# On the complexity of the Whitehead minimization problem*

Abdó Roig (abdo.roig@upc.edu)
Universitat Politècnica de Catalunya

Enric Ventura (enric.ventura@upc.edu)†
Universitat Politècnica de Catalunya

Pascal Weil (pascal.weil@labri.fr)‡
LaBRI (CNRS and Université Bordeaux-1)

## Abstract

The Whitehead minimization problem consists in finding a minimum size element in the automorphic orbit of a word, a cyclic word or a finitely generated subgroup in a finite rank free group. We give the first fully polynomial algorithm to solve this problem, that is, an algorithm that is polynomial both in the length of the input word and in the rank of the free group. Earlier algorithms had an exponential dependency in the rank of the free group. It follows that the primitivity problem – to decide whether a word is an element of some basis of the free group – and the free factor problem can also be solved in polynomial time.

*Part of this work was produced while the second author was a Visiting Professor at LaBRI, Université Bordeaux-1. The second author gratefully acknowledges partial support by DGI (Spain) through grant BFM2003-06613. The third author gratefully acknowledges support from the ESF programme AutoMathA.

†EPSEM, Universitat Politècnica de Catalunya – Av. Bases de Manresa 61 – 73 08242 Manresa, Barcelona – Spain

‡LaBRI – 351 cours de la Libération – 33405 Talence Cedex – France

Let $F$ be a finite rank free group and let $A$ be a fixed (finite) basis of $F$. The elements of $F$ can be represented by reduced words over the symmetrized alphabet $A \cup \bar{A}$, and the finitely generated subgroups of $F$ by certain finite graphs whose edges are labeled by letters from $A$ (obtained by the technique of so-called Stallings foldings [23], see [12] and Section 1). We measure the size of an element of $F$ by the length of the reduced word representing it, and the size of a finitely generated subgroup of $F$ by the number of vertices of the graph representing it. The *Whitehead minimization problem* consists in finding a minimum size element in the automorphic orbit of a given word or a given finitely generated subgroup. An important variant considers rather as input conjugacy classes of words (the so-called cyclic words) or subgroups.

As we will see (Section 1.3), the minimization problem for words, cyclic words and subgroups can be reduced to the problem for conjugacy classes of finitely generated subgroups, so we will often discuss only the latter problem.

The Whitehead minimization problem is a fragment of the larger *equivalence problem*, where one must decide, given two subgroups of $F$, whether they sit in the same automorphic orbit. More precisely, in view of a result of Gersten [5] (which generalizes to subgroups a classical result of Whitehead that applies to words [26], see [17, Sec. I.4]), the first part of the (only known) algorithm to solve the equivalence problem consists in finding minimum size elements in the automorphic orbits of the given subgroups, that is, in solving the Whitehead minimization problem for these two subgroups.

Moreover, any solution of the Whitehead minimization problem for words implies a solution of the *primitivity problem*: to decide whether a given word is an element of some basis of $F$. Indeed, a word is primitive if and only if the minimum length of an element in its automorphic orbit is 1. Similarly, a subgroup is a free factor of $F$ if and only if the minimum size of an element in its automorphic orbit is 1, so any solution of the Whitehead minimization problem for subgroups implies a solution of the *free factor problem*: to decide whether a given subgroup is a free factor of $F$.

As hinted above, a classical algorithm is known to solve the Whitehead minimization problems. The algorithm for the word case is based on Whitehead's theorem [26] (see [17, Prop. I.4.17]), and the algorithm for the subgroup case relies on a generalization of this theorem due to Gersten [5, Corol. 2] (Theorem 1.9 below). Both algorithms are polynomial in $n$, the size of the input, and exponential in $r$, the rank of the free group $F$, see Section 1.4 below and [18] for a more detailed analysis.

In a recent paper, Haralick, Miasnikov and Myasnikov [8] (see also Miasnikov and Myasnikov [18]) present a number of heuristics and experiments on different implementation strategies for the algorithm regarding words, that tend to show that the actual dependency of the problem in the rank of $F$ is much lower than exponential, at least in the generic case. Haralick and Miasnikov [7] pursue in that direction by giving a polynomial-time stochastic algorithm for the same problem. Another recent paper, by Silva and Weil [22] gives an exact algorithm for solving the free factor problem on input $H$, which is polynomial in the size of $H$ and exponential in $\mathsf{rank}(F) - \mathsf{rank}(H)$.

The main result of this paper confirms the intuition of [8, 18, 7] and improves the complexity result in [22]. Indeed, we give a fully polynomial solution to the Whitehead minimization problem, that is, an exact algorithm that is polynomial in both the size of the input and the rank of $F$. Interestingly, this result is obtained with only a small amount of new mathematical results. Our algorithm is in fact a minor modification of the classical Whitehead method (the algorithm mentioned above), and the study of its complexity relies on the conjunction of three ingredients:

(1) a representation of a Whitehead automorphism as a pointed cut of the set $A \cup \bar{A}$, that is, a partition of $A \cup \bar{A}$ into two disjoint subsets $Y$ and $Z$ and the choice of a letter $a \in Y$ such that $\bar{a} \in Z$,

(2) an exact computation of the effect of such an automorphism on the length of a cyclic word $u$ (resp. the size of a conjugacy class $H$ of finitely generated subgroups) in terms of the capacity of the associated cut on the Whitehead graph associated with $u$ (resp. a generalization of the Whitehead graph which we call the Whitehead hypergraph of $H$),

(3) and an algorithmic complexity result on finding a minimum capacity cut in a graph (resp. a hypergraph).

The first ingredient is classical in combinatorial group theory (see for instance [17, Prop. I.4.16]). The second ingredient can be described as a rewording of a formula of Gersten [5, Corol. 2] proved in [11, Prop. 10.3]. And the last one can be reduced to standard problems in combinatorial optimization, for which there exist several polynomial-time algorithms in the literature.

In Section 1, we fix the notation to handle words, cyclic words and subgroups of $F$, and to describe Whitehead automorphisms. We also discuss the foundational theorem of Whitehead and its generalization by Gersten, that gives rise to the known algorithm solving the Whitehead minimization problem. As indicated above this algorithm is polynomial in the size of the input and exponential in the rank of the ambient free group.

In Section 2, we introduce the Whitehead hypergraph of a cyclically reduced subgroup, and we present a rewording of a formula of Gersten, describing the effect of a Whitehead automorphism on the size of a conjugacy class of finitely generated subgroups (and generalizing a classical result of Whitehead on cyclic words, see [17, Prop. I.4.16]). This technical analysis helps show how the exponential dependency in the usual Whitehead algorithm can be removed, provided a certain graph-theoretic problem, namely the min-cut problem for undirected hypergraphs, can be solved in polynomial time.

We discuss existing polynomial-time algorithms to solve the min-cut problem in Section 3, thus completing our proof. Finally, we consider some consequences of our main result.

We conclude this introduction with a remark on complexity computation. Since the rank $r$ of $F$ is part of the input, we consider complexity functions under the *bit cost assumption*: $r$ is the cardinality of $A$ and each letter is identified by a bit string of length $\log r$, so that the elementary operations on $A$ (reading or writing a letter, comparing two letters, etc) require $O(\log r)$ units of time.

# 1 The Whitehead minimization problem

In this paper, $F$ denotes a finitely generated free group and $A$ denotes a fixed basis of $F$. We let $r = \mathsf{rank}(F) = \mathsf{card}(A)$.

## 1.1 Words, graphs and subgroups

Elements of $F$ can be represented as usual by *reduced words* on the symmetrized alphabet $\tilde{A} = A \cup \bar{A}$, and we write $u \in F(A)$ to indicate that the element $u$ of $F$ is given by a reduced word on alphabet $\tilde{A}$.

Recall that the mapping $a \mapsto \bar{a}$ is extended to the set of all words over the alphabet $\tilde{A}$ by letting $\bar{\bar{a}} = a$ for each $a \in A$, and $\overline{ua} = \bar{a}\,\bar{u}$ for each word $u$ and each letter $a \in \tilde{A}$.

### 1.1.1 Graphs

To represent finitely generated subgroups of $F$, we use finite $A$-graphs. An *$A$-graph* is a directed graph, whose edges are labeled by letters in $\tilde{A}$. More precisely, it is a pair $\Gamma = (V, E)$ with $E \subseteq V \times \tilde{A} \times V$. The elements of $V$ are called *vertices*, the elements of $E$ are called *edges*, and we say that there is an edge from $x$ to $y$ labeled $a$ if $(x, a, y) \in E$. We denote respectively by $\alpha$, $\lambda$ and $\omega$ the first, second and third coordinate projections from $E$ to $V$, $\tilde{A}$ and $V$. The map $\lambda$ is called the *labeling function*.

We measure the *size* of an $A$-graph $\Gamma$ by the number of its vertices, and we write $|\Gamma| = \mathsf{card}(V)$.

A *dual* $A$-graph is one in which for each $a \in A$, there is an edge from vertex $x$ to vertex $y$ labeled $a$ if and only if there is one from $y$ to $x$ labeled $\bar{a}$. That is, $(x, a, y) \in E$ if and only if $(y, \bar{a}, x) \in E$.

A dual $A$-graph is *reduced* if whenever there are $a$-labeled edges from $x$ to $y$ and from $x'$ to $y$, then $x = x'$ ($a \in \tilde{A}$, $x, x', y \in V$). It is *cyclically reduced* if it is reduced and, for each vertex $x$, there exist at least 2 edges into $x$.

If $\Gamma$ is an $A$-graph and $x \in V$, we define the *link* of $x$ to be the set of edges into $x$, and the *hyperlink* of $x$ to be the set of labels of these edges,

$$\mathsf{link}_\Gamma(x) = \{e \in E \mid \omega(e) = x\}, \quad \mathsf{hl}_\Gamma(x) = \{\lambda(e) \mid e \in \mathsf{link}_\Gamma(x)\}.$$

(The reason for the terminology *hyperlink* will become clear in Section 2.1.)

Let $\Gamma$ be a dual $A$-graph. By an immediate rewording of the definitions, we see that $\Gamma$ is reduced if and only if $\lambda$ establishes a bijection from $\mathsf{link}_\Gamma(x)$ to $\mathsf{hl}_\Gamma(x)$; and $\Gamma$ is cyclically reduced if in addition, $\mathsf{card}(\mathsf{hl}_\Gamma(x)) \geq 2$ for each $x \in V$. A vertex $x$ such that $\mathsf{card}(\mathsf{hl}_\Gamma(x)) \leq 1$, is called an *endpoint* of $\Gamma$.

We say that an $A$-graph is *connected* if the underlying undirected graph is connected. In the case of a dual $A$-graph, this is the case if and only if, for any vertices $x, y$, there exists a path from $x$ to $y$.

The following particular case will be important for our purpose. A *circular* $A$-graph is a connected dual $A$-graph in which the link of each vertex has exactly 2 elements. A cyclically reduced circular $A$-graph is called a *cyclic word*.

**Example 1.1** When representing $A$-graphs, we draw only the positively labeled edges, that is, those labeled by letters of $A$. Figure 1 shows such $A$-graphs. In that figure, $\Gamma_1$, $\Gamma_2$ and $\Gamma_3$ are reduced, $\Gamma_0$ is not. Vertices 2 and

3 are endpoints in $\Gamma_2$ and $\Gamma_3$ respectively, while $\Gamma_1$ has no endpoint. In $\Gamma_2$, the hyperlink of vertex 1 is $\{a, \bar{a}, \bar{b}\}$ and the hyperlink of vertex 2 is $\{\bar{a}\}$. □

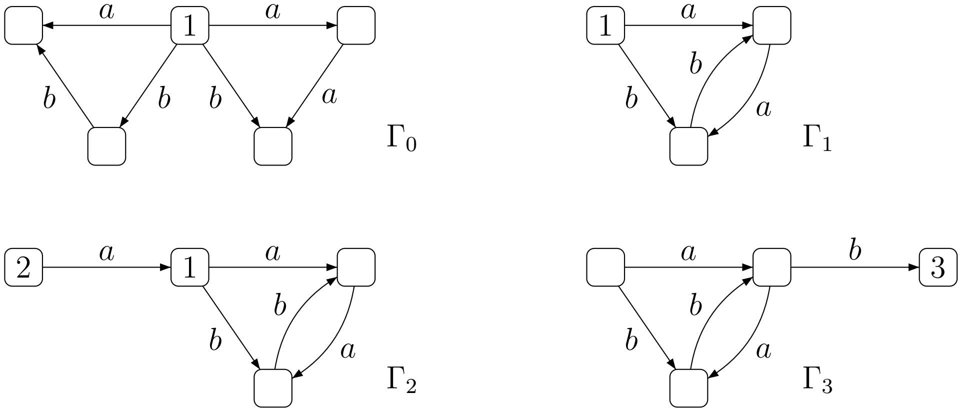


Figure 1: Four $A$-graphs

### 1.1.2 Reduction

Let $\Gamma = (V, E)$ be a dual $A$-graph, and let $x, y$ be distinct vertices. The $A$-graph obtained from $\Gamma$ *by identifying vertex $y$ to vertex $x$* is constructed as follows: its vertex set is $V \setminus \{y\}$; and its edge set is obtained from $E$ by replacing everywhere $y$ by $x$. The resulting $A$-graph is again dual. Note that identifying $y$ to $x$ or $x$ to $y$ yields isomorphic $A$-graphs.

If $\Gamma$ is not reduced, there exist pairs of distinct edges $(x, a, z)$ and $(y, a, z)$ (that is, edges with the same label that point to the same vertex). An *elementary reduction* of $\Gamma$ is the $A$-graph that results from identifying vertex $y$ to vertex $x$ in such a situation (which automatically implies the identification of the two distinct edges).

The *reduction of a dual $A$-graph* $\Gamma$ consists in repeatedly performing elementary reductions, as long as some are possible. This is a terminating process since we consider only finite graphs, and each elementary reduction properly decreases the size of the graph. The resulting graph, denoted by

$\mathsf{red}(\Gamma)$, is reduced, and it is well known that it does not depend on the sequence of elementary reductions that were performed (that is: the process of elementary reductions is confluent, see [23]).

If $\Gamma$ is reduced, an *elementary trimming* consists in removing an endpoint and the edges adjacent to it. The *cyclic reduction* of $\Gamma$ consists in repeatedly applying elementary trimmings, as long as it is possible. The resulting graph is cyclically reduced, it does not depend on the sequence of elementary trimmings performed, and it is called the *cyclic core* of $\Gamma$, written $\mathsf{cc}(\Gamma)$. Clearly, $\mathsf{cc}(\Gamma)$ is a subgraph of $\Gamma$.

If $x$ is a vertex of $\Gamma$, there exists a unique shortest path from $x$ to a vertex of $\mathsf{cc}(\Gamma)$. We call this path the *branch* of $\Gamma$ at $x$, and we denote by $b(x)$ the label of that path, and by $\beta(x)$ its extremity in $\mathsf{cc}(\Gamma)$. If $x$ is already in $\mathsf{cc}(\Gamma)$, then $\beta(x) = x$ and $b(x)$ is the empty word, $b(x) = 1$.

**Example 1.2** With reference to the $A$-graphs in Example 1.1 and Figure 1, we have

$$\mathsf{red}(\Gamma_0) = \Gamma_1 = \mathsf{cc}(\Gamma_2) = \mathsf{cc}(\Gamma_3).$$

In graph $\Gamma_2$, we have $\beta(2) = 1$ and $b(2) = a$. □

Let $u = a_1 \cdots a_n \in F(A)$. The word $u$ is said to be *cyclically reduced* if $a_n \neq \bar{a}_1$, if and only if the word $u^2$ is reduced. It is a standard observation that, in general, there exists a unique cyclically reduced word $w$ such that $u = vw\bar{v}$ for some $v \in F(A)$. The word $w$ is called the *cyclic core* of $u$, written $\mathsf{cc}(u)$. Let $\Gamma(u)$ be the circular graph with vertex set $\mathbb{Z}/n\mathbb{Z} = \{1, \ldots, n\}$ and with edges $(i, a_i, i+1)$ and $(i+1, \bar{a}_i, i)$ for each $1 \leq i \leq n$. Reducing $\Gamma(u)$ yields the graph shown in Figure 2, and $\mathsf{cc}(\Gamma(u)) = \Gamma(\mathsf{cc}(u))$.

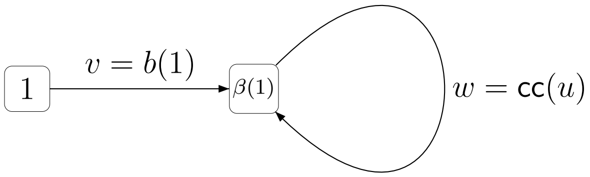


Figure 2: The graph $\mathsf{red}(\Gamma(u))$, where $u = vw\bar{v}$, $v = b(1)$ and $w = \mathsf{cc}(u)$

**Fact 1.3** It is verified in [22, Sec. 3.2] that reducing and trimming an $n$-vertex dual $A$-graph takes time $O(n^2 \log(nr))$. See Touikan [25] for a faster algorithm.

If $u \in F(A)$ has length $n$, its cyclic core $\mathsf{cc}(u)$ is computed in time $O(n \log r)$. In particular, reducing and trimming an $n$-vertex circular graph takes time $O(n \log r)$. □

### 1.1.3 Graphs and subgroups

It is classical to represent finitely generated subgroups of $F$ by pointed $A$-graphs. Let $H$ be a finitely generated subgroup of $F$ (we write $H \leq_{\mathsf{fg}} F$) and let $h_1, \ldots, h_m \in F(A)$ be a set of generators of $H$. Let $\Gamma_0(h_1, \ldots, h_m)$ be the dual $A$-graph which consists of a bouquet of $m$ loops, labeled by the $h_i$, around a distinguished vertex, usually denoted by 1. We denote by $\Gamma(H)$ the reduced $A$-graph $\mathsf{red}(\Gamma_0(h_1, \ldots, h_m))$. Observe that this construction coincides with the application of the so-called Stallings foldings [23, 12]: it is well-known that the pair $(\Gamma(H), 1)$ depends on $H$ only, not on the choice of the generating set $\{h_1, \ldots, h_m\}$, and we call it *the* (*graphical*) *representation of* $H$. $\Gamma(H)$ is a connected reduced $A$-graph, in which no vertex different from 1 is an endpoint.

Conversely, if $\Gamma$ is a connected reduced $A$-graph, 1 is a vertex of $\Gamma$ and $\Gamma$ has no endpoint except maybe for 1, there exists a unique subgroup $H \leq_{\mathsf{fg}} F$ such that $(\Gamma, 1)$ is the representation of $H$. In that situation, let $T$ be a spanning tree of the $A$-graph $\Gamma$, and for each vertex $x$, let $u_x$ be the only reduced word labeling a path from 1 to $x$ inside the tree $T$. For each positively labeled edge $e = (x, a, y)$ (that is, $a \in A$), let $h_e = u_x a \bar{u}_y$. Then a basis of $H$ consists of the elements $h_e$, where $e$ runs over the positively labeled edges not in $T$ [23, 12].

**Example 1.4** Let $H_1 = \langle a^2b^{-1}, b^2a^{-1} \rangle$. With reference to the graphs in Figure 1, we see that $\Gamma_0 = \Gamma_0(a^2b^{-1}, b^2a^{-1})$ (with distinguished vertex 1) and $(\Gamma_1, 1)$ is the graphical representation of $H_1$. Let $H_2 = \langle a^3b^{-1}a^{-1}, ab^2a^{-2} \rangle$ and $H_3 = \langle b^{-1}ab^{-1}ab, b^{-1}a^{-1}b^3 \rangle$. The graphical representation of $H_2$ is $(\Gamma_2, 2)$ and the graphical representation of $H_3$ is $(\Gamma_3, 3)$. □

**Fact 1.5** In view of Fact 1.3, and if $\sum_{i=1}^m |h_i| = n$, computing $\Gamma(H)$ takes time $O(n^2 \log(nr))$. Given an $n$-vertex reduced $A$-graph $\Gamma$ and a vertex 1, and letting $H$ be the subgroup represented by $(\Gamma, 1)$, one can compute in time $O(n^2 \log(nr))$ a basis of $H$, whose elements are words of length at most $2n$ [22, Sec. 3.3].

If $H = \langle u \rangle$ is a cyclic subgroup of $F$, generated by a word of length $n \neq 0$, then $\{u\}$ is a basis of $H$ and $\Gamma(\langle u \rangle) = \mathsf{red}(\Gamma(u))$ is computed in time $O(n \log r)$ by Fact 1.3. □

Let $H \leq_{\mathsf{fg}} F$, with representation $(\Gamma(H), 1)$. Let us say that $H$ *is cyclically reduced* if 1 is not an endpoint, that is, if $\Gamma(H)$ is cyclically reduced. In the general case, let $K$ be the subgroup represented by $(\mathsf{cc}(\Gamma(H)), \beta(1))$. It is well-known that $K = \chi_{b(1)}(H)$, where $\chi_u$ is the conjugation $x \mapsto u^{-1}xu$. It follows that, for each subgroup $H' \leq_{\mathsf{fg}} F$, $H'$ is a conjugate of $H$ if and only if $\mathsf{cc}(\Gamma(H')) = \mathsf{cc}(\Gamma(H))$. As a consequence, we say that $\mathsf{cc}(\Gamma(H))$ is *the (graphical) representation of the conjugacy class* $[H]$.

We note that if $H = \langle u \rangle$ is a cyclic subgroup, then the subgroup $H$ is cyclically reduced if and only if the word $u$ is cyclically reduced.

In the sequel, the size of $\Gamma(H)$ and of $\mathsf{cc}(\Gamma(H))$ will be taken to be measures of the *size* of $H$ and $[H]$, and we will write $|H| = |\Gamma(H)|$ and $|[H]| = |\mathsf{cc}(\Gamma(H))|$. In particular, $|[H]|$ is the minimum size of subgroups in the conjugacy class $[H]$, and it is equal to the size of any cyclically reduced conjugate of $H$.

**Example 1.6** Let $H_1$, $H_2$ and $H_3$ be the subgroups discussed in Example 1.4. In view of Figure 1, we see that $H_1$ is cyclically reduced, and that $H_2 = \chi_{a^{-1}}(H_1)$ and $H_3 = \chi_{ab}(H_1)$. □

## 1.2 Action of an automorphism

Let $\varphi \in \mathsf{Aut}(F)$ and let $\Gamma = (V, E)$ be a reduced $A$-graph with a designated vertex 1. Let $\varphi_\bullet(\Gamma)$ be the $A$-graph obtained after the following steps:

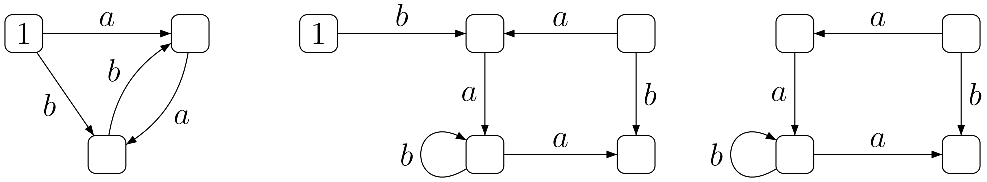
(a) replace each $a$-labeled edge by a path labeled by the word $\varphi(a)$ with the same endpoints: if $\varphi(a) = a_1 \cdots a_m$ $(a_i \in \tilde{A})$ and $(x_0, a, x_m) \in E$, remove the edges $(x_0, a, x_m)$ and $(x_m, \bar{a}, x_0)$, add new vertices $x_1, \ldots, x_{m-1}$ and add edges $(x_{i-1}, a_i, x_i)$ and $(x_i, \bar{a}_i, x_{i-1})$ for each $1 \leq i \leq m$;

(b) reduce the resulting $A$-graph;

(c) repeatedly trim all the endpoints different from 1.

If $\Gamma$ is cyclically reduced, we also denote by $\varphi(\Gamma)$ the cyclically reduced graph $\mathsf{cc}(\varphi_\bullet(\Gamma))$.

**Example 1.7** Let $\Gamma_1$ be as in Example 1.1 and let $\varphi \in \mathsf{Aut}(F)$ be given by $\varphi(a) = ba^{-1}$ and $\varphi(b) = bab^{-1}$. The graphs $\Gamma_1$, $\varphi_\bullet(\Gamma_1)$ and $\varphi(\Gamma_1)$ are shown in Figure 3. □



$\Gamma_1$ $\varphi_\bullet(\Gamma_1)$ $\varphi(\Gamma_1)$

Figure 3: Action of an automorphism on an $A$-graph

**Fact 1.8** Let $\ell$ be the maximum length of the image of a letter by an automorphism $\varphi$ (so $\varphi$ is given by a $r$-tuple of words of length at most $\ell$) and let $\Gamma$ be an $n$-vertex reduced $A$-graph. In view of Fact 1.3, the complexity of computing $\varphi_\bullet(\Gamma)$ is $O(n^2\ell^2r^2\log(n\ell r))$.

If $\Gamma$ is a cyclic word, $\Gamma = \Gamma(u)$ with $u \in F(A)$ cyclically reduced and $|u| = n$, then $\varphi_\bullet(\Gamma) = \mathsf{red}(\Gamma(\varphi(u)))$, which is computed in time $O(n\ell\log r)$. □

It is easy to verify that if $\varphi \in \mathsf{Aut}(F)$ and $H \leq_{\mathsf{fg}} F$, then $\varphi(H)$ is represented by $(\varphi_\bullet(\Gamma(H)), 1)$. If in addition $H$ is cyclically reduced, the conjugacy class $\varphi([H])$ is represented by $\varphi(\Gamma(H))$.

The *Whitehead minimization problem* (WMP) *for finitely generated subgroups* (resp. *for conjugacy classes of finitely generated subgroups*) consists in finding a minimum size element $X'$ of the automorphic orbit of a given finitely generated subgroup (resp. conjugacy class of finitely generated subgroups) $X$, and an automorphism $\varphi$ such that $X' = \varphi(X)$.

If the input of the WMP is a conjugacy class of cyclic subgroups, that is, a cyclic word, we talk of the *Whitehead minimization problem for cyclic words.* The *Whitehead minimization problem for words* consists in finding a minimum length element $u'$ of the automorphic orbit of a given reduced word $u$, and an automorphism $\varphi$ such that $u' = \varphi(u)$.

We will see in Section 1.3 that all these problems reduce to the problem for conjugacy classes of finitely generated subgroups.

## 1.3 Gersten's theorem and the Whitehead method

It is well-known that the group $\mathsf{Aut}(F)$ of automorphisms of $F$ is finitely generated. One particular finite set of generators of $\mathsf{Aut}(F)$ is the set of so-called Whitehead automorphisms (relative to the choice of the basis $A$), whose precise definition will be given in Section 1.4 below.

The first key element of the algorithm presented here, is the following statement, due to Gersten [5, Corol. 2].

**Theorem 1.9** *Let $H$ be a cyclically reduced subgroup of $F(A)$. If there exists an automorphism $\varphi$ of $F$ such that $|\varphi([H])| < |[H]|$, then there exists such an automorphism among the Whitehead automorphisms.*

**Remark 1.10** This theorem is a generalization of a fundamental result of Whitehead ([26], see [17, Sec. I.4]), which concerns the cyclic word case — the case where $H = \langle u \rangle$ for some cyclically reduced word $u$. □

This implies the following algorithm — the so-called *Whitehead method* — to solve the WMP for a conjugacy class of finitely generated subgroups $[H]$. The input of the algorithm is a cyclically reduced subgroup $H$, or rather the cyclically reduced $A$-graph $\Gamma(H)$. The output of the algorithm is a cyclically reduced subgroup $H'$ and a tuple $\vec{\varphi} = (\varphi_m, \ldots, \varphi_1)$ of Whitehead automorphisms, such that $[H'] = \varphi_m \circ \cdots \circ \varphi_1([H])$ and $[H']$ has minimum size in the automorphic orbit of $[H]$.

First let $\vec{\varphi} = (\mathsf{id})$ and $\Gamma = \Gamma(H)$. Then repeatedly apply the following steps: try every Whitehead automorphism $\psi$ until $|\psi(\Gamma)| < |\Gamma|$; if such a $\psi$ exists, replace $\Gamma$ by $\psi(\Gamma)$ and $\vec{\varphi}$ by $(\psi, \vec{\varphi})$; otherwise, stop and output $\Gamma$ and $\vec{\varphi}$. At each step, the size of $\Gamma$ decreases by at least one unit, so this procedure terminates after at most $|\Gamma|$ iterations. Finally, in order to output a basis of (a possible value of) $H'$, choose arbitrarily a vertex 1 in $\Gamma$ and use the procedure discussed in Section 1.1.3.

Let us give an estimate of the complexity of this algorithm.

**Fact 1.11** We first note that the cost of the construction of $\Gamma(H)$, if the input is a set of generators of $H$ of total length $n$, is $O(n^2 \log(nr))$ (Fact 1.5). Moreover, $\Gamma(H)$ has at most $n$ vertices.

As we will see, a Whitehead automorphism maps every letter to a word of length at most 3, so finding the image of a cyclically reduced graph of size $n$ under a Whitehead automorphism also takes time $O(n^2 r^2 \log(nr))$ (Fact 1.8).

Thus, if $f(r)$ is the cardinality of the set of Whitehead automorphisms (of a rank $r$ free group), each iterating step of the algorithm may require $f(r)$ steps, each of which consists in computing the image of a cyclically reduced graph of length at most $n$ under a Whitehead automorphism, and hence has complexity $O(n^2r^2\log(nr))$. There are at most $n$ iterating steps, so the iterating part of the algorithm has time complexity $O(n^3r^2f(r)\log(nr))$.

Finally, retrieving a basis of $H'$ from the ultimate value of $\Gamma$ takes time $O(n^2\log(nr))$ (Fact 1.5). That is negligible in front of $n^3r^2f(r)\log(nr)$, so the total complexity of the algorithm is $O(n^3r^2f(r)\log(nr))$, which is polynomial in $n$ and exponential in $r$ as we shall see in Section 1.4.

In the cyclic word case, that is, the case where $H = \langle u\rangle$ with $u \in F(A)$ cyclically reduced and $|u| = n$, the complexity is $O(n^2f(r)\log r)$. □

**Remark 1.12** Let $(H', \vec{\varphi})$ be the output of the algorithm on input $H$. The complexity discussion above shows that $\vec{\varphi} = (\varphi_m, \ldots, \varphi_1)$ with $m < n$. As we will see in Section 1.4, the length of the image of a letter in a Whitehead automorphism is at most 3, so the length of $\varphi_m \circ \cdots \circ \varphi_1(a)$ $(a \in A)$ may be exponential in $m$, and the computation of $\varphi_m \circ \cdots \circ \varphi_1$ may take time exponential in $m$. This possible exponential explosion is the reason why we choose to output a tuple of Whitehead automorphisms rather than their composition.

An easy example for this exponential explosion is provided by the (Whitehead) automorphisms $\alpha\colon a \mapsto ab;\ b \mapsto b$ and $\beta\colon a \mapsto a;\ b \mapsto ba$. Then $\beta\circ\alpha\colon a \mapsto aba;\ b \mapsto ba$, and the length of $(\beta\circ\alpha)^n(a)$ (resp. $(\beta\circ\alpha)^n(b)$) is the sum of the entries of the first (resp. second) column of $\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}^n$. It is well-known that the asymptotic behavior of these numbers as $n$ tends to infinity, is $O(\rho^n)$ where $\rho$ is the dominant eigenvalue of that matrix, namely, $\rho = (3+\sqrt{5})/2$. □

The above algorithm can be modified to solve the WMP for a finitely generated subgroup $H$ in $F(A)$. A minimum size element $K$ of the automorphic orbit of $H$ is in particular a minimum size element of its own conjugacy class, and hence $K$ is cyclically reduced. Moreover, the class $[K]$ must be a minimum size element in the automorphic orbit of $[H]$. Thus a minimum size element of the orbit of $H$ is obtained by computing a minimum size element of the orbit of $[H]$, say $[K]$, and choosing arbitrarily a cyclically reduced subgroup in $[K]$.

More precisely, the algorithm is as follows. First let $(\Gamma(H), 1)$ be the graphical representation of $H$, let $\Gamma = \mathsf{cc}(\Gamma(H))$ and $\vec{\varphi} = (\chi_{b(1)})$. We note that $\Gamma = {\chi_{b(1)}}_\bullet(\Gamma(H))$. Next, rename as 1 the vertex $\beta(1)$ of $\Gamma$. Then repeatedly apply the following steps: try every Whitehead automorphism $\psi$ until $|\psi(\Gamma)| < |\Gamma|$; if such a $\psi$ exists, consider the pointed $A$-graph $(\psi_\bullet(\Gamma), 1)$, replace $\vec{\varphi}$ by $(\chi_{b(1)} \circ \psi, \vec{\varphi})$, rename $\beta(1)$ as 1, and replace $\Gamma$ by $\psi(\Gamma) = \mathsf{cc}(\varphi_\bullet(\Gamma))$; otherwise, stop and output $\Gamma$ and $\vec{\varphi}$. Finally, construct a basis of the subgroup $H'$ represented by $(\Gamma, 1)$.

Finally, this last algorithm can be used to solve the WMP for words: a minimum length element in the automorphic orbit of a word $u \in F(A)$ is necessarily a cyclically reduced word $u'$ such that $\langle u' \rangle$ is a solution of the WMP on input $\langle u \rangle$. Therefore, it suffices to apply the above algorithm on input $\langle u \rangle$, letting $u' = u$ at the beginning of the algorithm, and updating $u'$ to $\chi_{b(1)} \circ \psi(u')$ at each iterating step. We note that the length of $u'$ never exceeds $|\Gamma|$, and hence never exceeds $|u| = n$.

**Fact 1.13** The extra work required by this modified algorithm (see Fact 1.11), namely to compute $\mathsf{cc}(\Gamma(H))$ and to compose at most $n$ Whitehead automorphisms with conjugations by words of length at most $n$, takes time $O(n^2 r \log r)$, which is negligible in front of $n^3 r^2 f(r) \log(nr)$. So the time complexity of this algorithm is again $O(n^3 r^2 f(r) \log(nr))$.

In the cyclic subgroup case, as well as in the word case, the complexity is $O(n^2 f(r) \log r)$. □

## 1.4 Whitehead automorphisms

We now review the definition of the *Whitehead automorphisms* of $F$, relative to the choice of the basis $A$, see for instance [17, Sec. I.4].

There are two kinds of Whitehead automorphisms. The *first kind* consists of the automorphisms that permute the set $\tilde{A}$. We observe that these are exactly the length-preserving automorphisms of $F(A)$, that is, the automorphisms $\varphi$ such that $|\varphi(u)| = |u|$ for each $u \in F(A)$. Each is uniquely specified by a permutation $\sigma$ of $A$ and an $A$-tuple $\mathbf{x} = (x_a)_{a \in A} \in \{\pm 1\}^A$: the automorphism specified by $\sigma$ and $\mathbf{x}$ maps each letter $a$ to $\sigma(a)^{x_a}$. In particular, there are $r!\,2^r$ length-preserving (Whitehead) automorphisms.

Let $v \in \tilde{A}$. We define a *$v$-cut of $\tilde{A}$* to be a subset $Y \subseteq \tilde{A}$ containing $v$ and avoiding $\bar{v}$. Each pair $(v, Y)$ of a letter $v \in \tilde{A}$ and a $v$-cut $Y$ defines a

*Whitehead automorphism of the second kind* $\varphi$ as follows: $\varphi(v) = v$ and for each $a \in A \setminus \{v, \bar{v}\}$,

$$\varphi(a) = v^{\gamma} a v^{\rho} \text{ where } \gamma = \begin{cases} -1 & \text{if } \bar{a} \in Y, \\ 0 & \text{otherwise;} \end{cases} \qquad \rho = \begin{cases} 1 & \text{if } a \in Y, \\ 0 & \text{otherwise.} \end{cases}$$

**Remark 1.14** By inverting both sides of this formula, which specifies the images of the letters in $A$ under $\varphi$, we find that the same formula also holds for the letters in $\bar{A}$: if $a \in \bar{A}$ (and $\bar{a} \in A$) and $a \neq v, \bar{v}$, then $\varphi(a) = \varphi(\bar{a})^{-1} = v^{\gamma} a v^{\rho}$ where $\gamma = -1$ if $\bar{a} \in Y$ and $\rho = 1$ if $a \in Y$. □

Observe that, if $Y$ is reduced to the singleton $\{v\}$, then the resulting Whitehead automorphism is the identity. Apart from this particular case, no Whitehead automorphism of the second kind is length-preserving, and the automorphisms specified by different pairs $(v, Y)$ and $(v', Y')$ are distinct. In particular, if $\mathbb{W}(A)$ denotes the set of non-identity Whitehead automorphisms of the second kind, then $\mathsf{card}(\mathbb{W}(A)) = 2r\,(2^{2r-2} - 1) = r\,(2^{2r-1} - 2)$.

Finally, we note that in the algorithms to solve the WMP discussed in Section 1.3, the set of all Whitehead automorphisms can be replaced throughout by $\mathbb{W}(A)$, since we care only about the length of words and cyclic words, and since $\mathbb{W}(A)$ is preserved by composition with the length-preserving Whitehead automorphisms. That is, the function $f(r)$ is Facts 1.11 and 1.13 can be taken equal to $r2^{2r}$. In particular, we have the following fact.

**Fact 1.15** The algorithms given in Section 1.3 to solve the WMP for conjugacy classes of subgroups or for subgroups, take time $O(n^3 r^3 4^r \log(nr))$.

The algorithms to solve the WMP for words, cyclic words and cyclic subgroups take time $O(n^2 r\, 4^r \log r)$. □

# 2 Choice of a best Whitehead automorphism

The algorithms given above to solve the WMP are exponential in the rank $r$ of $F$ because every element of $\mathbb{W}(A)$ may have to be tested at each iteration of the algorithm. Our point is to show that, given a cyclically reduced $A$-graph $\Gamma$, one can in polynomial time (in $r$ and in $|\Gamma|$) find an element $\varphi$ of $\mathbb{W}(A)$ that minimizes $|\varphi(\Gamma)|$ — thereby determining in particular whether $|\Gamma|$ is minimal. Our first tool for this purpose is a generalization of the classical Whitehead graph associated with a cyclic word.

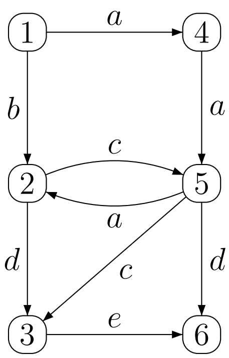
## 2.1 Whitehead hypergraph of a cyclically reduced $A$-graph

A *hypergraph* (here, an undirected one) is a triple $G = (B, D, \kappa)$ where $B$ and $D$ are sets and $\kappa\colon D \to \mathcal{P}(B)$ is such that $\kappa(d) \neq \emptyset$ for each $d \in D$. The elements of $B$ are called *vertices*, the elements of $D$ are called *hyperedges*. It is understood that there can be several hyperedges with the same $\kappa$-image. A hypergraph in which every hyperedge has cardinality at most (resp. exactly) 2 can be identified naturally with an undirected graph (resp. a loop-free undirected graph).

Let $\Gamma = (V, E)$ be a cyclically reduced $A$-graph. The *Whitehead hypergraph* of $\Gamma$, written $W_\Gamma$, is defined as follows. Its vertex set is $\tilde{A}$. Its hyperedge set $D$ is in bijection with $V$, $D = \{d_x \mid x \in V\}$, and for each vertex $x$ of $\Gamma$, $\kappa(d_x)$ is the hyperlink of $x$, $\kappa(d_x) = \mathsf{hl}(x) = \{\lambda(e) \mid e \in E,\ \omega(e) = x\}$.

Note that every hyperedge of $W_\Gamma$ has a $\kappa$-image with at least two elements since $\Gamma$ is cyclically reduced.

**Example 2.1** Let $\Gamma$ be the $A$-graph shown below, where $A = \{a, b, c, d, e\}$. Since the vertex set of $\Gamma$ is $\{1, 2, 3, 4, 5, 6\}$, the Whitehead hypergraph $W_\Gamma$ has vertex set $\tilde{A}$ and hyperedge set $\{d_i \mid 1 \leq i \leq 6\}$, with



$$\kappa(d_1) = \{\bar{a}, \bar{b}\}$$
$$\kappa(d_2) = \{a, b, \bar{c}, \bar{d}\}$$
$$\kappa(d_3) = \{c, d, \bar{e}\}$$
$$\kappa(d_4) = \{a, \bar{a}\}$$
$$\kappa(d_5) = \{a, \bar{a}, c, \bar{c}, \bar{d}\}$$
$$\kappa(d_6) = \{d, e\}$$

□

**Example 2.2** Let $u$ be a cyclically reduced word. Then the hypergraph $W_{\Gamma(\langle u \rangle)}$ is in fact a graph ($\kappa$ maps each hyperedge to a pair of distinct vertices), denoted by $W_u$, which coincides with the classical notion of the *Whitehead graph of a cyclically reduced word* [17, Sec. I.7]. □

**Fact 2.3** If $|\Gamma| = n$, then $W_\Gamma$ has $2r$ vertices, $n$ hyperedges and can be constructed in time $O(nr \log r)$. □

## 2.2 Applying a Whitehead automorphism

We now come to the technical core of this paper: given a cyclically reduced $A$-graph $\Gamma$ and a Whitehead automorphism $\varphi \in \mathbb{W}(A)$, specified by a pair $(v, Y)$, we give an exact formula for the size difference $|\varphi(\Gamma)| - |\Gamma|$. In fact, this formula is already known: it was established by Gersten [5, Prop. 1], proved in Kalajdžievski [11, Prop. 10.3], and it is a generalization of a result of Whitehead [17, Prop. I.4.16], which covers the cyclic word case. Our contribution here consists in rewording it in graph-theoretic terms, and possibly in a clearer demonstration. We note that this formula is an essential ingredient in the proof of Whitehead's theorem and its generalization by Gersten (Theorem 1.9 above), see [17, 5, 11].

Let $G$ be an undirected hypergraph, with vertex set $V(G)$. We define the *capacity* of a subset $Y \subseteq V(G)$ to be the number $\mathsf{cap}_G(Y)$ of hyperedges $e$ of $G$ such that $\kappa(e)$ meets both $Y$ and its complement $Y^c$. If $v \in V(G)$, the *degree of* $v$ is the number $\mathsf{deg}_G(v)$ of hyperedges whose $\kappa$-image contains $v$ (that is, that are adjacent to $v$). We show the following result.

**Proposition 2.4** *Let $\Gamma$ be a cyclically reduced $A$-graph, let $v \in \tilde{A}$ and let $Y \subseteq \tilde{A}$ be a $v$-cut of $\tilde{A}$ (i.e., a set containing $v$ and not $\bar{v}$). Let $\varphi$ be the Whitehead automorphism specified by the pair $(v, Y)$. Then we have*

$$|\varphi(\Gamma)| - |\Gamma| = \mathsf{cap}_{W_\Gamma}(Y) - \mathsf{deg}_{W_\Gamma}(v).$$

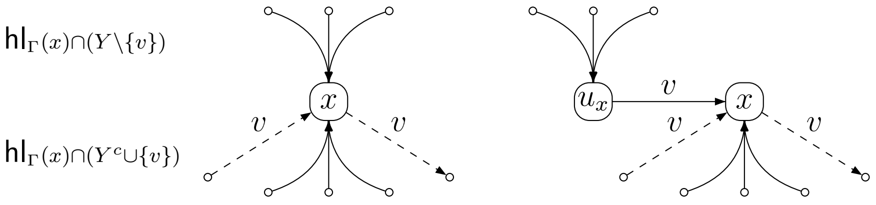
**Proof.** Let $\Gamma = (V, E)$, $v$, $Y$ and $\varphi$ be as in the statement. We first examine in detail the construction of $\varphi(\Gamma)$. The first step is to construct the $A$-graph $\Gamma' = (V', E')$ as follows. For each vertex $x \in V$, if $\mathsf{hl}_\Gamma(x) \cap (Y \setminus \{v\}) \neq \emptyset$, we let $u_x$ be a new vertex. If $\mathsf{hl}_\Gamma(x) \subseteq Y^c \cup \{v\}$, $u_x$ is undefined. We let $V' = V \cup \{u_x \mid u_x \text{ exists}\}$.

The set $E'$ consists of the following edges. All the $v$- and $\bar{v}$-labeled edges of $\Gamma$ are also in $E'$. Next, if $x \in V$ and $u_x$ exists, then there is a $v$-labeled edge from $u_x$ to $x$ and a $\bar{v}$-labeled edge from $x$ to $u_x$. Finally, for each $a$-labeled ($a \neq v, \bar{v}$) edge from $x$ to $y$ in $\Gamma$, there is an $a$-labeled edge in $\Gamma'$

- from $u_x$ to $u_y$ if $a, \bar{a} \in Y$,
- from $x$ to $u_y$ if $a \in Y$, $\bar{a} \notin Y$,
- from $u_x$ to $y$ if $a \notin Y$, $\bar{a} \in Y$,

- from $x$ to $y$ if $a, \bar{a} \notin Y$.

The transformation of $\Gamma$ into $\Gamma'$ is local in the following sense. Around each vertex $x$, we separate $\mathsf{hl}_\Gamma(x)$ into $\mathsf{hl}_\Gamma(x) \cap (Y \setminus \{v\})$ and $\mathsf{hl}_\Gamma(x) \cap (Y^c \cup \{v\})$ and, when the first set is non-empty, we push this fragment of $\mathsf{hl}_\Gamma(x)$ away from $x$ by introducing a new $v$-labeled edge (see Figure 4).


$\mathsf{hl}_{\Gamma(x)\cap(Y\setminus\{v\})}$
$\mathsf{hl}_{\Gamma(x)\cap(Y^c\cup\{v\})}$
$x$
$v$
$v$
$u_x$
$v$
$x$
$v$
$v$

in $\Gamma$ in $\Gamma'$

Figure 4: From $\Gamma$ to $\Gamma'$

An observation that will be important in the sequel is that, by construction, the vertices $x \in V$ satisfy $\mathsf{hl}_{\Gamma'}(x) \subseteq Y^c \cup \{v\}$, while the new vertices of the form $u_x$ satisfy $\mathsf{hl}_{\Gamma'}(u_x) \subseteq (Y \setminus \{v\}) \cup \{\bar{v}\}$.

We note that for each $a$-labeled edge from $x$ to $y$ in $\Gamma$, we now have a path in $\Gamma'$ from $x$ to $y$, labeled by the word $\varphi(a)$, and that $\Gamma'$ consists of the collection of these paths. Thus $\varphi(\Gamma)$ is obtained by first reducing $\Gamma'$, and then taking the cyclic core of the resulting $A$-graph, see Section 1.2.

We now consider whether $\Gamma'$ is reduced. Let $x \in V$ be such that $u_x$ exists. Then $\mathsf{link}_{\Gamma'}(u_x)$ consists of a $\bar{v}$-labeled edge, and a non-empty set in bijection with the set of edges in $\mathsf{link}_\Gamma(x)$ with a label in $Y \setminus \{v\}$. In particular, the labeling map $\lambda'$ is injective on $\mathsf{link}_{\Gamma'}(u_x)$. Moreover, $u_x$ is not an endpoint.

Now let $x$ be a vertex of $\Gamma'$, in $V$. There are 3 cases. We let

- $V_1$ be the set of $x \in V$ such that $u_x$ does not exist, that is, $\mathsf{hl}_\Gamma(x) \subseteq Y^c \cup \{v\}$;

- $V_2$ be the set of $x \in V$ such that $u_x$ exists and $\mathsf{link}_\Gamma(x)$ contains no $v$-labeled edge, that is, $\mathsf{hl}_\Gamma(x) \cap (Y \setminus \{v\}) \neq \emptyset$ and $v \notin \mathsf{hl}_\Gamma(x)$;

- $V_3$ be the set of $x \in V$ such that $u_x$ exists and $\mathsf{link}_\Gamma(x)$ contains a $v$-labeled edge, that is, $\mathsf{hl}_\Gamma(x) \cap (Y \setminus \{v\}) \neq \emptyset$ and $v \in \mathsf{hl}_\Gamma(x)$.

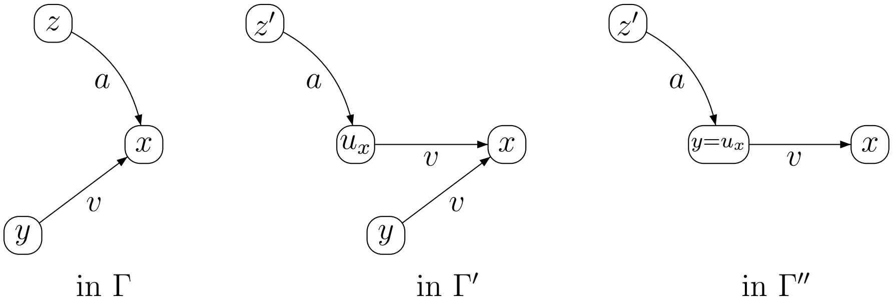
**Case 1:** $x \in V_1$. Then $\mathsf{link}_{\Gamma'}(x)$ is in bijection with $\mathsf{link}_{\Gamma}(x)$, the labeling map $\lambda'$ is injective on $\mathsf{link}_{\Gamma'}(x)$, and $x$ is not an endpoint.

**Case 2:** $x \in V_2$. Then $\mathsf{link}_{\Gamma'}(x)$ consists of a $v$-labeled edge plus a set in bijection with the subset of $\mathsf{link}_{\Gamma}(x)$ of all edges labeled by letters in $Y^c$. In particular, the labeling map $\lambda'$ is injective on $\mathsf{link}_{\Gamma'}(x)$. Moreover, $x$ is an endpoint in $\Gamma'$ if and only if $\mathsf{hl}_{\Gamma}(x) \subseteq Y$.

**Case 3:** $x \in V_3$. Then $\mathsf{link}_{\Gamma'}(x)$ consists of two $v$-labeled edges (one of them starting at $u_x$) and a set in bijection with the subset of $\mathsf{link}_{\Gamma}(x)$ consisting of the edges labeled by letters in $Y^c$.

Thus $\Gamma'$ is non-reduced if and only if $V_3 \neq \emptyset$, and in that case, the first step in reducing $\Gamma'$ consists in performing the elementary reductions that arise from the pairs of $v$-labeled edges into the vertices $x \in V_3$. We claim that the resulting graph, say $\Gamma''$, is already reduced.

In order to justify this claim, let us consider the effect of such an elementary reduction. Since $x \in V_3$, $\Gamma$ has a $v$-labeled edge from some $y \in V$ to $x$, and at least one $a$-labeled edge from some $z \in V$ to $x$, with $a \in Y \setminus \{v\}$, see Figure 5. In $\Gamma'$, there are $v$-labeled edges from $y$ and from $u_x$ to $x$, and an $a$-labeled edge from $z'$ to $u_x$ (with $z' \in \{z, u_z\}$). By a previous observation, $\mathsf{hl}_{\Gamma'}(y) \subseteq Y^c \cup \{v\}$ and $\mathsf{hl}_{\Gamma'}(u_x) \setminus \{\bar{v}\} \subseteq Y \setminus \{v\}$. Thus, after the elementary reduction identifying $u_x$ to $y$, the labeling function $\lambda''$ is injective on $\mathsf{link}_{\Gamma''}(y)$. It follows that $\Gamma''$ is reduced. Moreover $|\Gamma''| = |\Gamma| - \mathsf{card}(V_2)$.



in $\Gamma$ in $\Gamma'$ in $\Gamma''$

Figure 5: $x \in V_3$, $a \in Y \setminus \{v\}$

Next, we proceed to trimming $\Gamma''$, since we want to compute $\varphi(\Gamma)$, which is equal to $\mathsf{cc}(\Gamma'')$. The analysis above shows that $\Gamma''$ has two kinds of endpoints:

- vertices $x \in V_2$ such that $\mathsf{hl}_{\Gamma}(x) \subseteq Y$ (i.e., $\mathsf{hl}_{\Gamma}(x) \subseteq Y \setminus \{v\}$),

- vertices $x \in V_3$ such that $\mathsf{hl}_\Gamma(x) \subseteq Y$.

Suppose first that $x \in V_2$ and $\mathsf{hl}_\Gamma(x) \subseteq Y \setminus \{v\}$. Since $x$ is not an endpoint in $\Gamma$, $\mathsf{hl}_\Gamma(x)$ contains at least 2 elements $a \neq b$ and hence, $\mathsf{hl}_{\Gamma''}(u_x) = \mathsf{hl}_{\Gamma'}(u_x)$ has at least three elements, see Figure 6. Removing the vertex $x$ and the only adjacent edges (labeled $v$ from $u_x$ to $x$, and $\bar{v}$ from $x$ to $u_x$) leaves a non-singleton link at $u_x$, that is, trimming $x$ does not create a new endpoint.

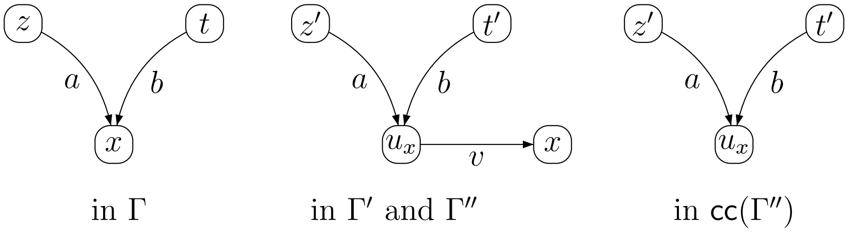


Figure 6: $x \in V_2$, $a, b \in Y \setminus \{v\}$, $a \neq b$

Suppose now that $x \in V_3$ and $\mathsf{hl}_\Gamma(x) \subseteq Y$, and let $y$ be the initial vertex of the $v$-labeled edge of $\Gamma$ into $x$. Since $x \in V_3$, there is an $a$-labeled edge of $\Gamma$ into $x$ (say, from vertex $z \in V$) for some $a \in Y \setminus \{v\}$, and therefore, $a \in \mathsf{hl}_{\Gamma'}(u_x) \cap (Y \setminus \{v\})$. Since $y$ is not an endpoint in $\Gamma$, $\mathsf{link}_\Gamma(y)$ contains an edge labeled $b \neq \bar{v}$, see Figure 7. Then $\mathsf{link}_{\Gamma'}(y)$ contains an edge labeled $b'$ with $b' = v$ if $b \in Y$, and $b' = b$ otherwise. In particular, $b' \in Y^c \cup \{v\} \setminus \{\bar{v}\}$, and hence $b' \neq a$. Finally, the vertices $u_x$ and $y$ are identified in $\Gamma''$, so $\mathsf{link}_{\Gamma''}(y)$ contains the distinct elements $a, b'$ and again, trimming $x$ does not create a new endpoint.

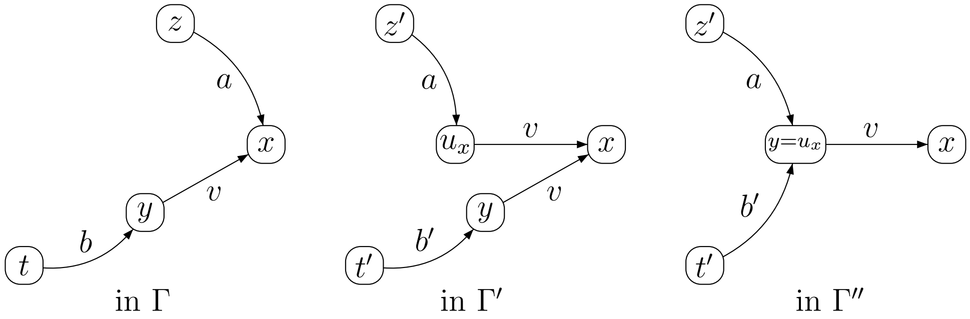


Figure 7: $x \in V_3$, $a \in Y \setminus \{v\}$, $b \neq \bar{v}$, $b' \in \{b, v\}$, $b' \in Y^c \cup \{v\} \setminus \{\bar{v}\}$

It follows that trimming these two families of endpoints suffices to yield $\mathsf{cc}(\Gamma'') = \varphi(\Gamma)$. The total number of vertices trimmed in this process is

$\mathsf{card}(\{x \in V_2 \cup V_3 \mid \mathsf{hl}_\Gamma(x) \subseteq Y\}) = \mathsf{card}(\{x \in V \mid \mathsf{hl}_\Gamma(x) \subseteq Y\})$, so

$$\begin{aligned} |\varphi(\Gamma)| - |\Gamma| &= |\mathsf{cc}(\Gamma'')| - |\Gamma| &= |\Gamma''| - \mathsf{card}(\{x \in V \mid \mathsf{hl}_\Gamma(x) \subseteq Y\}) - |\Gamma| \\ & &= \mathsf{card}(V_2) - \mathsf{card}(\{x \in V \mid \mathsf{hl}_\Gamma(x) \subseteq Y\}). \end{aligned}$$

In this count, we observe that each vertex $x \in V$ may contribute positively (if $x \in V_2$) and negatively (if $\mathsf{hl}_\Gamma(x) \subseteq Y$), that is

$$|\varphi(\Gamma)| - |\Gamma| = \sum_{x \in V} \delta(x),$$

where

$$\delta(x) = \begin{cases} +1 & \text{if } x \in V_2 \text{ and } \mathsf{hl}_\Gamma(x) \not\subseteq Y, \\ -1 & \text{if } x \notin V_2 \text{ and } \mathsf{hl}_\Gamma(x) \subseteq Y, \\ 0 & \text{otherwise.} \end{cases}$$

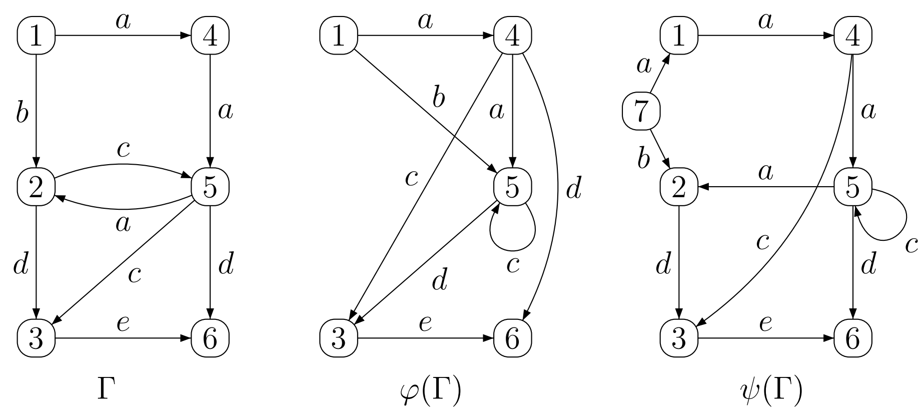
This is equivalent to

$$\delta(x) = \begin{cases} +1 & \text{if } \mathsf{hl}_\Gamma(x) \text{ meets } Y \text{ and } Y^c, \text{ and } v \notin \mathsf{hl}_\Gamma(x), \\ -1 & \text{if } \mathsf{hl}_\Gamma(x) \subseteq Y \text{ and } v \in \mathsf{hl}_\Gamma(x), \\ 0 & \text{otherwise,} \end{cases}$$

which yields the expected formula, $|\varphi(\Gamma)| - |\Gamma| = \mathsf{cap}_{W_\Gamma}(Y) - \mathsf{deg}_{W_\Gamma}(v)$. □

**Example 2.5** Consider the 6-vertex $A$-graph $\Gamma$ in Figure 8. Its Whitehead hypergraph $W_\Gamma$ was computed in Example 2.1 and we have $\mathsf{deg}_{W_\Gamma}(a) = 3$. Let $\varphi$ be the Whitehead automorphism specified by the pair $(a, \{a, b, \bar{c}, \bar{d}\})$. We note that $\mathsf{cap}_{W_\Gamma}(\{a, b, \bar{c}, \bar{d}\}) = 2$, and that $\varphi(\Gamma)$ then has $6{+}2{-}3 = 5$ vertices, in conformity with Proposition 2.4. The graph $\varphi(\Gamma)$ is shown in Figure 8, as well as the graph $\psi(\Gamma)$, where $\psi$ is the Whitehead automorphism specified by the pair $(a, \{a, \bar{b}, \bar{c}\})$. Since $\mathsf{cap}_{W_\Gamma}(\{a, \bar{b}, \bar{c}\}) = 4$ , $\psi(\Gamma)$ must have size $6 + 4 - 3 = 7$ by Proposition 2.4. □

**Remark 2.6** Let $\Gamma$ be a cyclically reduced $A$-graph, let $v \in \tilde{A}$ and let $Y \subseteq \tilde{A}$. It is easily verified that $Y$ is a $v$-cut if and only if its complement $Y^c$ is a $\bar{v}$-cut, $\mathsf{cap}_{W_\Gamma}(Y) = \mathsf{cap}_{W_\Gamma}(Y^c)$ and $\mathsf{deg}_{W_\Gamma}(v) = \mathsf{deg}_{W_\Gamma}(\bar{v})$ (equal to the number of $v$-labeled edges in $\Gamma$). □



$\Gamma$ $\varphi(\Gamma)$ $\psi(\Gamma)$

Figure 8: The graphs $\Gamma$, $\varphi(\Gamma)$ and $\psi(\Gamma)$

## 2.3 Relative complexity of the Whitehead minimization problem

The algorithms to solve the WMP discussed in Section 1.3, can now be modified as follows.

We first consider the case of conjugacy classes of finitely generated subgroups. Let $H$ be a cyclically reduced subgroup. First let $\vec{\varphi} = (\mathsf{id})$ and $\Gamma = \Gamma(H)$. Then repeatedly apply the following steps: compute the Whitehead hypergraph $W_\Gamma$ and for each $v \in A$, find a $v$-cut $Y_v$ of $\tilde{A}$ that minimizes $\mathsf{cap}_{W_\Gamma}(Y_v)$; if $\min_{v\in A}(\mathsf{cap}_{W_\Gamma}(Y_v) - \mathsf{deg}_{W_\Gamma}(v)) < 0$, let $\psi$ be the Whitehead automorphism specified by $(v, Y_v)$ where $v$ realizes the above minimum, and replace $\Gamma$ by $\psi(\Gamma)$ and $\vec{\varphi}$ by $(\psi, \vec{\varphi})$; otherwise, stop and output $\Gamma$ and $\vec{\varphi}$. Finally, choose arbitrarily a vertex 1 in $\Gamma$ and use the procedure discussed in Section 1.1.3 to output a basis of $H'$.

The difference with the algorithm in Section 1.3 lies in the fact that instead of trying every Whitehead automorphism to find one that decreases the size of the cyclically reduced $A$-graph, we directly select one that will yield the maximum size decrease. The fact that we consider only $v$-cuts where $v \in A$ is justified by Remark 2.6.

Passing from the above algorithm to one that solves the WMP for subgroups, is done as in Section 1.3.

In order to estimate the complexity of the reworded algorithm, we let $g(n, r)$ be the complexity of the following problem, which we call the *Whitehead hypergraph min-cut problem*:

if $\mathsf{card}(A) = r$, given $W_\Gamma$, the Whitehead hypergraph of a cyclically reduced $A$-graph $\Gamma$ of size $n$ and a letter $v \in A$, find a $v$-cut $Y$ of $\tilde{A}$ minimizing $\mathsf{cap}_{W_\Gamma}(Y)$.

**Fact 2.7** As we already saw in Fact 1.11, the cost of the construction of $\Gamma(H)$, if the input is a set of generators of $H$ of total length $n$, is $O(n^2 \log(nr))$, and $\Gamma(H)$ has at most $n$ vertices. The computation of the image of a cyclically reduced graph of size $n$ under a Whitehead automorphism takes time $O(n^2 r^2 \log(nr))$. Moreover, the Whitehead hypergraph of a size $n$ cyclically reduced $A$-graph has $n$ hyperedges, and is computed in time $O(nr \log r)$ (Fact 2.3). Finding the degree of a vertex in such a hypergraph takes time $O(nr \log r)$.

Then the complexity of each iterating step of our algorithm is at most $O(nr \log r + r(g(n,r) + nr \log r) + n^2 r^2 \log(nr)) = O(n^2 r^2 \log(nr) + rg(n,r))$. Since there are at most $n$ iterating steps, the complexity of the full algorithm is $O(n^2 \log(nr) + n(n^2 r^2 \log(nr) + rg(n,r)) + n^2 \log(nr))$, that is, $O(n^3 r^2 \log(nr) + nrg(n,r))$. □

**Fact 2.8** Let the *Whitehead graph min-cut problem* be an instance of the Whitehead hypergraph min-cut problem where the input is the Whitehead graph $W_u$ of a cyclic word $u$, and let $g'(n,r)$ be the complexity function of this problem. Reasoning as in Fact 2.7, we find that the complexity of our algorithm to solve the WMP for cyclic words is $O(n^2 \log r + nrg'(n,r))$. □

# 3 Main result

To conclude our work, we need to find algorithms to solve the Whitehead hypergraph min-cut problem and its graph analogue in time polynomial in $n$ and $r$.

## 3.1 On minimizing the capacity of a cut

### 3.1.1 The general case

The solution of the Whitehead hypergraph min-cut problem can be reduced to a standard problem in combinatorial optimization, that of the minimization of submodular functions. A real-valued function $f$, defined on the powerset of a set $B$, is said to be *submodular* if $f(Y \cup Z) + f(Y \cap Z) \leq f(Y) + f(Z)$ for any $Y, Z \subseteq B$. We first verify the following fact.

**Lemma 3.1** *Let $W = (B, D, \kappa)$ be a hypergraph. The map $Y \longmapsto \mathsf{cap}_W(Y)$, defined on the powerset of $B$, is submodular.*

**Proof.** Let $d$ be a hyperedge of $W$. The contribution of $d$ to $\mathsf{cap}_W(Y)$ is 0 if $\kappa(d) \subseteq Y$ or $\kappa(d) \subseteq Y^c$, and 1 otherwise. In particular:

- if the contribution of $d$ to $\mathsf{cap}_W(Y \cup Z) + \mathsf{cap}_W(Y \cap Z)$ is 2, then $\kappa(d)$ meets $Y \cup Z$, $(Y \cup Z)^c$, $Y \cap Z$ and $(Y \cap Z)^c$; then it meets $Y$, $Y^c$, $Z$ and $Z^c$, so that the contribution of $d$ to $\mathsf{cap}_W(Y) + \mathsf{cap}_W(Z)$ is 2 as well;

- if the contribution of $d$ to $\mathsf{cap}_W(Y) + \mathsf{cap}_W(Z)$ is 0, then $\kappa(d)$ is contained in $Y$ or in $Y^c$, and it is contained in $Z$ or in $Z^c$; equivalently, it is contained in $Y \cap Z$, $Y^c \cap Z$, $Y \cap Z^c$ or $Y^c \cap Z^c$; in particular, the contribution of $d$ to $\mathsf{cap}_W(Y \cup Z) + \mathsf{cap}_W(Y \cap Z)$ is 0 as well;

- in all other cases, the contribution of $d$ to $\mathsf{cap}_W(Y \cup Z) + \mathsf{cap}_W(Y \cap Z)$ is at most 1 and its contribution to $\mathsf{cap}_W(Y) + \mathsf{cap}_W(Z)$ is at least 1.

This concludes the proof. □

Let $v$ and $W_\Gamma$ be an instance of the Whitehead hypergraph min-cut problem. For each $Y \subseteq V \setminus \{v, \bar{v}\}$, let $f(Y) = \mathsf{cap}_{W_\Gamma}(Y \cup \{v\})$. We note that $Y$ minimizes function $f$ if and only if $Y \cup \{v\}$ has minimum capacity among the $v$-cuts of $\tilde{A}$. It is easily derived from Lemma 3.1 that $f$ is submodular.

It follows from results of Grötschel, Lovász and Schrijver [6] that $f$ can be minimized by an algorithm that makes a polynomial (in $r$) number of oracle calls (queries to evaluate $f$ on a given argument). In our situation, given a subset $Y \subseteq V \setminus \{v, \bar{v}\}$, computing $f(Y) = \mathsf{cap}_{W_\Gamma}(Y \cup \{v\})$ takes time $O(nr \log r)$, so the Whitehead hypergraph min-cut problem can be solved in time polynomial in $n$ and $r$. According to Queyranne [20, pp. 3-4], the number of oracle calls is $O(r^4)$, so the running time of the algorithm is $O(nr^5 \log r)$.

A more recent result of Cunningham [1] gives a minimization algorithm with running time $O(Mr^3 \log(Mr))$, where $M$ is an upper bound on the maximum value of $f$. In our case, the value of $f$ is at most the number of hyperedges in $W_\Gamma$, namely $n$, so Cunningham's algorithm runs in time $O(nr^3 \log(nr))$. We may take $g(n, r) = nr^3 \log(nr)$.

**Remark 3.2** The efficient minimization of submodular functions is an active research topic, and more recent work offers different algorithms which can be used for our purpose just as well as Cunningham's. We refer the reader for instance to Iwata, Fleischer and Fujishige [10], Schrijver [21] and Iwata [9]. □

### 3.1.2 The graph case

In the case where the cyclically reduced graph $\Gamma$ is a cyclic word, the Whitehead hypergraph $W_\Gamma$ is in fact a graph. The Whitehead graph min-cut problem is a particular case of the more general *min-cut problem*, also a standard problem in operational research, for the solution of which there exists a vast literature.

In its generality, the min-cut problem for graphs is the following. We are given a directed graph $G = (V, E)$ with vertex set $V$ and edge set $E$, and a pair $(s, t)$ of distinct vertices of $G$. In this problem, there may be several edges from a vertex $x$ to a vertex $y$. An *$(s,t)$-cut* of $G$ is a subset $Y$ of $V$, containing $s$ and avoiding $t$. The *capacity* $\mathsf{cap}_G(Y)$ of such a set is equal to the number of edges that start in $Y$ and end in the complement of $Y$. The min-cut problem consists in finding an $(s,t)$-cut $Y$ that minimizes $\mathsf{cap}_G(Y)$.

There are many algorithms to efficiently solve the min-cut problem, see below. In order to solve the Whitehead min-cut problem on instance $W_u$ and $v$ (see Section 2.3), we may first turn $W_u$ into a directed graph $W_u^+$ as follows: we replace each undirected edge between vertices $x$ and $y$ by a pair of directed edges, one from $x$ to $y$ and the other from $y$ to $x$. Next, we observe that a $v$-cut in the sense of Section 1.4 is a $(v, \bar{v})$-cut in the sense of the min-cut problem, and conversely. Finally, we verify that if $Y$ is a $(v, \bar{v})$-cut, then both notions of capacity of $Y$ coincide, that is, $\mathsf{cap}_{W_u}(Y) = \mathsf{cap}_{W_u^+}(Y)$.

Thus a $(v, \bar{v})$-cut with minimum capacity in $W_u^+$ is also a $v$-cut with minimum capacity in $W_u$. In particular, we may take $g'(n, r)$ to be the time complexity of any algorithm solving the min-cut problem in a directed graph with $n$ edges and $2r$ vertices.

Finally, we note that Dinic's algorithm solves the min-cut problem in time $O(nr^2)$ [2], see [15, p. 97], that is, we may take $g'(n, r) = nr^2$.

**Remark 3.3** There are many polynomial time algorithms to solve the min-cut problem, and we refer the reader to Kozen's book [15, Chaps. 15–17] for a review of some of those algorithms that rely on the max-flow min-cut theorem (Ford and Fulkerson [3]), that is, that consist in maximizing a flow function associated with the graph. Dinic's algorithm mentioned above falls in that category. We note also that Galil's more recent algorithm [4] works in time $O(n^{2/3}\,r^{5/3})$. □

## 3.2 Fully polynomial algorithms

Putting together the results of Sections 2.3 and 3.1, we get the expected result.

**Theorem 3.4** *One can solve the WMP in time polynomial in the size $n$ of the input and the rank $r$ of the ambient free group.*

More precisely, on the basis of Facts 2.7 and 2.8, the discussion in Section 3.1 implies the following.

**Fact 3.5** The WMP for finitely generated subgroups and for conjugacy classes of finitely generated subgroups can be solved in time $O((n^2r^4+n^3r^2)\log(nr))$, where $n$ is the size of the input and $r = \mathsf{rank}(F)$. □

**Fact 3.6** The WMP for words and for cyclic words can be solved in time $O(n^2r^3)$, where $n$ is the size of the input and $r = \mathsf{rank}(F)$. □

Our main concern in this paper is the fact that the above complexity functions are polynomial in $n$ and $r$, and we are less concerned with the exact polynomial that can be achieved. In fact, we have phrased our algorithms in a modular way: an algorithm solving the Whitehead (hypergraph or graph) min-cut problem is called by our algorithm, and any improvement in the efficiency of the computation of a min-cut leads to an improvement in the efficiency of our algorithm.

It is also worth noting that in the input of the WMP, we may assume $r \leq n$. Indeed, letters of $A$ that do not occur in the input word or subgroup may be ignored, for instance by restricting ourselves to Whitehead automorphisms that fix them (say, leaving them and their inverses outside any $v$-cut). This implies immediately the following more compact results.

**Fact 3.7** The WMP for finitely generated subgroups and for conjugacy classes of finitely generated subgroups can be solved in time $O(n^6\log n)$, where $n$ is the size of the input, independently of the rank of the ambient free group. □

**Fact 3.8** The WMP for words and for cyclic words can be solved in time $O(n^5)$, where $n$ is the size of the input, independently of the rank of the ambient free group. □

## 3.3 Consequences

Recall that a subgroup $H \leq_{\mathsf{fg}} F(A)$ is a *free factor* of $F$ if any of its bases can be extended to a basis of $F$. The *free factor problem* consists in deciding, given $H$, whether $H$ is a free factor of $F$. It is immediate that this is the case if and only if the minimum size of an element of the automorphic orbit of $H$ is 1. Therefore we obtain the following corollary.

**Corollary 3.9** *There is an algorithm that decides the free factor problem (for a subgroup given by a set of generators of total length $n$ in a rank $r$ free group) in time polynomial in both $n$ and $r$.*

It is interesting to compare this result with that obtained by Silva and Weil [22]. These authors give a purely graph-theoretic algorithm to solve the free factor problem on input $H$ in time $O(n^{2d+2}\log(nr))$, where $d = r - \mathsf{rank}(H)$. According to the theoretical complexity functions, the result in Corollary 3.9 is stronger in general, but Silva and Weil's algorithm may be more efficient on large size, large rank inputs. Computer experiments might be interesting, especially as the latter algorithm is simpler to implement, and might yield smaller constants.

A word $u \in F(A)$ is *primitive* if it is an element of some basis of $F(A)$. That is, $u$ is primitive if and only if $\langle u \rangle$ is a free factor of $F$. So we also have the following corollary.

**Corollary 3.10** *There is an algorithm that decides primitivity (of a word of length $n$ in a rank $r$ free group) in time polynomial in both $n$ and $r$.*

Observe that the formula proved in Proposition 2.4 is additive, in the following sense: if $\mathbf{\Gamma} = (\Gamma_1, \ldots, \Gamma_m)$ is a tuple of cyclically reduced $A$-graphs, and if $W_{\mathbf{\Gamma}}$ is the Whitehead hypergraph of this tuple (the union of the $W_{\Gamma_i}$), if $v \in \tilde{A}$, $Y$ is a $v$-cut of $\tilde{A}$ and $\varphi$ is the Whitehead automorphism determined by $(v, Y)$, then

$$\sum_{i=1}^{m} |\varphi(\Gamma_i)| - \sum_{i=1}^{m} |\Gamma_i| = \mathsf{cap}_{W_{\mathbf{\Gamma}}}(Y) - \mathsf{deg}_{W_{\mathbf{\Gamma}}}(v).$$

This additivity extends to Gersten's theorem (Theorem 1.9 above) as observed in [5]. That is, if some automorphism of $F$ reduces the total size of a tuple of cyclically reduced $A$-graphs, then some Whitehead automorphism does [5, Corol. 2], generalizing Whitehead's result [17, Prop. I.4.20]). Our argument then also carries over to the complexity of the *Whitehead minimization problem for tuples of conjugacy classes of finitely generated subgroups* (to find a tuple of conjugacy classes with minimum total size, in the automorphic orbit of a given tuple).

**Corollary 3.11** *There is an algorithm that solves the WMP for tuples of conjugacy classes of finitely generated subgroups in time polynomial in both $n$ (the sum of the sizes of the given conjugacy classes) and $r$ (the rank of the ambient free group).*

**Corollary 3.12** *There is an algorithm that solves the WMP for tuples of cyclic words in time polynomial in both $n$ (the sum of the sizes of the given conjugacy classes) and $r$ (the rank of the ambient free group).*

## 3.4 A few open questions

**A cut-vertex theorem?** Connectedness and connected components are defined in hypergraphs as in graphs: two vertices $b, b'$ of a hypergraph $W = (B, D, \kappa)$ are connected if there exist a sequence of hyperedges $d_1, \ldots, d_\ell$ such that $b \in \kappa(d_1)$, $\kappa(d_i) \cap \kappa(d_{i+1}) \neq \emptyset$ for all $1 \leq i < \ell$, and $b' \in \kappa(d_\ell)$. Let $\Gamma$ be a cyclically reduced $A$-graph, and say that $v \in \tilde{A}$ is a *cut-vertex* of $W_\Gamma$ if removing $v$ and the hyperedges adjacent to it, yields a hypergraph $W'$ with more connected components than $W_\Gamma$.

If $v$ is a cut-vertex, then the connected component of $W$ containing $v$ splits into at least two non-empty connected components when $v$ is removed; let $Y'$ be one of them, not containing $\bar{v}$, let $Y = Y' \cup \{v\}$ and let $\varphi \in \mathbb{W}(A)$ be specified by $(v, Y)$. By definition of a cut-vertex, the hyperedges connecting $Y$ and $Y^c$ form a proper subset of the hyperedges adjacent to $v$, that is, $\mathsf{cap}_{W_\Gamma}(Y) < \mathsf{deg}_{W_\Gamma}(v)$. It follows that $|\varphi(\Gamma)| < |\Gamma|$. In particular, if $\Gamma = \Gamma(H)$, then the size of $H$ is not minimal.

This generalizes to subgroups a simple part of Whitehead's celebrated cut-vertex theorem: if $u$ is a cyclically reduced word and $W_u$ has a cut-vertex, then the length of $u$ is not minimal. It is known that the converse does not hold, that is, the Whitehead graphs of some non-minimal words present no cut-vertex (the word $u = abbaab \in F(\{a, b\})$ provides an example). However, the Whitehead cut-vertex Lemma (see Stallings [24, Theorem 2.4]) states the much deeper result that if $u$ is primitive then its Whitehead graph $W_u$ is either disconnected (in which case some conjugate of $u$ is contained in a proper free factor of $F$, [24, Prop. 2.2]) or it has a cut-vertex. It would be interesting to find an analogous statement for subgroups (and cut-vertices in the Whitehead hypergraph), and to see whether these statements can be derived from the combinatorial arguments discussed here.

**The hard part of the equivalence problem** In the so-called hard-part of Whitehead's algorithm to solve the equivalence problem, say for cyclic words, one considers two cyclic words $u$ and $v$ of minimum length in their automorphic orbit, and one needs to decide whether a sequence of Whitehead automorphisms takes $u$ to $v$ without ever changing the cyclic length. At first sight, this might require exploring all words of length $|u|$, which yields an algorithm that is exponential in both $n$ and $r$. Myasnikov and Shpilrain [19] (see also Khan [14]) establish that the complexity is in fact polynomial in $n$ for $r = 2$, and more recent work by Lee [16] suggests that this is probably true for all values of $r$. Kapovich, Schupp and Shpilrain [13] develop a remarkable study of the generic-case complexity of this problem. One might hope to use our method to get rid of the exponential dependency in $r$ as well. This would require being able to find *all* minimal cuts in the Whitehead graph of a cyclic word *of minimal length*, and it would be interesting to investigate whether that can be done in polynomial time.

Extending that investigation to minimal cuts in Whitehead hypergraphs would naturally be equally interesting.

**Is the greedy algorithm the optimal size reduction technique?** In the Whitehead minimization problem, one may be interested in minimizing the number of Whitehead automorphisms one needs to apply to a conjugacy class $[H]$ in order to find a minimum size element of its orbit. The algorithm discussed in this paper follows the so-called *greedy* paradigm: at each step of the iteration, one chooses a Whitehead automorphism that maximizes the size decrement. This does not a priori imply that the number of steps is minimized. It would be interesting to verify whether such a greedy algorithm is in fact optimal also in the number of steps, and to get estimates of that number of steps.

# References

[1] W.H. Cunningham. On submodular function minimization, *Combinatorica* **5** (1985) 185-192.

[2] E.A. Dinic. Algorithm for solution of a problem of maximal flow in a network with power estimation, *Soviet Math. Doklady* **11** (1970) 1277-1280.

[3] L.R. Ford Jr., D.R. Fulkerson. Maximal flow through a network, *Canad. J. Math.* **8** (1956) 399-404.

[4] Z. Galil. An $O(V^{\frac{5}{3}}E^{\frac{2}{3}})$ algorithm for the maximal flow problem, *Acta Informatica* **14** (1980) 221-242.

[5] S. Gersten. On Whitehead's algorithm, *Bull. Am. Math. Soc.* **10** (1984) 281-284.

[6] M. Grötschel, L. Lovász, A. Schrijver. *Geometric algorithms and combinatorial optimization*, Springer (1988).

[7] R.M. Haralick, A.D. Myasnikov. A hybrid search algorithm for the Whitehead minimization problem, preprint, `arXiv:math.GR/0604206`, 2006.

[8] R.M. Haralick, A.D. Miasnikov, A.G. Myasnikov. Heuristics for the Whitehead minimization problem, *Experiment. Math.* **14:1** (2005) 7-14.

[9] S. Iwata. A fully combinatorial algorithm for submodular function minimization, *SODA* 2002.

[10] S. Iwata, L. Fleischer, S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions, *J. Assoc. Comput. Mach.* **48** (2001) 761-777.

[11] S. Kalajdžievski. Automorphism group of a free group: centralizers and stabilizers, *J. Algebra* **150** (1992) 435-502.

[12] I. Kapovich, A. Miasnikov. Stallings foldings and subgroups of free groups, *J. Algebra* **248** (2002) 608-668.

[13] I. Kapovich, P. Schupp, V. Shpilrain. Generic properties of Whitehead's algorithm and isomorphism rigidity of random one-relator groups, *Pacific J. Math.* **223** (2006) 113-140.

[14] B. Khan. The structure of automorphic conjugacy in the free group of rank two, in *Proc. Special Session on interactions between logic, group theory and computer science*, Contemporary Mathematics **349**, Amer. Math. Soc. (2004).

[15] D. Kozen. *The design and analysis of algorithms*, Springer (1991).

[16] Donghi Lee. A tighter bound for the number of words of minimum length in an automorphic orbit, preprint, 2006.

[17] R. Lyndon, P. Schupp. *Combinatorial group theory*, Springer (1977, reprinted 2001).

[18] A.D. Miasnikov, A.G. Myasnikov. Whitehead method and genetic algorithms, *Contemporary Mathematics* **349** (2004) 89-114.

[19] A.V. Myasnikov, V. Shpilrain. Automorphic orbits in free groups, *J. Algebra* **269** (2003) 18-27.

[20] M. Queyranne. Minimizing symmetric submodular functions, *Mathematical Programming* **82** (1998) 3-12.

[21] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *J. Combinat. Theory*, Series B **80** (2000) 346-355.

[22] P. Silva, P. Weil. On an algorithm to decide whether a free group is a free factor of another, `arXiv:math.GR/0609552`, 2005.

[23] J. R. Stallings. Topology of finite graphs, *Inventiones Math.* **71** (1983) 551-565.

[24] J.R. Stallings. Whitehead graphs on handlebodies, in *Geometric group theory down under (Canberra, 1996)*, 317-330, de Gruyter, Berlin, 1999.

[25] N. Touikan. A fast algorithm for Stalling's folding process, preprint, `www.math.mcgill.ca/~touikan/crypto_seminar/FastFolding.pdf`, 2005.

[26] J.H.C. Whitehead. On equivalent sets of elements in a free group, *Annals of Mathematics* **37** (1936) 782-800.