The lattice of cycles of an undirected graph

G. Averkov [a,1], A. Chavez [b,2], J.A. De Loera [b,3], B. Gillespie [c,*]

[a] *Fakultät 1, BTU Cottbus-Senftenberg, Germany*
[b] *Department of Mathematics, University of California, Davis, USA*
[c] *Department of Mathematics, Colorado State University, USA*

A R T I C L E   I N F O

A B S T R A C T

We study bases of the lattice generated by the cycles of an undirected graph, defined as the integer linear combinations of the 0/1-incidence vectors of cycles. We prove structural results for this lattice, including explicit formulas for its dimension and determinant, and we present efficient algorithms to construct lattice bases, using only cycles as generators, in quadratic time. By algebraic considerations, we relate these results to the more general setting with coefficients from an arbitrary Abelian group. Our results generalize classical results for the vector space of cycles of a graph over the binary field to the case of an arbitrary field.

* Corresponding author.
E-mail addresses: gennadiy.averkov@gmail.com (G. Averkov), amrchavez@ucdavis.edu (A. Chavez), deloera@math.ucdavis.edu (J.A. De Loera), bryan.gillespie@colostate.edu (B. Gillespie).

1. Introduction

The structure of the cycles of a graph is a rich topic with challenging problems. Consider for instance three famous covering problems on the set of all cycles of a graph: The *double cover conjecture* states that for any bridgeless graph there exists a list of cycles that contains every edge twice. *Goddyn's conjecture* further states that if G is a bridgeless graph and C is a cycle in G, then there exists a double cover of G containing the cycle C. An *m-cycle k-cover* is a list of m Eulerian subgraphs covering each edge exactly k times. For example, every bridgeless graph admits a 7-cycle 4-cover, but it is an open problem to decide whether every cubic bridgeless graph has a 6-cycle 4-cover. For details see [1–3] and references therein. Motivated by such *covering and packing problems* using cycles, and relying on the linear structure, this paper studies the lattice generated by the cycles of an undirected connected graph G, i.e., the set of all integer linear combinations of 0/1-incidence vectors of cycles of G. We call it the *cycle lattice* of the graph G.

Studying the lattices generated by incidence vectors of combinatorial objects is a technique which has been used to model combinatorial problems. A classical example is of course the case of directed graphs, which has many applications (see the survey [4]). The technique has also been used in several cases for undirected graphs, including for matchings, cuts, and cycles (see [5–8] and references therein). From these last examples we take inspiration; we provide theoretical and computational results about the bases of the cycle lattice of an undirected connected graph, and some consequences.

In what follows $G = (V, E)$ will denote a connected undirected graph with vertices V and edges E. In general, we allow loops and parallel edges unless otherwise noted. A *cycle* is a connected subgraph of G with each vertex having degree two, and we write $\mathcal{C}(G)$ for the collection of cycles of G. We will usually regard cycles and trees as subsets of E.

If $A \subseteq E$, then let $\chi_A \in \mathbb{Z}^E$ denote the characteristic vector of A. For a collection $\mathcal{A}$ of subsets of E, define the *lattice* of $\mathcal{A}$ by

$$\mathrm{Lat}(\mathcal{A}) := \left\{ \sum_{A \in \mathcal{A}} n_A \chi_A \ : \ n_A \in \mathbb{Z} \right\} \subseteq \mathbb{Z}^E.$$

If K is a field, or more generally an Abelian group, then we define the *K-linear hull* of $\mathcal{A}$ to be

$$\mathrm{Lin.\,Hull}_K(\mathcal{A}) := \mathrm{Lat}(\mathcal{A}) \otimes_{\mathbb{Z}} K = \left\{ \sum_{A \in \mathcal{A}} n_A \chi_A \ : \ n_A \in K \right\}.$$

We are interested in studying the properties of these spaces when $\mathcal{A} = \mathcal{C}(G)$. The lattice $\mathrm{Lat}(\mathcal{C}(G))$ we call the *cycle lattice* of G, and the $\mathbb{Q}$-linear hull $\mathrm{Lin.\,Hull}_{\mathbb{Q}}(\mathcal{C}(G))$ we call the *rational cycle space* of G. In Section 3 we give structural and algorithmic

results on the cycle lattice of a graph, while in Section 4 we explore the consequences of these results on the K-linear hulls of graph cycles for different choices of K.

The classical *binary cycle space* of G in particular fits into this framework as the linear hull for the choice $K = \mathbb{Z}/2\mathbb{Z}$. In our more general setting, we are able to give a dimension formula for arbitrary fields K, and we describe a structural characterization of linear hulls for general Abelian groups which sheds light on the special role played in this theory by fields of characteristic 2. It is also worth mentioning that several authors have considered the case of *directed graphs*, which have different behavior. For more details on the other linear spaces generated by cycles, see [4].

The following result by Goddyn (see [5, Prop. 2.1]) is our starting point. It characterizes the rational cycle space via series classes of E (i.e., $e, f \in E$ are *in series* if they are in the same cycles).

Proposition 1.1. *Let* $G = (V, E)$ *be a graph. Then the rational cycle space* $\mathrm{Lin.\,Hull}_{\mathbb{Q}}(\mathcal{C}(G))$ *is given by*

$$\left\{ p \in \mathbb{Q}^E : p_e = 0 \text{ for any bridge } e, \text{ and } p_e = p_f \text{ for } e \text{ and } f \text{ in series} \right\}.$$

In particular, Proposition 1.1 implies that the rational cycle space and the cycle lattice of G are full-dimensional if and only if G has no bridges and no nontrivial series classes, or equivalently when the graph is 3-edge-connected. In our analysis we will see that there is no loss of generality in assuming that G is 3-edge-connected (in particular see Lemma 2.7).

Our contributions

Our first main result on the cycle lattice is a key building block utilized throughout the paper.

Theorem 1. *Let* $G = (V, E)$ *be a 3-edge-connected graph, and let* $T \subseteq E$ *be a spanning tree of* G. *Consider the sets*

$$\mathcal{C}_T := \left\{ \chi_{\mathrm{ci}(e,T)} : e \in E \setminus T \right\} \quad \text{and} \quad X_T := \left\{ 2\chi_t : t \in T \right\},$$

where $\mathrm{ci}(e, T)$ *denotes the unique cycle contained in* $T \cup e \subset E$. *Then the collection* $\mathcal{C}_T \cup X_T$ *is a basis for the cycle lattice of* G. *Moreover, the determinant of this lattice is given by*

$$\det\left(\mathrm{Lat}(\mathcal{C}(G)) \right) = 2^{|T|}.$$

The lattice bases provided by this result are a natural extension to the well-known *fundamental cycle bases* of the binary cycle space, but have the disadvantage that they

include elements which are not cycles. With some additional work, we are able to algorithmically produce lattice bases consisting only of cycles. Note that this is in stark contrast to the lattices of other natural collections such as matchings [7] and cuts [6], which do not always have bases consisting of sets in the generating collection.

A collection of cycles of G is called a *lattice cycle basis* if its indicator vectors form a basis of the cycle lattice. In Section 3, we show the existence of lattice cycle bases of a graph G, and we give two algorithmic constructions for such bases.

A lattice cycle basis is called a *semi-fundamental basis* with respect to a spanning forest F if it consists of all of the fundamental cycles of F, along with some additional cycles containing exactly two edges outside of F, called *semi-fundamental cycles*. In Section 3.2, we describe an efficient algorithm to compute semi-fundamental lattice cycle bases of a graph G with respect to a choice of spanning forest in quadratic time, producing the following.

Theorem 2. *Let G be a connected graph with m edges and n vertices. Then a lattice cycle basis of G exists, and can be constructed in time $O(mn)$. If T is any spanning tree of G, then the basis may be chosen to be semi-fundamental with respect to T.*

A potentially useful property of the semi-fundamental bases given by this algorithm is that all cycles included have length bounded by $2 \operatorname{diam}(T)$; see Corollary 3.8. In particular, this bound may be quite nontrivial for common classes of graphs with high connectivity such as expander graphs and small-world networks [9,10]. Several papers investigate the variety of different cycle lengths possible in a graph, for example [11–13], and our length bound indicates that in many cases, the cycles in a lattice basis may be chosen from a very restricted set. Additionally, lattice generators with sparse support are relevant to applications in several areas of optimization; see [14] and references.

If G, H are graphs, then G is called a *topological one-edge extension* of H if it is obtained from H by connecting two vertices, either existing in H or created by dividing edges of H in two, by a new edge. (See Definition 3.9.) If G is 3-edge-connected, then a sequence of topological one-edge extensions starting at the single-vertex graph and ending at G is called a *topological extension sequence* of G. In particular, a graph G is known to be 3-edge-connected if and only if it admits a topological extension sequence, and we present an algorithm to produce such a sequence.

Theorem 3. *Let G be a 3-edge-connected graph with n vertices and m edges. Then a topological extension sequence for G exists, and can be constructed in time $O(mn)$.*

If $G_0, G_1, \ldots, G_k$ is a topological extension sequence of a graph G, then a nested sequence $(\mathcal{C}_i)$ with $\mathcal{C}_i$ a lattice cycle basis of G_i is called a *compatible chain* of lattice cycles bases. In Section 3.3, we give a different algorithm for a basis of the cycle lattice which produces a topological extension sequence and a compatible chain of lattice cycle bases.

Theorem 4. *Let G be a 3-edge-connected graph with m edges and n vertices. Then a topological extension sequence of G and a compatible chain of lattice cycle bases can be constructed in time $O(mn)$.*

The construction of the above algorithm can additionally be extended to general connected graphs using the reduction of Lemma 2.7.

Finally, in Section 4 we relate the cycle lattice of a graph to the A-linear hull for A an Abelian group. The main structural result is given by the following.

Theorem 5. *Let $G = (V, E)$ be a 3-edge-connected graph with m edges and n vertices, and let A be an Abelian group. Then*

$$\mathrm{Lin.\,Hull}_A(\mathcal{C}(G)) \simeq (2A)^{n-1} \oplus A^{m-n+1}.$$

This result is applied in Theorem 6 to the case when A is a field, generalizing known results about the classical binary cycle space:

Theorem 6. *Let $G = (V, E)$ be a 3-edge-connected graph with m edges and n vertices, and let K be a field of characteristic p. Then $\mathrm{Lin.\,Hull}_K(\mathcal{C}(G))$ is a K-vector space of dimension*

$$\dim_K\big(\mathrm{Lin.\,Hull}_K(\mathcal{C}(G))\big) = \begin{cases} m, & \text{if } p \neq 2, \\ m - n + 1, & \text{if } p = 2. \end{cases}$$

If $p \neq 2$, then any lattice basis of $\mathrm{Lat}(\mathcal{C}(G))$ reduces modulo p to a linear basis of $\mathrm{Lin.\,Hull}_K(\mathcal{C}(G))$. If $p = 2$, then any basis of the classical binary cycle space maps to a linear basis of $\mathrm{Lin.\,Hull}_K(\mathcal{C}(G))$ under the natural inclusion map.

The remainder of the paper is organized as follows. In Section 2 we give a brief overview of relevant background material and prior results in graph theory, and discuss preliminary computational results and our computational model. In Section 3.1 we study the basic structure of the cycle lattice and derive Theorem 1, and in Sections 3.2 and 3.3 we present two approaches for producing lattice cycle bases of graphs, in particular proving Theorems 2, 3 and 4. In Section 4 we summarize several consequences of our results for linear hulls of cycles with respect to fields and Abelian groups, and we give proofs of Theorems 5 and 6.

2. Preliminaries

In the following, we introduce preliminary material which will be needed throughout the remainder of the work. First we give a brief overview of the ideas in graph theory that will be assumed as background. After this, we discuss the computational model that will be used for our algorithmic assertions, and we develop a computational reduction, Lemma 2.7, which will be essential for our analysis in Section 3.

2.1. Graph theory

We briefly provide the graph theory concepts necessary to understand our results. For standard background from graph theory, including basic definitions and notations, we refer to [2].

Here and throughout this work, graphs are allowed to have loops and parallel edges. Let $G = (V, E)$ be an undirected graph with vertices V and edges E. If v is a vertex of G, then its degree $\deg(v)$ is the number of non-loop edges incident to v plus twice the number of loops incident to v. A (simple) path P of length k in G is a subgraph of G with distinct vertices $\{x_0, x_1, \ldots, x_k\}$ and edges $\{x_0 x_1, x_1 x_2, \ldots, x_{k-1} x_k\}$. A (simple) **cycle** of length $k + 1$ in G is a subgraph of G consisting of a simple path along with an additional edge connecting its endpoints.

If $e \in E$, then the **deletion** of e from G is the graph $G \setminus e$ obtained from G by removing the edge E, and the **contraction** of e in G is the graph G/e obtained from G by combining its endpoints into a single vertex, and removing e from the result. The cycles of $G \setminus e$ are exactly the cycles of G which do not contain e, and the cycles of G/e are the inclusion-minimal nonempty subgraphs within the set of graphs $\{C/e : C \text{ a cycle of } G\}$.

If $E_1, E_2 \subseteq E$ are disjoint sets of edges, then a graph may be obtained by deleting the edges of E_1 and contracting the edges of E_2 in any order. The resulting graph is independent of the order chosen, and is denoted $G \setminus E_1/E_2$. A graph which can be obtained in this way from G is called a **minor** of G.

An edge $e \in E$ is called a **bridge** if e is contained in no cycle of G, and edges $e, f \in E$ are said to be in **series** if $e \in C$ implies $f \in C$ for every cycle C. The relation of being in series is an equivalence relation on E whose equivalence classes are called the **series classes** of G. A series class is called **non-trivial** if it has more than one element, and **trivial** otherwise.

The graph G is **connected** if it contains a path between any two vertices, and it is **k-edge-connected** if $G \setminus E_1$ is connected for any set E_1 of $k - 1$ or fewer edges. Most importantly for our purposes, a connected graph is 2-edge-connected if and only if it has no bridges, and is 3-edge-connected if and only if it has no bridges and no nontrivial series classes.

Edge connectivity of a graph is related to the following version of the well-known *Menger's theorem* (see for instance [2, Sec. 3.3]), which is critical to our proofs.

Proposition 2.1 *(Menger's theorem, edge version). Let G be an undirected graph and let $u, v \in V(G)$ be distinct vertices. Then the minimum number of edges which can be deleted from G to disconnect u and v is equal to the maximum number of edge-disjoint paths connecting u and v.*

In particular, if G is k-edge-connected and $u, v \in V$, then there exist k edge-disjoint paths connecting u and v.

Suppose G is connected, and let $T \subseteq E$ be a spanning tree of G. Recall that for each edge $e \in E \setminus T$, there is a unique cycle contained in the edge set $T \cup e$, which is called the **fundamental cycle** of e with respect to T and is denoted $\mathrm{ci}(e, T)$. For each edge $t \in T$, the forest $T \setminus t$ has two connected components, which induces a cut of G between the corresponding vertex sets. This cut is called the **fundamental cut** or **fundamental bond** of t with respect to T, and is denoted $\mathrm{bo}(t, T)$. Fundamental cycles and fundamental cuts exhibit the following duality (see e.g., [15, Lem. 7.3.1]).

Lemma 2.2. *Let $G = (V, E)$ be a connected graph and let $T \subseteq E$ be a spanning tree. If $t \in T$ and $e \in E \setminus T$, then*

$$e \in \mathrm{bo}(t, T) \text{ if and only if } t \in \mathrm{ci}(e, T).$$

The well-studied **binary cycle space** of G is defined as $\mathrm{Lat}(\mathcal{C}(G)) \otimes (\mathbb{Z}/2\mathbb{Z})$, and can be thought of as the vector subspace of $(\mathbb{Z}/2\mathbb{Z})^E$ spanned by the indicator vectors of $\mathcal{C}(G)$. It is known that the collection of all fundamental cycles with respect to a fixed spanning tree of G gives a basis of this space; the following result summarizes this and other related properties.

Proposition 2.3 *([2, Sec. 1.9]). Let $G = (V, E)$ be a connected graph with binary cycle space $\mathcal{B}$. Then:*

- $\mathcal{B}$ *is the collection of characteristic vectors of Eulerian subgraphs of G.*
- *If $T \subseteq E$ is a spanning tree of G, then $\left\{ \chi_{\mathrm{ci}(e,T)} : e \in E \setminus T \right\}$ is a basis of $\mathcal{B}$.*
- *The dimension of $\mathcal{B}$ is $|E| - |V| + 1$.*

2.2. Computational model and graph reductions

The structure of graph cycle lattices can in many cases be reduced to the case of 3-edge-connected graphs. In the following we give details of this reduction, and describe the computational model we use for algorithmic complexity bounds. The key result connecting cycle lattices of 3-edge-connected graphs with those of general graphs is found in Lemma 2.7.

Let $G = (V, E)$ be a graph with m edges and n vertices. We will assume that V is a totally ordered set, and that G is represented as an ordered adjacency list. In particular, inspecting the edges of G adjacent to a vertex v takes time $O(\deg(v))$, and inspecting all of the edges of G takes time $O(m)$. If $E_1, E_2 \subseteq E$ are disjoint, then the graph minor $G \setminus E_1 / E_2$ can be computed in time $O(mn)$ by deleting all of the edges of $E_1 \cup E_2$ from G and merging the remaining adjacencies of vertices connected by a path in E_2.

Lemma 2.4. *Let $G = (V, E)$ be a connected graph with m edges and n vertices, and let T be a spanning tree of G. Then the fundamental cycles and fundamental cuts of T can be computed in time $O(mn)$.*

Proof. We will record the fundamental cycles of T by computing the $T \times (E \setminus T)$ binary matrix X with values

$$X_{t,e} = \begin{cases} 1, & t \in \mathrm{ci}(e, T) \\ 0, & \text{otherwise} \end{cases}.$$

This simultaneously computes the fundamental cycles and the fundamental cuts of T because $t \in \mathrm{ci}(e, T)$ if and only if $e \in \mathrm{bo}(t, T)$ by Lemma 2.2.

The algorithm proceeds as follows. Initialize X to all zeros, and pick an arbitrary vertex v_0 of T. Traverse T to compute for each vertex $v \in V$ the path P_v in T from v_0 to v. For each edge $e \in E \setminus T$, let v, v' be its endpoints. Find the first edge at which P_v and $P_{v'}$ differ, and set $X_{t,e}$ to 1 for the subsequent edges of these paths.

The elements of T in a fundamental circuit $\mathrm{ci}(e, T)$ are given by the unique path in T between the endpoints of e, so we see that the edges recorded in this way represent the fundamental circuits of T as desired. The computational time $O(mn)$ follows because each path P_v has at most $n-1$ edges, and the number of edges $e \in E \setminus T$ is bounded by m. $\square$

Lemma 2.5. *Let $G = (V, E)$ be a connected graph with m edges and n vertices. Then the bridge elements and series classes of G can be computed in time $O(mn)$.*

Proof. Let T be a spanning tree of G, and let X be the $T \times (E \setminus T)$ fundamental cycle matrix from Lemma 2.4, which can be computed in time $O(mn)$. The bridge elements of G are those which appear in no fundamental cycle of T, and so can be identified as the indices of the all-zero rows of X.

For the series classes, note that if $x, y \in E$ are in different series classes, then there are elements of the binary cycle space for which the x and y coordinates differ. In particular, since the fundamental cycles of T generate the binary cycle space, this implies that there is a *fundamental* cycle of T containing one of x, y but not the other. Thus to compute the series classes of G, it is sufficient to compute the partition of E corresponding to the common refinement of the partitions $\{C, E \setminus C\}$ where C is a fundamental cycle with

respect to T. Because each cycle C has length at most n and there are $m-n+1 = O(m)$ such cycles, this refinement can be computed in time $O(mn)$. $\square$

Lemma 2.6. *Let $G = (V, E)$ be a connected graph, and let $e \in E$. Then:*

1. *If e is contained in a non-trivial series class of G, then $\mathcal{C}(G)$ and $\mathcal{C}(G/e)$ are in bijection by the map $C \mapsto C/e$.*
2. *If e is a bridge of G, then $\mathcal{C}(G) = \mathcal{C}(G \setminus e)$.*

Let $\pi : \mathbb{Z}^E \to \mathbb{Z}^{E \setminus e}$ be the standard projection map. In both of the cases above, π induces a lattice isomorphism between the cycle lattice of G and the cycle lattice of the corresponding graph minor.

Proof. Part 1 follows because the cycles of G/e are the nonempty subgraphs in $\{C/e : C \in \mathcal{C}(G)\}$ which are minimal under inclusion of edge sets, and Part 2 follows because no cycle contains a bridge, and the cycles of $G \setminus e$ are those of G not containing e. The projection π induces a lattice isomorphism in each case because it maps the indicator vector of a cycle in G to the indicator vector of the corresponding cycle in G/e or $G \setminus e$. $\square$

If G is a connected graph, let $\hat{G}$ denote a graph obtained from G by deleting all bridges of G and contracting all but one element from each nontrivial series class of edges in G. The graph $\hat{G}$ is called a **cosimplification** of G. This induces a projection map $\pi : E(G) \to E(\hat{G}) \cup \{\epsilon\}$, where ϵ is a formal symbol disjoint from $E(G)$, given by

$$\pi : e \mapsto \begin{cases} \epsilon, & e \text{ a bridge element} \\ \hat{e}, & \text{otherwise} \end{cases},$$

where $\hat{e}$ denotes the representative of the series class in G of an edge e in $\hat{G}$. The connected components of $\hat{G}$ are in particular 3-edge-connected, and this gives a graph reduction which is useful in studying cycles and cycle bases.

Lemma 2.7. *Let $G = (V, E)$ be a connected graph with m edges and n vertices. Then a cosimplification $\hat{G}$ can be computed in time $O(mn)$. Let $G_1, \ldots, G_k$ be the connected components of $\hat{G}$, and let $\mathcal{C}_i$ be a lattice cycle basis of G_i for each i. Then a lattice cycle basis of G is given by $\mathcal{C} = \bigcup_i \{\pi^{-1}(C) : C \in \mathcal{C}_i\}$, which can be computed in time $O(mn)$.*

Proof. From Lemma 2.5, we can compute the bridges and series classes of G in time $O(mn)$, from which the graph minor $\hat{G}$ can be constructed.

Lemma 2.6 implies that π induces a lattice isomorphism from $\text{Lat}(\mathcal{C}(G))$ to $\text{Lat}(\mathcal{C}(\hat{G}))$. Since the cycle lattice of a graph admits a direct sum decomposition over connected components, the collection $\bigcup_i \mathcal{C}_i$ yields a lattice cycle basis of $\hat{G}$. Thus, π^{-1} lifts to a lattice cycle basis of G.

The computation of π^{-1} can be accomplished by checking each cycle for the presence of the series class representative for a non-trivial series class and extending, if the representative is found, to include the whole series class in G. There are $O(m)$ cycles in the basis of $\hat{G}$, and $O(n)$ nontrivial series classes, yielding a time bound of $O(mn)$ for the computation. $\square$

3. The cycle lattice of a graph

We now develop results relating to the cycle lattice of a graph, with an emphasis on understanding the structure of and algorithms for producing lattice bases. In Section 3.1, we prove basic structural results of the cycle lattice, and produce a simple lattice basis extending the fundamental cycle basis of the classical binary cycle space. In Section 3.2 we present an algorithm to produce *semi-fundamental* lattice cycle bases, and in Section 3.3 we present a different algorithm which sequentially expands lattice cycle bases for the graph minors in a *topological extension sequence*. Both algorithms will be seen to produce a lattice cycle basis in time $O(mn)$, where m is the number of edges of the graph and n is the number of vertices.

3.1. Lattice structure and a non-cycle basis

As a first step toward understanding the lattice structure, we make the following observation.

Lemma 3.1. *Let G be a 3-edge-connected graph. For each $e \in E$, the vector $2\chi_e$ is an element of $\mathrm{Lat}(\mathcal{C}(G))$. In particular, $2\mathbb{Z}^E \subseteq \mathrm{Lat}(\mathcal{C}(G))$.*

Proof. Let $e \in E$, and without loss of generality suppose e connects distinct vertices u and v. Since G is 3-edge-connected, any minimal cut in $G \setminus e$ disconnecting vertices u and v contains at least two edges. By Menger's theorem, there are edge-disjoint simple paths P and Q between u and v which exclude the edge e.

In particular, $P \cup e$ and $Q \cup e$ are cycles whose only common edge is e. Additionally, $P \cup Q$ is a (potentially non-simple) cycle of G, which can be written as a disjoint union of cycles, $P \cup Q = C_1 \cup \cdots \cup C_k$. From this we obtain that

$$2\chi_e = \chi_{P \cup e} + \chi_{Q \cup e} - \sum_i \chi_{C_i}$$

is in $\mathrm{Lat}(\mathcal{C}(G))$. $\square$

As a first corollary, we obtain a basis-free description of the cycle lattice, analogous to Proposition 1.1.

Corollary 3.2. *Let $G = (V, E)$ be a graph. Then $\mathrm{Lat}(\mathcal{C}(G))$ is given by the collection of $p \in \mathbb{Z}^E$ such that*

- $p_e = 0$ *for any bridge* e.
- $p_e = p_f$ *for* e *and* f *in series.*
- $\sum_{e \in N(v)} p_e$ *is even for each vertex* v*, where* $N(v)$ *is the neighborhood of edges incident to* v*.*

Proof. The indicator vector of any cycle of G satisfies all of the above conditions, so this is likewise true for any element of the cycle lattice. Now suppose $p \in \mathbb{Z}^E$ satisfies the conditions above, and let $E_1 = \{e \in E : p_e \text{ is odd}\}$. By the parity condition,

$$0 \equiv \sum_{e \in N(v)} p_e \equiv \sum_{e \in N(v) \cap E_1} p_e \equiv |N(v) \cap E_1| \pmod 2.$$

Since each vertex v is incident to an even number of edges of E_1, this implies that E_1 induces an Eulerian subgraph of G, which thus can be decomposed into a disjoint union of a collection $\mathcal{C}'$ of cycles. Letting $p' = p - \sum_{C \in \mathcal{C}'} \chi_C$, we see that p' has only even coordinates.

Passing to a cosimplification $\hat{G}$ of G, the projection of p' lies in the cycle lattice of $\hat{G}$ by Lemma 3.1. Because p' is zero on bridges and equal on all elements of a series class, the cycle decomposition of the projection lifts to a cycle decomposition of p' in G. Thus, p' lies in the cycle lattice of G, and consequently so does p. □

The fact that $2\mathbb{Z}^E$ is a sublattice of $\mathrm{Lat}(\mathcal{C}(G))$ for 3-edge-connected G means that we can view the quotient $\mathrm{Lat}(\mathcal{C}(G))/2\mathbb{Z}^E$ as a subspace of the binary vector space $(\mathbb{Z}/2\mathbb{Z})^E$. This allows us to directly compute the determinant of the lattice.

Proposition 3.3. *Let* $G = (V, E)$ *be a 3-edge-connected graph with cycle lattice* $\mathcal{L} = \mathrm{Lat}(\mathcal{C}(G))$*. Then* $\det(\mathcal{L}) = 2^{|V|-1}$*.*

Proof. The determinant of $\mathcal{L}$ can be expressed in terms of group indices as $\left[\mathbb{Z}^E : \mathcal{L}\right]$, which implies

$$\frac{2^{|E|}}{\det(\mathcal{L})} = \left[\mathcal{L} : 2\mathbb{Z}^E\right] = 2^{\dim_{\mathbb{Z}/2\mathbb{Z}} \left(\mathcal{L}/2\mathbb{Z}^E\right)}.$$

Here, the first equality is by comparison of determinants, and the second is by interpreting the lattice quotient $\mathcal{L}/2\mathbb{Z}^E$ as a vector subspace of $\mathbb{Z}^E/2\mathbb{Z}^E$. By Proposition 2.3, the space of cycles over $\mathbb{Z}/2\mathbb{Z}$ has dimension $|E| - |V| + 1$, from which we can compute $\det(\mathcal{L})$ directly. □

With explicit formulas for the dimension and the determinant of $\mathrm{Lat}(\mathcal{C}(G))$, we are now able to produce an explicit lattice basis by producing an appropriate number of lattice vectors with the correct determinant. This is accomplished by extending the collection of indicator vectors of fundamental cycles $\mathrm{ci}(e, T)$ with respect to a fixed spanning tree T, as follows.

Proposition 3.4. *If $G = (V, E)$ is a 3-edge-connected graph and T is a spanning tree of G, let*

$$\mathcal{C}_T := \{\chi_{\mathrm{ci}(e,T)} : e \in E \setminus T\} \quad and \quad X_T := \{2\chi_t : t \in T\}.$$

Then the collection $\mathcal{C}_T \cup X_T$ is a basis for $\mathrm{Lat}(\mathcal{C}(G))$.

We present two short proofs of this fact, one of which directly applies our computation for the determinant of $\mathrm{Lat}(\mathcal{C}(G))$, and a second of independent interest which uses the duality between fundamental cycles and fundamental cuts.

Proof 1. We have that $X_T \subseteq \mathrm{Lat}(\mathcal{C}(G))$ by Lemma 3.1. Note that the vectors in $\mathcal{C}_T \cup X_T$ are naturally identified with the edges of G. Then for a fixed ordering on E for which the elements of T come before the elements of $E \setminus T$, the matrix of column vectors of $\mathcal{C}_T \cup X_T$ induced by this ordering on both the rows and columns can be seen to have the following block structure,

$$\begin{bmatrix} 2I_{|T|} & A \\ 0 & I_{|E \setminus T|} \end{bmatrix},$$

where I_k is the $k \times k$ identity matrix, and A is the matrix whose columns are the indicator vectors of $\mathrm{ci}(e, T) \setminus e$ for $e \in E \setminus T$. From this block structure we see that the determinant of these lattice vectors is $2^{|V|-1}$. Thus, by Proposition 3.3, the collection forms a lattice basis of $\mathrm{Lat}(\mathcal{C}(G))$, as desired. $\square$

Our second proof of Proposition 3.4 relies on the notion of fundamental bonds we discussed earlier.

Proof 2. We have that $X_T \subseteq \mathrm{Lat}(\mathcal{C}(G))$ by Lemma 3.1. Since the lattice has dimension $|E| = |\mathcal{C}_T \cup X_T|$ by Proposition 1.1, it is sufficient to show that $\mathcal{C}_T \cup X_T$ generates any cycle of G.

For this, let $p = \chi_C$ be the indicator vector of a cycle in G, and note that since the intersection of a cycle with a cut has even cardinality, we have

$$\sum_{e \in D} p_e \equiv 0 \pmod{2},$$

for any cut D of G. Letting $q = \sum_{e \in E \setminus T} p_e \chi_{\mathrm{ci}(e,T)}$, we see that for each $e \in E \setminus T$ the e-components of q and p are equal. Further, for each $t \in T$, the t-component of q can be written as

$$q_t = \sum_{\substack{e \in E \setminus T \\ \text{s.t. } t \in \mathrm{ci}(e,T)}} p_e = \sum_{e \in \mathrm{bo}(t,T)} p_e - p_t.$$

Denoting the sum in the latter expression above by S_t, note that $S_t + p_t$ is the sum of the components of p across the fundamental cut $\mathrm{bo}(t, T)$, which is even. This implies that $S_t - p_t$ is even. Letting $\alpha_t = (S_t - p_t)/2$, we have

$$p = q + \sum_{t \in T} \alpha_t (2\chi_t).$$

Thus $p = \chi_C$ is generated by $\mathcal{C}_T \cup X_T$, and this concludes the proof. $\quad\square$

Note that while the above material is formulated in the setting of graphs, the results may be extended to the more general class of binary matroids satisfying the **lattice of circuits** property (see [5, Sec. 2]). We omit these details here, but the arguments involved are substantially similar.

Proposition 3.4 provides a lattice basis of $\mathrm{Lat}(\mathcal{C}(G))$ which is useful for many applications, but includes elements outside of the generating collection of cycle indicator vectors. We next consider lattice bases consisting only of cycle indicator vectors.

Definition 3.5. If $\mathcal{C}'$ is a collection of cycles of a graph G whose indicator vectors form a basis of the lattice $\mathrm{Lat}(\mathcal{C}(G))$, we call $\mathcal{C}'$ a **lattice cycle basis** of G.

We emphasize that in the above definition, the additive structure is over $\mathbb{Z}$ rather than over $\mathbb{Z}/2\mathbb{Z}$ as in the classical binary cycle space, so that a priori it is not clear if a lattice cycle basis of a graph always exists. Indeed, in the context of lattices, a generating set does not always contain a basis. For example, the set $\{2, 3\}$ generates the lattice $\mathbb{Z}$, but $\{2, 3\}$ contains no basis of $\mathbb{Z}$. Somewhat surprisingly, such a phenomenon never occurs for the generator set $\{\chi_C : C \in \mathcal{C}(G)\}$ of the lattice $\mathrm{Lat}(\mathcal{C}(G))$. In the following sections, we provide two constructions for such lattice cycle bases.

3.2. Semi-fundamental lattice cycle bases

We now describe an algorithm to produce a cycle basis using the fundamental cycles of a spanning tree, and some additional cycles which we call *semi-fundamental*.

Definition 3.6. Let $G = (V, E)$ be a connected graph, let T be a spanning tree of G, and let $e, f \in G \setminus T$, $e \neq f$. If $\mathrm{ci}(e, T)$ and $\mathrm{ci}(f, T)$ intersect in at least one edge of T, we call the symmetric difference

$$\mathrm{ci}(ef, T) := \mathrm{ci}(e, T) \,\Delta\, \mathrm{ci}(f, T)$$

a **semi-fundamental cycle** of G with respect to the tree T.

A lattice cycle basis $\mathcal{C}'$ of G is called **semi-fundamental** with respect to T if $\mathcal{C}'$ contains all of the fundamental cycles with respect to T, and all other cycles of $\mathcal{C}'$ are semi-fundamental with respect to T.

The following lemma provides a key inductive step for the subsequent algorithm.

Lemma 3.7. *Let $G = (V, E)$ be a 3-edge-connected graph with at least two vertices, and let T be a spanning tree of G. Then there exist two fundamental cycles of T that intersect in a single edge of T.*

Proof. We first prove that if there exist two fundamental cycles C, C' whose intersection is a path P of positive length k, then there exist two fundamental cycles whose intersection is a single edge of this path. If P has length 1 then C and C' are already sufficient to conclude, so suppose that P has length at least 2.

Let $v_0, \ldots, v_k$ be the vertices of P occurring on P in this order, let e be the edge joining v_0 with v_1 and e' the edge joining v_{k-1} with v_k. Since $e \neq e'$, the forest $T \setminus \{e, e'\}$ has three connected components. Let H_P be the component which contains $P \setminus \{e, e'\}$, and let H and H' be the connected components containing the vertices v_0 and v_k, respectively.

Since G is 3-edge-connected, $G \setminus \{e, e'\}$ is connected, and some edge $x \notin T$ connects H_P to either H or H'. Without loss of generality, assume that x connects H_P to H. In particular, the fundamental cycle $C'' = \mathrm{ci}(x, T)$ contains e and avoids e'. Consequently, $P \cap C''$ is a nonempty proper subpath of P that contains the edge e and does not contain e'.

Since C and C' diverge at v_0, the edges of $C \setminus P$ and $C' \setminus P$ incident to v_0 are distinct. In particular, at least one of these edges is distinct from the edge of $C'' \setminus P$ incident to v_0. So, suppose without loss of generality C and C'' diverge at v_0. Since e' is in C but not in C'', there is another vertex v_j with $1 \leq j < k$ at which C and C'' diverge. Consequently, the intersection $P' = C \cap C''$ of fundamental cycles C and C'' is a proper sub-path of P of length strictly less than k.

To conclude the lemma, note that if $t \in T$, then the 3-edge-connectivity of G implies that $\mathrm{bo}(t, T)$ contains at least two distinct edges $e_1, e_2 \in E \setminus T$. In particular $\mathrm{ci}(e_1, T) \cap \mathrm{ci}(e_2, T) \ni t$ is a nonempty path of T. Thus there exist two fundamental cycles of T with nonempty intersection, so by reverse induction we conclude that there exist fundamental cycles sharing exactly one edge. $\square$

We next prove Theorem 2, giving an algorithm to efficiently produce a semi-fundamental lattice cycle basis.

Theorem 2. *Let G be a connected graph with m edges and n vertices. Then a lattice cycle basis of G exists, and can be constructed in time $O(mn)$. If T is any spanning tree of G, then the basis may be chosen to be semi-fundamental with respect to T.*

Proof. Assume first that $G = (V, E)$ is 3-edge-connected. We inductively construct sequences

$$\left(t_k\right)_{k \in \{1, \ldots, n-1\}}, \quad \left(e_k\right)_{k \in \{1, \ldots, n-1\}}, \quad \left(f_k\right)_{k \in \{1, \ldots, n-1\}}, \tag{1}$$

where the sequence $(t_k)_{k \in \{1,\ldots,n-1\}}$ gives an ordering of the edges of T, while e_k and f_k are edges outside of T. Suppose that $t_1, \ldots, t_{i-1}$ have already been constructed, and let

$$G_i := G/\{t_1, \ldots, t_{i-1}\} \qquad \text{and} \qquad T_i := T/\{t_1, \ldots, t_{i-1}\}. \quad (2)$$

In particular, G_i is 3-edge-connected since this property is preserved by graph contraction, and T_i is a spanning tree of G_i. By Lemma 3.7, there exist edges $e_i, f_i \in G_i \setminus T_i = G \setminus T$ such that the fundamental cycles of e_i and f_i with respect to T_i in G_i satisfy

$$\mathrm{ci}(e_i, T_i) \cap \mathrm{ci}(f_i, T_i) = \{t_i\}$$

for some $t_i \in T_i$. We will show that the pairs of edges (e_i, f_i) define semi-fundamental cycles of T which produce a semi-fundamental lattice cycle basis.

The fundamental cycles of T are preserved under tree contractions, in the sense that

$$\mathrm{ci}(e, T_k) = \mathrm{ci}(e, T)/\{t_1, \ldots, t_{k-1}\}$$

holds for every k and every $e \in E \setminus T$. Consequently, t_k is a common edge of the cycles $\mathrm{ci}(e_k, T)$ and $\mathrm{ci}(f_k, T)$, which implies that the semi-fundamental cycles $\mathrm{ci}(e_k f_k, T)$ are well-defined. Thus, let

$$\mathcal{C}' := \{\mathrm{ci}(e, T) \ : \ e \in G \setminus T\} \cup \{\mathrm{ci}(e_k f_k, T) \ : \ k \in \{1, \ldots, n-1\}\},$$

and let Λ be the lattice generated by $\mathcal{C}'$. For $A_k := \mathrm{ci}(e_k, T) \cap \mathrm{ci}(f_k, T)$, we have

$$2\chi_{A_k} = \chi_{\mathrm{ci}(e_k, T)} + \chi_{\mathrm{ci}(f_k, T)} - \chi_{\mathrm{ci}(e_k f_k, T)} \in \Lambda.$$

By construction, $t_k \in A_k \subseteq \{t_1, \ldots, t_k\}$. Thus, $2\chi_{t_k} = 2\chi_{A_k} - \sum_{t \in A_k \setminus \{t_k\}} 2\chi_t$. By induction on k we see that $2\chi_{t_k} \in \Lambda$ for every $k \in \{1, \ldots, n-1\}$.

It follows that Λ contains the basis of $\mathrm{Lat}(\mathcal{C}(G))$ from Proposition 3.4, and consequently, $\Lambda = \mathrm{Lat}(\mathcal{C}(G))$. Taking into account that $\mathcal{C}'$ consists of $m = \dim(\mathrm{Lat}(\mathcal{C}(G)))$ cycles, we conclude that $\mathcal{C}'$ is a lattice cycle basis of G.

To verify the algorithmic part of the assertion, we need to expand on the procedure of constructing $\mathcal{C}'$ suggested above. The procedure relies on the constructive proof of Lemma 3.7, so we first explain how to convert the proof of Lemma 3.7 into an efficient algorithm.

In order to find two fundamental cycles that have common edges, pick an edge t of T, determine the two trees in the forest $T \setminus \{t\}$ and then iterate through the edges $e \in E$ to detect those edges that connect the two trees in $T \setminus \{t\}$. There will be at least two such edges by 3-edge-connectedness of G. This procedure gives a pair C, C' of intersecting fundamental cycles. The proof of Lemma 3.7 continues by explaining how to decrease the number of edges in the intersection $C \cap C'$ by exchanging one of the two fundamental cycles C, C' with another one. Every exchange is based on deletion of two edges e, e'

from T and looking at edges that connect the trees in the forest $T \setminus \{e, e'\}$. Every cycle exchange can be carried out in time $O(m)$ using similar ideas.

The algorithm emerging from Lemma 3.7 produces a sequence (C_i, C_i'), $i = 1, \ldots, s$ of pairs of fundamental cycles with $\{t\} := C_s \cap C_s' \subsetneq \ldots \subsetneq C_1 \cap C_1'$, spending $O(m)$ time units for each pair. After contraction of the edge t in the main iteration, one can reuse the pairs $(C_i/t, C_i'/t)$, $i < s$ of contracted cycles in the following iterations. With this approach, every pair of cycles so computed is used as a pair $\mathrm{ci}(e_i, T_i)$, $\mathrm{ci}(f_i, T_i)$ for some i. Thus, the algorithm spends $O(m)$ time units per edge of T, and this amounts to the total running time $O(mn)$.

Now suppose G is a general connected graph with spanning tree T. By Lemma 2.7, a cosimplification $\hat{G}$ of G may be constructed in time $O(mn)$. We further require that $\hat{G}$ be constructed so that only edges of T are contracted, which is possible because each nontrivial series class of G contains at most one edge outside of T. Under this construction, the edges in $T \cap \hat{G}$ form a spanning forest of the cosimplification.

Let $G_i = (V_i, E_i)$, $i = 1, \ldots, k$ be the connected components of $\hat{G}$ with $m_1, \ldots, m_k$ edges and $n_1, \ldots, n_k$ vertices respectively, and let $T_i = T \cap E_i$ be the spanning tree of G_i induced by T under the cosimplification. Each G_i is 3-edge-connected, so a semifundamental lattice cycle basis $\mathcal{C}_i$ of G_i with respect to T_i may be constructed in time $O(m_i n_i)$. By the second part of Lemma 2.7, these lattice cycle bases may be lifted to a lattice cycle basis of G in time $O(mn)$, for a total computational time of $O(mn + \sum_i m_i n_i) = O(mn)$.

Last, we argue that this lifted lattice cycle basis is semi-fundamental with respect to T. If $\hat{C} \in \mathcal{C}_i$ and C is the lifting of $\hat{C}$ to G, then $\hat{C} \cap T_i = C \cap T$ because the edges contracted to form $\hat{G}$ were all edges of T. Hence fundamental and semi-fundamental cycles of T_i in G_i are lifted to fundamental and semi-fundamental cycles of T in G. Additionally, because the edges $E \setminus T$ are given by $\bigcup_i E_i \setminus T_i$, each fundamental $\mathrm{ci}_G(e, T)$ is given by the lifting of a fundamental cycle $\mathrm{ci}_{G_i}(e_i, T_i)$ for some i and some $e_i \in E_i$. $\quad\square$

The following observation highlights a potentially useful property of the lattice cycle bases produced by the above algorithm: the lengths of cycles in a semi-fundamental basis are controlled by the diameter of the underlying spanning tree.

Corollary 3.8. *Let $G = (V, E)$ be a connected graph with m edges and $n \geq 2$ vertices, and let T be a spanning tree of G. Then a lattice cycle basis of G may be constructed in time $O(mn)$ such that each cycle has length at most $2 \operatorname{diam}(T)$.*

Proof. Any semi-fundamental lattice cycle basis with respect to T has the desired property. In particular, a fundamental cycle of T has length at most $\operatorname{diam}(T) + 1$, and a semi-fundamental cycle, as the symmetric difference of two intersecting fundamental cycles, has length at most $2 \operatorname{diam}(T)$. $\quad\square$

3.3. Lattice cycle bases by topological extension

We now present a different approach for the construction of lattice cycle bases. This approach is dynamic in the sense that lattice bases are built up for a sequence of successively larger graph minors, and the extensions at each step may be chosen independently. Throughout this section, if $F \subseteq E$ are sets and $x \in \mathbb{Q}^E$, then we write $(x)_F$ for the standard projection of x onto $\mathbb{Q}^F$.

Definition 3.9. For graphs G and H, we write $H \xrightarrow{e} G$, and say that G is a **topological one-edge extension** of H, if G is obtained from H by one of the following operations:

(A) A new edge e is added between existing, possibly equal, vertices a and b of H.
(B) An edge f of H is divided into two edges f_1, f_2 by a new vertex a, and a new edge e is added between a and an existing vertex b of H.
(C) Two distinct edges f and g of H are each divided into two edges f_1, f_2 and g_1, g_2 by new vertices a and b respectively, and a new edge e is added between a and b.

We say that the **type** of the topological one-edge extension $H \xrightarrow{e} G$ is one of (A), (B), or (C), depending on which of the above operations G is derived from. From the definition, we see that H is a graph minor of G obtained by deleting the new edge e and contracting one edge from each split pair, depending on the type of the extension.

Example 3.10. The complete graph K_4 on four vertices is a topological one-edge extension of the 3-edge bond graph. In particular, the extension is of type (C), and can be realized by picking f and g to be any pair of distinct edges in the 3-edge bond.

For a topological one-edge extension $H \xrightarrow{e} G$, the cycle lattice of H naturally embeds into the cycle lattice of G. Specifically, H is obtained from $G \setminus e$ by contracting zero, one, or two edges, and each contracted edge is part of a non-trivial series class of $G \setminus e$. The isomorphism of cycle lattices of H and $G \setminus e$ then follows from Lemma 2.6, and the cycle lattice of $G \setminus e$ canonically embeds into the cycle lattice of G because $\mathcal{C}(G \setminus e) \subseteq \mathcal{C}(G)$.

The embedding of cycle lattices may be described by a linear map $\mathbb{Q}^{E(H)} \to \mathbb{Q}^{E(G)}$, which depends on the type of the extension as follows:

$$(A): \quad x \mapsto \begin{bmatrix} x \\ 0 \end{bmatrix} \begin{matrix} E(H) \\ e \end{matrix} \qquad (B): \quad x \mapsto \begin{bmatrix} (x)_{E(H) \setminus f} \\ x_f \\ x_f \\ 0 \end{bmatrix} \begin{matrix} E(H) \setminus f \\ f_1 \\ f_2 \\ e \end{matrix}$$

$$(C): \quad x \mapsto \begin{bmatrix} (x)_{E(H)\setminus\{f,g\}} \\ x_f \\ x_f \\ x_g \\ x_g \\ 0 \end{bmatrix} \begin{array}{l} E(H)\setminus\{f,g\} \\ f_1 \\ f_2 \\ g_1 \\ g_2 \\ e \end{array}$$

Topological one-edge extension can be seen to be compatible with 3-edge-connectivity by the following result, whose proof is straightforward.

Lemma 3.11. *If $H \xrightarrow{e} G$ and H is 3-edge-connected, then G is 3-edge-connected as well.*

In fact, 3-edge-connectivity of a graph can be characterized in terms of topological one-edge extensions. The following result is essentially the case $l = 1$ of a theorem of Mader [16] characterizing $(2l + 1)$-edge-connected graphs, and can be viewed as a counterpart of Tutte's 1966 theorem characterizing 3-vertex-connectivity. See [17, Thm. 7.13] and [2, Thm. 3.2.2] for details.

Proposition 3.12 (*Growing a 3-edge-connected graph from a single vertex*). *A graph G is 3-edge-connected if and only if*

$$G_0 \xrightarrow{e_1} G_1 \xrightarrow{e_2} \cdots \xrightarrow{e_k} G_k = G$$

holds for some graphs $G_0, \dots, G_k$, where G_0 is a single vertex.

We call a sequence of topological one-edge extensions as in the above a **topological extension sequence** for G. The following describes an algorithm to efficiently produce such a sequence.

Theorem 3. Let G be a 3-edge-connected graph with n vertices and m edges. Then a topological extension sequence for G exists, and can be constructed in time $O(mn)$.

Proof. If H is any graph, let $\text{Top}(H)$ denote the *topological representative* of H, defined as the graph obtained from H by suppressing vertices of degree 2. We will produce a topological extension sequence for G presented as a decomposition of $E(G)$ into a sequence of edge-disjoint paths $P_0, \dots, P_k$, where the graphs $G_i := \text{Top}(P_0 \cup \dots \cup P_i)$ satisfy

$$G_0 \xrightarrow{e_1} G_1 \xrightarrow{e_2} \cdots \xrightarrow{e_k} G_k = G.$$

To start, let P_0 be any vertex of G, and let P_1 be a path starting and ending at the vertex of P_0. Throughout, let $H_i := P_0 \cup \dots \cup P_i$. For $i \geq 2$, suppose $i - 1$ paths $P_0, \dots P_{i-1}$ and corresponding subgraphs $H_0, \dots, H_{i-1}$ have already been constructed.

If $H_{i-1} = G$, then the algorithm terminates. Otherwise, $E(G) \setminus E(H_{i-1})$ is nonempty, and because G is connected, there is a vertex v_i of H_{i-1} which is incident to an edge not contained in H_{i-1}. We then use depth-first search to generate a path $P_i \subseteq G \setminus H_{i-1}$ starting at v_i, ending at some vertex of H_{i-1}, and containing no edges or interior vertices in H_{i-1}.

If $\deg_{H_{i-1}}(v_i) > 2$ or if H_{i-1} is a cycle graph, then any path P_i using only edges and interior vertices outside H_{i-1} is sufficient. If $\deg_{H_{i-1}}(v_i) = 2$ and H_{i-1} is not a cycle graph, then v_i is an interior vertex of a path Q_i with end vertices of degree at least 3 in H_{i-1}, and with interior vertices of degree exactly 2 in H_{i-1}. In this case, we search for a path P_i starting at v_i and ending at some vertex w_i of H_{i-1} which is not also an interior vertex of Q_i. This ensures that P_i induces a topological one-edge extension of G_{i-1} of type either (B) or (C), and does not connect two new vertices on the same edge in G_{i-1}. Such a path exists because otherwise the removal of the two edges of Q_i incident with its endpoints would disconnect G, contradicting 3-edge-connectedness.

As an optimization, after each path is completed in this way, inspect the edges of G for edges incident to two vertices of degree at least 3 in H_i. If e is such an edge, then the additional path P consisting only of the edge e may be added to the output sequence. The additional paths of this form do not require depth-first search in G, and the overall computational cost of adding these paths can be seen to be $O(mn)$.

The correctness of the algorithm follows from Lemma 3.11. For the running time, we note that the number of iterations requiring a depth-first search is at most $3n$. This follows because each path P_i formed by a depth-first search either contains a vertex outside of H_{i-1}, or has an endpoint in H_{i-1} of degree 2. In either of these cases, some vertex of G with degree less than 3 in H_{i-1} has strictly larger degree in H_i, and this can occur at most $3n$ times throughout the construction. Thus the search-based steps are computed in time $O(mn)$. Because the single-edge paths between degree 3 vertices are also computed in time $O(mn)$, the result follows. $\square$

In the following we provide an explicit construction to extend a basis of the cycle lattice from a 3-edge-connected graph to a topological one-edge extension. This provides the primary tool which we will use to inductively construct lattice cycle bases from scratch.

Proposition 3.13. *Let H be a 3-edge-connected graph, and let G be a topological one-edge extension of H with m edges. Then a set of cycles exists whose indicator vectors extend any basis of $\mathrm{Lat}(\mathcal{C}(H))$ to a basis of $\mathrm{Lat}(\mathcal{C}(G))$. Such a set can be computed in time $O(m)$.*

Proof. Let $\mathcal{B} = \{b_1, \ldots, b_m\}$ be a lattice basis of $\mathrm{Lat}(\mathcal{C}(H))$. We argue by cases for the three possible types of the topological one-edge extension $H \xrightarrow{e} G$. We will use the notation for such extensions described in Definition 3.9, and we will write Φ for the natural linear map embedding $\mathrm{Lat}(\mathcal{C}(H))$ into $\mathrm{Lat}(\mathcal{C}(G))$. In each case, once an extending

collection of cycles $\mathcal{C}' \subseteq \mathcal{C}(G)$ of the correct cardinality has been determined, we verify that the collection $\Phi(\mathcal{B}) \cup \{\chi_C : C \in \mathcal{C}'\}$ is a basis of $\mathrm{Lat}(\mathcal{C}(G))$ by checking that the new collection has determinant equal to $\det\big(\mathrm{Lat}(\mathcal{C}(G))\big)$.

For type (A), a cycle C in G which contains e can be constructed in time $O(m)$ by finding a simple path P between a and b in H. Then we have

$$\det(\Phi(\mathcal{B}), \chi_C) = \det \left[\begin{array}{ccc|c} b_1 & \cdots & b_m & \chi_P \\ 0 & \cdots & 0 & 1 \end{array} \right] \begin{array}{c} E(H) \\ e \end{array}$$

$$= \det(\mathcal{B}).$$

Thus the determinant of the new collection is equal to $\det\big(\mathrm{Lat}(\mathcal{C}(H))\big) = \det\big(\mathrm{Lat}(\mathcal{C}(G))\big)$, and we can let $\mathcal{C}' = \{C\}$.

For type (B), we can construct $C_1, C_2 \in \mathcal{C}(G)$ satisfying $C_i \cap \{f_1, f_2, e\} = \{f_i, e\}$ in time $O(m)$ as follows. If v_i is the endpoint of f in H incident to f_i in G, then a simple path P_i can be found between v_i and b in the connected graph $H \setminus f$, and we can set $C_i = P_i \cup \{f_i, e\}$. Setting $\Delta = \det(\Phi(\mathcal{B}), \chi_{C_1}, \chi_{C_2})$, we have

$$\Delta = \det \left[\begin{array}{ccc|cc} (b_1)_{E(H)\setminus f} & \cdots & (b_m)_{E(H)\setminus f} & \chi_{P_2} & \chi_{P_1} \\ (b_1)_f & \cdots & (b_m)_f & 0 & 1 \\ (b_1)_f & \cdots & (b_m)_f & 1 & 0 \\ 0 & \cdots & 0 & 1 & 1 \end{array} \right] \begin{array}{c} E(H)\setminus f \\ f_1 \\ f_2 \\ e \end{array}$$

$$= \det(\mathcal{B}) \cdot \det \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix},$$

where the last line is by subtracting the row indexed by f_1 from the row indexed by f_2 and taking determinants along the block diagonal. In this case, the determinant of the new collection is given by $2\det\big(\mathrm{Lat}(\mathcal{C}(H))\big) = \det\big(\mathrm{Lat}(\mathcal{C}(G))\big)$, so the collection $\mathcal{C}' = \{C_1, C_2\}$ satisfies the desired conditions.

For type (C), we construct three cycles C_1, C_2, and C satisfying $C_i \cap \{f_1, f_2, g_1, g_2, e\} = \{f_i, g_i, e\}$ for $i = 1, 2$, and $C \cap \{f_1, f_2, g_1, g_2, e\} = \{f_1, g_2, e\}$. For $i = 1, 2$, let v_i be the endpoint of f incident to f_i in G, and let w_i be the endpoint of g incident to g_i in G. Since H is 3-edge-connected, the deletion $H \setminus \{f, g\}$ remains connected. Thus we can find simple paths P_1, P_2, P in $H \setminus \{f, g\}$ in time $O(m)$ where P_i joins v_i and w_i and P joins v_1 and w_2. Then we can set $C_i = P_i \cup \{f_i, g_i, e\}$ and $C = P \cup \{f_1, g_2, e\}$. Setting $\Delta = \det(\Phi(\mathcal{B}), \chi_{C_1}, \chi_{C_2}, \chi_C)$, we have

$$\Delta = \det \left[\begin{array}{ccc|ccc} (b_1)_{E(H)\setminus\{f,g\}} & \cdots & (b_m)_{E(H)\setminus\{f,g\}} & \chi_{P_1} & \chi_{P_2} & \chi_P \\ (b_1)_f & \cdots & (b_m)_f & 1 & 0 & 1 \\ (b_1)_g & \cdots & (b_m)_g & 1 & 0 & 0 \\ (b_1)_f & \cdots & (b_m)_f & 0 & 1 & 0 \\ (b_1)_g & \cdots & (b_m)_g & 0 & 1 & 1 \\ 0 & \cdots & 0 & 1 & 1 & 1 \end{array} \right] \begin{array}{c} E(H)\setminus\{f,g\} \\ f_1 \\ g_1 \\ f_2 \\ g_2 \\ e \end{array}$$

$$= \det(\mathcal{B}) \cdot \det \begin{bmatrix} -1 & 1 & -1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix},$$

where the last line is by subtracting the rows indexed by f_1 and g_1 from the rows indexed by f_2 and g_2 and taking determinants along the block diagonal. In this case, the determinant of the new collection is given by $4 \det\big(\mathrm{Lat}(\mathcal{C}(H))\big) = \det\big(\mathrm{Lat}(\mathcal{C}(G))\big)$, so the collection $\mathcal{C}' = \{C_1, C_2, C\}$ satisfies the desired conditions. $\square$

If $G_0 \xrightarrow{e_1} G_1 \xrightarrow{e_2} \cdots \xrightarrow{e_k} G_k$ is a topological extension sequence, define a **compatible chain** of lattice cycle bases to be a nested sequence $\mathcal{C}_0 \subset \mathcal{C}_1 \subset \cdots \subset \mathcal{C}_k$, where $\mathcal{C}_i$ is a lattice cycle basis of G_i for each i. The following aggregates the preceding results to give an algorithm which produces a topological extension sequence and a compatible chain of lattice cycles bases of a 3-edge-connected graph G. In particular, this produces a lattice cycle basis of G.

Theorem 4. Let G be a 3-edge-connected graph with m edges and n vertices. Then a topological extension sequence of G and a compatible chain of lattice cycle bases can be constructed in time $O(mn)$.

Proof. From Theorem 3, a topological extension sequence $G_0 \xrightarrow{e_1} G_1 \xrightarrow{e_2} \cdots \xrightarrow{e_k} G_k = G$ of G can be computed in time $O(mn)$, and such an extension sequence has length $O(m)$. Inductively we produce a lattice cycle basis $\mathcal{C}_i$ of G_i for each i by extending $\mathcal{C}_{i-1}$.

As an auxiliary data structure, we maintain a spanning tree T_i of G_i at each step of the induction as follows. For each edge x divided in two by the topological extension $G_{i-1} \xrightarrow{e_i} G_i$, if $x \in T_{i-1}$, then include both of the resulting edges in T_i, and if $x \notin T_{i-1}$, then include only one of the resulting edges in T_i. In particular, $e_i \notin T_i$ for any i.

For each topological one-edge extension $G_{i-1} \xrightarrow{e_i} G_i$ of type (B) or (C), Proposition 3.13 shows how to construct $\mathcal{C}_i$ from $\mathcal{C}_{i-1}$ in time $O(m)$. Any topological one-edge extension of these types adds at least one new vertex, so there are at most $O(n)$ such extensions in the sequence, and the overall time needed for computing these lattice cycle bases is $O(mn)$.

For topological one-edge extensions of type (A), an extending cycle can be produced from any path between the endpoints of e_i excluding this new edge. In particular, we make use of the path between the endpoints of e_i in the spanning tree T_i, which can be computed in time $O(n)$. Since there are $O(m)$ such extensions in the extension sequence, the overall time needed for computing these lattice cycle bases is also $O(mn)$. $\square$

As for Theorem 2, this $O(mn)$ algorithm for producing a lattice cycle basis for 3-edge-connected graphs extends by Lemma 2.7 to an $O(mn)$ algorithm for arbitrary connected graphs. It is interesting to note that the extensions at each step of the above construction are independent of the other steps, which implies a multiplicative mode of growth for

the number of lattice cycle bases of a 3-edge-connected graph in terms of its number of edges.

4. Applications to linear hulls

The results discussed so far in this paper focus on the Abelian group structure of the cycle lattice, whose properties are closely connected to the combinatorics of the underlying graph. We now present some consequences of these results to the A-linear hull of cycles of a graph G with respect to an Abelian group A. In the following, we view $\mathrm{Lin.\,Hull}_A(\mathcal{C}(G))$ as a subgroup of $A^E \simeq \mathbb{Z}^E \otimes_{\mathbb{Z}} A$.

Theorem 5. Let $G = (V, E)$ be a 3-edge-connected graph with m edges and n vertices, and let A be an Abelian group. Then

$$\mathrm{Lin.\,Hull}_A(\mathcal{C}(G)) \simeq (2A)^{n-1} \oplus A^{m-n+1}.$$

Proof. Let T be a spanning tree of G, and let $\phi_0 : A^E \to A^E$ be defined by

$$\phi_0 : a \mapsto \sum_{t \in T} a_t \chi_t + \sum_{e \in E \setminus T} a_e \chi_{\mathrm{ci}(e,T)}.$$

By Proposition 3.4, the vectors $\mathcal{C}_T = \left\{ \chi_{\mathrm{ci}(e,T)} : e \in E \setminus T \right\}$ with the vectors $X_T = \{ 2\chi_t : t \in T \}$ give a lattice basis of $\mathrm{Lat}(\mathcal{C}(G))$. Consequently, for $a \in A^T$ and $b \in A^{E \setminus T}$,

$$\phi_0(2a, b) = \sum_{t \in T} a_t(2\chi_t) + \sum_{e \in E \setminus T} b_e \chi_{\mathrm{ci}(e,T)}$$

is an element of $\mathrm{Lin.\,Hull}_A(\mathcal{C}(G))$, and in particular ϕ_0 restricts to a homomorphism $\phi : (2A)^T \oplus A^{E \setminus T} \to \mathrm{Lin.\,Hull}_A(\mathcal{C}(G))$.

Because $X_T \cup \mathcal{C}_T$ is a basis of $\mathrm{Lat}(\mathcal{C}(G))$, we conclude that ϕ is surjective. To show that ϕ is injective, suppose $a \in A^T$ and $B \in A^{E \setminus T}$ such that $\phi(2a, b) = 0$. Each coordinate $e \in E \setminus T$ has a nonzero value in the sum $\phi(2a, b)$ only for the summand corresponding to the cycle indicator vector $\chi_{\mathrm{ci}(e,T)}$, so we conclude that $b = 0$. Subsequently, each coordinate $t \in T$ has a nonzero value in the remaining sum $\phi(2a, 0)$ only for the summand corresponding to the indicator vector χ_t, so we likewise have $2a = 0$. $\square$

For finite Abelian groups, the previous result gives a simple method to determine the primary decomposition of $\mathrm{Lin.\,Hull}_A(\mathcal{C}(G))$ from the primary decomposition of A. Each p-primary group C_{p^k}, $p \neq 2$ in the decomposition of A introduces m copies in the primary decomposition of $\mathrm{Lin.\,Hull}_A(\mathcal{C}(G))$, and each 2-primary group C_{2^k} introduces $m - n + 1$ copies of C_{2^k} along with $n - 1$ copies of $C_{2^{k-1}}$.

As another application of Theorem 5, we may apply the result to an arbitrary field to obtain the following.

Theorem 6. Let $G = (V, E)$ be a 3-edge-connected graph with m edges and n vertices, and let K be a field of characteristic p. Then $\mathrm{Lin.\,Hull}_K(\mathcal{C}(G))$ is a K-vector space of dimension

$$\dim_K \big(\mathrm{Lin.\,Hull}_K(\mathcal{C}(G)) \big) = \begin{cases} m, & \text{if } p \neq 2, \\ m - n + 1, & \text{if } p = 2. \end{cases}$$

If $p \neq 2$, then any lattice basis of $\mathrm{Lat}(\mathcal{C}(G))$ reduces modulo p to a linear basis of $\mathrm{Lin.\,Hull}_K(\mathcal{C}(G))$. If $p = 2$, then any basis of the classical binary cycle space maps to a linear basis of $\mathrm{Lin.\,Hull}_K(\mathcal{C}(G))$ under the natural inclusion map.

Proof. Let $U = \mathrm{Lin.\,Hull}_K(\mathcal{C}(G))$. By Theorem 5, the K-dimension of U is given by $\dim_K\big((2K)^{n-1} \oplus K^{m-n+1}\big)$. When K has characteristic 2, we have $2K = \{0\}$, and otherwise, $2K = K$.

For the vector space bases, the argument of Theorem 5 shows that the image of the lattice basis $\mathcal{B}$ of Proposition 3.4 in U generates the linear hull. If $p \neq 2$, then $\mathcal{B}$ has as many vectors as the dimension, and thus forms a vector space basis of U. If $\mathcal{B}'$ is a lattice basis of $\mathrm{Lat}(\mathcal{C}(G))$, then its image in U is a $\mathbb{Z}$-invertible linear transformation of $\mathcal{B}$, and thus also forms a vector space basis.

If $p = 2$, then the $n - 1$ vectors $2\chi_t$ of $\mathcal{B}$ map to 0 in U, and the remaining $m - n + 1$ vectors $\mathcal{B}_0$ form a basis of U, again by dimension considerations. The collection $\mathcal{B}_0$ forms a basis of the binary cycle space of G by Proposition 2.3, so any basis $\mathcal{B}_0'$ of the binary cycle space is an invertible transformation of $\mathcal{B}_0$ over $\mathbb{Z}/2\mathbb{Z} \subseteq K$, and thus likewise maps to a vector space basis of U. □

Declaration of competing interest

The authors declare that they have no known competing interests that could have appeared to influence the work reported in this paper.

References

[1] J. Bermond, B. Jackson, F. Jaeger, Shortest coverings of graphs with cycles, J. Comb. Theory, Ser. B 35 (3) (1983) 297–308.
[2] R. Diestel, Graph Theory, 5th edition, Graduate Texts in Mathematics, Springer-Verlag, Heidelberg, 2017.
[3] C. Zhang, Integer Flows and Cycle Covers of Graphs, Pure and Applied Mathematics, vol. 205, Marcel Dekker, 1997.
[4] T. Kavitha, C. Liebchen, K. Mehlhorn, D. Michail, R. Rizzi, T. Ueckerdt, K. Zweig, Cycle bases in graphs characterization, algorithms, complexity, and applications, Comput. Sci. Rev. 3 (4) (2009) 199–243.
[5] L.A. Goddyn, Cones, lattices and Hilbert bases of circuits and perfect matchings, in: Graph Structure Theory, in: Contemporary Mathematics, vol. 147, American Mathematical Society, 1993, pp. 419–439.
[6] L.A. Goddyn, T. Huynh, T. Deshpande, On Hilbert bases of cuts, Discrete Math. 339 (2) (2016) 721–728.

[7] L. Lovász, Matching structure and the matching lattice, J. Comb. Theory, Ser. B 43 (2) (1987) 187–222.

[8] P.D. Seymour, Sums of circuits, in: J. Bondy, U. Murty, W. Tutte (Eds.), Graph Theory and Related Topics, Academic Press, 1979, pp. 341–355.

[9] S. Hoory, N. Linial, A. Wigderson, Expander graphs and their applications, Bull. Am. Math. Soc. 43 (4) (2006) 439–561.

[10] B. Bollobás, O. Riordan, The diameter of a scale-free random graph, Combinatorica 24 (1) (2004) 5–34.

[11] A. Gyárfás, J. Komlós, E. Szemerédi, On the distribution of cycle lengths in graphs, J. Graph Theory 8 (4) (1984) 441–462.

[12] P. Erdős, R.J. Faudree, C.C. Rousseau, R.H. Schelp, The number of cycle lengths in graphs of given minimum degree and girth, Discrete Math. 200 (1999) 55–60.

[13] J. Verstraëte, On the number of sets of cycle lengths, Combinatorica 24 (4) (2004) 719–730.

[14] I. Aliev, G. Averkov, J.A. De Loera, T. Oertel, Optimizing sparsity over lattices and semigroups, arXiv:1912.09763, 2020.

[15] A. Björner, The homology and shellability of matroids and geometric lattices, in: Matroid Applications, in: Encyclopedia of Mathematics and Its Applications, vol. 40, Cambridge University Press, 1992.

[16] W. Mader, A Reduction Method for Edge-Connectivity in Graphs, Annals of Discrete Mathematics, vol. 3, Elsevier, 1978, pp. 145–164.

[17] A. Frank, Connectivity and network flows, in: Handbook of Combinatorics 1, 1995, pp. 111–177.