, the proportion of σ_1-non-intrusive braids is at least $1-\mu^k$ for sufficiently large k. Let k_0 be the biggest value of k such that the mentioned proportion is smaller than $1-\mu^k$. Then, for $l > k_0$, we can decompose

$$\mathbf{B}(l) = \Bigg(\bigsqcup_{k=0}^{k_0} \bigsqcup_{\eta=-l}^{l-k} B_n^{\eta,k}\Bigg) \bigsqcup \Bigg(\bigsqcup_{k=k_0+1}^{l} \bigsqcup_{\eta=-l}^{l-k} B_n^{\eta,k}\Bigg).$$

The size of the set on the left is $O((k_0+1)(2l+1)\lambda^{k_0})$, that is, $O(2l+1)$. In the set on the right, the number of braids which are *not* σ_1-non-intrusive is

$$O\big((2l-k_0)(\lambda\mu)^{k_0+1} + (2l-k_0-1)(\lambda\mu)^{k_0+2} + \cdots \\ + (l+2)(\lambda\mu)^{l-1} + (l+1)(\lambda\mu)^{l}\big).$$

Now notice that the number of elements in $\mathbf{B}(l)$ is at least $|B_n^{0,l}| = \Theta(\lambda^l)$. Therefore, the proportion of braids which are *not* σ_1-non-intrusive in $\mathbf{B}(l)$ is

$$O\bigg(\frac{2l+1}{\lambda^l} + \frac{(2l-k_0)\mu^{k_0+1}}{\lambda^{l-k_0-1}} + \frac{(2l-k_0-1)\mu^{k_0+2}}{\lambda^{l-k_0-2}} + \cdots \\ + \frac{(l+2)\mu^{l-1}}{\lambda} + (l+1)\mu^l\bigg) \le O\big((2l+1)(l+1)(\max(\lambda^{-1},\mu))^l\big).$$

This tends exponentially fast to 0, hence the proportion of σ_1-non-intrusive braids in $\mathbf{B}(l)$ tends exponentially fast to 1, as l tends to infinity. □

4 Minimal ultra summit set

Once we showed that σ_1-non-intrusive braids are generic, we will see that their ultra summit sets are particularly simple. Recall the directed graph Γ_x related to the ultra summit set $\mathrm{USS}(x)$ of a braid x, and recall also that the arrows in Γ_x can be black or grey (or both). Since $\mathrm{USS}(x) \subset \mathrm{SSS}(x)$, the canonical length of all elements in $\mathrm{USS}(x)$ is the same, namely $\ell_s(x)$.

Definition 4.1. Let $x \in B_n$. We say that $\mathrm{USS}(x)$ is *minimal* if $\ell_s(x) > 1$ and, for every vertex y in Γ_x there is only one black arrow starting at y, corresponding to $\iota(y)$, and only one grey arrow starting at y, corresponding to $\iota(y^{-1}) = \partial(\varphi(y))$.

Lemma 4.2. *Let $x \in B_n$. If* $\mathrm{USS}(x)$ *is minimal, then all elements in* $\mathrm{USS}(x)$ *are rigid.*

Proof. Recall that y is rigid if and only if $\mathfrak{p}(y) = \iota(y) \wedge \iota(y^{-1}) = 1$. Suppose that $\mathrm{USS}(x)$ is minimal, that is, for every $y \in \mathrm{USS}(x)$, $\iota(y)$ and $\iota(y^{-1})$ are minimal simple elements for y. This means that $y^{\iota(y)}$ and $y^{\iota(y^{-1})}$ belong to $\mathrm{USS}(x)$ and no proper positive prefix of either conjugating element conjugates y to $\mathrm{USS}(x)$. Let $s = \iota(y) \wedge \iota(y^{-1})$. By Theorem 2.17, $y^s \in \mathrm{USS}(x)$. Since s is a prefix of both $\iota(y)$ and $\iota(y^{-1})$, which are minimal, there are just two possible cases: Either $s = \iota(y) = \iota(y^{-1})$ or $s = \iota(y) \wedge \iota(y^{-1}) = 1$.

Suppose that $s = \iota(y) = \iota(y^{-1})$. That is, $\iota(y) = \partial(\varphi(y))$. We would then have $\varphi(y)\iota(y) = \varphi(y)\partial(\varphi(y)) = \Delta$. But then, if $\Delta^p y_1 \cdots y_l$ is the left normal form of y, we would have $\mathbf{c}(y) = \Delta^p y_2 \cdots y_{l-1}\varphi(y)\iota(y) = \Delta^p y_2 \cdots y_{l-1}\Delta$, hence the canonical length of $\mathbf{c}(y)$ would be smaller than the canonical length of y, contradicting the fact that $y \in \mathrm{USS}(x)$. Hence, $\iota(y) \wedge \iota(y^{-1}) = 1$, meaning that y is rigid for every $y \in \mathrm{USS}(x)$. □

Suppose that $\mathrm{USS}(x)$ is minimal, so by Lemma 4.2 it consists of rigid elements. By definition, at every vertex y of Γ_x there are two outgoing arrows, a black one corresponding to $\iota(y)$, and a grey one corresponding to $\iota(y^{-1}) = \partial(\varphi(y))$. But once the conjugation by $\iota(y)$ is performed, as y is rigid, the last factor of the element obtained is precisely $\iota(y)$. This means that every vertex of Γ_x has exactly one incoming black arrow, which corresponds to its final factor. In the same way, as y^{-1} is rigid (rigidity is preserved by taking inverses), the same argument applies to the grey arrows: Every vertex of Γ_x has exactly one incoming grey arrow, which corresponds to the final factor of its inverse.

Therefore, the graph Γ_x, locally at y, is as sketched in Figure 1, where the grey arrows are represented by dashed lines.

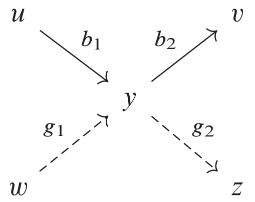

Figure 1. Black and grey arrows in Γ_x, locally at a vertex y in a minimal ultra summit set.

Lemma 4.3. *Let* $\mathrm{USS}(x)$ *be minimal and let* $y \in \mathrm{USS}(x)$*. Using the notations in Figure* 1*, the following conditions hold:*

(i) $b_1 b_2$ *is left-weighted.*

(ii) $g_1 g_2$ *is left-weighted.*

(iii) $b_1 g_2 = \Delta$.

(iv) $g_1 b_2 = \Delta$.

Proof. By Lemma 4.2, we know that y is rigid, and then y^{-1} is also rigid. We recall that each black arrow is the initial factor of its source, and the final factor of its target. Also, each grey arrow is the initial factor of the inverse of its source, and the final factor of the inverse of its target. Hence:

(i) $b_1 = \varphi(y)$ and $b_2 = \iota(y)$. Since y is rigid, $b_1 b_2$ is left-weighted.

(ii) $g_1 = \varphi(y^{-1})$ and $g_2 = \iota(y^{-1})$. Since y^{-1} is rigid, $g_1 g_2$ is left-weighted.

(iii) $b_1 = \varphi(y)$ and $g_2 = \iota(y^{-1}) = \partial(\varphi(y))$ by Lemma 2.8. Hence $b_1 g_2 = \Delta$.

(iv) $b_2 = \iota(y)$ and $g_1 = \varphi(y^{-1}) = \partial^{-1}(\iota(y))$ by Lemma 2.8. Hence we have $g_1 b_2 = \Delta$. □

Corollary 4.4. *If* USS(x) *is minimal, then one of the following conditions holds:*

(i) USS(x) *has two orbits under cycling, conjugate to each other by* Δ

(ii) USS(x) *has one orbit under cycling, conjugate to itself by* Δ.

Proof. By Lemma 4.2, USS(x) consists of rigid elements. As every rigid element belongs to USS(x) (cycling is roughly a cyclic permutation of its factors), it follows that USS(x) is the set of rigid conjugates of x. (This was already shown in [20, Theorem 1], anyway).

Let $y \in$ USS(x). As USS(x) is minimal, every black arrow in Γ_x corresponds to a cycling. Hence, iteratively conjugating y by black arrows, we obtain the whole orbit of y under cycling. Now consider the grey arrow g_2 starting at y, and denote by b_1 the black arrow ending at y as in Figure 1. Denote by u the source of b_1 and by z the target of g_2. By Lemma 4.3, $b_1 g_2 = \Delta$, hence $u^\Delta = z$. Since u belongs to the orbit of y under cycling, say $\mathcal{O}_y$, it follows that z belongs to $\tau(\mathcal{O}_y)$. This happens for all elements in USS(x): Every grey arrow in Γ_x sends an element in an orbit $\mathcal{O}$, to an element in $\tau(\mathcal{O})$.

Therefore, if we start at an orbit $\mathcal{O}$, conjugating by a black arrow we stay in $\mathcal{O}$, and conjugating by a grey arrow we pass to the orbit $\tau(\mathcal{O})$. As all elements in USS(x) are connected by black and grey arrows, and Δ^2 is a central element, it follows that in USS(x) there are at most two orbits: If the orbit $\mathcal{O}$ is conjugate to itself by Δ, there will be just one orbit; otherwise, there will be two orbits conjugate to each other by Δ. □

Suppose that USS(x) is minimal and has one orbit which is conjugate to itself by Δ. For simplicity, we will assume that $x \in$ USS(x). We will distinguish two cases, depending on whether or not Δ commutes with x. If $\Delta \in Z(x)$ (the centralizer of x) then, as conjugation by Δ sends normal forms to normal forms, every factor in the left normal form of x commutes with Δ, hence every element in the orbit of x is conjugate to itself by Δ. That is, $\tau(y) = y$ for every element $y \in \mathrm{USS}(x)$. If $\Delta \notin Z(x)$, then x is not conjugate to itself by Δ and this implies that the length of its orbit is an even number, as shown in the following result.

Proposition 4.5. *Let $x \in \mathrm{USS}(x)$ be a braid such that $\mathrm{USS}(x)$ is minimal and has one orbit which is conjugate to itself by Δ. Let $x = X_1, X_2, \dots, X_k$ be the elements in this orbit, where $\mathbf{c}(X_j) = X_{j+1}$ (indices are considered modulo k). If $\Delta \notin Z(x)$, then k is even and $\tau(X_j) = X_{j+\frac{k}{2}}$ for $j = 1, \dots, k$.*

Proof. Let $\Delta^p x_1 \cdots x_l$ be the left normal form of x, and suppose that p is even. Then one has $\tau(x) = \Delta^p \tau(x_1) \cdots \tau(x_l) \in \mathrm{USS}(x)$ and $\tau(x) \neq x$ as $\Delta \notin Z(x)$. Since $\mathrm{USS}(x)$ has a single orbit, one can go from x to $\tau(x)$ by iterated cycling, and since x is rigid and p is even, each cycling corresponds to a cyclic permutation of the factors in the left normal form of x. Hence, there is some r, $0 < r < k$, such that $\tau(x) = \mathbf{c}^r(x)$. That is, $\tau(x_j) = x_{j+r}$ for all $j = 1, \dots, l$, where indices are taken modulo l. This implies that $x_j = \tau^2(x_j) = x_{j+2r}$, and hence $\mathbf{c}^{2r}(x) = x$. Since the orbit of x under cycling has length k, it follows that $k \mid 2r$. But since $k > r$, we finally obtain $k = 2r$.

Therefore, the length of the orbit is an even number and $\tau(X_j) = X_{j+\frac{k}{2}}$ for every $X_j \in \mathrm{USS}(x)$. Notice that

$$x = \Delta^p (x_1 \cdots x_r)(\tau(x_1) \cdots \tau(x_r)) \cdots (x_1 \cdots x_r)(\tau(x_1) \cdots \tau(x_r)).$$

Now suppose that p is odd. In this case, since x is rigid, the left normal form of x^2 is

$$x^2 = \Delta^{2p} \tau(x_1) \cdots \tau(x_l) x_1 \cdots x_l,$$

and x^2 is also rigid. Hence, cycling x^2 corresponds to a cyclic permutation of the factors in its left normal form and, for every $i \geq 0$, the normal form of $\mathbf{c}^i(x^2)$ has $2l$ non-Δ factors, the last l being the factors of $\mathbf{c}^i(x)$ and the first l being their conjugates by Δ. Therefore, the lengths of the orbits of x and x^2 coincide, and the orbit of x^2 is precisely $x^2 = (X_1)^2, (X_2)^2, \dots, (X_k)^2$.

Since the infimum of x^2 is even, we can apply the previous case to x^2 and we obtain that k is even, and that $\tau((X_j)^2) = (X_{j+\frac{k}{2}})^2$, which is equivalent to $\tau(X_j) = X_{j+\frac{k}{2}}$ for $j = 1, \dots, k$ (as the braids involved are rigid). □

Now that we know some properties of the structure of minimal ultra summit sets, we proceed to show that this case is generic. Let us first show that the minimality of the ultra summit set is a local property.

Theorem 4.6. *Let x be a rigid braid with $\ell(x) > 1$. Then* $\mathrm{USS}(x)$ *is minimal if and only if $\iota(x)$ and $\iota(x^{-1})$ are minimal simple elements for x.*

Proof. The condition is clearly necessary. To show that it is sufficient, suppose that $\iota(x)$ and $\iota(x^{-1})$ are minimal simple elements for x. Let $\mathcal{O}_x$ be the orbit of x under cycling, and let $\widetilde{\mathcal{O}}_x$ be the orbit of x under twisted decycling.

It is shown in [4] that if x is rigid and $\ell(x) > 1$, then $\mathrm{USS}(x)$ is the set of rigid conjugates of x. Then $\mathrm{USS}(x^{-1})$ is the set of rigid conjugates of x^{-1}, so one has $\mathrm{USS}(x^{-1}) = \mathrm{USS}(x)^{-1}$. Also, as all elements are rigid, decycling is just the inverse of cycling, hence twisted decycling sends any element in $\mathcal{O}_x$ to an element in $\tau(\mathcal{O}_x)$, and viceversa. It follows that $\mathcal{O}_x \cup \tau(\mathcal{O}_x) = \widetilde{\mathcal{O}}_x \cup \tau(\widetilde{\mathcal{O}}_x)$. Call this set $\mathcal{U}$.

By Corollary 2.27, as $\iota(x)$ is a minimal simple element for x, all elements in the normal form of x are minimal, so $\iota(y)$ is a minimal simple element for y, for every $y \in \mathcal{O}_x$. Conjugating the whole picture by Δ, it follows that $\iota(z)$ is a minimal simple element for z, for every z in $\tau(\mathcal{O}_x)$. So we cannot escape from $\mathcal{U} = \mathcal{O}_x \cup \tau(\mathcal{O}_x)$ conjugating by black arrows, as all black arrows correspond to cyclings.

Now notice that one can obtain $\Gamma_{x^{-1}}$ from Γ_x just by replacing each vertex y with y^{-1}. The arrows will have the same labels, but their colors will be exchanged. Since $\iota(x^{-1})$ is a minimal simple element for x, it is also a minimal simple element for x^{-1}. Applying Corollary 2.27 to x^{-1}, we have that all black arrows in $\mathcal{O}_{x^{-1}} \cup \tau(\mathcal{O}_{x^{-1}})$ correspond to cyclings. Notice that applying twisted decycling to a braid y is equivalent to applying cycling to its inverse (the conjugating element is $\iota(y^{-1})$ in both cases). Hence $(\mathcal{O}_{x^{-1}})^{-1} = \widetilde{\mathcal{O}}_x$. Therefore, all grey arrows in $\widetilde{\mathcal{O}}_x \cup \tau(\widetilde{\mathcal{O}}_x)$ correspond to twisted decyclings. So we cannot escape from $\mathcal{U} = \widetilde{\mathcal{O}}_x \cup \tau(\widetilde{\mathcal{O}}_x)$ conjugating by grey arrows.

We have then shown that for every element $y \in \mathcal{U}$, $\iota(y)$ and $\iota(y^{-1})$ are minimal simple elements for y. Hence y only admits these two minimal simple elements (as every minimal simple element must be a prefix of one of these, by Corollary 2.18), and conjugating y by any of these will take us again into $\mathcal{U}$. As we can obtain the whole $\mathrm{USS}(x)$ starting with any element and conjugating by minimal simple elements, it follows that $\mathcal{U} = \mathrm{USS}(x)$. Therefore, $\mathrm{USS}(x)$ is minimal. □

Theorem 4.7. *Let x be a σ_1-non-intrusive braid. Then* $\mathrm{USS}(x)$ *is minimal.*

Proof. By hypothesis, x admits a non-intrusive conjugation to a rigid pseudo-Anosov braid y, where σ_1 and $\partial(\sigma_1)$ are simple factors of the normal form of y. Clearly σ_1 is minimal as it cannot be decomposed, hence by Corollary 2.27 all simple factors in the normal form of y are minimal. Therefore, $\iota(y)$ is a minimal simple element for y.

Now $\partial(\sigma_1)$ is also a simple factor of the normal form of y. This implies, using the relation between the normal forms of y and y^{-1}, that some factor in the normal form of y^{-1} is equal to either σ_1 or σ_{n-1}. Then all simple factors in the normal form of y^{-1} are minimal. Therefore $\iota(y^{-1})$ is a minimal simple element for y^{-1}, so it is a minimal simple element for y (recall that y is rigid).

Therefore, both $\iota(y)$ and $\iota(y^{-1})$ are minimal simple elements for y. Also, $\ell(y) > 1$ as its left normal form contains both σ_1 and $\partial(\sigma_1)$. By Theorem 4.6, $\mathrm{USS}(y)$, that is, $\mathrm{USS}(x)$, is minimal. □

The next result follows immediately from Theorem 4.7 and Corollary 3.6.

Theorem 4.8. *The proportion of braids in* $\mathbf{B}(l)$ *whose ultra summit set is minimal tends to 1 exponentially fast, as* l *tends to infinity.*

5 The centralizer of generic braids

We now have all the ingredients to describe the centralizer of a generic braid. We know from Theorem 4.8 that braids with minimal ultra summit sets are generic. Hence, we will consider braids x such that $\mathrm{USS}(x)$ is minimal, and we will describe an explicit set of generators for $Z(x)$, the centralizer of x. According to Corollary 4.4, we can have two different situations, depending on whether $\mathrm{USS}(x)$ has one or two orbits under cycling.

Recall that one can obtain an element $y \in \mathrm{USS}(x)$ by applying iterated cyclic slidings (or iterated cyclings and decyclings) to x. This gives a conjugating element c such that $y = c^{-1}xc$, and then $Z(y) = c^{-1}Z(x)c$. Therefore, in order to describe $Z(x)$ it suffices to describe $Z(y)$, where $y \in \mathrm{USS}(x)$. We will then assume, for the rest of this section, that $x \in \mathrm{USS}(x)$ and, in particular, that x is rigid.

Following the ideas in [15], we see that in order to compute a generating set for $Z(x)$, we just need to know Γ_x. Suppose that $a \in Z(x)$. Then $a = \Delta^{-2t}b$ for some $t \geq 0$, where b is a positive braid. Then b can be decomposed as a product of minimal simple elements, which corresponds to an oriented path in Γ_x that starts and finishes at x. The positive braid Δ^2 can also be decomposed as a loop in Γ_x based at x. Hence, every element in $Z(x)$ can be decomposed as a product of loops in Γ_x, so a set of generators for $Z(x)$ is obtained from a set of generators of the fundamental group $\pi_1(\Gamma_x, x)$, replacing each loop by the braid it represents.

Computing a set of generators of $\pi_1(\Gamma_x, x)$ is a well-known procedure [24]: Choose a maximal tree T in Γ_x. For every vertex $v \in \Gamma_x$, call γ_v the only simple path in T going from x to v. Let A be the set of arrows in $\Gamma_x \setminus T$ and, for every $\lambda \in A$, denote by $s(\lambda)$ and $t(\lambda)$ the source and the target of λ, respectively.

Then $\pi_1(\Gamma_x, x)$ is generated by

$$F = \{\gamma_{s(\lambda)}\lambda\gamma_{t(\lambda)}^{-1} \mid \lambda \in A\}.$$

If we denote by ρ the homomorphism which maps $\pi_1(\Gamma_x, x)$ onto $Z(x)$, which sends each path to its associated element, then $\rho(F)$ is a generating set for $Z(x)$.

Let us then study the graphs Γ_x with detail. We shall need the following.

Definition 5.1. Given $x \in B_n$, we define the *preferred cycling conjugator* $\mathrm{PC}(x)$ of x as the product of conjugating elements corresponding to iterated cycling until the first repetition. That is, if t is the smallest positive integer such that $\mathbf{c}^t(x) = \mathbf{c}^i(x)$ for some $0 \leq i < t$, then

$$\mathrm{PC}(x) = \iota(x)\iota(\mathbf{c}(x))\cdots\iota(\mathbf{c}^{t-1}(x)).$$

Notice that if $x \in \mathrm{SSS}(x)$ and one conjugates x by $\mathrm{PC}(x)$, one obtains an element in $\mathrm{USS}(x)$. Notice also that if $x \in \mathrm{USS}(x)$, then $\mathrm{PC}(x)$ is the conjugating element along the whole cycling orbit of x. In particular, if $x \in \mathrm{USS}(x)$, then $\mathrm{PC}(x)$ commutes with x. Notice also that if x is rigid, the above expression of $\mathrm{PC}(x)$ is in left normal form as written. That is, the product $\iota(\mathbf{c}^{i-1}(x))\iota(\mathbf{c}^i(x))$ is left-weighted for every $i \geq 0$.

Let us distinguish cases depending on the number of orbits in $\mathrm{USS}(x)$.

5.1 Ultra summit set with two orbits

Let x be a rigid braid such that $\mathrm{USS}(x)$ is minimal and has two orbits under cycling, conjugated to each other by Δ (see Corollary 4.4). Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be these two orbits, and let k be the length of each one. We can assume that $x \in \mathcal{O}_1$ and $\mathcal{O}_2 = \tau(\mathcal{O}_1)$. Notice that $k = \ell(\mathrm{PC}(x))$. Denote by $X_{1,j}$ and $X_{2,j}$ the elements in orbits $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively, where $\tau(X_{i,j}) = X_{3-i,j}$ and $\mathbf{c}(X_{i,j}) = X_{i,j+1}$ for every $j = 1, \ldots, k$, where $i = 1, 2$ and the second subindex is considered modulo k.

Notation 5.2. Let Γ_x be the graph associated to $\mathrm{USS}(x)$. We denote by a_j the black arrow in $\mathcal{O}_1$ starting at $X_{1,j}$, b_j the black arrow in $\mathcal{O}_2$ starting at $X_{2,j}$, α_j the grey arrow going from $\mathcal{O}_1$ to $\mathcal{O}_2$ ending at $X_{2,j}$ and β_j the grey arrow going from $\mathcal{O}_2$ to $\mathcal{O}_1$ ending at $X_{1,j}$, for every $j = 1, \ldots, k$. See Figure 2, where grey arrows are represented by dashed lines.

Theorem 5.3. *Let x be a rigid braid such that* $\mathrm{USS}(x)$ *minimal and has two orbits under cycling. Then* $Z(x) = \langle \mathrm{PC}(x), \Delta^2 \rangle$.

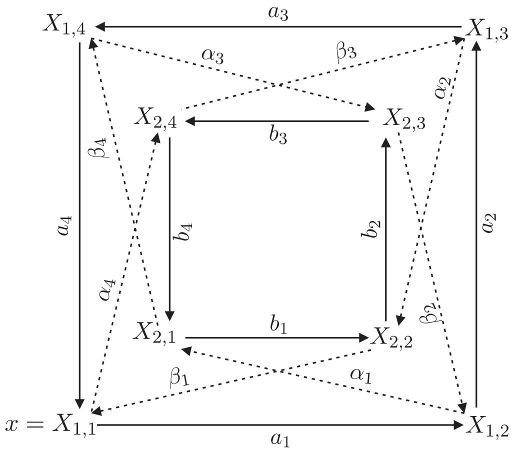

Figure 2. The ultra summit set $\mathrm{USS}(x)$ with two orbits, for $k = 4$.

Proof. The inclusion

$$\langle \mathrm{PC}(x), \Delta^2 \rangle \subseteq Z(x)$$

is clear, as Δ^2 is central and $\mathrm{PC}(x) \in Z(x)$ by construction. Let us show the converse inclusion by obtaining a set of generators for $Z(x)$, coming from a generating set for $\pi_1(\Gamma_x, x)$. Let T be the maximal subtree of Γ_x whose arrows are $a_1, \dots, a_{k-1}, b_1, \dots, b_{k-1}, \alpha_1$. Every arrow $\lambda \in \Gamma_x \setminus T$ will produce a generator $F_\lambda = \gamma_{s(\lambda)} \lambda \gamma_{t(\lambda)}$ (by abuse of notation, we will identify the path in Γ_x with the braid it represents). We need to show that F_λ can be written as a product of $\mathrm{PC}(x)$, Δ^2 and their inverses, for every $\lambda \in \Gamma_x \setminus T = \{a_k, b_k, \alpha_2, \dots, \alpha_k, \beta_1, \dots, \beta_k\}$.

First, $F_{a_k} = a_1 \cdots a_k = \mathrm{PC}(x)$, so the claim holds when $\lambda = a_k$. Now

$$F_{b_k} = a_1 \alpha_1 b_1 \cdots b_{k-1} b_k (a_1 \alpha_1)^{-1}.$$

By Lemma 4.3, $a_j \alpha_j = \Delta$ for every $j = 1, \dots, k$. Then

$$F_{b_k} = \Delta b_1 \cdots b_k \Delta^{-1}.$$

Recall that the two orbits under cycling are conjugate by Δ and $\tau(X_{1,j}) = X_{2,j}$ for every $j = 1, \dots, k$. This implies $\tau(a_j) = b_j$, hence

$$F_{b_k} = a_1 \cdots a_k = \mathrm{PC}(x).$$

Next, for $j = 2, \dots, k-1$ one has

$$\begin{aligned} F_{\alpha_j} &= a_1 \cdots a_j \alpha_j b_{j-1}^{-1} \cdots b_1^{-1} (a_1 \alpha_1)^{-1} \\ &= a_1 \cdots a_{j-1} \Delta b_{j-1}^{-1} \cdots b_1^{-1} \Delta^{-1} \\ &= a_1 \cdots a_{j-1} a_{j-1}^{-1} \cdots a_1^{-1} = 1. \end{aligned}$$

Now

$$\begin{aligned} F_{\alpha_k} &= \alpha_k b_{k-1}^{-1} \cdots b_1^{-1} \alpha_1^{-1} a_1^{-1} \\ &= \alpha_k b_k b_k^{-1} b_{k-1}^{-1} \cdots b_1^{-1} \alpha_1^{-1} a_1^{-1} \\ &= \Delta b_k^{-1} \cdots b_1^{-1} \Delta^{-1} \\ &= a_k^{-1} \cdots a_1^{-1} = \mathrm{PC}(x)^{-1}. \end{aligned}$$

It remains to show that the loops determined by the arrows β_j belong to the set $\langle \mathrm{PC}(x), \Delta^2 \rangle$. For $j = k$,

$$F_{\beta_k} = a_1 \alpha_1 \beta_k a_{k-1}^{-1} \cdots a_1^{-1}.$$

Notice that $b_k \beta_k = \Delta$ by Lemma 4.3. Hence

$$F_{\beta_k} = a_1 \alpha_1 b_k^{-1} \Delta a_{k-1}^{-1} \cdots a_1^{-1} = \Delta b_k^{-1} \Delta a_{k-1}^{-1} \cdots a_1^{-1} = \Delta^2 \mathrm{PC}(x)^{-1}.$$

Finally, for $j = 1, \ldots, k-1$, we have

$$F_{\beta_j} = a_1 \alpha_1 b_1 \cdots b_j \beta_j a_{j-1}^{-1} \cdots a_1^{-1} = \Delta b_1 \cdots b_{j-1} \Delta a_{j-1}^{-1} \cdots a_1^{-1} = \Delta^2. \qquad \square$$

5.2 Ultra summit set with one orbit

Consider now a rigid braid x such that $\mathrm{USS}(x)$ is minimal and has only one orbit $\mathcal{O}_1$ of length k, conjugated to itself by Δ (see Corollary 4.4). Recall that $k = \ell(\mathrm{PC}(x))$. Denote by $X_{1,j}$ the elements in this orbit, for every $j = 1, \ldots, k$, where $x = X_{1,1}$. We will distinguish two different cases, depending on whether Δ belongs to $Z(x)$ or not. By Proposition 4.5, if Δ does not belong to $Z(x)$, then k is even and $\tau(X_{1,j}) = X_{1,j+\frac{k}{2}}$ (where $j + \frac{k}{2}$ is considered modulo k) for every $j = 1, \ldots, k$. Otherwise, $\tau(X_{1,j}) = X_{1,j}$, for all $j = 1 \ldots, k$. Figure 3 illustrates the two cases.

Notation 5.4. Let Γ_x be the graph which represents the whole set $\mathrm{USS}(x)$ in any of the above two cases. We denote by a_j the black arrow starting at $X_{1,j}$, and by α_j the grey arrow starting at $X_{1,j+1}$, so that $a_j \alpha_j = \Delta$, for every $j = 1, \ldots, k$.

Theorem 5.5. *Let x be a rigid braid such that* $\mathrm{USS}(x)$ *is minimal and has only one orbit under cycling of length k. One has:*

(i) *If $\Delta^{-1} x \Delta \neq x$, then $Z(x) = \langle a_1 \cdots a_{\frac{k}{2}} \Delta^{-1}, \Delta^2 \rangle$.*

(ii) *If $\Delta^{-1} x \Delta = x$, then $Z(x) = \langle \mathrm{PC}(x), \Delta \rangle$.*

Proof. In both cases, the inclusions

$$\langle a_1 \ldots a_{\frac{k}{2}} \Delta^{-1}, \Delta^2 \rangle \subseteq Z(x) \quad \text{and} \quad \langle \mathrm{PC}(x), \Delta \rangle \subseteq Z(x)$$

are immediate.

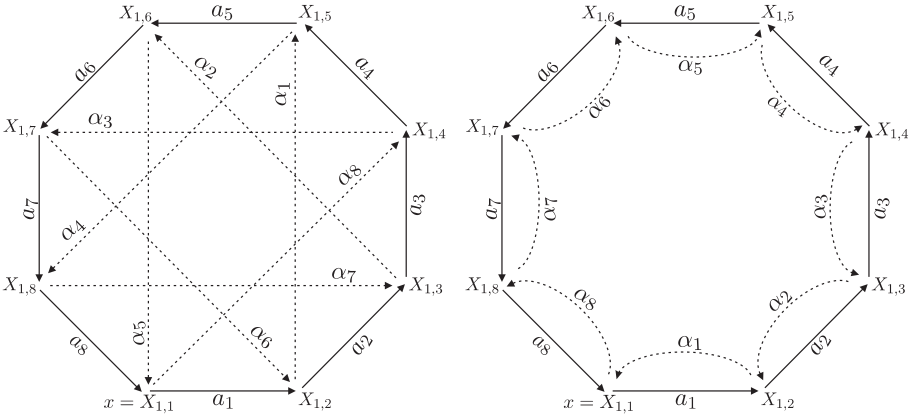

Figure 3. The ultra summit set $\mathrm{USS}(x)$ with one orbit such that $\Delta^{-1}x\Delta \neq x$ (left) and $\Delta^{-1}x\Delta = x$ (right), for $k = 8$.

We obtain a generating set for $Z(x)$ in a similar way as it is done in the proof of Theorem 5.3. Let T be the maximal tree in Γ_x, with base vertex $x = X_{1,1}$, containing the edges $a_1,\dots,a_{k-1}$.

Suppose first that $\Delta^{-1}x\Delta \neq x$. We need to show that

$$F_{\alpha_1},\dots,F_{\alpha_k},F_{a_k} \in \langle a_1\dots a_{\frac{k}{2}}\Delta^{-1},\Delta^2\rangle.$$

Write $g = a_1\cdots a_{\frac{k}{2}}\Delta^{-1}$. We claim that

$$a_1\cdots a_j\big(\alpha_j a^{-1}_{j+\frac{k}{2}-1}a^{-1}_{j+\frac{k}{2}-2}\cdots a^{-1}_{j+1}\big)a_j^{-1}\cdots a_1^{-1} = g^{-1}$$

for every $j = 1,\dots,k$, where every subindex is considered modulo k. In order to prove this claim, applying Lemma 4.3 and Proposition 4.5, we recall that $a_j\alpha_j = \Delta$ and $\tau(X_{1,j}) = X_{1,j+\frac{k}{2}}$. Hence $\tau(a_j) = a_{j+\frac{k}{2}}$ and then the left-hand side of the formula is equal to

$$\begin{aligned} a_1\cdots a_{j-1}\Delta a^{-1}_{j+\frac{k}{2}-1}\cdots a_1^{-1} &= \Delta a_{1+\frac{k}{2}}\cdots a_{j-1+\frac{k}{2}}a^{-1}_{j+\frac{k}{2}-1}\cdots a_1^{-1} \\ &= \Delta a^{-1}_{\frac{k}{2}}\cdots a_1^{-1} = g^{-1}, \end{aligned}$$

proving the claim. It follows immediately that $F_{\alpha_j} = g^{-1}$ for $j = 1,\dots,\frac{k}{2}$. For $j = \frac{k}{2}+1,\dots,k-1$, one has

$$\begin{aligned} F_{\alpha_j} &= a_1\cdots a_j\alpha_j a^{-1}_{j-\frac{k}{2}-1}\cdots a_1^{-1} = a_1\cdots a_{j-1}\Delta a^{-1}_{j-\frac{k}{2}-1}\cdots a_1^{-1} \\ &= a_1\cdots a_{j-1}a^{-1}_{j-1}\cdots a^{-1}_{\frac{k}{2}+1}\Delta = a_1\cdots a_{\frac{k}{2}}\Delta = g\Delta^2. \end{aligned}$$

For $j=k$,

$$\begin{aligned}F_{\alpha_k} &= \alpha_k a_{\frac{k}{2}-1}^{-1}\cdots a_1^{-1} = a_k^{-1}(a_k\alpha_k)a_{\frac{k}{2}-1}^{-1}\cdots a_1^{-1}\\ &= \Delta a_{\frac{k}{2}}^{-1}\cdots a_1^{-1} = g^{-1}.\end{aligned}$$

Finally,

$$\begin{aligned}F_{a_k} &= a_1\cdots a_k = a_1\cdots a_{\frac{k}{2}}a_{\frac{k}{2}+1}\cdots a_k\\ &= a_1\cdots a_{\frac{k}{2}}(\Delta^{-1}a_1\cdots a_{\frac{k}{2}-1}\Delta) = g^2\Delta^2.\end{aligned}$$

This shows the result when $\Delta^{-1}x\Delta \neq x$.

Suppose now that $\Delta^{-1}x\Delta = x$. We have to show that

$$F_{\alpha_1},\dots,F_{\alpha_k},F_{a_k} \in \langle \mathrm{PC}(x),\Delta\rangle.$$

For $j=1,\dots,k-1$ we have

$$F_{\alpha_j} = a_1\cdots a_{j-1}(a_j\alpha_j)a_{j-1}^{-1}\cdots a_1^{-1}.$$

Recall that $a_j\alpha_j=\Delta$ and that $\tau(X_{1,j})=X_{1,j}$, so $\tau(a_i)=a_i$ for every i. Hence $F_{\alpha_j}=\Delta$ for $j=1,\dots,k-1$. Now

$$F_{\alpha_k} = \alpha_k a_{k-1}^{-1}\cdots a_1^{-1} = (\alpha_k a_k)a_k^{-1}\cdots a_1^{-1} = \Delta(\mathrm{PC}(x))^{-1}.$$

Finally, $F_{a_k}=a_1\cdots a_k=\mathrm{PC}(x)$. □

6 An algorithm to compute the centralizer of a braid

Our goal in this section is to present an algorithm to compute a generating set for the centralizer of a braid, based on the results from previous sections, which generically terminates in polynomial time and yields a minimal set of generators consisting of two elements.

Given $y\in B_n$, the first step will be to find $x\in \mathrm{USS}(y)$ and a conjugating element c such that $c^{-1}yc=x$. This is achieved by iterated applications of cyclic sliding to y (see [20]). Next we check whether x is rigid. If x is rigid, the algorithm continues by computing the minimal simple elements for x (see [16]). If $\iota(x)$ and $\iota(x^{-1})$ are equal to the minimal simple elements for x, then $\mathrm{USS}(y)=\mathrm{USS}(x)$ is minimal by Theorem 4.6. At this point, one needs to determine whether $\mathrm{USS}(y)$ has one or two orbits, and in the latter case whether x commutes with Δ, in order to apply Theorem 5.3 or Theorem 5.5. For that, we will check whether or not $\tau(x)$ belongs to the orbit of x, and in particular whether $\tau(x)$ is equal to x. This yields a generating set for $Z(x)$ given by Theorem 5.3 or Theorem 5.5 and one obtains a generating set for $Z(y)$ from the equality $Z(y)=cZ(x)c^{-1}$.

We remark that if x is not rigid or if $\mathrm{USS}(y)$ is not minimal (a case which is not generic, as shown in Theorem 4.8), the centralizer of y can be determined by the algorithm given by Franco and González-Meneses [15]. The algorithm in [15] refers to $\mathrm{SSS}(x)$ instead of $\mathrm{USS}(x)$, but it can be equally applied to $\mathrm{USS}(x)$, or even to $\mathrm{SC}(x)$, considerably improving its efficiency.

The pseudocode of the algorithm described in the previous paragraphs is shown in Algorithm 1.

Algorithm 1. Compute a generating set for the centralizer of a braid.

```
Input: A braid $y \in B_n$ given in left normal form.
Output: A generating set for $Z(y)$.
Compute $x \in \mathrm{USS}(y)$ and $c \in B_n$ such that $c^{-1}yc = x$ by iterated cyclic sliding [19,20].
if $x$ is rigid then
    Applying iterated cycling to $x$, compute $k$ = length of the orbit of $x$ under cycling.
    Compute the minimal simple elements for $x$, see [18].
    if $\iota(x)$ and $\iota(x^{-1})$ are the minimal simple elements for $x$ then
        Compute $\tau(x)$.
        if $\tau(x) = x$ then
            $A = \mathrm{PC}(x)$, $B = \Delta$. [Theorem 5.5]
        else if $k$ is even and $\tau(x) = \mathbf{c}^{\frac{k}{2}}(x)$ then
            $A = a_1 \cdots a_{\frac{k}{2}} \Delta^{-1}$, $B = \Delta^2$. [Theorem 5.5]
        else
            $A = \mathrm{PC}(x)$, $B = \Delta^2$. [Theorem 5.3]
        end if
        return $cAc^{-1}, cBc^{-1}$.
    else
        Compute $Z(x)$ applying the algorithm in [15] (using $\mathrm{SC}(x)$).
    end if
else
    Compute $Z(x)$ applying the algorithm in [15] (using $\mathrm{SC}(x)$).
end if
```

In order to study the complexity of this algorithm, we naturally measure the input braids by their canonical length. But we will also see how the number n of strands affects the computation, as this is important in practice, and because the algorithms in braid groups are usually programmed allowing inputs with an arbitrary number of strands.

We saw in Theorem 4.8 that σ_1-non-intrusive braids are generic in B_n, so we will study the complexity of computing the centralizer of such a braid.

Proposition 6.1. *Let $y \in B_n$ be a σ_1-non-intrusive braid such that* $\mathrm{USS}(y)$ *is minimal. If $l = \ell(y)$, the complexity of computing a generating set for $Z(y)$ using Algorithm* 1 *is $O(l^2 n^4 \log n)$.*

Proof. The first step of Algorithm 1 consists of applying iterated cyclic sliding until the first repeated element x is obtained, computing at the same time the conjugating element c, which is the product of the conjugating elements for each cyclic sliding. By [19, Algorithm 1], the cost of computing x and c is $O(Tln \log n)$, where T is the number of cyclic slidings applied.

Recall from Lemma 4.2 that, as $\mathrm{USS}(y)$ is minimal, x is rigid. In [20] it is shown that, if a braid y is conjugate to a rigid braid x, then the product of all conjugating elements corresponding to iterated cyclic sliding is the shortest positive element conjugating y to a rigid braid. On the other hand, as $y = \Delta^p y_1 \cdots y_l$ is σ_1-non-intrusive, there is a positive prefix of $\tau^{-p}(y_1 \cdots y_l)$ conjugating y to a rigid element. This implies that the number of cyclic slidings needed to conjugate y to a rigid braid is smaller than the letter length of $\tau^{-p}(y_1 \cdots y_l)$, that is, smaller than $l\|\Delta\| = \frac{1}{2}ln(n-1)$. Since the obtained element x is rigid, the next cyclic sliding yields $\mathfrak{s}(x) = x$ and the repeated element is immediately obtained. Hence $T = O(ln^2)$, and the first step of the algorithm takes time $O(l^2n^3 \log n)$.

The following step is to check whether x is rigid, which means to determine whether $\mathfrak{p}(x) = 1$. This was already made in the previous step, when $\mathfrak{s}(x)$ was computed. So this step has no cost.

Let $x = \Delta^q x_1 \cdots x_r$ in left normal form. Since $x \in \mathrm{USS}(y) \subset \mathrm{SSS}(y)$, we have $q \geq p$ and $r \leq l$. Since x is rigid, the next step in Algorithm 1 is to apply iterated cycling to x in order to compute k, the length of its orbit under cycling. As x is rigid, the left normal form of $\mathbf{c}(x)$ is precisely $\Delta^p x_2 \cdots x_r \tau^{-p}(x_1)$. In order to compute this normal form, one just needs to apply τ^{-p} to x_1. If p is even, then τ^{-p} is trivial, and if p is odd, τ^{-p} is equal to τ. If the simple factors are stored, as usual, as a list of n numbers (from 1 to n) representing the permutation they induce, applying τ to x_1 is $O(n)$. Hence, each cycling takes $O(n)$. After each cycling, one need to check whether the obtained element equals x. Comparing two normal forms is $O(ln \log n)$ (we compare two lists of $rn \leq ln$ numbers between 1 and n, and each number has length at most $\log n$). We know that $\mathbf{c}^{2r}(x) = x$, hence $k \leq 2r \leq 2l$, so the number of total comparisons we will perform is $O(l)$. Therefore, in order to compute k one performs $k \leq 2l$ cyclings (total cost $O(ln)$) and $O(l)$ comparisons (total cost $O(l^2 n \log n)$). Hence, the cost of computing k is $O(l^2 n \log n)$.

The next step consists of computing the minimal simple elements for x. The complexity of this computation is described in [5, Proposition 4.10] for general elements in a Garside group, but in our case we are in a simpler situation: We are computing the minimal simple elements for a braid $x \in B_n$ which is rigid. We can proceed as follows: For every generator σ_i $(i = 1, \dots, n-1)$, compute the minimal positive element ρ_i such that $\sigma_i \preccurlyeq \rho_i$ and $x^{\rho_i} \in \mathrm{SSS}(x)$. By [19, Proposition 3.4], in order to compute this element we start with $\rho = \sigma_i$, and while $\ell(x^\rho) > r$ we replace ρ with $\rho(1 \vee (x^\rho)^{-1}\Delta^p \vee x^\rho\Delta^{-p-r})$. The final value of ρ is precisely ρ_i. The element $1 \vee (x^\rho)^{-1}\Delta^p$ is the leftmost factor in the right normal form of $(x^\rho)^{-1}\Delta^p$, and $1 \vee x^\rho\Delta^{-p-r}$ is the leftmost factor in the right normal form of $x^\rho\Delta^{-p-r}$. Hence, the element $1 \vee (x^\rho)^{-1}\Delta^p \vee x^\rho\Delta^{-p-r}$ can be obtained at the cost of computing two right normal forms and the least common multiple of two simple elements. The total cost of one iteration is $O(l^2 n \log n)$.

Once ρ_i is computed, we can use [20, Theorem 2]: Since $x^{\rho_i} \in \mathrm{SSS}(y)$ and it is conjugate to a rigid element, the shortest positive element α_i conjugating x^{ρ_i} to a rigid element is precisely the product of all conjugating elements for iterated cyclic sliding of x^{ρ_i}. By construction $\rho_i\alpha_i$ will be the minimal element which admits σ_i as a prefix and conjugates x to a rigid element. Since Δ admits σ_i as a prefix and conjugates x to a rigid element, it follows that ρ_i is a prefix of Δ, so it is simple. Hence, the total number of iterations of the procedure in this paragraph and the procedure in the previous paragraph is bounded by $\frac{1}{2}n(n-1)$. Since the cost of a single cyclic sliding is $O(ln \log n)$, it follows that the complexity of computing $\rho_i\alpha_i$ is $O(l^2 n^3 \log n)$. This is done for $i = 1, \dots, n-1$, and the set of minimal simple elements for x is just the set of minimal elements in $\{\rho_1\alpha_1, \dots, \rho_{n-1}\alpha_{n-1}\}$, hence we can compute this set in $O(l^2 n^4 \log n)$.

The next steps, computing $\tau(x)$ and comparing it with x or $\mathbf{c}^{\frac{k}{2}}(x)$ (which has been computed in a previous step, when k was obtained), are negligible compared with the previous ones.

Finally, it remains to conjugate the generators A and B by c^{-1} to obtain $Z(y)$. Since y is σ_1-non-intrusive, $\ell(c) \leq \frac{2}{5}\ell(x) = O(l)$, and on the other hand $\ell(A)$ is at most k in all possible cases, where $k \leq 2l$. Then the cost of conjugating A and B by c^{-1}, is $O(l^2 n \log n)$ (see [19]).

Therefore, the total time complexity of the whole algorithm, when applied to the σ_1-non-intrusive braid y, is $O(l^2 n^4 \log n)$. □

Corollary 6.2. *There exists an algorithm to compute a generating set for the centralizer of a braid $y \in B_n$, whose generic-case complexity is $O(l^2 n^4 \log n)$, where $l = \ell(y)$.*

Notice that, when the braid group B_n is fixed, the generic-case complexity of Algorithm 1 is quadratic on the canonical length of the input braid. But one can implement the algorithm letting n vary, and the generic-case complexity still remains polynomial.

Bibliography

[1] E. Artin, Theorie der Zöpfe, *Abh. Math. Semin. Univ. Hambg.* **4** (1925), no. 1, 47–72.

[2] E. Artin, Theory of braids, *Ann. of Math. (2)* **48** (1947), 101–126.

[3] J. Birman, K. H. Ko and S. J. Lee, A new approach to the word and conjugacy problems in the braid groups, *Adv. Math.* **139** (1998), no. 2, 322–353.

[4] J. S. Birman, V. Gebhardt and J. González-Meneses, Conjugacy in Garside groups. I. Cyclings, powers and rigidity, *Groups Geom. Dyn.* **1** (2007), no. 3, 221–279.

[5] J. S. Birman, V. Gebhardt and J. González-Meneses, Conjugacy in Garside groups. II. Structure of the ultra summit set, *Groups Geom. Dyn.* **2** (2008), no. 1, 13–61.

[6] N. Bourbaki, *Éléments de mathématique. Fasc. XXXVII. Groupes et algèbres de Lie. Chapitre II: Algèbres de Lie libres. Chapitre III: Groupes de Lie*, Act. Sci. Indust. 1349, Hermann, Paris, 1972.

[7] E. Brieskorn and K. Saito, Artin-Gruppen und Coxeter-Gruppen, *Invent. Math.* **17** (1972), 245–271.

[8] S. Caruso, On the genericity of pseudo-Anosov braids I: Rigid braids, *Groups Geom. Dyn.* **11** (2017), no. 2, 533–547.

[9] S. Caruso and B. Wiest, On the genericity of pseudo-Anosov braids II: Conjugations to rigid braids, *Groups Geom. Dyn.* **11** (2017), no. 2, 549–565.

[10] R. Charney and J. Meier, The language of geodesics for Garside groups, *Math. Z.* **248** (2004), no. 3, 495–509.

[11] P. Dehornoy, Groupes de Garside, *Ann. Sci. Éc. Norm. Supér. (4)* **35** (2002), no. 2, 267–306.

[12] P. Dehornoy and L. Paris, Gaussian groups and Garside groups, two generalisations of Artin groups, *Proc. Lond. Math. Soc. (3)* **79** (1999), no. 3, 569–604.

[13] E. A. El-Rifai and H. R. Morton, Algorithms for positive braids, *Quart. J. Math. Oxford Ser. (2)* **45** (1994), no. 180, 479–497.

[14] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson and W. P. Thurston, *Word Processing in Groups*, Jones and Bartlett, Boston, 1992.

[15] N. Franco and J. González-Meneses, Computation of centralizers in braid groups and Garside groups, *Rev. Mat. Iberoam.* **19** (2003), no. 2, 367–384.

[16] N. Franco and J. González-Meneses, Conjugacy problem for braid groups and Garside groups, *J. Algebra* **266** (2003), no. 1, 112–132.

[17] F. A. Garside, The braid group and other groups, *Quart. J. Math. Oxford Ser. (2)* **20** (1969), 235–254.

[18] V. Gebhardt, A new approach to the conjugacy problem in Garside groups, *J. Algebra* **292** (2005), no. 1, 282–302.

[19] V. Gebhardt and J. González-Meneses, Solving the conjugacy problem in Garside groups by cyclic sliding, *J. Symbolic Comput.* **45** (2010), no. 6, 629–656.

[20] V. Gebhardt and J. González-Meneses, The cyclic sliding operation in Garside groups, *Math. Z.* **265** (2010), no. 1, 85–114.

[21] J. González-Meneses and B. Wiest, On the structure of the centralizer of a braid, *Ann. Sci. Éc. Norm. Supér. (4)* **37** (2004), no. 5, 729–757.

[22] N. V. Ivanov, *Subgroups of Teichmüller Modular Groups*, Transl. Math. Monogr. 115, American Mathematical Society, Providence, 1992.

[23] E.-K. Lee and S.-J. Lee, Periodic elements in Garside groups, *J. Pure Appl. Algebra* **215** (2011), no. 10, 2295–2314.

[24] R. C. Lyndon and P. E. Schupp, *Combinatorial Group Theory*, Classics Math., Springer, Berlin, 2001.

[25] G. S. Makanin, The normalizers of a braid group, *Mat. Sb. (N.S.)* **86(128)** (1971), 171–179.

[26] J. D. McCarthy, Normalizers and cetralizers of pseudo-Anosov mapping classes, unpublished manuscript (1992), http://users.math.msu.edu/users/mccarthy/publications/normcent.pdf.

Received February 14, 2017.

Author information

Juan González-Meneses, Departamento de Álgebra, Instituto de Matemáticas (IMUS),
Universidad de Sevilla, Av. Reina Mercedes s/n, 41012 Sevilla, Spain.
E-mail: meneses@us.es

Dolores Valladares, Departamento de Álgebra, Instituto de Matemáticas (IMUS),
Universidad de Sevilla, Av. Reina Mercedes s/n, 41012 Sevilla, Spain.
E-mail: valladaresg@us.es