



HAL
open science

Computing the fully optimal spanning tree of an ordered bipolar directed graph

Emeric Gioan, Michel Las Vergnas

► **To cite this version:**

Emeric Gioan, Michel Las Vergnas. Computing the fully optimal spanning tree of an ordered bipolar directed graph. *Discrete Mathematics*, In press. lirmm-04387449v2

HAL Id: lirmm-04387449

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-04387449v2>

Submitted on 2 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computing the fully optimal spanning tree of an ordered bipolar directed graph[☆]

Emeric Gioan^{*,a}, Michel Las Vergnas^{**,b}

^a*LIRMM, Univ. Montpellier, CNRS, Montpellier, France*

^b*CNRS, Paris, France*

Abstract

It was previously shown by the authors that a directed graph on a linearly ordered set of edges (ordered graph) with adjacent unique source and sink (bipolar digraph) has a unique fully optimal spanning tree, that satisfies a simple criterion on fundamental cycle/cocycle directions. This result is related to a strengthening of the notion of optimality in linear programming. Furthermore, this result yields, for any ordered graph, a canonical bijection between bipolar orientations and spanning trees with internal activity 1 and external activity 0 in the sense of the Tutte polynomial. This bijection can be extended to all orientations and all spanning trees, yielding the active bijection, presented for graphs in other papers. In this paper, we specifically address the problem of the computation of the fully optimal spanning tree of an ordered bipolar digraph. In contrast with the inverse mapping, built by a straightforward single pass over the edge set, the direct computation is not easy and had previously been left aside. We give two independent constructions. The first one is a deletion/contraction recursion, involving an exponential number of minors. It is structurally significant but it is efficient only for building the whole bijection (i.e., all images) at once. The second one is more complicated and is the main contribution of the paper. It involves just one minor for each edge of the resulting spanning tree, and it is a translation and an adaptation to the case of graphs, in terms of weighted cocycles, of a general geometric linear programming type algorithm, which allows for a polynomial time complexity.

Keywords: bipolar directed graph, spanning tree, computational complexity, linear programming, active bijection, Tutte polynomial

1. Introduction

This paper is in the continuation of a series of papers by the authors involving the active bijection and fundamental properties of directed graphs on a linearly ordered set of edges (and, more generally, of oriented matroids on a linearly ordered ground set), including [4–14]. This paper is focused on a special kind of directed graph which is central in the series. It answers computational problems that were previously left aside, and can be read independently of the rest of the series, though it is rather technical and specialized. Parts of the formalism and of the technique that we

[☆]Research supported by the ANR Grant DIGRAPHS (Directed graphs, ANR-19-CE48-0013).

*Email: emeric.gioan@lirmm.fr

**R.I.P.

use are inspired by (oriented) matroid theory, but the paper is written in terms of graphs and can be read by a graph specialist. Also, it relies upon specificities of graphs.

In a previous paper [7] (see also [9]), we showed that a directed graph \vec{G} on a linearly ordered set of edges, with adjacent unique source and sink connected by the smallest edge (ordered bipolar digraph), has a unique special spanning tree that satisfies a simple criterion on fundamental cycle/cocycle directions, and that we call *the fully optimal spanning tree* $\alpha(\vec{G})$ of \vec{G} . Precise definitions of these notions are given in Section 2.

As motivation, let us point out that this notion is important for various reasons. First, it is related to a strengthening of the notion of optimality in linear programming. Briefly, a bipolar digraph is a particular case of a bounded region of a hyperplane arrangement, and the fully optimal spanning tree is a particular case of the fully optimal basis, which is a uniquely determined basis that contains and extensively strengthens the notion of optimal vertex of a bounded region; see Remark 4.7, and see [6, 9, 10, 13]. Second, ordered bipolar digraphs are the basic object of a general canonical decomposition of directed graphs into bipolar minors, from which one gets, by considering all orientations of a graph (that is, all possible directed graphs having this graph as underlying undirected graph): an expression for the Tutte polynomial of the underlying undirected graph in terms of products of β -invariants of minors, an important partition of the set of orientations into activity classes, and a simple expression for the Tutte polynomial using four orientation activity parameters; see [11, Section 3]. Third, the fully optimal spanning tree of an ordered bipolar digraph is the basic object that allows us to build a canonical bijection, as well as various significant bijections, involving all orientations and all spanning trees of a graph; see [11, Section 4]. Briefly, associating bipolar orientations of an ordered graph with their fully optimal spanning trees provides a canonical bijection with spanning trees with internal activity 1 and external activity 0 in the sense of the Tutte polynomial [17]. It is a classical result from [18] that those two sets have the same size, also known as the β -invariant $\beta(G)$ of the graph [3]. We call this bijection *the uniactive bijection of G* . This bijection can be extended to all orientations and all spanning trees, yielding *the active bijection*, introduced in terms of graphs first in [7], and detailed next in [11] (see also [8] for the case of chordal graphs). Beyond graphs, the general context of the active bijection is oriented matroids, as studied by the authors in a series of papers, including [4–14]. See [5] for a detailed and practical overview, or see the introductions of [11] or [12] for a general overview, more information and references. The general purpose of this work is to study graphs (or hyperplane arrangements, or oriented matroids) on a linearly ordered set of edges (or ground set), under various structural and enumerative aspects.

In the present paper we address the problem of computing the fully optimal spanning tree. Its existence and uniqueness is a difficult combinatorial theorem, proved in [7, Theorem 4] (see also [11, Section 4.1] for a short summary in graphs; see also [9, Theorem 4.5] for a generalization and a geometric interpretation in oriented matroids; see also [12] for a summary of various interpretations and implications of this theorem). As shown in [7] and recalled in Section 2.2, the inverse mapping, producing a bipolar orientation for which a given spanning tree is fully optimal, is very easy to compute by a single pass over the ordered set of edges. But the direct computation is complicated and it had not been addressed in previous papers. When generalized to real hyperplane arrangements, the problem contains and strengthens the linear programming problem (as shown in [9], hence the name *fully optimal*), which is already a non-trivial computational problem, as there are known polynomial time algorithms using numerical methods, but no known purely

combinatorial polynomial time algorithm. This “one way function” feature is a noteworthy aspect of the active bijection. Here, we give two independent constructions to compute the mapping, that is, to compute the fully optimal spanning tree of an ordered bipolar digraph.

The first construction, in Section 3, is recursive, by deletion/contraction of the greatest element. It was recently given in [11, Section 6.1] in the same way, but we recall it for the sake of completely addressing the problem and detailing computational complexity aspects. Let us observe that it is usual to have some deletion/contraction constructions when the Tutte polynomial is involved. This construction of the mapping has a short statement and proof, and it can be used to efficiently build the whole bijection at once (i.e., all the images simultaneously; see Remark 3.5). So, it is satisfying for structural understanding and for a global approach, but it is not satisfying in terms of computational complexity for building one single image as it involves an exponential number of minors. Let us mention that this construction is part of a general deletion/contraction framework [11, Section 6], and that it generalizes to oriented matroids [4, 12].

The second construction, in Section 4, is more technical and is the main contribution of the paper. It is efficient from the computational complexity viewpoint because it involves only one minor for each edge of the resulting spanning tree, and it consists of searching, successively in each minor, for the smallest cocycle with respect to a linear ordering of the set of signed cocycles induced by a suitable weight function. In this way, the fully optimal spanning tree can be computed in polynomial time. In detail, seeing that the successive fundamental cocycles of the fully optimal spanning tree induce such optimal cocycles in successive suitable minors is the core of the construction (Theorem 4.3). Actually, this algorithm is a translation and an adaptation to the case of graphs (implicitly using that graphic matroids are binary) of a general geometric construction obtained by elaborations on pseudo/real linear programming (in oriented matroids / real hyperplane arrangements). In particular, searching for a minimum weight signed cocycle amounts to solving a lexicographic multiobjective linear program (see Remark 4.7 and details in the proof of Lemma 4.9; see Remark 4.11 for related digraph problems). Since this is repeated in a number of minors which is bounded by the size of a spanning tree, we get a polynomial time construction (Theorem 4.10). Let us mention that, just as we use that graphic matroids are binary, we could directly provide a similar construction for regular matroids, but we shall not detail this. To go further: see [13] for the general geometric construction; see [10] for a short formulation of the same algorithm in terms of real hyperplane arrangements; see [9] for general relations between the full optimality criterion and usual linear programming optimality; see also [6] on the uniform case (for real vectors in general position, the construction is equivalent to standard linear programming).

In addition, let us recall from [7, Section 4] (see also [11, Section 4.1] for more details) that the bijection between bipolar orientations and their fully optimal spanning trees directly yields a bijection between cyclic-bipolar orientations (the strongly connected orientations obtained from bipolar orientations by reversing the source-sink edge) and spanning trees with internal activity 0 and external activity 1 (obtained from spanning trees with internal activity 1 and external activity 0 by exchanging in the spanning tree the smallest edge with the second smallest edge of the graph). Hence, the algorithms developed here can also be used for this second bijection. Let us mention that those two bijections are also connected by an important duality property, called *the active duality*, which strengthens linear programming duality; see [11, Section 4.1] and [9, Section 5].

Lastly, it is important to point out that the two aforementioned constructions of the fully optimal spanning tree do not give a new proof of its existence and uniqueness: on the contrary, this crucial fundamental result is used to ensure the correctness of these two constructions.

2. Preliminaries

2.1. Usual terminology and tools from oriented matroid theory

All graphs considered in this paper will be connected. They can have loops and multiple edges. A *digraph* is a directed graph, and an *ordered graph* is a graph $G = (V, E)$ on a linearly ordered set of edges E . Edges of a directed graph are *directed* (or *oriented*) from one endpoint to the other. We still call them *edges* rather than *arcs*. A directed graph will be denoted with an arrow, \vec{G} , and the underlying undirected graph without arrow, G . In this setting, \vec{G} is called an *orientation* of G .

The cycles, cocycles, and spanning trees of a graph $G = (V, E)$ are considered as subsets of E , hence their edges can be called their *elements*. The cycles and cocycles of G are always understood as being minimal by inclusion. Given $F \subseteq E$, we denote by $G(F)$ the graph obtained by restricting the edge set of G to F , that is, the minor $G \setminus (E \setminus F)$ of G . A minor $\vec{G}/\{e\}$, or $\vec{G} \setminus \{e\}$, for $e \in E$, can be denoted for brevity by \vec{G}/e , or $\vec{G} \setminus e$, respectively. For $e \in E$, we denote by $-_e \vec{G}$ the digraph obtained by reversing the direction of the edge e in \vec{G} .

When connectivity properties and connected components are involved for a digraph, in a construction or a proof, they always concern the undirected underlying graph. In particular, saying that $\vec{G}(A)$ is connected for $A \subseteq E$ means that $G(A)$ is connected and has no isolated vertex in V .

Let G be an ordered (connected) graph and let T be a spanning tree of G . For $t \in T$, the *fundamental cocycle* of t with respect to T , denoted by $C_G^*(T; t)$, or $C^*(T; t)$ for brevity, is the cocycle joining the two connected components of $T \setminus \{t\}$. Equivalently, it is the unique cocycle contained in $(E \setminus T) \cup \{t\}$. For $e \notin T$, the *fundamental cycle* of e with respect to T , denoted by $C_G(T; e)$, or $C(T; e)$ for brevity, is the unique cycle contained in $T \cup \{e\}$.

The technique used in the paper is adapted from oriented matroid techniques, which means in particular that it focuses on edges, whereas vertices are usually not used. Given an orientation \vec{G} of a graph G , we will have to deal with directions of edges in cycles and cocycles of the underlying graph G , and, sometimes, to deal with combinations of cycles or cocycles. To achieve this, it is convenient to use some practical notation and classical properties from oriented matroid theory [2].

A *signed edge subset* is a subset $C \subseteq E$ provided with a partition into a positive part C^+ and a negative part C^- . A cycle, or cocycle, of G provides two opposite signed edge subsets called *signed cycles*, or *signed cocycles* (respectively), of \vec{G} by giving a sign in $\{+, -\}$ to each of its elements with respect to the orientation \vec{G} of G in the natural way. In detail: traversing a cycle in a given direction yields positive signs for edges whose direction agree with the traversing direction, and negative signs for other edges; and considering the partition of the vertex set induced by a cocycle yields positive signs for edges going from one given part to the other one, and negative signs for other edges. In particular, a directed cycle, or a directed cocycle, of \vec{G} corresponds to a signed cycle, or a signed cocycle (respectively), all the elements of which are positive, and to its opposite, all the elements of which are negative. We will often use the same notation C either for a signed edge subset (formally a pair (C^+, C^-) , e.g., a signed cycle) or for the underlying subset $(C^+ \uplus C^-)$, e.g., a graph cycle). Given a spanning tree T of G and an edge $t \in T$, the fundamental cocycle $C^*(T; t)$ induces two opposite signed cocycles of \vec{G} ; then, by convention, the signed fundamental cocycle $C^*(T; t)$ is considered to be the one in which t is positive. Similarly, given an edge $e \notin T$, the signed fundamental cycle $C(T; e)$ is considered to have a positive e . The point of this convention is just to determine which of the two opposite signed cycles/cocycles is the fundamental cycle/cocycle.

The next three tools can be skipped in a first reading, as they will only be used in the proof of the main result of the paper, namely Theorem 4.3. First, let us recall the definition of the *composition* $C \circ D$ between two signed edge subsets as the edge subset $C \cup D$ with signs inherited from C for the element of C and inherited from D for the elements of $D \setminus C$. This operation is extended by associativity to define the composition of a sequence of signed subsets. We will use the classical *orthogonality* property between a cocycle D and a composition C of cycles of \vec{G} , that is: $C \cap D \neq \emptyset$ implies $(C^+ \cap D^+) \cup (C^- \cap D^-) \neq \emptyset$ and $(C^- \cap D^+) \cup (C^+ \cap D^-) \neq \emptyset$.

Second, we recall that, given two cocycles C and C' of \vec{G} , and an element $f \in C \cup C'$ which does not have opposite signs in C and C' , there exists a cocycle D obtained by *elimination between C and C' preserving f* such that $f \in D$, $D^+ \subseteq C^+ \cup C'^+$, $D^- \subseteq C^- \cup C'^-$, and D contains no element of $C \cap C'$ having opposite signs in C and C' . This last property is a strengthening of the oriented matroid elimination property in the particular case of digraphs, a short proof of which is the following. Assume C defines the partition (V_1, V_2) of the set of vertices, and C' defines the partition (V'_1, V'_2) , with a positive sign given to edges from V_1 to V_2 in C and from V'_1 to V'_2 in C' . Then the edges having opposite signs in C and C' are those joining $V_1 \cap V'_2$ and $V_2 \cap V'_1$, then, with $V' = (V_1 \cap V'_2) \cup (V_2 \cap V'_1)$, the cut defined by the partition $(V', E \setminus V')$ contains a cocycle answering the problem.

Third, we recall the following easy property. Let $A, B \subseteq E$ with $A \cap B = \emptyset$, such that the minor $G/B \setminus A$ is connected (or equivalently: $G/B \setminus A$ has the same rank as G/B , that is, the same number of edges for a spanning forest). If D' is a cocycle of $G/B \setminus A$, then there exists a unique cocycle D of G such that $D \cap B = \emptyset$ and $D \setminus A = D'$. If the graphs are directed then D has the same signs as D' on the elements of D' . We say that D' is *induced by D* , or that D *induces D'* . (In particular, when we consider a cocycle D of G inducing a cocycle D' of $G(E')$ for some $E' \subseteq E$, it implies that $D' = D \cap E'$ is a cocycle of $G(E')$, a property which is not true in general for any cocycle D of G .)

2.2. Bipolar orientations and fully optimal spanning trees

We say that a directed graph \vec{G} on the edge set E is *bipolar with respect to $p \in E$* if \vec{G} is acyclic and has a unique source and a unique sink which are the endpoints of p . For brevity, we might write bipolar w.r.t. p . In particular, if \vec{G} consists of a single edge p which is not a loop, then \vec{G} is bipolar w.r.t. p . Equivalently, \vec{G} is bipolar w.r.t. p if and only if G is acyclic and every edge lies on a directed path joining the endpoints of p . Equivalently, \vec{G} is bipolar w.r.t. p if and only if every edge of \vec{G} is contained in a directed cocycle and every directed cocycle contains p (that is, in other words, \vec{G} has dual-orientation-activity 1 and orientation-activity 0, in the sense of [16]; see [11, Section 2.4] for more information). Observe that if \vec{G} is bipolar w.r.t. p , then \vec{G} has no loop, and \vec{G} has no coloop unless p is its only edge.

Definition 2.1. Let $\vec{G} = (V, E)$ be a directed graph, on a linearly ordered set of edges, which is bipolar with respect to the minimal element p of E . The fully optimal spanning tree $\alpha(\vec{G})$ of \vec{G} is the unique spanning tree T of G such that the following sign criterion is satisfied:

- for all $b \in T \setminus p$, the signs of b and $\min(C^*(T; b))$ are opposite in $C^*(T; b)$;
- for all $e \in E \setminus T$, the signs of e and $\min(C(T; e))$ are opposite in $C(T; e)$.

The existence and uniqueness of such a spanning tree, along with the next theorem, are the main results from [7, 9]. Notice that a directed graph and its opposite are mapped onto the same

spanning tree. We say that spanning tree T has *internal activity 1 and external activity 0*, or equivalently that T is *uniactive internal*, if: $\min(E) \in T$; for every $t \in T \setminus \min(E)$ we have $t \neq \min(C^*(T; t))$; and for every $e \in E \setminus T$ we have $e \neq \min(C(T; e))$.

In this paper, it is not necessary to define further the notion of activities of spanning trees, which comes from the theory of the Tutte polynomial (see [7, 11, 12]). For information (not used in the paper), the number of uniactive internal spanning trees of G does not depend on the linear ordering of E and is known as the β -invariant $\beta(G)$ of G [3], while the number of bipolar orientations with respect to p does not depend on p and is equal to $2 \cdot \beta(G)$ [18].

Theorem 2.2 (Key Theorem [7, 9]). *Let G be a graph on a linearly ordered set of edges E with $\min(E) = p$. The mapping $\vec{G} \mapsto \alpha(\vec{G})$ yields a bijection between all bipolar orientations of G with respect to p , with the same fixed orientation for p , and all uniactive internal spanning trees of G .*

The bijection of Theorem 2.2 is called *the uniactive bijection* of the ordered graph G .

For completeness of the paper (though not used thereafter), let us recall that, from the constructive viewpoint, this bijection was built in [7, 9] by the inverse mapping, provided by a single pass algorithm over T and fundamental cocycles, or dually over $E \setminus T$ and fundamental cycles. This algorithm is illustrated in [7, Figure 1], on the same example that we will use in Section 4. Equivalently, the inverse mapping can be obviously built by a single pass over E , choosing edge directions one by one so that the criterion of Definition 2.1 is satisfied. We recall this algorithm below (as done also in [9] and [11, Section 5.2]).

Proposition 2.3 (self-dual reformulation of [7, Proposition 3]). *Let G be a graph on a linearly ordered set of edges $E = \{e_1, \dots, e_n\}_<$. For a uniactive internal spanning tree T of G , the two opposite bipolar orientations of G whose image under α is T are computed by the following algorithm.*

Orient e_1 arbitrarily.

For k from 2 to n do

if $e_k \in T$ then

let $a = \min(C^*(T; e_k))$

orient e_k so that a and e_k have opposite signs in $C^*(T; e_k)$

if $e_k \notin T$ then

let $a = \min(C(T; e_k))$

orient e_k so that a and e_k have opposite signs in $C(T; e_k)$

Lastly, in the proof of the main result Theorem 4.3, we will use the following property of the fully optimal spanning tree, which is given by [7, Proposition 3] or by [9, Proposition 3.3] (and which can be used for an alternative formulation of Definition 2.1). Let $\vec{G} = (V, E)$ be an ordered directed graph, which is bipolar w.r.t. $p = \min(E)$. Let $T = \alpha(\vec{G})$. Recall the convention that an edge has a positive sign in its fundamental cycle or cocycle with respect to a spanning tree. With $T = b_1 < b_2 < \dots < b_r$, and with $E \setminus T = \{c_1 < \dots < c_{n-r}\}$, we have:

- $b_1 = p = \min(E)$;
- for every $1 \leq i \leq r$, all elements of $C^*(T; b_i) \setminus \cup_{j \leq i-1} C^*(T; b_j)$ are positive in $C^*(T; b_i)$;
- for every $1 \leq i \leq n-r$, all elements of $C(T; c_i) \setminus \cup_{j \leq i-1} C(T; c_j)$ are positive in $C(T; c_i)$ except $p = \min(E)$.

Equivalently, as formulated in [9], in terms of compositions of signed subsets (see Section 2.1), the two latter properties can be formulated in the following way: $C^*(T; b_1) \circ \cdots \circ C^*(T; b_r)$ is positive; and $C(T; c_1) \circ \cdots \circ C(T; c_{n-r})$ is positive except on p .

3. Recursive construction by deletion/contraction

This section investigates a recursive deletion/contraction construction of the fully optimal spanning tree of an ordered bipolar digraph. Theorem 3.2 below is also stated in [11, Section 6.1]. For the sake of completeness, we give its (short) proof in both papers, including Lemma 3.1 below. In the present paper, we focus on developing computational remarks from these results (Remarks 3.4, 3.5, 3.6, 3.7 below). This construction is developed further in [11, Section 6] by giving deletion/contraction constructions involving all orientations and spanning trees (see [4, 12] for generalizations to oriented matroids).

Lemma 3.1. *Let \vec{G} be a digraph, on a linearly ordered set of edges E , which is bipolar w.r.t. $p = \min(E)$. Assume $|E| > 1$ and let ω be the greatest element of E . Let $T = \alpha(\vec{G})$. If $\omega \in T$ then \vec{G}/ω is bipolar w.r.t. p and $T \setminus \{\omega\} = \alpha(\vec{G}/\omega)$. If $\omega \notin T$ then $\vec{G} \setminus \omega$ is bipolar w.r.t. p and $T = \alpha(\vec{G} \setminus \omega)$. In particular, we get that \vec{G}/ω is bipolar w.r.t. p or $\vec{G} \setminus \omega$ is bipolar w.r.t. p .*

Proof. First, let us recall that if a spanning tree of a directed graph satisfies the criterion of Definition 2.1, then this directed graph is necessarily bipolar with respect to its smallest edge. This is implied by [7, Propositions 2 and 3], or also stated explicitly in [9, Proposition 3.2], and this is easy to see: if the criterion is satisfied, then the spanning tree is internal uniactive (by definitions of internal/external activities) and the digraph is determined up to reversing all edges (see Proposition 2.3), which implies that the digraph is in the inverse image of T by the uniactive bijection of Theorem 2.2 and that it is bipolar with respect to its smallest edge.

Notice first that ω is not a loop nor a coloop of \vec{G} (otherwise \vec{G} would not be bipolar).

Assume that $\omega \in T$. Obviously, the fundamental cocycle of $b \in T \setminus \{\omega\}$ w.r.t. $T \setminus \{\omega\}$ in G/ω is the same as the fundamental cocycle of b w.r.t. T in G . And the fundamental cycle of $e \notin T$ w.r.t. $T \setminus \{\omega\}$ in G/ω is obtained by removing ω from the fundamental cycle of e w.r.t. T in G . Hence, those fundamental cycles and cocycles in G/ω satisfy the criterion of Definition 2.1, hence \vec{G}/ω is bipolar w.r.t. p and $T \setminus \{\omega\} = \alpha(\vec{G}/\omega)$.

Similarly (dually in fact), assume that $\omega \notin T$. The fundamental cocycle of $b \in T$ w.r.t. $T \setminus \{\omega\}$ in $G \setminus \omega$ is obtained by removing ω from the fundamental cocycle of b w.r.t. T in G . And the fundamental cycle of $e \notin T \setminus \{\omega\}$ w.r.t. $T \setminus \{\omega\}$ in $G \setminus \omega$ is the same as the fundamental cycle of e w.r.t. T in G . Hence, those fundamental cycles and cocycles in $G \setminus \omega$ satisfy the criterion of Definition 2.1, hence $\vec{G} \setminus \omega$ is bipolar w.r.t. p and $T \setminus \{\omega\} = \alpha(\vec{G} \setminus \omega)$. \square

Theorem 3.2. *Let \vec{G} be a digraph, on a linearly ordered set of edges E , which is bipolar with respect to $p = \min(E)$. The fully optimal spanning tree $\alpha(\vec{G})$ of \vec{G} satisfies the following inductive definition, where $\omega = \max(E)$.*

If $|E| = 1$ then $\alpha(\vec{G}) = \omega$.

If $|E| > 1$ then:

If \vec{G}/ω is bipolar w.r.t. p but not $\vec{G} \setminus \omega$ then $\alpha(\vec{G}) = \alpha(\vec{G}/\omega) \cup \{\omega\}$.

If $\vec{G} \setminus \omega$ is bipolar w.r.t. p but not \vec{G}/ω then $\alpha(\vec{G}) = \alpha(\vec{G} \setminus \omega)$.

If both $\vec{G} \setminus \omega$ and \vec{G}/ω are bipolar w.r.t. p then:

let $T' = \alpha(\vec{G} \setminus \omega)$, $C = C_{\vec{G}}(T'; \omega)$ and $e = \min(C)$;
if e and ω have opposite directions in C then $\alpha(\vec{G}) = \alpha(\vec{G} \setminus \omega)$;
if e and ω have the same directions in C then $\alpha(\vec{G}) = \alpha(\vec{G}/\omega) \cup \{\omega\}$.

or equivalently:

let $T'' = \alpha(\vec{G}/\omega)$, $D = C_{\vec{G}}^*(T'' \cup \omega; \omega)$ and $e = \min(D)$;
if e and ω have opposite directions in D then $\alpha(\vec{G}) = \alpha(\vec{G}/\omega) \cup \{\omega\}$;
if e and ω have the same directions in D then $\alpha(\vec{G}) = \alpha(\vec{G} \setminus \omega)$.

Proof. By Lemma 3.1, at least one minor among $\{\vec{G}/\omega, \vec{G} \setminus \omega\}$ is bipolar w.r.t. p . If exactly one minor among $\{\vec{G}/\omega, \vec{G} \setminus \omega\}$ is bipolar w.r.t. p , then by Lemma 3.1 again, the above definition is implied. Assume now that both minors are bipolar w.r.t. p .

Consider $T' = \alpha(\vec{G} \setminus \omega)$. Fundamental cocycles of elements in T' w.r.t. T' in \vec{G} are obtained by removing ω from those in $\vec{G} \setminus \omega$. Hence they satisfy the criterion from Definition 2.1. Fundamental cycles of elements in $E \setminus (T' \cup \{\omega\})$ w.r.t. T' in \vec{G} are the same as in $\vec{G} \setminus \omega$. Hence they satisfy the criterion from Definition 2.1. Let C be the fundamental cycle of ω w.r.t. T' . If e and ω have opposite directions in C , then C satisfies the criterion from Definition 2.1, and $\alpha(\vec{G}) = T'$. Otherwise, we have $\alpha(\vec{G}) \neq T'$, and, by Lemma 3.1, we must have $\alpha(\vec{G}) = \alpha(\vec{G}/\omega) \cup \{\omega\}$.

The second condition involving $T'' = \alpha(\vec{G}/\omega)$ is proved in the same manner. Since it yields the same mapping α , then this second condition is actually equivalent to the first one, and so it can be used as an alternative. Note that the fact that these two conditions are equivalent is difficult and proved here in an indirect way (actually this fact is equivalent to the key result that α yields a bijection); see Remark 3.7. \square

Corollary 3.3. *Using the notation of Theorem 3.2, if $-\omega \vec{G}$ is bipolar w.r.t. p then the above algorithm of Theorem 3.2 builds at the same time $\alpha(\vec{G})$ and $\alpha(-\omega \vec{G})$, and we have:*

$$\left\{ \alpha(\vec{G}), \alpha(-\omega \vec{G}) \right\} = \left\{ \alpha(\vec{G} \setminus \omega), \alpha(\vec{G}/\omega) \cup \{\omega\} \right\}.$$

Also, we have that $-\omega \vec{G}$ is bipolar w.r.t. p if and only if $\vec{G} \setminus \omega$ and \vec{G}/ω are bipolar w.r.t. p .

Proof. Direct by Theorem 3.2 and Theorem 2.2 (bijection property). \square

Remark 3.4 (exponential computational complexity for one image). The algorithm of Theorem 3.2 is exponential time, as it may involve an exponential number of minors. Indeed, in general, one needs to compute both $\alpha(\vec{G} \setminus \omega)$ and $\alpha(\vec{G}/\omega)$ in order to compute $\alpha(\vec{G})$ (because one might compute $T' = \alpha(\vec{G} \setminus \omega)$ and finally set $\alpha(\vec{G}) = \alpha(\vec{G}/\omega) \cup \{\omega\}$, or equivalently one might compute $T'' = \alpha(\vec{G}/\omega)$ and finally set $\alpha(\vec{G}) = \alpha(\vec{G} \setminus \omega)$). Hence, in general, one may need to compute $\alpha(\vec{G} \setminus \omega \setminus \omega')$, $\alpha(\vec{G} \setminus \omega/\omega')$, $\alpha(\vec{G}/\omega \setminus \omega')$ and $\alpha(\vec{G}/\omega/\omega')$, with $\omega' = \max(E \setminus \{\omega\})$, and so on... Finally, with $|E| = n$, the number of calls to the algorithm to build $\alpha(\vec{G})$ is $O(2^0 + 2^1 + \dots + 2^{n-1})$, that is, $O(2^n)$. In contrast, the algorithm provided in Section 4 involves a linear number of minors (and yields a polynomial time algorithm; see Theorem 4.10). However, the algorithm of Theorem 3.2 is efficient in terms of computational complexity for building the images of all bipolar orientations of G at once, see Remark 3.5 below.

Remark 3.5 (efficient computational complexity for building the whole bijection at once). Let us explain how the algorithm of Theorem 3.2 allows to build the $O(2^n)$ images of all bipolar orientations in just $O(n \cdot 2^n)$ steps. By Corollary 3.3, the construction of Theorem 3.2 can be used to build the whole active bijection for G (i.e., the 1 – 1 correspondence between all bipolar orientations of G w.r.t. p with fixed orientation, and all spanning trees of G with internal activity 1 and external activity 0), from the whole active bijections for G/ω and $G \setminus \omega$. For each pair of bipolar orientations $\{\vec{G}, -\omega \vec{G}\}$, the algorithm provides which “choice” is right to associate one orientation with the orientation induced in G/ω and the other with the orientation induced in $G \setminus \omega$. We mention that this “choice” notion is developed further in [11, Section 6] (and [14]) as the basic component for a deletion/contraction framework. This ability to build the whole bijection at once is interesting from a structural viewpoint, but also from a computational complexity viewpoint. Indeed, with $|E| = n$, the number of calls to the algorithm to build one image is $O(2^n)$ (see Remark 3.4), but the number of calls to the algorithm to build the $O(2^n)$ images of all bipolar orientations in the above way is just $O(n \cdot 2^n)$. In detail, by the choice process described above, in order to handle all (bipolar) orientations of G , one considers (at most) all orientations of G/ω and $G \setminus \omega$, that is, $2^{n-1} \times 2$ digraphs. To this end, one considers (at most) all orientations of $G \setminus \omega \setminus \omega'$, $G \setminus \omega / \omega'$, $G / \omega \setminus \omega'$ and $G / \omega / \omega'$, with $\omega' = \max(E \setminus \{\omega\})$, that is, $2^{n-2} \times 4$ digraphs, and so on... So, with $E = e_1 < \dots < e_n$, the choice process described above amounts to consider (at most) every orientation of every graph of the form $G/A \setminus A'$ where $A \cap A' = \emptyset$ and $A \cup A' = \{e_i, \dots, e_n\}$ for $2 \leq i \leq n$, that is, $2^{n-i+1} \times 2^{i-1}$ digraphs, for $2 \leq i \leq n$. Globally, we thus use $O(2^{n-1} \times 2^1 + 2^{n-2} \times 2^2 + \dots + 2^2 \times 2^{n-2} + 2^1 \times 2^{n-1}) = O(n \cdot 2^n)$ elementary steps to build the $O(2^n)$ images.

Remark 3.6 (connection with linear programming). As the full optimality notion strengthens linear programming optimality (see Section 1 and Remark 4.7), the deletion/contraction algorithm of Theorem 3.2 corresponds to a refinement of the classical linear programming solving by constraint/variable deletion; see [6, 13].

Remark 3.7 (non-trivial equivalence in Theorem 3.2). The equivalence of the two formulations in the algorithm of Theorem 3.2 is a deep result, difficult to prove if no property of the computed mapping is known. Here, to prove it, we implicitly use that α is already well-defined by Definition 2.1, and bijective for bipolar orientations (Key Theorem 2.2). But if one defines a mapping α from scratch as in the algorithm (with either one of the two formulations) and then investigates its properties, then it turns out that the above equivalence result is equivalent in some way to the existence and uniqueness of the fully optimal spanning tree as defined in Definition 2.1. See [14] for details.

4. Direct computation by optimization

This section gives a direct and efficient construction of the fully optimal spanning tree of an ordered bipolar digraph, in terms of an optimization based on a linear ordering of cocycles in a minor for each edge of the resulting spanning tree (Theorem 4.3). It is completely independent of Section 3. This construction is an adaptation for graphs of a general construction by elaborations on linear programming detailed for oriented matroids in [12] (see Section 1 and Remark 4.7). In this paper, a translation of the notion of optimal cocycle in terms of linear programming is only used in the proof of Lemma 4.9 to ensure that a polynomial time algorithm can be used. In this

way, the computation of the fully optimal spanning tree can be performed in polynomial time (Theorem 4.10). The question of relating the present setting to a usual setting in the digraph field is addressed in Remark 4.11.

Lemma 4.1. *Let G be a graph. Given a spanning tree T of G and $U \subseteq T$, there exists at most one cocycle C of G such that $C \cap T = U$. Given a spanning tree T of G and two cocycles C and D of G , there exists $f \in T$ belonging to $C \Delta D$.*

Proof. Let us prove the first claim. Assume such a cocycle C exists. Then it is defined by a partition of the set of vertices of G into two sets A and B , such that $G[A]$ and $G[B]$ are connected (since C is minimal by inclusion). Each connected component of $T \setminus U$ is contained either in $G[A]$ or in $G[B]$. On the other hand, if an edge $e = (x, y)$ is in U then it is in C and has an endpoint in $G[A]$ and the other in $G[B]$. Hence, as T is spanning, the partition into the sets A and B is completely determined by T and U , which implies the uniqueness of C . Now, let us prove the second claim. Assume $T \cap (C \Delta D) = \emptyset$, then we have $T \cap C \subseteq D$ and $T \cap D \subseteq C$, so $T \cap C = T \cap D = T \cap C \cap D$, which is a contradiction with the first claim for $U = T \cap C = T \cap D$. \square

Definition 4.2. We define an *optimizable digraph* to be a connected digraph $\vec{G} = (V, E \cup F)$, possibly with $E \cap F \neq \emptyset$, given with:

- an edge $p \in E \setminus F$ called the *infinity edge*;
- a subset E of edges called the *ground set*, such that the digraph $\vec{G}(E)$ is connected and acyclic;
- a linearly ordered subset F of edges called the *objective set*, such that $F \cup \{p\}$ is a spanning tree of G .

Given an optimizable digraph \vec{G} as defined above, we define a *linear ordering for the signed cocycles of \vec{G} containing p* as follows. Let C and D be two signed cocycles (see Section 2.1) of \vec{G} containing p . By Lemma 4.1, since $F \cup \{p\}$ is a spanning tree of G , there exists an element of F that belongs to $C \Delta D$. Let f be the smallest element of F such that either f belongs to $C \Delta D$, or f belongs to $C \cap D$ and has opposite signs in C and D . If f is positive in C or negative in D then set $C > D$. If f is negative in C or positive in D then set $D > C$.

Equivalently, we set $C > D$ if f is a positive element of C and not an element of D , or f is a positive element of C and a negative element of D , or f is not an element of C and a negative element of D , where f is the smallest possible element of F that allows for setting $C < D$ or $D < C$ in this way.

Equivalently, it is easy to see that this ordering is provided by the weight function w on signed cocycles of G defined as follows. Let us denote the elements of F by $f_2 < \dots < f_r$, and let us denote by f_i or \bar{f}_i , for $2 \leq i \leq r$, the element with a positive or negative sign, respectively. For $2 \leq i \leq r$, set $w(f_i) = 2^{r-i}$ and $w(\bar{f}_i) = -2^{r-i}$, and set $w(e) = 0$ if $e \in E \setminus F$. Then define the weight $w(C)$ of a signed cocycle C as the sum of weights of its elements. The above linear ordering is given by: $C > D$ if and only if $w(C) > w(D)$.

We define the *optimal cocycle* of \vec{G} as the maximal signed cocycle of \vec{G} containing p with positive sign, and inducing a directed cocycle of $\vec{G}(E)$. It exists since $\vec{G}(E)$ is acyclic, and it is unique since the ordering is linear.

Theorem 4.3. *Let $\vec{G} = (V, E)$ be an ordered bipolar digraph with respect to $p = \min(E)$. The fully optimal spanning tree $\alpha(\vec{G}) = p < t_2 < \dots < t_r$ is computed by the following algorithm.*

(The two comments inserted below state important properties and help to follow what is built in the algorithm.)

(1) Initialize \vec{G} as the optimizable digraph given by:

- the infinity edge p
- the ground set E
- the objective set $F = f_2 < \dots < f_r$ where $p < f_2 < \dots < f_r$ is the smallest lexicographic spanning tree of \vec{G} (and the linear ordering on F is induced by the linear ordering on E).

(2) For i from 2 to r do:

(2.1) Let C_{opt} be the optimal cocycle of \vec{G} .

First comment/property: C_{opt} is the cocycle induced in the current digraph \vec{G} by the fundamental cocycle of the element t_{i-1} (computed at the previous step, $t_1 = p$) w.r.t. the fully optimal spanning tree of the initial digraph.

(2.2) Let

$$t_i = \min(E \setminus C_{opt})$$

Second comment/property: the i -th edge t_i of $\alpha(\vec{G})$ is the smallest edge not contained in fundamental cocycles of smaller edges w.r.t. $\alpha(\vec{G})$ (this is because the spanning tree $\alpha(\vec{G})$ is internal uniactive).

(2.3) Let

$$p' = t_i$$

(2.4) Let

$$E' = E \setminus C_{opt}$$

(2.5) If $p' \in F$ then let $F' = F \setminus \{p'\}$, and if $p' \notin F$ then let:

$$F' = F \setminus \max \left(F \cap C_G(F \cup \{p\}; p') \right).$$

Equivalently, F' is obtained by removing from F the greatest possible element such that the following property holds:

$p' \notin F'$ and $F' \cup \{p'\}$ is a spanning tree of the minor \vec{G}' defined below.

(2.6) Set

$$\vec{G}' = \vec{G} (E' \cup F' \cup \{p\}) / \{p\}$$

as the optimizable digraph given by:

- the infinity edge p'
- the ground set E'
- the objective set F'

(2.7) Update $\vec{G} \leftarrow \vec{G}'$; $p \leftarrow p'$; $E \leftarrow E'$; $F \leftarrow F'$.

(3) Output

$$\alpha(\vec{G}) = p < t_2 < \dots < t_r.$$

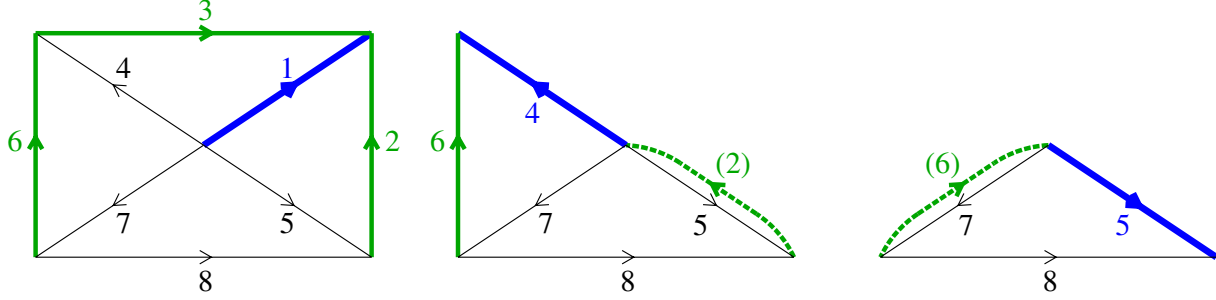


Figure 1: illustration for Example 4.4. The digraph $\vec{G} = \vec{G}_1$ is on the left, the digraph $\vec{G}_2 = \vec{G}_1/1\setminus 3$ is in the middle, and the digraph $\vec{G}_3 = \vec{G}_2/4\setminus 2$ is on the right. At each step: the boldest edge is p , the other bold edges form the set F , and the dashed edges are in F but no more in E (their labels are in brackets).

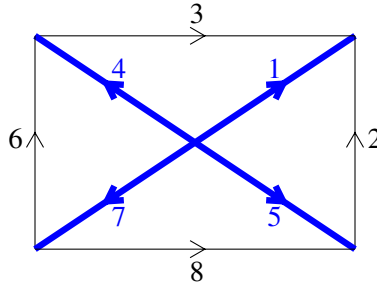


Figure 2: the fully optimal spanning tree $\alpha(\vec{G}) = 1457$ of the digraph \vec{G} of Figure 1 (the same as in [7]).

Example 4.4. Theorem 4.3 might seem rather technical in a pure graph setting, so let us first illustrate it on an example before proving it. Let us apply the algorithm of Theorem 4.3 to the same example as in [7], where it illustrated the inverse algorithm. The steps of the algorithm are depicted on Figure 1. The output is depicted on Figure 2. One can keep in mind that the successive optimal cocycles built in the algorithm correspond to the fundamental cocycles with respect to the successive edges of the fully optimal spanning tree (cf. first comment/property in the algorithm; see details in Property (P2_i) in the proof of Theorem 4.3 below).

— *Computation of t_2 .* Initially $\vec{G} = \vec{G}_1$ is a digraph with set of edges $E = \{1 < 2 < 3 < 4 < 5 < 6 < 7 < 8\}$. The minimal spanning tree is $\{1 < 2 < 3 < 6\}$. Hence $p = 1$ and $F = 236$. The linearly ordered directed cocycles of \vec{G}_1 containing $p = 1$ are: $123 > 1246 > 1358 > 14568 > 1457$. The maximal is $C_{\text{opt}} = 123$ (which is equal, at this first step, to the smallest directed cocycle for the lexicographic ordering; see Proposition 4.12 below). We get $t_2 = \min(E \setminus C_{\text{opt}}) = 4$.

— *Computation of t_3 .* We now consider the optimizable digraph $\vec{G}_2 = \vec{G}_1/\{1\}\setminus\{3\}$ given with $p = 4$, $E = 45678$, and $F = 26$ (the edge 3 is deleted as the greatest belonging to the circuit 134 and to the previous $F = 236$). The linearly ordered signed cocycles of \vec{G}_2 positive on E and containing $p = 4$ are: $46 > \bar{2}457$ (where the bar upon elements represents negative elements). The maximal is $C_{\text{opt}} = 46$. We get $t_3 = \min(E \setminus C_{\text{opt}}) = 5$.

— *Computation of t_4 .* We now consider the optimizable digraph $\vec{G}_3 = \vec{G}_2/\{4\}\setminus\{2\}$ given with $p = 5$, $E = 578$, and $F = 6$ (the edge 2 is deleted as the greatest belonging to the circuit 25 and to the previous $F = 26$). The linearly ordered signed cocycles of \vec{G}_3 positive on E and containing $p = 5$ are: $58 > 5\bar{6}7$. The maximal is $C_{\text{opt}} = 58$. We get $t_4 = \min(E \setminus C_{\text{opt}}) = 7$.

— *Output.* We get finally $\alpha(\vec{G}) = 1457$ (and one can check that the fundamental cocycles $C^*(1457; 1) = 123$, $C^*(1457; 4) = \bar{3}46$, and $C^*(1457; 5) = \bar{2}58$ induce the successive optimal cocycles 123, 46 and 58 in the successive considered minors).

Notation for what follows. The proof of Theorem 4.3 is given below, after two lemmas. In all this framework, we will use the following notation. We denote by \vec{G} , \vec{G}' , etc., the variables as they are used during the algorithm, meaning they are considered at any given step of the algorithm with their current value. We denote by \vec{G}_1 the initial optimizable digraph \vec{G} , and \vec{G}_i the variable optimizable digraph \vec{G} updated at step (2.7) with respect to variable i , with parameters $p_i = t_i$ as p , E_i as E and F_i as F , for all $1 \leq i \leq r$.

Lemma 4.5. *The algorithm of Theorem 4.3 is well-defined.*

Proof. We shall prove by induction that, at the beginning of each stage, the digraph \vec{G} is well-defined as an optimizable digraph. Initially, the optimizable digraph $\vec{G} = \vec{G}_1$ is obviously well defined. One has thus to check that the optimizable digraph \vec{G}' defined at each call to step (2.6) is well-defined, assuming by the induction hypothesis that \vec{G} is an optimizable digraph. First, $\vec{G}(E)$ is connected, and C_{opt} is a cocycle of \vec{G} inducing a cocycle of $\vec{G}(E)$. Also, we have $p \in C_{\text{opt}}$ by definition of C_{opt} , and we have $E' = E \setminus C_{\text{opt}}$ by step (2.4). Hence p is the unique edge in $E' \cup \{p\}$ joining the two connected components of $\vec{G}(E')$. Hence $\vec{G}(E' \cup \{p\})/\{p\}$ is connected, and so is $\vec{G}'(E')$ (implying that \vec{G}' is also connected as they have the same vertices).

By definition at steps (2.2)(2.3)(2.4), we have $p' = \min(E')$, and by definition of F' at step (2.5), we have $p' \notin F$, hence $p' \in E' \setminus F'$, as required for an optimizable digraph. Since $F \cup \{p\}$ is a spanning tree of \vec{G} , then F is a spanning tree of \vec{G}' , and, by definition of F' at step (2.5), $F' \cup \{p'\}$ is a spanning tree of \vec{G}' , as required for an optimizable digraph. By assumption, at each call to step (2.6), the digraph $\vec{G}(E)$ is acyclic. Since C_{opt} is a cocycle of \vec{G} with $p \in C_{\text{opt}}$ as above, then $\vec{G}(E' \cup \{p\})$ is also acyclic, and so is $\vec{G}'(E')$, as required for an optimizable digraph. \square

Lemma 4.6. *At any step of the algorithm of Theorem 4.3, the following invariant is maintained: the smallest element of every cocycle of G belongs to $F \cup \{p\}$, and this element is equal to the smallest element of the cocycle of G_1 inducing this cocycle (in the minor G of G_1 ; see Section 2.1).*

Proof. The property is true at the initial step since $F \cup \{p\} = F_1 \cup \{p_1\}$ is defined as the smallest spanning tree of $G = G_1$. Assume it is true for G , we want it true also for G' as defined at step (2.6). Let D' be a cocycle of G' . By construction, $G' = G \setminus A/\{p\}$ for some set A such that G' is connected, so there exists a cocycle D of G inducing D' , such that $p \notin D$ and $D \setminus A = D'$ (see Section 2.1). Since $F \cap A = F \setminus (E' \cup F' \cup \{p\})$ by definition of G' , and $F \setminus F'$ contains exactly one element f by definition of F' at step (2.5), then $F \cap A$ contains at most this element f . By the induction hypothesis, we have $\min(D) \in F$. By definition of D , we have $D \setminus A = D'$. Assume for a contradiction that $\min(D') \neq \min(D)$. Then $\min(D) \in A \cap F$, implying $F \cap A = \{f\}$ and $\min(D) = f$. There are two cases for defining f at step (2.5). In the first case, we have $f = p' = \min(E') \in F$, which implies $f \in E'$ and which contradicts $\{f\} = F \cap A = F \setminus (E' \cup F' \cup \{p\})$. In the second case, f is defined as the greatest element of the unique cycle C of G contained in $F \cup \{p, p'\}$. In this case, let $f' \in D \cap C$. We have $f' \leq f$ since $f' \in C$ and the greatest element of C is f . And we have $f' \geq f$ since $f' \in D$ and $\min(D) = f$. Hence $f' = f$, and we have proved $D \cap C = \{f\}$, which contradicts the orthogonality of the cycle C and the cocycle D (see Section 2.1).

So we have $\min(D') = \min(D)$. Since the cocycle D_1 of G_1 inducing D in G also induces D' in G' , and since $\min(D) = \min(D_1)$ by the induction hypothesis, we get $\min(D') = \min(D_1)$.

Finally, if $\min(D') \notin F'$, since $\min(D') = \min(D) \in F$, then $\min(D') = f$. As above, there are two cases for defining f at step (2.5). In the first case, we have $f = p' = \min(E')$, hence $\min(D') = f \in F' \cup \{p'\}$, which is the property that we have to prove. In the second case, we have $\min(D) = f$ and the same argument as above leads again to $C \cap D = \{f\}$ and to the same contradiction. The invariant is now proved. \square

Proof of Theorem 4.3. The present proof is a condensed but complete version for graphs of the general geometric proof that will be given in [13], making use of the linear ordering of cocycles defined above (which is possible in graphs but not in general; see Remark 4.7). We will extensively use the notion of cocycle induced in a minor of a graph by a cocycle of this graph; see Section 2.1. Since \vec{G}_1 is bipolar, its fully optimal spanning tree $\alpha(\vec{G}_1)$ exists and satisfies the properties recalled in Section 2.2. By Lemma 4.5, the algorithm given in the theorem statement is well defined. Now, we have to prove that $\alpha(\vec{G}_1)$ is necessarily equal to the output of this algorithm. The proof consists of proving by induction that, for every $2 \leq i \leq r$, the two following properties (P1_i) and (P2_i) are true. The property (P1_i) for $2 \leq i \leq r$ means that the algorithm actually returns $\alpha(\vec{G}_1)$. The property (P2_i) for $2 \leq i \leq r$ means that the optimal cocycles computed in the successive minors are actually induced by the fundamental cocycles of the fully optimal spanning tree $\alpha(\vec{G}_1)$. (This is the first comment/property in the algorithm statement.) Then, the proof is not long but rather technical. Let us denote the fully optimal spanning tree of \vec{G}_1 by $T = \alpha(\vec{G}_1) = \{b_1 < \dots < b_r\}$. We already know that $b_1 = \min(E)$.

- (P1_i) We have $b_i = t_i$, where t_i is defined at step (2.2) of the algorithm and b_i is the i -th element of $T = \alpha(\vec{G}_1)$.
- (P2_i) The optimal cocycle C_{opt} of \vec{G}_{i-1} , defined at step (2.1) of the algorithm equals the cocycle denoted by C_{i-1} of \vec{G}_{i-1} induced by the fundamental cocycle $C^*(T; b_{i-1})$ of $p_{i-1} = t_{i-1} = b_{i-1}$ w.r.t. T in \vec{G}_1 , that is: $C_{i-1} = C^*(T; b_{i-1}) \cap (E_{i-1} \cup F_{i-1})$, where $(E_{i-1} \cup F_{i-1})$ is the edge set of \vec{G}_{i-1} .

First, observe that C_{i-1} is a well defined induced cocycle in property (P2_i) as soon as (P1_j) is true for all $j < i - 1$ (implying $t_j = b_j$), since $b_j \notin C^*(T; b_{i-1})$ for $j < i - 1$ and $\vec{G}_{i-1} = \vec{G}_1 / \{p_1, t_2, \dots, t_{i-2}\} \setminus A$ for some A .

Second, let $2 \leq k \leq r$. Assume that the property (P2_i) is true for all $2 \leq i \leq k$ and the property (P1_i) is true for all $2 \leq i \leq k - 1$. Then we directly have that the property (P1_i) is also true for $i = k$. Indeed, as shown in [7, Proposition 2] (that can be proved easily), the fact that the spanning tree $T = b_1 < b_2 < \dots < b_r$ has external activity 0 implies that, for all $1 \leq i \leq r$, we have $b_i = \min(E \setminus \cup_{j \leq i-1} C^*(T; b_j))$. (This is the second comment/property in the algorithm statement.) So we have $b_k = \min(E_{k-1} \setminus C_{k-1}) = \dots = \min(E_1 \setminus \cup_{j \leq k-1} C_j) = \min(E_1 \setminus \cup_{j \leq k-1} C^*(T; b_j))$, and so we have $t_k = b_k$.

Now, it remains to prove that, under the above assumption, the property (P2_i) is true for $i = k + 1$. Consider C_{opt} , denoting the optimal cocycle of \vec{G}_k , and C_k , denoting the cocycle of \vec{G}_k induced by the fundamental cocycle of $p_k = t_k = b_k$ w.r.t. T in \vec{G}_1 . Assume for a contradiction that $C_{opt} \neq C_k$.

Since properties (P1_{*i*}) and (P2_{*i*}) are true for all $2 \leq i \leq k$ by assumption, and reformulating definition of \vec{G}' at step (2.6), we have:

$$\begin{aligned}\vec{G}_k &= \vec{G}_{k-1}/\{b_{k-1}\} \setminus \left(C_{k-1} \setminus (F_k \cup \{b_{k-1}\}) \right) \\ &= \vec{G}_{k-1}/\{b_{k-1}\} \setminus \left(C^*(T; b_{k-1}) \setminus (F_k \cup \{b_{k-1}\}) \right)\end{aligned}$$

and hence, inductively, we have:

$$\vec{G}_k = \vec{G}_1/\{b_1, \dots, b_{k-1}\} \setminus A$$

where A is the union of all fundamental cocycles of b_i w.r.t. T for $1 \leq i \leq k-1$ minus $F_k \cup \{b_1, \dots, b_{k-1}\}$. That is:

$$A = \left(\bigcup_{1 \leq i \leq k-1} C^*(T; b_i) \right) \setminus \left(F_k \cup \{b_1, \dots, b_{k-1}\} \right).$$

In particular, we have $C_k = C^*(T; b_k) \setminus A$. And we also have $A \cap T = \emptyset$.

As recalled in Section 2.2, the definition of T implies that C_k is positive on E_k , that is, positive except maybe on elements of $F_k \setminus E_k$, just the same as C_{opt} . By assumption and property (P1), we have $b_k = t_k = \min(E_k)$. Then, by definition of $C_k = C^*(T; b_k) \cap (E_k \cup F_k)$, we have $b_k \in C_k$. Hence, the cocycle C_k has been taken into account in the linear ordering of cocycles of the optimizable digraph \vec{G}_k defining C_{opt} . So $C_{\text{opt}} > C_k$ in this linear ordering by definition of C_{opt} .

By the definition of this linear ordering, we may choose f to be the smallest element of F_k with the property of being positive in C_{opt} and not belonging to C_k , or positive in C_{opt} and negative in C_k , or not belonging to C_{opt} and negative in C_k . The edge f does not have opposite signs in C_{opt} and $-C_k$. So let D' be a cocycle of \vec{G}_k obtained by elimination between C_{opt} and $-C_k$ preserving f (see Section 2.1). Necessarily, f is positive in D' by the definition of f . By the definitions of C_{opt} and C_k , the element b_k is positive in C_{opt} and in C_k . Moreover, by the definition of f , every edge in F smaller than f belonging to C_{opt} (or to C_k) also belongs to C_k (or to C_{opt} , respectively) with the same sign. Hence, by properties of elimination, the cocycle D' does not contain b_k nor any element of F smaller than f belonging to C_{opt} or C_k . Since the smallest edge in D' belongs to $F \cup \{b_k\}$, as shown by the invariant of Lemma 4.6, and since we have $b_k \notin D'$, then we have $\min(D') = f$. The negative elements of D' are either elements of $F_k \setminus E_k$, or elements of $C_k \setminus \{b_k\}$. In the first case, since $E_k = E_1 \setminus A$, the negative elements belong to $F_k \cap A \subseteq A \subseteq E_1 \setminus T$. In the second case, the negative elements also belong to $E_1 \setminus T$ by definition of a fundamental cocycle. Finally, let D be the cocycle of \vec{G}_1 inducing D' in \vec{G}_k , such that $D \cap \{b_1, \dots, b_k\} = \emptyset$ and $D \setminus A = D'$. The negative elements of D belong to A or are negative in D' , then, in every case, they belong to $E_1 \setminus T$. Moreover, as shown by the invariant of Lemma 4.6, we have $\min(D) = \min(D') = f$.

So we have built a cocycle D such that:

- (i) $D \cap \{b_1, \dots, b_k\} = \emptyset$
- (ii) $\min(D) = f$
- (iii) f is positive in D
- (iv) the negative elements of D are in $E_1 \setminus T$

In a first case, we assume that $f \notin T$. Let us use the notation $E_1 \setminus T = \{c_1 < \dots < c_{n-r}\}$, as at the end of Section 2. Then $f = c_j$ for some $c_j \in E_1 \setminus T$. Let $C = C(T; c_1) \circ \dots \circ C(T; c_j)$. As

recalled at the end of Section 2, this composition of cycles has only positive elements except the first one $p_1 = b_1$. The edge f is positive in C and D , hence, by orthogonality (see Section 2.1), there exists an edge $e \in C \cap D$ with opposite signs in C and D . We have $b_1 \notin D$, hence $e \neq b_1$, and hence e is positive in C and negative in D . Hence $e \in E_1 \setminus T$. Since $e \in C$, we must have $e = c_i$ for some $i \leq j$, that is, $e \leq f$. But $f = \min(D)$ implies $e = f$, which is a contradiction.

In a second case, we assume that $f \in T$. Let $a = \min(C^*(T; f))$. Since $f \in F$, we have $f \neq \min(E_1)$, and so $f \neq a$ by definition of T , so we have $a \notin T$. Then $a = c_j$ for some $c_j \in E_1 \setminus T$. Let $C = C(T; c_1) \circ \dots \circ C(T; c_j)$, which has only positive elements except the first one $p_1 = b_1$. Since $a \in C^*(T; f)$, we have $f \in C(T; c_j)$. As above, by orthogonality, the edge f positive in C and D , together with $p_1 \notin D$, implies an edge $e \in C \cap D$ positive in C and negative in D . So $e \in E_1 \setminus T$, and so $e = c_i$ for some $i \leq j$, so $e \leq a$, which implies $e \leq f$ by definition of a , leading to the same contradiction as above with $f = \min(D)$.

We obtained a contradiction after assuming that $C_{\text{opt}} \neq C_k$. So we proved $C_{\text{opt}} = C_k$, which ends the proof of property (P2_{*i*}) for $i = k + 1$, and ends the proof of properties (P1_{*i*}) and (P2_{*i*}) by induction for all $2 \leq i \leq r$. \square

Remark 4.7 (linear programming). As mentioned in Section 1, the algorithm of Theorem 4.3 can be seen as an elaboration on linear programming, available more generally in oriented matroids or real hyperplane arrangements. In this setting, we actually optimize a sequence of nested faces (corresponding to the sequence of fundamental cocycles of the fully optimal spanning tree, that is, to the sequence of optimal cocycles in the above algorithm, a process that we call flag programming), each with respect to a sequence of objective functions (the linearly ordered objective set in the above algorithm, a process called lexicographic multiobjective programming, see Lemma 4.9). This yields a unique fully optimal basis for any bounded region. This refines standard linear programming where just one vertex is optimized with respect to just one objective function. However, this can be computed inductively using standard pseudo/real linear programming. See Section 1 for references.

Lemma 4.8. *For each optimizable digraph \vec{G} considered in the algorithm of Theorem 4.3, the following property is satisfied. If C is a signed cocycle of \vec{G} , such that C induces a positive directed cocycle of $\vec{G}(E)$ and such that $p \notin C$, then $\min(F \cap C)$ has a negative sign in C .*

Proof. We prove the result by induction. The result is trivially true for the initial $\vec{G} = \vec{G}_1$ as it is a bipolar digraph w.r.t. p whose edge-set is E , hence all its positive directed cocycles contain p . Now we assume that the result is true for \vec{G} , and we prove that it is true for \vec{G}' defined at step (2.6). Assume for a contradiction that C' is a signed cocycle of \vec{G}' , such that C' induces a positive directed cocycle of $\vec{G}'(E')$, $p' \notin C'$, and $f' = \min(F' \cap C')$ is positive in C' . Let C be the cocycle of \vec{G} inducing C' in the minor \vec{G}' of \vec{G} (defined at the end of Section 2.1).

Let us first prove that $f' = \min(F \cap C)$. By the definitions of \vec{G}' and C , we have $p \notin C$ and $C' = C \cap (E' \cup F')$, so $C' \cap F' = C \cap F'$. By the definition of F' at step (2.5), we have either $F' = F \setminus \{p'\}$ or $F' = F \setminus \{f\}$ where f is the greatest element of the unique cycle $C(F \cup \{p\}; p')$ contained in $F \cup \{p, p'\}$. In the first case, since $p' \notin C'$, we have $C' \cap F' = C \cap F$, hence $f' = \min(F \cap C)$. In the second case, we have $C' \cap F' = (C \cap F) \setminus \{f\}$. This implies $f' = \min(C \cap F)$ unless $f = \min(C \cap F)$. In the latter case, consider an element $e \in C \cap C(F \cup \{p\}; p')$. Then $e \leq f$ by definition of f . Furthermore, we have $e \in C \cap F$ since $p \notin C$ and $p' \notin C$, hence $f \leq e$ by

$f = \min(C \cap F)$. So we get $e = f$, and thus the cocycle C and the cycle $C(F \cup \{p\}; p')$ intersect on one only edge, which is impossible. Finally, we have proven that $f' = \min(F \cap C)$.

Now, let us consider the optimal cocycle C_{opt} of \vec{G} . As a cocycle, it is the set of edges joining the two parts A and B of a partition of the vertex set of \vec{G} , and deleting C_{opt} thus disconnects the graph \vec{G} in two connected components $\vec{G}[A]$ and $\vec{G}[B]$. By assumption, C' induces a cocycle of $\vec{G}'(E')$. Since $\vec{G}'(E') = \vec{G}(E' \cup \{p\})/\{p\}$ with $E' = E \setminus C_{\text{opt}}$ and $p \in C_{\text{opt}}$, the cocycle induced by C' in $\vec{G}'(E')$ is a cocycle for one of the two above connected components. Without loss of generality, we can assume that it joins two parts B_1 and B_2 of B , and that the endpoints of p are in A and B_1 . For convenience, we will also assume that p is directed from A to B_1 , but the reasoning below can be done in the same way when p is directed from B_1 to A by reversing all considered directions.

The cocycle C' of \vec{G}' is induced by the cocycle C of \vec{G} , so that, in \vec{G} , we have the following situation: C_{opt} is the cocycle joining A and $B_1 \cup B_2$, with all edges in E directed from A to $B_1 \cup B_2$, and C is the cocycle joining $A \cup B_1$ and B_2 , with $C \cap E'$ joining B_1 and B_2 .

By the definition of C , we have $p \notin C$ and f' is positive in C . Since $f' = \min(F \cap C)$ (as shown above), the cocycle C cannot induce a directed cocycle of $\vec{G}(E)$, otherwise it would be a contradiction with the induction hypothesis. So there exists $e \in E$ with negative sign in C . Since C' induces a positive directed cocycle of $\vec{G}'(E')$, we must have $e \notin E'$. Then e must join A and B_2 . Since e is directed from A to B_2 (as p is directed from A to B), we thus have that all edges in $C \cap E'$ are directed from B_2 to B_1 (their sign is opposite to the sign of e in C).

Now let us consider the cocycle D joining $A \cup B_2$ and B_1 . We have $D = C_{\text{opt}} \Delta C$, and $D \cap E' = C \cap E'$. In particular, D contains p , all its edges in $E \setminus E'$ are directed from A to B_1 (as in C_{opt}), and all its edges in E' are directed from B_2 to B_1 (as in C). It thus induces a positive directed cocycle of $\vec{G}(E)$ containing p . Let f be an edge in C_{opt} smaller than f' (if it exists). Since $f' = \min(C \cap F)$ (as shown above), the edge f does not belong to C , hence it joins A and B_1 , hence it belongs both to C_{opt} and D , with the same sign in both. The edge f' belongs to C' , hence it joins B_1 and B_2 , and it is positive in C' (by our initial assumption), hence it is directed from B_2 to B_1 , and it does not belong to C_{opt} . So, finally, by definition of the ordering of directed cocycles of $\vec{G}(E)$ containing p , we have $C_{\text{opt}} < D$, which is a contradiction with the definition of the optimal cocycle C_{opt} . \square

Lemma 4.9. *Let $\vec{G} = (V, E \cup F)$ be an optimizable digraph satisfying the property stated in Lemma 4.8. Then its optimal cocycle can be computed in polynomial time.*

Proof. This proof is based on geometry and linear programming. It consists of translating the notion of optimal cocycle in terms of lexicographic multiobjective linear programming, which can be solved in polynomial time by numerical methods. We detail the construction and the geometric translation for the convenience of a reader not familiar with such a setting. (The construction is also a special case of a general construction for oriented matroids detailed in [13]; see Remark 4.7.)

Let us first recall the classical representation of a directed graph as a signed real hyperplane arrangement (which is usual in terms of oriented matroids, see for instance [2, Chapter 1]). Consider an optimizable digraph $\vec{G} = (V, E \cup F)$. First, consider the vector space $\mathbb{R}^{|V|}$ generated by variables $x_i, i \in V$. Each undirected edge (u, v) of \vec{G} is identified to the hyperplane $x_v - x_u = 0$ of the real vector space $\mathbb{R}^{|V|-1}$ which is orthogonal to $(1, \dots, 1)$ in $\mathbb{R}^{|V|}$. Specifying the direction of the edge from u to v amounts to specifying a positive halfspace $x_v - x_u > 0$ and a closed positive halfspace

$x_v - x_u \geq 0$. Second, consider the infinity edge p , its associated hyperplane in $\mathbb{R}^{|V|-1}$, and an affine hyperplane called H parallel to this hyperplane in its positive halfspace. The intersections of hyperplanes of $\mathbb{R}^{|V|-1}$ with H thus form a set of affine hyperplanes of the ambient affine space $H \sim \mathbb{R}^{|V|-2}$. Each edge except p corresponds to an affine hyperplane of H . (We can assume that no edge in \vec{G} is parallel to p , otherwise it can be ignored as it does interfere with the construction.) Finally, in this affine representation, we call points the 0-dimensional faces obtained as hyperplane intersections, and they exactly correspond to signed cocycles of \vec{G} containing p with positive sign. The sign of an edge in a cocycle is positive if and only if the point corresponding to the cocycle belongs to the positive halfspace corresponding to the edge. The edge p is not represented and the cocycles not containing p correspond to points at infinity (we call them points at infinity rather than directions).

From now on, we work in the ambient affine space H , where each directed edge of \vec{G} except p is seen as a positive halfspace of H , and (signed) cocycles of \vec{G} containing p (with positive sign) are seen as points in H . Since $\vec{G}(E)$ is acyclic, then the intersection of closed positive halfspaces in $E \setminus \{p\}$ forms a full-dimensional region R of H (every edge belongs to a directed cocycle). By the property seen above, the extreme points of R (at infinity or not) correspond to positive cocycles.

Let us now turn into a standard linear programming setting. A linear program consists of maximizing a linear form in a region of the space delimited by affine inequalities. Here, the optimization will take place in the region R defined above. As a full-dimensional region of H , it contains at least one point, and the linear program is thus called *feasible*. Let $f = f_2$ be the smallest element of F . It is considered as the objective function for the optimization (precisely, its associated hyperplane is the kernel of the linear form to maximize, increasing in the direction from the negative to the positive halfspace). The linear program (or the region R) is called *bounded* (w.r.t. f) if the region R contains no point at infinity on the positive side of f (being a bounded region of the affine space is sufficient to yield a bounded linear program but it is not necessary). Since the property stated in Lemma 4.8 is satisfied, we can check that R is bounded w.r.t. f_2 . Assume that it is not the case, then there exists a positive cocycle C of $\vec{G}(E)$ not containing p (i.e., a point at infinity in R) with positive sign for f_2 . Since $f_2 \in C$, we have $f_2 = \min(F \cap C)$, yielding a contradiction with the property of Lemma 4.8. So, the linear program is feasible and bounded w.r.t. $f = f_2$.

The fundamental result of linear programming states that a linear program which is both feasible and bounded has an optimal solution (a point in R having a maximal value for the objective function f). Actually, the region R can have several optimal points, all lying in the face of R parallel to the hyperplane f and containing an optimal point. In lexicographic multiobjective linear programming, if this face is not just made of one single point, then one starts another linear program on this face with respect to another independent linear form. In our case, we use the next element f_3 of F . The optimization is still made on a feasible region (as a face of R), and on a bounded region w.r.t. f_3 (by the property of Lemma 4.8 again, for a point at infinity on a face parallel to f_2 , corresponding to the cocycle C , we have $f_2 \notin C$, and then $f_3 = \min(F \cap C)$ as soon as $f_3 \in C$). The same construction can be repeated with successive elements of F until a unique optimal point is returned (which happens since $F \cup \{p\}$ is a spanning tree of \vec{G} , meaning that F forms a maximal independent set of linear forms for this setting).

It remains to prove that the optimal cocycle C_{opt} of \vec{G} defined in terms of cocycle ordering corresponds to the optimal point of R defined above in terms of lexicographic multiobjective linear programming. Let us consider two adjacent vertices v_C and v_D in the skeleton of R . Let L be the line formed by v_C and v_D . Let $f \in F$ be an affine hyperplane which is not parallel to the line L ,

and let v_f be the intersection of L and f . Since the initial hyperplane arrangement was built from a graph, the uniform matroid $U_{2,4}$ is an excluded minor of the underlying matroid, which implies that a line in the considered affine hyperplane arrangement has at most two intersection points with hyperplanes (plus one intersection point at infinity with p). Hence, among the three vertices v_C , v_D and v_f of L , at least two of them are equal, hence $v_f = v_C$ or $v_f = v_D$, which implies $f \in C\Delta D$. (Note that this is where we use the fact the oriented matroid is graphical, and this is why the construction of this paper would directly generalize to regular matroids; see also [13].)

Now let us apply the definition of the ordering of cocycles of the optimizable digraph \vec{G} . Assume that f is the smallest element of F belonging to $C\Delta D$. We have $C > D$ if f is positive in C (hence with $f \notin D$) or negative in D (hence with $f \notin C$). Equivalently, in geometrical terms, we have $C > D$ if either v_C is in the positive halfspace of f and v_D is in the hyperplane f , or v_C is in the hyperplane f and v_D is in the negative halfspace of f . In any case, the ordering either $C > D$ or $D < C$ indicates if v_C has either a bigger or a smaller value than v_D , respectively, with respect to the linear form defined by f . Finally, the overall cocycle ordering is consistent with the above lexicographic multiobjective linear programming setting where the comparison between v_C and v_D is made with respect to f_2 , then with respect to f_3 if L is parallel to f_2 (that is, if $f_2 \notin C$ and $f_2 \notin D$), and so on. (One can observe that we do not use here the situation where $f \in C \cap D$. This is because we focus on the skeleton of R . But we handle the case where $f \in C \cap D$ in the definition of the ordering of cocycles so that this ordering is easily defined for every pair of cocycles. This is done in a consistent way: if f is positive in C and negative in D , then, by elimination of f , there exists C' not containing f such that $C > C' > D$.)

To conclude, it is well-known that finding an optimal point of a feasible bounded linear program can be done in polynomial time using numerical methods. Hence, finding (at most) $r - 1$ successive optimal points in $r - 1$ successive regions with respect to f_2, \dots, f_r , until the unique final one corresponding to C_{opt} is found, can also be done in polynomial time. \square

Theorem 4.10. *The fully optimal spanning tree of an ordered directed graph which is bipolar with respect to its smallest edge can be computed in polynomial time.*

Proof. By Lemma 4.8, each of the optimizable digraphs considered in the algorithm of Theorem 4.3 satisfies the property required in Lemma 4.9. So, we apply Lemma 4.9 in each of these $r - 1$ successive optimizable digraphs, each of them being built as a minor of the initial digraph. \square

Remark 4.11 (minimum weight cuts in directed graphs). As we shall see below, the problem of finding the optimal cocycle of an optimizable digraph has some relation with the problem of finding a minimum weight cut in a digraph, a problem which is related to various usual digraph problems. Note that these kinds of problems can be addressed from a pure graph viewpoint, but can also often be translated and addressed in a linear programming setting, allowing for polynomial time numerical algorithms (as we did in Lemma 4.9). On this approach, one can see for instance the book [15], and, in particular, concerning digraph cuts, its Sections 7.1, 8.3, and 8.4.

Here, interesting questions could be to investigate the notion of optimal cocycle of an optimizable digraph in a usual digraph setting, and to build it in polynomial time while staying at the graph level (with or without the property of Lemma 4.8, which is not satisfied in general).

Let us give an answer which is available for the first computed optimal cocycle in Theorem 4.3, which is the fundamental cocycle of $p = \min(E)$ w.r.t. the fully optimal spanning tree of \vec{G} (by the first comment/property in Theorem 4.3 applied to the initial digraph). For the initial optimizable

digraph \vec{G} , the ground set is the whole edge set E , hence the optimal cocycle C_{opt} is a directed cocycle of $\vec{G}(E) = \vec{G}$. In this case, no negative element has to be taken into account when defining C_{opt} from the weight function of Definition 4.2. Finally, C_{opt} can be seen as the directed cocycle of maximal weight in a bipolar acyclic digraph \vec{G} for a certain weight function on (undirected) edges. In what follows, for convenience, we rather consider that the weight has to be minimized.

In such a situation, in order to build such a C_{opt} cocycle, one can use the celebrated *Max-flow-Min-cut Theorem* of digraphs. We refer the reader to [1, Section 4.5] for details. Beware that the notion of cut in a directed graph used in this theorem is not the same as ours, so we will call it *dicut* (to us, a cocycle is an inclusion-minimal cut of the undirected graph with added signs depending on the directions, while a dicut is a set of edges directed from one part of the vertices to the complementary part). Briefly, start with the acyclic digraph with weights on edges, and add all opposite edges with infinite weights. By this theorem, computing a minimal dicut is equivalent to computing a maximum flow, hence it can be done in polynomial time, while staying at the graph level (e.g., by a Ford-Fulkerson method, which by the way can be also understood in terms of linear programming, and which is even simpler in our case where the endpoints of p are the only source and sink). Since the resulting minimum dicut has a finite weight, then it necessarily corresponds to a directed cocycle of the initial digraph (removing edges with an infinite weight), that is, to C_{opt} .

However, this construction cannot be directly applied to compute the optimal cocycle of a general optimizable digraph (nor to compute the next fundamental cocycles of the fully optimal spanning tree), since the optimal cocycle is not in general a directed cocycle of the optimizable digraph (only of its restriction to the ground set E) and since weights of edges in F may have to be counted negatively depending on their direction. We leave as an open question to find suitable adaptations of usual digraph constructions for this problem.

Independently of the above discussion, using the linear ordering on E and the fact that the objective set is built from the smallest lexicographic spanning tree, one can see that the first C_{opt} is also the lexicographically smallest directed cocycle; we state this complementary result below.

Proposition 4.12. *Let $\vec{G} = (V, E)$ be an ordered bipolar digraph w.r.t. $p = \min(E)$. The fundamental cocycle of p with respect to the fully optimal spanning tree of \vec{G} , that is, $C^*(\alpha(\vec{G}); p)$, is the lexicographically smallest directed cocycle of \vec{G} .*

Proof. Let C_{opt} be the optimal cocycle of \vec{G} seen as an optimizable digraph as in the initial step of Theorem 4.3. By this theorem (and as mentioned in Remark 4.11), C_{opt} is a directed cocycle of \vec{G} and $C_{\text{opt}} = C^*(\alpha(\vec{G}); p)$. Let C_m be the smallest lexicographic directed cocycle of \vec{G} . We want to prove that $C_{\text{opt}} = C_m$. Both C_{opt} and C_m contain $p = \min(E)$. Assume for a contradiction that $C_{\text{opt}} \neq C_m$. Let $e = \min(C_{\text{opt}} \Delta C_m)$. We have $e \neq p$, since $p \in C_{\text{opt}} \cap C_m$. We have $e \in C_m \setminus C_{\text{opt}}$, otherwise C_m would not be lexicographically smaller than C_{opt} . So, we have $e \notin F$, otherwise we would have $C_m > C_{\text{opt}}$ by definition of the ordering of cocycles, contradicting that C_{opt} is maximal in this ordering. Now, since $e \notin F \cup \{p\}$, let C be the fundamental cycle of e with respect to the spanning tree $F \cup \{p\}$, that is, the unique cycle contained in $F \cup \{p, e\}$. Let $f \in C$ with $f \neq e$, then $f < e$, otherwise the spanning tree $(F \setminus \{f\}) \cup \{p, e\}$ (obtained by exchanging f and e in $F \cup \{p\}$) would be lexicographically smaller than $F \cup \{p\}$, contradicting the definition of F . Since $e = \min(C_{\text{opt}} \Delta C_m)$, every element of E smaller than e belongs to C_{opt} if and only if it belongs to C_m . So, as every element f of $C \setminus \{e\}$ is smaller than e , we get that $C \cap C_{\text{opt}} \setminus \{e\} = C \cap C_m \setminus \{e\}$.

Since $e \in C_m \setminus C_{\text{opt}}$, we get $(C \cap C_{\text{opt}}) \cup \{e\} = C \cap C_m$. The intersection of a cycle and a cocycle always has an even size, so we get a contradiction because either $C \cap C_{\text{opt}}$ or $C \cap C_m$ has an odd size. \square

Acknowledgments. Emeric Gioan wishes to thank an anonymous reviewer for numerous useful comments and English corrections, and to thank Jørgen Bang-Jensen and Stéphane Bessy for communicating how to use the *Max-flow-Min-cut Theorem* in Remark 4.11. This research was supported by the ANR Grant DIGRAPHS (Directed graphs, ANR-19-CE48-0013).

References

- [1] J. Bang-Jensen, and G. Gutin, *Digraphs - Theory, Algorithms and Applications*, 2nd ed., Springer Monographs in Mathematics, Springer-Verlag London, 2009. See Section 4.5 *The Maximum Flow problem*.
- [2] A. Björner, M. Las Vergnas, B. Sturmfels, N. White and G. Ziegler, *Oriented matroids*, 2nd ed., Encyclopedia of Mathematics and its Applications 46, Cambridge University Press, Cambridge, UK 1999.
- [3] H.H. Crapo, *A higher invariant for matroids*, J. Combinatorial Theory **2** (1967), 406–417.
- [4] E. Gioan, *Correspondance naturelle entre bases et réorientations des matroïdes orientés*, Ph.D. thesis, University of Bordeaux 1, 2002.
- [5] E. Gioan. *The Tutte polynomial of oriented matroids*. Handbook of the Tutte Polynomial and Related Topics, J. Ellis-Monaghan and I. Moffatt ed., Chapman and Hall/CRC, Chapter 31, 565–589, 2022.
- [6] E. Gioan and M. Las Vergnas, *Bases, reorientations and linear programming in uniform and rank 3 oriented matroids*, Adv. in Appl. Math. **32** (2004), 212–238. (Special issue Workshop on Tutte polynomials, Barcelona 2001.)
- [7] E. Gioan and M. Las Vergnas, *Activity preserving bijections between spanning trees and orientations in graphs*, Discrete Math. **298** (2005), 169–188. (Special issue FPSAC 2002.)
- [8] E. Gioan and M. Las Vergnas, *The active bijection between regions and simplices in supersolvable arrangements of hyperplanes*, Electronic Journal of Combinatorics **11 (2)** (2006), #R30, 39p. (Stanley Festschrift.)
- [9] E. Gioan and M. Las Vergnas, *The active bijection in graphs, hyperplane arrangements, and oriented matroids 1. The fully optimal basis of a bounded region*, European Journal of Combinatorics **30 (8)** (2009), 1868–1886. (Special issue: Combinatorial Geometries and Applications: Oriented Matroids and Matroids.) (Erratum available at <http://www.lirmm.fr/~gioan>.)
- [10] E. Gioan and M. Las Vergnas, *A linear programming construction of fully optimal bases in graphs and hyperplane arrangements*, Electronic Notes in Discrete Mathematics **34** (2009), 307–311. (Proceedings EuroComb 2009, Bordeaux.)
- [11] E. Gioan and M. Las Vergnas, *The active bijection for graphs*. Advances in Applied Mathematics **104** (2019), 165–236.
- [12] E. Gioan and M. Las Vergnas, *The active bijection 2.b - Decomposition of activities for oriented matroids, and general definitions of the active bijection*. Preprint available at arXiv:1807.06578.
- [13] E. Gioan and M. Las Vergnas, *The active bijection in hyperplane arrangements and oriented*

- matroids 3. Fully optimal bases and elaborations on pseudo/real linear programming.* In preparation.
- [14] E. Gioan and M. Las Vergnas, *The active bijection in hyperplane arrangements and oriented matroids 4. Deletion/contraction and characterizations.* In preparation.
- [15] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization.* Berlin, Springer, 1988. See Section 7.1 *Flows and Cuts*, Section 8.3 *Flows, Paths, Chains, and Cuts*, and Section 8.4 *Trees, Branchings, and Rooted and Directed Cuts*.
- [16] M. Las Vergnas, *The Tutte polynomial of a morphism of matroids II. Activities of orientations*, Progress in Graph Theory (J.A. Bondy & U.S.R. Murty, ed.), Academic Press, Toronto, Canada, 1984, (Proc. Waterloo Silver Jubilee Conf. 1982), pp. 367–380.
- [17] W.T. Tutte, *A contribution to the theory of chromatic polynomials*, Canad. J. Math. **6** (1954), 80–91.
- [18] T. Zaslavsky, *Facing up to arrangements: Face-count formulas for partitions of space by hyperplanes*, Mem. Amer. Math. Soc., No. 154, Amer. Math. Soc., Providence, R.I., 1975.