COMPUTING CYCLIC INVARIANTS FOR MOLECULAR GRAPHS[*]

FRANZISKA BERGER[†], PETER GRITZMANN[‡], AND SVEN DE VRIES[§]

Abstract. Ring structures in molecules belong to the most important substructures for many applications in Computational Chemistry. One typical task is to find an implicit description of the ring structure of a molecule. We present efficient algorithms for cyclic graph invariants that may serve as molecular descriptors to accelerate database searches. Another task is to construct a well-defined set of rings of a molecular graph explicitly. We give a new algorithm for computing the set of *relevant cycles* of a graph.

Key words. chemical graphs, invariants, minimum cycle basis, algorithm for optimal set

AMS subject classifications. 68R10, 92C40, 92C42

1. Introduction.

Structural information about chemical compounds such as molecules is available in molecular databases whose number and size is growing rapidly. Consequently, the necessity for efficient computational methods to search these databases becomes increasingly important. Most available databases offer search and retrieval functionality for molecules by structure, i.e., the user draws or uploads a molecular graph S, and then the database is searched for molecules whose structure is identical to S or which contain substructures identical to S.

This identification involves the notorious (sub-)graph isomorphism problem for S for each molecule in the database: see e.g. [Coo71, GJ79a, KST93] and [Bab16] for recent developments. A common strategy to avoid solving many graph isomorphism problems exactly is to eliminate, in a preprocessing step, as many candidate structures as possible by comparing *graph invariants* known in this context as *molecular descriptors*. The most popular descriptors opinvolve the number of atoms and bonds and the different atom types. Graphs from the database whose invariants do not coincide with those of S (or do not allow a substructure isomorphic to S) are immediately rejected.

Other techniques to exclude candidate structures are similarity methods such as fragment-based search and finger-printing methods [WBD98]. In fragment-based search, the adjacency matrices of the two graphs are examined to compare the number of certain small but significant substructures (e.g. paths). Finger-printing means to derive a short vector from the graph which

[*] This is the pre-peer reviewed version of the following article: Berger, Gritzmann, de Vries, "Computing Cyclic Invariants for Molecular Graphs", Networks, 70:2 (2017) 116–131, which has been published in final form at doi:10.1002/net.21757. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Self-Archiving. Submitted to the Networks on Oct. 2016, accepted July 2017

Funding: The authors were supported by the DFG Schwerpunktprogramm Nr. 1126, grants no. GR 933/8-1, GR 933/8-2.

Prior publication: An early version of this work was part of the PhD-thesis by one of the authors [Ber04]; but already there this collaborative work is referred to. Furthermore, in the paper [BGdV09] the authors report of several applications of cycle bases and therefore the chemistry application is touched upon briefly; while some results from the present paper are quoted there, the reader is referred to the present paper for the full account and all proofs.

[†]VIA Software GmbH & Co KG, Robert-Bosch-Str. 30, 63303 Dreieich (berger@via-software.de)

[‡]Zentrum Mathematik, Technische Universität München, 80290 München, Germany (gritzmann@tum.de)

[§]Fb IV – Mathematik, Universität Trier, D-54286 Trier, Germany (devries@uni-trier.de).

encodes structural information, for instance, the number of paths up to a given length [JWD00]. For several other similarity indices, see [WBD98]. Recent applications in chemistry can be found in [Tol14], [MS14] and [May14].

If the structure of a given molecule is very simple, one might successively replace each cycle by a vertex and compare the resulting tree structure. However, this only works when the graph resulting from the sequence of contractions is independent of their order, which is rarely the case.

In the present paper we give efficient algorithms for computing graph invariants associated to ring substructures in the molecule. They are derived from a new algorithm for generating the set of *relevant cycles* of a graph. This term was introduced in [Plo71].

In the past, considerable research effort has been invested into methods for computing all "chemically meaningful" rings of a molecule. Since it is not exactly specified what this should mean, several different ring sets have been proposed [BFG$^+$04]. The most well-known is perhaps the *Smallest Set of Smallest Rings* (SSSR), see [GJ79b, DGHL89b, BP94]. This set is a minimum cycle basis of the model graph of the molecule. A specific set of rings which contains an SSSR and is, in addition, uniquely defined, is the set of *relevant cycles*, the union of all SSSRs, see [Plo71, Vis97, GLS03]. The set has exponential size in general, but for most molecular graphs (having small maximum degree) it can be determined with reasonable computational effort.

Our algorithm generates and classifies the relevant cycles with respect to an equivalence relation which also allows a detailed investigation of specific ring structures leading to two new invariants. These invariants, introduced in Section 2.2, can be computed in polynomial time; thus they can be used efficiently in practice even for larger instances.

The paper is organized as follows. The preliminary Section 2 introduces the relevant graph theoretic notation and reviews some general background material (2.1), discusses ring sets and corresponding cyclic invariants (2.2), and states the main results (2.3). Section 3 starts out with some mathematical tools (3.1), presents algorithms for computing the set of relevant cycles (3.2) and the set of essential cycles, the intersection of all SSSR's. Also it provides a polynomial-time algorithm for the first graph invariant $\vec{\epsilon}(G)$ (3.3). Section 4 studies a partitioning of the set of relevant cycles which allows us to devise in Section 5 a polynomial-time algorithm for the second graph invariant $\vec{\beta}(G)$. Section 6 shows how the new invariants operate on different molecular examples.

2. Structural and Algorithmic Preliminaries.

2.1. Definitions and Notation. Molecular graphs can be modeled as undirected multigraphs $G = (V, E)$. In the context of the present paper the term graph is always used as an abbreviation for multigraph.

We denote the number of edges of G by m, the number of vertices by n and set $V = \{v_1, \ldots, v_n\}$, $E = \{e_1, \ldots, e_m\}$.

Let $w : E \to (0, \infty)$ be a function which assigns strictly positive weights to the edges in E. The function w can, for instance, be used to distinguish different kinds of bonds [Fuj88, Fuj87, DGHL89a] such as single, double or triple bonds or carbon-carbon bonds versus carbon-noncarbon bonds. In the case that no such information is provided, we assume that $w(e) = 1$

for all $e \in E$. This will also be assumed in most examples throughout the paper.

A *cycle* $C = (V_C, E_C)$ is a graph in which every vertex has even degree. Cycles that differ only in vertices of degree 0 are indistinguishable with respect to E. As we are only interested in the cardinality and weight of the edges of cycles, we will assume henceforth that all vertices in V_C have even and *nonzero* degree. Then every cycle is uniquely determined by its edge set. (Note that this assumption does not exclude the empty graph.) We call a cycle *simple* if it is connected and if each vertex has degree two. A *cycle in* a given graph G is a subgraph of G which is a cycle. The most intuitive notion of a ring substructure corresponds to an induced subgraph that is a simple cycle.

Simple cycles can be regarded as important building blocks in the following sense.

Remark 1. *A sequence W of edges $(e_{i_1}, \ldots, e_{i_k})$ with the property that every one of its vertices is incident to an even number of edges (for example a closed walk) can be decomposed into a set of edge-disjoint simple cycles $D_1, \ldots, D_l$ and a collection of some pairs (f, f) with $f \in \{e_{i_1}, \ldots, e_{i_k}\}$ of those edges which appear multiple times in the sequence.*

A cycle $C = (V_C, E_C)$ in a graph $G = (V, E)$ is defined as a subgraph (rather than as a sequence of edges). Hence it can be identified with its edge-incidence vector $(b_i(C))_{i=1,\ldots,m}$, whose components are defined by:

$$b_i(C) = \begin{cases} 0, & \text{if } e_i \notin C \\ 1, & \text{if } e_i \in C. \end{cases}$$

The incidence vectors of cycles span a binary vector space $\mathcal{C}_{GF(2)}(G)$, the *cycle space* of the graph, subsequently abbreviated by $\mathcal{C}(G)$. The binary addition of two cycles C_1 and C_2 in $\mathcal{C}(G)$ corresponds to the symmetric difference of their edge sets $E(C_1)$ and $E(C_2)$:

$$C_1 \oplus C_2 = (E(C_1) \cup E(C_2)) \setminus E(C_1 \cap C_2)$$

and yields again a cycle. The following theorem is well known (see e.g. [Die97]).

Theorem 2. *The dimension of $\mathcal{C}(G)$ is $\mu(G) = m - n + \mathrm{c}(G)$, where $\mathrm{c}(G)$ is the number of connected components of G.*

The invariant $\mu := \mu(G)$ is called the *cyclomatic number* of G. In the case of molecules, $\mathrm{c}(G)$ is usually one. On the other hand, databases of molecular graphs often contain entries consisting of more than one molecule, for instance proteins or salts in solution. But as the single molecules in each entry may be treated separately, in the following we assume *connectivity* of all graphs. Furthermore, as this is trivial to verify, we will assume $n = n'$ and $m = m'$ for graphs G, G' that we investigate for isomorphism.

The *weight* of a cycle C with respect to w is defined as $w(C) := \sum_{e \in C} w(e)$, whereas its *length* is the number $|E(C)|$ of its edges. Note, that omitting zero-degree vertices (as we do) does neither change the weight nor the length of a cycle. More generally, the *weight* of a sequence of edges $W = (e_{i_1}, \ldots, e_{i_k})$ is defined as $w(W) := \sum_{l=1}^{k} w(e_{i_l})$ and its length is k.

A set of cycles is called *linearly independent* if their incidence vectors are linearly independent over $GF(2)$. A *cycle basis* $\mathcal{B}$ is a basis of $\mathcal{C}(G)$. It is called *minimum cycle basis* if its weight $w(\mathcal{B}) := \sum_{C\in\mathcal{B}} w(C)$ is minimal. The following lemma is well-known and easy to show, see e.g. [LS98].

Proposition 3. *Each cycle in a minimum cycle basis is simple.*

A cycle C is *relevant* if it belongs to some minimum cycle basis of G. In the following, $\mathcal{R}(G)$ denotes the set of relevant cycles in G. When there is no risk of confusion, we will simply write $\mathcal{R}$. A useful observation of [Vis97] characterizes $\mathcal{R}(G)$ as the set of simple cycles that cannot be represented as a sum of cycles of strictly smaller weight. More precisely:

Lemma 4 ([Vis97]). *Let G be a graph and let $C \in \mathcal{C}(G)$ be a simple cycle. C is relevant if and only if there do not exist simple cycles $C_1, \dots, C_k$ with the property that $C = C_1 \oplus \dots \oplus C_k$ and $w(C_i) < w(C)$ for all $i = 1, \dots, k$.*

The sets of linearly independent cycles in G constitute the independent sets of a matrix matroid; see e.g. [Oxl92]. This simple but important observation is used for many results in the following sections.

An immediate consequence is the fact that minimum cycle bases can be computed by the greedy algorithm, if the (potentially exponentially sized) list of all cycles were available. Moreover, if G is a graph and $\mathcal{B} = \{C_1, \dots, C_\mu\}$ is a minimum cycle basis of G ordered so that $w(C_1) \leq \dots \leq w(C_\mu)$, then the *ordered weight vector* $\vec{w}(G) = (w(C_1), \dots, w(C_\mu))$ of $\mathcal{B}$ is independent of the chosen minimum cycle basis: $\vec{w}(G)$ is a graph invariant.

Therefore, if G_1 and G_2 are two graphs with $\vec{w}(G_1) \neq \vec{w}(G_2)$, then they are not isomorphic, hence, in a database search for G_1, the graph G_2 should immediately be rejected. As an example take the non-isomorphic molecules G_1 and G_2 from Fig. 1[1] which have the same molecular formula and hence the same number of edges and vertices and also, since they are connected, the same cyclomatic number, but differ in their ordered weight vectors.

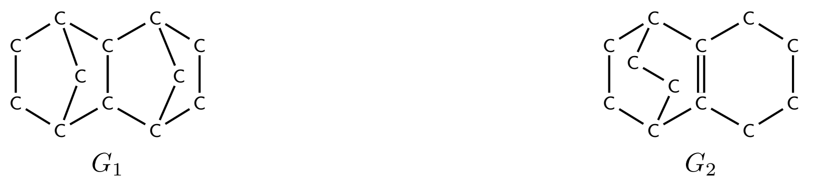


Fig. 1. *The molecular graphs of 1,4:5,8-decahydro-dimethanonaphthalene (G_1) and 1,2,3,4,5,6,7,8-octahydro-1,4-ethanonapthalene (G_2), both with the molecular formula $C_{12}H_{18}$. Assuming unit edge-weights, we have $\vec{w}(G_1) = (5,5,5,5)$ and $\vec{w}(G_2) = (2,6,6,6)$.*

$\vec{w}(G)$ can be computed in polynomial time. Currently, the fastest known algorithm for general graphs takes time $O(m^2 n/\log n + mn^2)$ [MM09]. If G is *planar*, a minimum cycle basis can be found in time $O(m^2 + n\log n)$, [HM93]. Note, however, that there are non-planar molecules (e.g. *Kuratowskiphane,* see [Kuc97]).

[1]The molecular graphs in this paper are represented schematically. Carbon, nitrogen or oxygen atoms are noted explicitly by their symbols C, N and O. Hydrogen atoms are left out almost entirely as is customary in chemistry and as they are irrelevant for the cyclic structure.

In the following sections, we denote the complexity of computing a sorted minimum cycle basis for arbitrary graphs by $\mathrm{MCB}(m,n)$ in terms of its asymptotics with respect to m and n, the number of edges and vertices.

2.2. Cyclic Graph Invariants. The graph invariant $\vec{w}(G)$ from the previous section is already quite effective. However it does not always distinguish two non-isomorphic molecules; see Fig. 2 for a first example. In the following, we study two invariants that provide additional information.

Similarly to $\vec{w}(G)$, the ordered weight vector of the set $\mathcal{R}(G)$ of relevant cycles is a graph invariant, that can be used to further refine $\vec{w}(G)$. However, since the number of relevant cycles can be exponentially large (see Fig. 3 for a *planar* construction), the union of all minimum cycle bases is in general not suitable for database search.

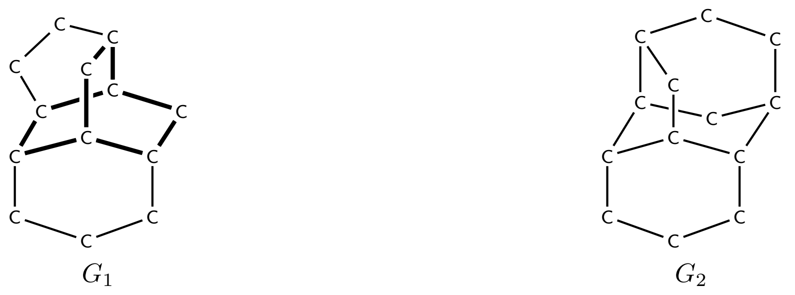


FIG. 2. *The two molecules G_1 and G_2 (rearrangements of tetracyclotridecane with the molecular formula $C_{13}H_{20}$) have $\vec{w} = (5,6,6,6)$. On the other hand, $\vec{\epsilon}(G_1) = (5,6)$, since of the three hexagons in the subgraph of G_1 formed by the bold edges, any two can be chosen and add up to the third, whereas $\vec{\epsilon}(G_2) = (5,6,6,6)$; here, all basic cycles are essential. Thus, $\vec{\epsilon}$ distinguishes the different molecules in this case.*

Instead, we consider the *intersection* of all minimum cycle bases. In the notation of [GLS00], we say that a cycle is *essential* if it is contained in this intersection and consequently belongs to *every* minimum cycle basis. Let $\mathcal{E}(G)$ denote the set of all essential cycles of G.

Note that $|\mathcal{E}(G)| \leq \mu$, and thus the ordered weight vector $\vec{\epsilon}(G)$ of essential cycles is a polynomially-sized graph invariant. In Section 3.3, we give a polynomial-time algorithm for computing $\vec{\epsilon}(G)$.

For an example of graphs where $\vec{\epsilon}(G)$ is stronger than $\vec{w}(G)$ see Fig. 2. However, as Fig. 4[1]) shows, there are also cases where $\vec{w}(G)$ is stronger than $\vec{\epsilon}(G)$. So, in general, neither $\vec{\epsilon}(G)$ nor $\vec{w}(G)$ dominate the other invariant.

To introduce the second graph invariant, let us take a closer look at $\mathcal{R}$. Since $\vec{w}$ is an invariant, the weight of every relevant cycle of G must be equal to the weight of some cycle in any fixed minimum cycle basis $\{C_1, \ldots, C_\mu\}$. Hence, the weights of relevant cycles can take

[1]These and other examples are are taken from PubChem, a public database provided by the US National Center for Biotechnology Information. It contains information on the biological activities of currently over 19 million small molecules, see http://pubchem.ncbi.nlm.nih.gov/

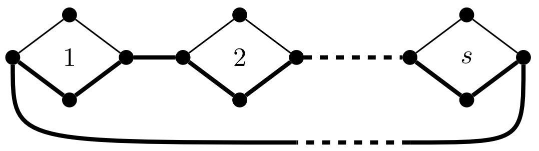
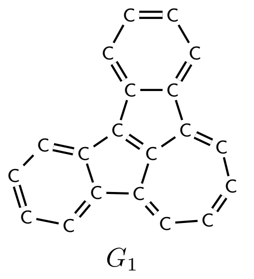


FIG. 3. *A minimum cycle basis of this graph consists of all quadrangles numbered from* 1 *to* s *and of any one of the* 2^s *cycles of length* $3s$ *which string all quadrangles together (one of them is indicated with bold edges). Hence the set of relevant cycles of* G *has exponential size.*



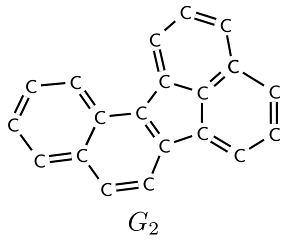


FIG. 4. G_1 *(identifier PubChem10245780) and* G_2 *(Benzo-12,13-fluoranthene) with molecular formula* $C_{20}H_{12}$*. We have* $\vec{w}(G_1) = (2,2,2,2,2,2,2,2,2,2,5,5,6,6,7) \neq (2,2,2,2,2,2,2,2,2,2,5,6,6,6,6) = \vec{w}(G_2)$*. On the other hand,* $\vec{\epsilon}(G_1) = \vec{\epsilon}(G_2) = (2,2,2,2,2,2,2,2,2,2)$*. The cycles of length greater than 2 in both graphs are not essential since we may add any of the 2-cycles to them to obtain different cycles of the same weights.*

values only from the set $\{w(C_1),\dots,w(C_\mu)\}$. Let a positive number κ be called *admissible* if $\kappa \in \{w(C_1),\dots,w(C_\mu)\}$.

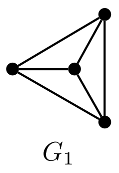
We consider the partition of $\mathcal{R}$ into sets of cycles with different weight κ. Every such set can be further partitioned into equivalence classes with respect to the *interchangeability relation* introduced by [GLS00, Gle01]. In the following, this latter relation is derived differently, using the concept of matroid connectivity. (Recall that the elements of the underlying matroid in our context are the cycles in the given graph or, equivalently, their incidence vectors.)

A *circuit* of a matroid is a minimal dependent set. For each element a of a matroid M let

$$\mathcal{A}(a) = \{a\} \cup \{b \in M : M \text{ has a circuit containing both } a \text{ and } b\}.$$

The connectivity relation $\sim$ is then defined by saying that $a \sim b$ iff $a \in \mathcal{A}(b)$. This is an equivalence relation, see e.g. [Oxl92], and its equivalence classes are called the *connected components* of M. Let us stress that connectivity in matroids is more akin to 2-connectedness in graphs than to 1-connectedness.

Now we introduce a refinement of a restriction of the connectivity relation $\sim$ on the cycle matroid of G. Let $\mathcal{R}_\kappa$ denote the submatroid on the subset of relevant cycles with admissible weight at most κ. We restrict the connectivity relation to cycles of weight exactly κ as follows. Two relevant cycles C, C' of weight κ are called *(κ)-interchangeable*, denoted by $C_1 \sim_\kappa C_2$, if



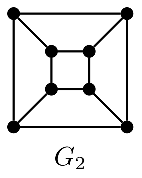


FIG. 5. *We have $\vec{\epsilon}(G_1) = \vec{\epsilon}(G_2) = ()$, but $\mu(G_1) = 3$ and $\mu(G_2) = 5$.*

there is a circuit in $\mathcal{R}_\kappa$ which contains both, i.e., if there is a minimal representation

$$C \oplus C' \oplus \bigoplus_{D \in \mathcal{I}} D = \mathbf{0}$$

of $\mathbf{0}$ with $\mathcal{I} = \emptyset$ or $\emptyset \neq \mathcal{I} \subseteq \mathcal{R}_\kappa$ and every non-trivial subfamily[1] of $\mathcal{I} \cup \{C, C'\}$ is linearly independent. The following lemma shows that the above definition is equivalent to [GLS00, Def. 6].

LEMMA 5. *Let C, C' be two relevant cycles of weight κ. Then $C \sim_\kappa C'$ if and only if there is a representation $C = C' \oplus \bigoplus_{D \in \mathcal{I}} D$, where $\mathcal{I} \subset \mathcal{R}_\kappa$ and the family $\mathcal{I} \cup \{C'\}$ is linearly independent.*

Proof. Since for $C = C'$ there is nothing to show, suppose that $C \neq C'$.

Let $C \sim_\kappa C'$. Then there is a circuit in $\mathcal{R}_\kappa$ of the form $C \oplus C' \oplus \bigoplus_{D \in \mathcal{I}} D = \mathbf{0}$. Minimality implies that all proper subfamilies of $\mathcal{I} \cup \{C, C'\}$ are linearly independent. Thus, by adding C to both sides we obtain a representation as claimed.

Conversely, suppose that $C = C' \oplus \bigoplus_{D \in \mathcal{I}} D$ is a representation as stated. Therefore in particular, $C, C' \notin \mathcal{I}$. By adding C to both sides we obtain the identity $0 = C \oplus C' \oplus \bigoplus_{D \in \mathcal{I}} D$. If the family $\mathcal{I} \cup \{C, C'\}$ is not minimally dependent, there exists a proper subfamily $\mathcal{I}'$ of $\mathcal{I} \cup \{C, C'\}$ of cycles with $\bigoplus_{D \in \mathcal{I}'} D = \mathbf{0}$. Of course, $C \in \mathcal{I}' \setminus \mathcal{I} \subset \{C, C'\}$.

Substituting for C, we get

$$0 = C \oplus C' \oplus \bigoplus_{D \in \mathcal{I}} D = \left(\bigoplus_{D \in \mathcal{I}' \setminus \{C\}} D \right) \oplus \left(C' \oplus \bigoplus_{D \in \mathcal{I}} D \right) = C' \oplus \bigoplus_{D \in \mathcal{I} \setminus \mathcal{I}'} D,$$

contradicting the linear independence of $\mathcal{I} \cup \{C'\}$. ∎

Obviously, essential cycles are not interchangeable with any other cycle in $\mathcal{R}$. Hence each essential cycle C forms its own equivalence class with respect to $\sim_\kappa$ where $\kappa = w(C)$. On the other hand, if $C \in \mathcal{R}$ is not essential, there always exists a relevant cycle C' such that C and C' are $\sim_\kappa$-interchangeable for $\kappa = w(C)$. In particular, if a minimum cycle basis $\mathcal{B}$ is fixed and if $C \in \mathcal{B}$ is not essential, there is a relevant *non-basis* cycle C' with $C' \sim_\kappa C$. Additionally, from each equivalence class $\mathcal{W}^\kappa$ with respect to $\sim_\kappa$, at least one cycle C is contained in $\mathcal{B}$. Moreover,

[1]To be more precise, if for $\mathcal{I} = \{D_1, \ldots, D_\ell\}$, we speak of the *family* $\mathcal{I} \cup \{C, C'\}$, we do refer to the family $(D_1, \ldots, D_\ell, C, C')$.

the number of cycles from each equivalence class $\mathcal{W}^\kappa \subseteq \mathcal{R}$ is the same for every minimum cycle basis $\mathcal{B}$. Hence we obtain (the slight generalization of) [GLS00, Thm. 11] to the weighted case:

Lemma 6. *Let $\kappa > 0$ be admissible, let $\mathcal{B}, \mathcal{B}'$ be two different minimum cycle bases and let $\mathcal{W}^\kappa$ be an equivalence class for $\sim_\kappa$. Then $|\mathcal{B} \cap \mathcal{W}^\kappa| = |\mathcal{B}' \cap \mathcal{W}^\kappa|$.*

Therefore one may call $|\mathcal{B} \cap \mathcal{W}^\kappa|$ the *relative rank* of $\mathcal{W}^\kappa$ in $\mathcal{R}$, in symbols $\operatorname{rank}_{\mathcal{R}_\kappa} \mathcal{W}^\kappa$. The relative ranks of the $\sim_\kappa$ equivalence classes for all admissible values $\kappa \in \mathbb{R}$, ordered by increasing κ and then by increasing class size form a graph invariant.

We use this relative-rank invariant to refine $\vec{w}(G)$: We take $\vec{w}(G)$ and separate its entries by vertical lines according to the $\sim_\kappa$ equivalence classes. The relative rank of each class corresponds to the number of entries between two vertical lines. This 'vector' will be called $\vec{\beta}(G)$; see Fig. 6 for an example.

The relative ranks describe how many cycles of equal weight in a minimum cycle basis are related with respect to interchangeability. In graphs where all cycles are disjoint, every cycle is essential and thus all relative ranks are 1. Thus, intuitively, $\vec{\beta}(G)$ measures the degree of interconnectivity or local density of G's cyclic structure.

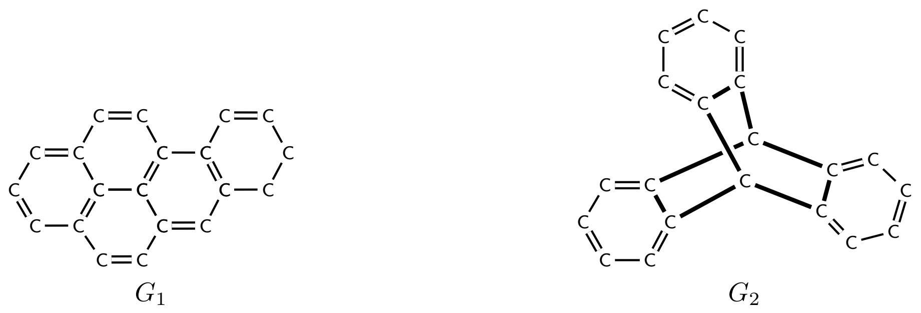


Fig. 6. *Molecules G_1 (7,8-Dihydrobenzo(a)pyrene) and G_2 (Identifier PubChem10225564), both $C_{20}H_{14}$, with $\vec{\beta}(G_1) = (2|2|2|\ 2|2|2|2|2|2|6|6|6|6|6|)$ and $\vec{\beta}(G_2) = (2|2|2|2|2|2|2|2|2|6|6|6|6, 6|)$. In the center part of G_2, any two of the three bold hexagons add up to the third, hence they are interchangeable. The relative rank of their equivalence class is two. In G_1, the relative rank of each hexagon is one.*

In general, $\vec{\beta}(G)$ refines the information of $\vec{w}(G)$ and $\vec{\epsilon}(G)$. Even though the graphs in Figure 6 have $\vec{w}(G_1) = \vec{w}(G_2)$ and $\vec{\epsilon}(G_1) = \vec{\epsilon}(G_2) = (2, 2, 2, 2, 2, 2, 2, 2, 2)$ they cannot be isomorphic because of $\vec{\beta}(G_1) \neq \vec{\beta}(G_2)$.

Now we state two important properties of relevant cycles which will be used in the next section. Here and in the following, we use the convention that in the basis representation of a cycle C by cycles of a fixed cycle basis $\mathcal{B}$, only cycles with non-zero coefficients are noted explicitly.

Lemma 7. *Let $\mathcal{B}$ be a minimum cycle basis of a graph G and let $C \in \mathcal{R}$ be a relevant cycle. Then the basis representation of C with respect to $\mathcal{B}$ contains at least one basis cycle of weight $w(C)$ and no cycles of weight strictly larger than $w(C)$.*

Proof. By Lemma 4 (and Proposition 3), C is relevant if and only if in every representation of C as a binary sum of cycles in G, at least one cycle has weight at least $w(C)$. Consequently, also the (unique) representation of a relevant cycle C as a sum of basis cycles cannot consist only of strictly shorter cycles. Further, if a basis cycle D of weight strictly greater than $w(C)$ were contained in this representation, it is possible to replace D by C in $\mathcal{B}$ to obtain a smaller cycle basis, contradicting the minimality of $\mathcal{B}$. ◻

The second property we will use is the subject of the following corollary.

Corollary 8. *Let $\mathcal{B}$ be a minimum cycle basis, and let C and C' be different relevant cycles of weight κ with $C \sim_\kappa C'$. Then there exists a representation $C = C' \oplus \bigoplus_{D\in\mathcal{I}} D$ where $\mathcal{I} \subset \mathcal{R}_\kappa$ and the family $\mathcal{I} \cup \{C'\}$ is linearly independent and in which all cycles with strictly smaller weight than κ belong to $\mathcal{B}$.*

Proof. Lemma 5 provides a representation $C = C' \oplus \bigoplus_{D\in\mathcal{I}'} D$, where $\mathcal{I}' \subset \mathcal{R}_\kappa$ and the family $\mathcal{I} \cup \{C'\}$ is linearly independent. We replace all cycles $D \in \mathcal{I}'$ with weight $w(D) < \kappa$ of this sum with their basis representations with respect to $\mathcal{B}$ and remove linearly dependent subsets. By Lemma 7, no cycle in these representations has weight greater than or equal to κ. Hence we obtain a representation of C with the asserted properties. ◻

2.3. Main Results. This paper presents the first polynomial time algorithms for computing the graph invariants $\vec{\epsilon}(G)$ and $\vec{\beta}(G)$ introduced in the previous section.

Clearly, $\vec{\epsilon}(G)$ and $\vec{\beta}(G)$ can be computed from the equivalence classes of $\sim$. A straightforward way to compute the two invariants would therefore be to generate the entire set of relevant cycles $\mathcal{R}$ (Algorithm 1) and then to partition it into $\sim_\kappa$ equivalence classes (Algorithm 3) for all relevant values of κ. The running time of Algorithm 1 is $O(|\mathcal{R}|mn^2)$ while Algorithm 3 requires time $O(mn(|\mathcal{R}|n + m^3))$. But as mentioned previously, $\mathcal{R}$ can be exponentially large. In fact, in the graph of Fig. 3 with $n = 4s$, all 2^s cycles of length $3s$ are relevant. Therefore we give direct algorithms, Algorithm 1 combined with Procedure 2, and Algorithm 5, which do not require the explicit construction of $\mathcal{R}$. They take time $O(\max\{\mathrm{MCB}(m,n), m^\omega, m^2n, mn^2\log n\})$ and $O(m^4n)$, respectively. Here, ω denotes the matrix multiplication constant (currently, $\omega \leq 2.3729$: see [Gal14]). Which of the four terms in the expression above dominates the others depends on the graph class.

In some applications, $\mathcal{R}$ will, however, be needed explicitly. The only other known algorithm for computing $\mathcal{R}$ needs time $O(m^4)+O(n|\mathcal{R}|)$, [Vis97]. While this algorithm is in general faster than Algorithm 1, the latter has two advantages.

First, Algorithm 1 permits a faster subsequent partition of $\mathcal{R}$ into $\sim_\kappa$ equivalence classes. To compute this partition, it is also possible to apply Vismara's algorithm and then use a coloring technique proposed in [GLS00]. The coloring part takes time $O(|\mathcal{R}|^2m^2)$ after $\mathcal{R}$ is constructed. Our algorithm, however, needs only polynomial time $O(m^4n)$. Hence the overall running time of $O(m^4n + |\mathcal{R}|mn^2)$ for obtaining all equivalence classes is faster than previous methods; in particular, it is *linear* in $|\mathcal{R}|$, which is the right order of magnitude in $|\mathcal{R}|$ given that $\mathcal{R}$ has to be listed.

The most important advantage of our algorithm is that it can be used to compute specific subsets of $\mathcal{R}$ directly: all relevant cycles of fixed weight, single $\sim_\kappa$-equivalence classes, or just the relative ranks of the equivalence classes; it is not clear how to achieve this with the previously known approaches.

3. Computing the Sets of Relevant and Essential Cycles. Before we study the sets of relevant and essential cycles in detail we provide some useful tools.

3.1. Some Tools. As it turns out, when looking at the linear (in)dependence of cycles, it suffices to restrict attention to the entries corresponding to the non-tree-edges of a fixed spanning tree of the graph. Henceforth we will assume that we have fixed a spanning tree $T = (V_T, E_T)$ for this purpose.

Let $\mathcal{B}$ be a cycle basis. The cycle-edge incidence matrix $A \in GF(2)^{\mu \times m}$ of $\mathcal{B}$ has as rows the (incidence vectors of the) cycles in $\mathcal{B}$. We denote the rows of A by $C_i, i = 1, \ldots, \mu$, and use the same notation for the corresponding cycles in $\mathcal{B}$. Further, let L be the square submatrix of A on the columns corresponding to the non-tree edges. Note that L is nonsingular. In fact, suppose L were singular and, hence, there were a subset of rows of L that sum up to $\mathbf{0}$. Since the sum of cycles is a cycle and T is cycle-free, the sum of the corresponding rows of A must be $\mathbf{0}$, too, contradicting the fact that $\mathcal{B}$ is a cycle basis.

Now, let U be the inverse of L, let $A^{(-i)}$ be the matrix that results from A after removing the i'th row C_i, and let u_i be the i-th column of U padded with $m - \mu$ zeros (in place of the tree-edges). Clearly each u_i is in the kernel of $A^{(-i)}$. Even more: it is the unique nonzero vector in the kernel of the matrix $A^{(-i)}$ with support restricted to the non-tree edges. As noted by [KMMP04], the vectors u_i, $i = 1, \ldots, \mu$, can be computed in time $O(m^\omega)$.

The following trivial remark gives an easy-to-check criterion for whether a cycle C can replace a basis cycle $C_i \in \mathcal{B}$, using u_i.

Remark 9. *Let $\mathcal{B}$ be a cycle basis. For a cycle $C_i \in \mathcal{B}$, let $A^{(-i)}$ be the incidence matrix of $\mathcal{B} \setminus \{C_i\}$, and let $u_i \in \ker A^{(-i)}$ be the vector defined above. A cycle $C \notin \mathcal{B}$ is linearly independent of the row-space of $A^{(-i)}$ if and only if $\langle C, u_i \rangle = 1$.*

Here, $\langle \cdot, \cdot \rangle$ denotes the standard inner product of $GF(2)^m$; we may extend it to sequences of edges $W = (e_{i_1}, \ldots, e_{i_k})$ by defining $\langle W, a \rangle := \bigoplus_{j=1}^{k} a_{i_j}$, where a_{i_j} denotes the component of a that corresponds to the edge e_{i_j}. (This definition ensures that we have $\langle W, a \rangle = \bigoplus_{j=1}^{l} \langle D_j, a \rangle$ for the decompositions of Remark 1.)

If, in addition to $\langle C, u_i \rangle = 1$, we have $w(C) = w(C_i)$; then C will be called *feasible* for C_i, and $\mathcal{R}_{C_i}$ will denote the set of all feasible cycles for C_i; this is the set of all cycles that could replace C_i in a minimum basis while preserving minimality. Note, that feasibility depends on the chosen minimum basis.

The following corollary characterizes the relevance of cycles through feasibility.

Corollary 10. *A cycle C is relevant if and only if it is feasible for some basis cycle C_i of $\mathcal{B}$.*

We will now show that, given the vector $u := u_i$, we can construct the set $\mathcal{R}_{C_i}$ by computing specific paths of weight $w(C_i)$ in an auxiliary graph G_u. The graph G_u is constructed from G as follows (for an example see Figure 7): The vertex set of G_u is $V \times GF(2)$. For each edge $e = \{x, y\}$ in G let u_e denote the component of u that corresponds to e, and add the two edges

$\{(x,0),(y,0\oplus u_e)\}$ and $\{(x,1),(y,1\oplus u_e)\}$ to G_u. Define a weight function $w_u : G_u \to (0,\infty)$ by assigning to each edge the weight $w(e)$ of the corresponding edge e in G it comes from. G_u has $2n$ vertices and $2m$ edges; it can be constructed in time $O(m)$. For a vertex v of G and a simple $(v,0)$–$(v,1)$ path P_v in G_u let $W(P_v)$ denote the closed walk in G obtained by replacing each vertex $(x,\delta) \in V(G_u)$ of P_v by $x \in V(G)$. For the next observation see [BGdV04].

LEMMA 11. *Let v be a vertex of G, and let P_v be a simple $(v,0)$–$(v,1)$ path in G_u. Then the closed walk $W(P_v)$ contains a simple cycle C in G with $\langle C,u\rangle = 1$.*

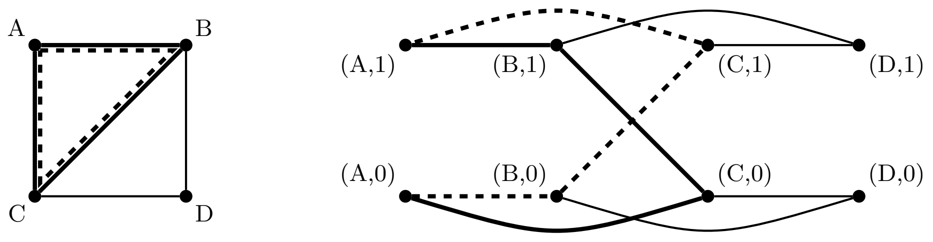


FIG. 7. *On the left, a graph G with unit weights is depicted. Let C_1, C_2 denote the two cycles $B-D-C-B$ and $A-B-D-C-A$, respectively, set $\mathcal{B} = \{C_1, C_2\}$, and let T be the tree $A-B-D-C$. Then $\mathcal{B}$ is the fundamental basis for T. Now, let the vector $u = u_1$ have the entry 1 precisely for the edge $B-C$. On the right the derived graph G_u is depicted (bold and dashed). The bold path and the dashed path $(A,0)-(A,1)$ in G_u both correspond to the same feasible cycle for C_1 in G.*

By Remark 9 it is clear that for each cycle that is feasible for C_i and $v \in C_i$ the construction produces a simple $(v,0)$–$(v,1)$ path in G_u. As a consequence of the following lemma, the converse is also true, i.e., any $(v,0)$–$(v,1)$ path of weight $w(C_i)$ in G_u corresponds to a feasible cycle for C_i.

LEMMA 12. *Let $\mathcal{B}$ be a minimum cycle basis, let C_i be a cycle of $\mathcal{B}$, $v \in V$, and let P_v be a simple $(v,0)$–$(v,1)$-path in G_u of weight $w(C_i)$. Then $W(P_v)$ is a simple cycle in G that is feasible for C_i.*

Proof. Suppose that $W(P_v)$ were not simple. By Lemma 11, there must then be a strictly shorter cycle corresponding to a subpath of P_v (as edge weights are assumed to be strictly positive) which is feasible for C_i. This contradicts the minimality of the cycle basis $\mathcal{B}$. ∎

All feasible cycles for C_i can therefore be generated by computing for each $v \in V$ all possible $(v,0)$–$(v,1)$ paths of weight $\kappa = w(C_i)$ in G_u. This can, for instance, be achieved by applying a k-shortest path algorithm for sufficiently large k to all pairs of vertices $\{(v,0),(v,1)\}$ in G_u. In fact, we stop it as soon as the next computed path is longer than κ.

Even though the number k of shortest paths necessary is not known a priorily, the use of Eppstein's k-shortest path algorithm permits to adjust k on the fly in time $O(kn+m+n\log n)$, see [Epp98]. Note that there are exactly two $(v,0)$–$(v,1)$ paths for every feasible cycle C through v, corresponding to its two different orientations (see Fig. 7). It thus follows from Lemma 12 that k is bounded by $2|\mathcal{R}|$.

3.2. The Set of Relevant Cycles. In this section, we give an algorithm for computing $\mathcal{R}$ by applying the results of Subsection 3.1.

Let $\mathcal{B} = \{C_1, \dots, C_\mu\}$ be a fixed minimum cycle basis. Using Corollary 10 and Lemma 12, we start with the incidence matrix A of $\mathcal{B}$, and determine, for $i = 1, \dots, \mu$, and with the help of $A^{(-i)}$, the set $\mathcal{R}_{C_i}$ of all feasible cycles for C_i. The basic scheme is given in Algorithm 1.

```
Input: Connected undirected edge-weighted graph $G$
Output: The set of relevant cycles $\mathcal{R}$
1: Compute a minimum cycle basis $\mathcal{B} = \{C_1, \dots, C_\mu\}$ of $G$.
2: Let $A$ be the cycle-edge incidence matrix of $\mathcal{B}$.
3: Compute kernel vectors $u_i$ of the submatrices $A^{(-i)}, i = 1, \dots, \mu$.
4: for $i = 1$ to $\mu$ do
5:   Set $\mathcal{R}_{C_i} := \emptyset$.
6:   For each $v \in V$, find all $(v,0)$–$(v,1)$ paths $P_v$ in $G_{u_i}$ with weight $w(C_i)$, and add all
     cycles $W(P_v)$ to the set $\mathcal{R}_{C_i}$.
7: end for
8: Output $\mathcal{R} := \bigcup_{i=1}^{\mu} \mathcal{R}_{C_i}$.
```

Algorithm 1: Computing the set of relevant cycles $\mathcal{R}$.

Note that the sets $\mathcal{R}_{C_i}, i = 1, \dots, \mu$, themselves are not invariant as they depend on the particular minimum cycle basis $\mathcal{B}$. However, as the next lemma shows, the sets $\mathcal{R}_{C_i}$ are compatible with the relation $\sim_k$.

LEMMA 13. *For $C, D \in \mathcal{R}_{C_i}$ of weight κ we have $C \sim_\kappa D$.*

Proof. $(\mathcal{B} \setminus \{C_i\}) \cup \{C\}$ and $(\mathcal{B} \setminus \{C_i\}) \cup \{D\}$ are optimal bases. Hence D can replace C in $(\mathcal{B} \setminus \{C_i\}) \cup \{C\}$. Therefore, there exists a dependency of C, D and other elements of $\mathcal{B} \setminus \{C_i\}$. If any of these other elements would be of larger weight than C and D, this would contradict optimality of $(\mathcal{B} \setminus \{C_i\}) \cup \{C\}$. Hence $C \sim_\kappa D$. ∎

THEOREM 14. *Algorithm 1 computes the set $\mathcal{R} = \bigcup_{i=1}^{\mu} \mathcal{R}_{C_i}$ of relevant cycles in time $O(\max\{MCB(m,n), m^\omega, m^2 n, mn^2 \log n\} + |\mathcal{R}| mn^2)$.*

Proof. Correctness of the algorithm follows from Remark 9 and Lemma 12. The time necessary for computing a minimum cycle basis is denoted by $\text{MCB}(m,n)$. As mentioned previously, the vectors u_i can be computed in time $O(m^\omega)$. This accounts for the first two terms.

At most $2|\mathcal{R}_{C_i}|$ paths of weight $w(C_i)$ are found for every vertex $v \in G$, corresponding to at most $|\mathcal{R}_{C_i}|$ cycles. Each such set of paths can be computed by the use of Eppstein's algorithm in time $O(n|\mathcal{R}_{C_i}| + m + n \log n)$. Since $|\mathcal{R}_{C_i}| \leq |\mathcal{R}|$ for all $i = 1, \dots, \mu$, the remaining terms are given by

$$\begin{aligned}\sum_{i=1}^{\mu} O(n(n|\mathcal{R}_{C_i}| + m + n \log n)) &= O(n^2 \sum_{i=1}^{\mu} |\mathcal{R}_{C_i}|) + O(mn(m + n \log n)),\\ &= O(n^2 m|\mathcal{R}| + m^2 n + mn^2 \log n).\end{aligned}$$ ∎

Note that the additional factor n in the above sum comes from considering all vertices v separately. Fig. 8 shows that, indeed, we cannot restrict the vertices to those belonging to C_i.

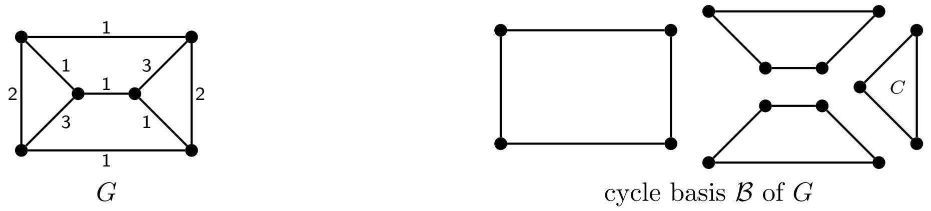


FIG. 8. *Graph G and cycle basis $\mathcal{B}$. The triangle C in $\mathcal{B}$ can be replaced by the other triangle in G. But the other triangle is disjoint from C.*

Clearly, it is possible to apply Algorithm 1 only to basis cycles of fixed weight κ, so that only certain subsets of $\mathcal{R}$ are generated. This may be of advantage if one is interested in specific substructures of a molecule, for instance all relevant carbon rings of a given length, say 6. Further, depending on the algorithm used for computing a minimum cycle basis, Step 1 may already provide the vectors u_i. Then Step 3 can be omitted

3.3. Computing the Set of Essential Cycles and the invariant $\vec{\epsilon}(G)$.

The set of essential cycles is a subset of $\mathcal{R}$. The following characterization generalizes [GLS00, Lemma 5] to the weighted case.

LEMMA 15. *Let $C_i \in \mathcal{B}$. Then $|\mathcal{R}_{C_i}| = 1$ if and only if C_i is essential.*

Proof. Clearly, if C_i is essential, then $|\mathcal{R}_{C_i}| = 1$.

Now, assume for a contradiction that $|\mathcal{R}_{C_i}| = 1$ but that there exists a different relevant cycle C with $C_i \sim_\kappa C$, where $\kappa = w(C_i)$. Of course, C can be expressed as the sum of cycles of some subset $\mathcal{B}'$ of the fixed minimum cycle basis $\mathcal{B}$ i.e., $C = \bigoplus_{D\in\mathcal{B}'} D$. If one of the elements of $\mathcal{B}'$ had weight greater than κ, then we would obtain a better basis by replacing it with C, contrary to the optimality of $\mathcal{B}$. If $C_i \in \mathcal{B}'$ then C would be feasible for C_i contrary to the assumption that $|\mathcal{R}_{C_i}| = 1$.

Because of $C_i \sim_\kappa C$, there exists a set $\mathcal{I}$ of relevant cycles of weight at most κ with $C = C_i \oplus \bigoplus_{D\in\mathcal{I}} D$ and $C_i \notin \mathcal{I}$. If each $D \in \mathcal{I}$ is expressed in terms of cycles from the basis, after cancellations, the representation $C = \bigoplus_{D\in\mathcal{B}'} D$ has to result. But then, the basis representation of at least one of the cycles $D \in \mathcal{I}$ involves C_i, as C_i is absent in the final sum. Hence there exists a set $\mathcal{B}'' \subset \mathcal{B}$ with $D = C_i \oplus \bigoplus_{F\in\mathcal{B}''} F$. By choice, $w(D) \leq \kappa$. If $w(F) > w(D)$ for some $F \in \mathcal{B}''$, a better basis could be obtained by replacing F by D. If $w(D) < w(C_i) = \kappa$, then the optimal basis could be improved by replacing C_i by D. Consequentially, $C_i \sim_\kappa D$ (and $C_i \neq D$) which contradicts $|\mathcal{R}_{C_i}| = 1$. ∎

Hence, if there is a basis cycle $C_i \in \mathcal{B}, i \in \{1, \ldots, \mu\}$, for which Algorithm 1 does not produce any cycle of weight $w(C_i)$ except copies of C_i itself, i.e., if $\mathcal{R}_{C_i} = \{C_i\}$, we know that this cycle is essential. Therefore, the set of essential cycles may be computed by a modification of the *for* loop in Algorithm 1 according to Procedure 2.

```
4: for $i = 1$ to $\mu$ do
5:     Set $\mathcal{R}_{C_i} := \emptyset$.
6:     For each $v \in V$ find the three shortest $(v, 0)$–$(v, 1)$ paths $P_v$ in $G_{u_i}$ with weight $w(C_i)$ if
       they exist, and add all cycles $W(P_v)$ to the set $\mathcal{R}_{C_i}$.
7:     If $\mathcal{R}_{C_i} = \{C_i\}$, mark $C_i$ as essential.
8: end for
```

Procedure 2: Replacing lines 4–8 of Algorithm 1 with this procedure gives an algorithm for computing the set of essential cycles.

Corollary 16. *With the* for *loop 4–8 of Algorithm 1 replaced by Procedure 2, the set of essential cycles and $\vec{\epsilon}$ may be computed in time $O(\max\{MCB(m, n), m^\omega, m^2 n, mn^2 \log n\})$.*

Proof. In Procedure 2, instead of computing all feasible cycles for a basis cycle C_i, we generate only the three shortest $(v, 0)$–$(v, 1)$ paths for each vertex $v \in G$ if they exist. This takes time $O(n(3n + m + n \log n)) = O(mn + n^2 \log n)$. For any fixed iteration i of the main loop, if $v \in C_i$, two of the paths correspond to the two different cyclic orders of C_i. We then have to check whether a third path P_v exists and whether it corresponds to a cycle $W(P_v)$ with strictly larger or equal weight than $w(C_i)$. In the latter case C_i is nonessential.

As, again, Fig. 8 shows, we also need to consider vertices $v \notin C_i$. For such a vertex it suffices to check whether a shortest $(v, 0)$–$(v, 1)$ path corresponds to a cycle of weight equal to or strictly larger than $w(C_i)$. If only cycles of strictly larger weight are found, we conclude by Lemma 15 that $\mathcal{R}_{C_i} = \{C_i\}$, and C_i is essential. Otherwise, C_i is not essential. The time necessary for the comparison is dominated by the terms for computing the cycles. Thus, the overall running time is $O(\max\{\mathrm{MCB}(m, n), m^\omega, m^2 n, mn^2 \log n\})$. ∎

4. Partitioning $\mathcal{R}$ into $\sim_\kappa$ Equivalence Classes. Now, we give an algorithm for partitioning the set of relevant cycles into $\sim_\kappa$-equivalence classes (see Algorithm 3).

Let $\mathcal{B} = \{C_1, \dots, C_\mu\}$ be a minimum cycle basis. By Corollary 10, every relevant cycle C is assigned to at least one of the sets of feasible cycles $\mathcal{R}_{C_i}, i = 1, \dots, \mu$, of equal weight, to $\mathcal{R}_{C_k}$, say. Hence, with $\kappa = w(C_k)$, cycle C belongs to the $\sim_\kappa$ equivalence class of the basis cycle C_k. It remains to determine which non-essential cycles from the minimum cycle basis $\mathcal{B}$ belong to a common $\sim_\kappa$ equivalence class, i.e., which sets $\mathcal{R}_{C_i}$ of cycles with equal weight can be merged if any.

This will be done in two steps. In Subsection 4.1, we treat the "easy" case that there are two sets of feasible cycles $\mathcal{R}_{C_j}$ and $\mathcal{R}_{C_k}$ with $\mathcal{R}_{C_j} \cap \mathcal{R}_{C_k} \neq \emptyset$ (and consequently, $w(C_j) = w(C_k)$). Lemma 13 and the transitivity of $\sim_\kappa$ imply that they belong to the same equivalence class. Merging these sets may still not yield the $\sim_\kappa$-equivalence classes, but only "pre-classes" as it may occur that sets of feasible cycles with the same weight and empty intersection nevertheless belong to the same equivalence class. This case is treated in Subsection 4.2.

4.1. Non-empty Intersection. Let us fix an admissible value $\kappa > 0$ and consider all basis cycles of weight κ. If only one such cycle exists, then $\mathcal{R}_{C_i}$ is already the entire $\sim_\kappa$ equivalence class. Therefore we assume that there are at least two different basis cycles of weight κ. We check for every pair C_j, C_k of them whether $\mathcal{R}_{C_j} \cap \mathcal{R}_{C_k}$ is non-empty. Then, by

Lemma 13 and the transitivity of $\sim_\kappa$, C_j and C_k belong to the same equivalence class. The intersection check can be done during the construction of the sets $\mathcal{R}_{C_i}$ in Algorithm 1. Using suitable data structures (like (a,b)-trees, see e.g. [MN99]) we then form the union of the two sets $\mathcal{R}_{C_j}$ and $\mathcal{R}_{C_k}$ as soon as a copy of a cycle $C \in \mathcal{R}_{C_j}$ is found in $\mathcal{R}_{C_k}$.

Now we describe a more efficient way to check whether the sets $\mathcal{R}_{C_j}$ and $\mathcal{R}_{C_k}$ for two different basis cycles C_j and C_k of equal weight have non-empty intersection. As we will see, for this purpose, $\mathcal{R}_{C_j}$ and $\mathcal{R}_{C_k}$ need not even be explicitly known. This fact will be utilized later in Subsection 4.3 and also in Section 5 for computing $\vec{\beta}(G)$. The next remark follows from the definition of the sets $\mathcal{R}_{C_i}$ and from Remark 9.

Remark 17. *Let $C_j, C_k \in \mathcal{B}$ be basis cycles of equal weight. Let $C \in \mathcal{R}$ be a cycle with $w(C) = w(C_j)$, and let u_j and u_k be the kernel vectors corresponding to C_j and C_k, respectively. Then the following two statements are equivalent:*

(i) $C \in \mathcal{R}_{C_j} \cap \mathcal{R}_{C_k}$.

(ii) $\langle C, u_j\rangle = 1$ and $\langle C, u_k\rangle = 1$.

A modification (proposed in a more general form by [CdV04]) of the construction leading to Lemmas 11 and 12 (cf. Fig. 7) allows us to compute a cycle with odd parity with respect to two kernel vectors u_j, u_k at once (see Fig. 9).

We define $G_{u_j,u_k} := (G_{u_j})_{u_k}$ by applying the construction of Subsection 3.1 to the graph G_{u_j} and the vector u_k. We obtain a graph with the vertex set $V \times GF(2)^2$ i.e., every vertex v of G is replaced by the four vertices $(v,(0,0))$, $(v,(0,1))$, $(v,(1,0))$, $(v,(1,1))$. Similarly, each edge $e = \{v, v\} \in G$ leads to four edges in G_{u_j,u_k}, namely $\{(v,y),(v', y \oplus (u_j(e), u_k(e))\}$ for each $y \in GF(2)^2$. The edge-weights $w(e)$ for $e \in G$ are carried over to all copies of e in G_{u_j,u_k}. For a path P in G_{u_j,u_k} let $W(P)$ denote the walk in G obtained by replacing each vertex (x,y) in P by x. Then the following generalization of Lemma 11 holds.

Lemma 18. *Let C be a simple cycle in G. Then the following two statements are equivalent.*

(i) $\langle C, u_j\rangle = 1 = \langle C, u_k\rangle$.

(ii) For every vertex $v \in C$ there exist two simple $(v,(0,0))$–$(v,(1,1))$ paths P_v and P'_v in G_{u_j,u_k} with $W(P_v) = W(P'_v) = C$ so that for all $x \in V \setminus \{v\}$, P_v and P'_v intersect the set $\{x\} \times GF(2)^2$ in at most one element and the set $\{v\} \times GF(2)^2$ exactly in $\{(v,(0,0)),(v,(1,1))\}$.

Proof. The backward direction is clear; so let us directly turn to the forward direction. Let (without loss of generality) $C = \{e_1, \ldots, e_r\}$ be a simple cycle in G satisying both, $\langle C, u_j\rangle = 1$ and $\langle C, u_k\rangle = 1$. Let $e_1 = \{v, v_1\}, \ldots, e_i = \{v_{i-1}, v_i\}, \ldots, e_r = \{v_{r-1}, v\}$ and set $z(e) = (u_j(e), u_k(e))$ for all edges e of G. We construct a path P_v in G_{u_j,u_k} according to the assertion. For e_1 choose the edge $\{(v,(0,0)),(v,(0,0) \oplus z(e_1))\}$. For every subsequent edge e_i in C take the edge $\{(v_{i-1}, y),(v_i, y \oplus z(e_i))\}$, where (v_{i-1}, y) (with $y \in GF(2)^2$) is the vertex reached in the previous step. Since C is simple, only one vertex from the set $\{(v_i, y) : y \in GF(2)^2\}$, $i = 2, \ldots, r-1$ appears in the path. Since $\langle C, u_j\rangle = 1$ and $\langle C, u_k\rangle = 1$, this procedure must end with the vertex $(v,(1,1))$.

The second path P'_v is derived from P_v by replacing each vertex $(x,y) \in P_v$ by the vertex $((x, y \oplus (1,1))$, and replacing the edges accordingly. ◻

While according to Lemma 18, simple cycles C with $\langle C, u_j\rangle = 1$ and $\langle C, u_k\rangle = 1$ lead to

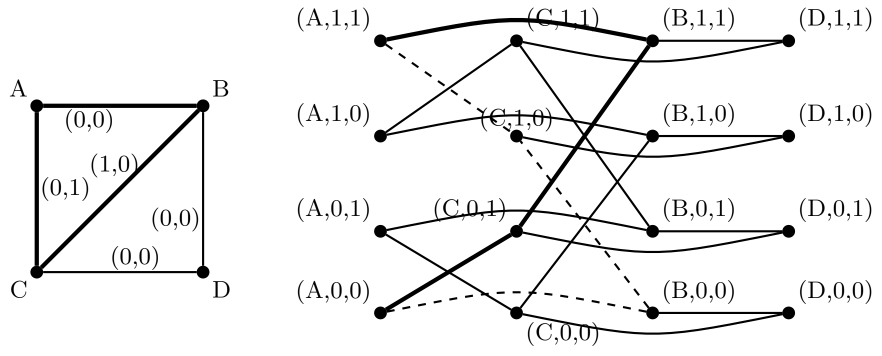


FIG. 9. *On the left the graph G of Fig. 7 is depicted again. As before, we use the spanning tree $A-B-D-C$ and the fundamental basis $\mathcal{B}=\{C_1,C_2\}$ consisting of the two cycles $B-C-D-B$ and $A-B-D-C-A$. If e_4 and e_5 denote the edges $B-C$ and $A-C$, we obtain $u_1=(0,0,0,0,1)$ and $u_2=(0,0,0,1,0)$. The corresponding edges e of G are labeled with the pairs $(u_1(e),u_2(e))$. On the right, the expanded Graph G_{u_1,u_2} is depicted. The bold $(A,(0,0))-(A,(1,1))$ path corresponds to the cycle $C=\{\{A,C\},\{C,B\},\{B,A\}\}$ in G which satisfies $\langle C,u_1\rangle=\langle C,u_2\rangle=1$. The dashed path correponds to the same cycle C.*

simple paths in G_{u_j,u_k}, not every simple $(v,(0,0))$–$(v,(1,1))$ path $P_v\in G_{u_j,u_k}$ must correspond to a simple cycle in G with the above property. However, as the next lemma shows, the full correspondence holds if the weight of P_v equals $w(C_j)$ (compare also Lemma 12).

LEMMA 19. *Let $w(C_j)\geq w(C_k)$, and let P_v be a simple $(v,(0,0))$–$(v,(1,1))$ path in G_{u_j,u_k}. Then P_v has weight at least $w(C_j)$. If P_v has weight equal to $w(C_j)$, then $W(P_v)$ is a simple cycle in G.*

Proof. For the first assertion, assume that P_v has weight strictly less than $w(C_j)$. By Remark 1, the closed walk $W(P_v)$ is the edge-disjoint union of some $\ell\in\mathbb{N}$ simple cycles $D_1,\dots,D_\ell$ with $w(C_j)>w(D_i)$ for $i=1,\dots,\ell$ and, possibly, some additional edge-pairs. As $\langle W(P_v),u_j\rangle=1$ there has to be an index i with $\langle D_i,u_j\rangle=1$. Since $w(C_j)>w(D_i)$, this contradicts the minimality of $\mathcal{B}$, as D_i is feasible for C_j by these properties.

Now, let P_v have weight $w(C_j)$. Using that all weights are positive, it follows analogously that $W(P_v)$ is a simple cycle. ∎

Lemma 18 enables us to check the existence of a cycle in $\mathcal{R}_{C_k}$ contained also in $\mathcal{R}_{C_j}$ by computing shortest $(v,(0,0))$–$(v,(1,1))$ paths in G_{u_j,u_k} for all $v\in G$. If no such path exists or if all such paths have weight strictly larger than κ, Lemma 19 implies that $\mathcal{R}_{C_j}\cap\mathcal{R}_{C_k}=\emptyset$.

Let us stress that Lemma 19 has been formulated in such generality that it is applicable for two basis cycles C_j,C_k of different weight. This will be used in Subsection 4.3.

4.2. Empty Intersection. Of course, if C_j and C_k do not belong to the same equivalence class with respect to $\sim_\kappa$ then $\mathcal{R}_{C_j} \cap \mathcal{R}_{C_k} = \emptyset$. The latter condition is, in general, only necessary for the former, not sufficient. [GLS00, Fig. 4] construct an example that has one interchangeability class $\mathcal{W}^\kappa$ with a partition $(\mathcal{D}_1, \mathcal{D}_2)$ into disjoint pre-classes where each $\mathcal{D}_i$ is equal to $\bigcup_{k \in \mathcal{I}_i} \mathcal{R}_{C_k}$ for non empty disjoint index sets $\mathcal{I}_i \subseteq \{1, \dots, \mu\}$, $i = 1, 2$. As a service to the reader, Fig. 10 provides an (even smaller) example.

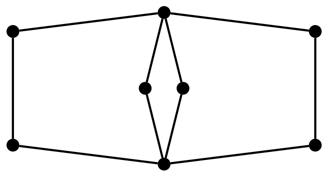


FIG. 10. *The depicted graph has 4 cycles of length 5 and one cycle of length 4; all of them are relevant. Every minimum cycle basis has to contain one of the left two 5-cycles, one of the right two 5-cycles and the central* 4 *cycle. Let* $\mathcal{B}$ *be the minimum cycle basis of this unweighted graph that consists of the three bounded faces of this drawing. Clearly, the left two* 5*-cycles lie in one pre-class; also, the right two* 5*-cycles lie in one pre-class. However, no* 5*-cycle on the left can be replaced by a* 5*-cycle on the right, so these two preclasses are distinct. Nevertheless, all four* 5*-cycles are* $\sim_5$*-equivalent, as they add up to* 0*, but every triple of them is linearly independent.*

We now explain how to check whether the cycles from two explicitly given pre-classes $\mathcal{D}_1$ and $\mathcal{D}_2$ of equal weight κ, but with empty intersection, belong to the same $\sim_\kappa$ equivalence class.

By transitivity, it is sufficient to determine whether an arbitrary pair $D_1 \neq D_2$ of cycles with $D_1 \in \mathcal{R}_{C_1} \subset \mathcal{D}_1$ and $D_2 \in \mathcal{R}_{C_2} \subset \mathcal{D}_2$ is interchangeable. By Lemma 15 we may assume that none of the classes $\mathcal{R}_{C_1}$ and $\mathcal{R}_{C_2}$ is essential. Hence we can choose *non-basic* cycles D_1 and D_2 for our investigation and consider their basis representations with respect to $\mathcal{B}$.

We distinguish the cases whether D_1 and D_2 have a common basis cycle C^* in their basis representations (Case I) or not (Case II). In the situation (I) we consider the cases (a) that $w(C^*) = w(D_1)$ and (b) that $w(C^*) < w(D_1)$, and show that D_1 and D_2 are always interchangeable.

Case (Ia): Suppose that the basis representations of the cycles D_1 and D_2, respectively, share a basis cycle C^* of weight $w(C^*) = \kappa$. Recall that $\mathcal{R}_{C^*}$ consists of all cycles of weight $w(C^*)$ that are linearly independent of $\mathcal{B} \setminus \{C^*\}$. Since basis representations are unique, D_1 and D_2 both have this property. Therefore $D_1 \in \mathcal{R}_{C_1} \cap \mathcal{R}_{C^*} \neq \emptyset$ and $D_2 \in \mathcal{R}_{C_2} \cap \mathcal{R}_{C^*} \neq \emptyset$, hence all three sets belong to the same pre-class, which implies $\mathcal{D}_1 = \mathcal{D}_2$.

Case (Ib): Suppose now that D_1 and D_2 have a common basis cycle C^* in their basis representations of weight *smaller* than κ. We show that D_1 and D_2 are interchangeable by deriving a representation according to Lemma 5.

By adding the two basis representations we obtain $D_1 \oplus D_2 = \bigoplus_{C \in \mathcal{K}} C$ for some $\mathcal{K} \subseteq$

$\mathcal{B} \setminus \{C^*\}$. Adding D_2 to both sides leads to

$$D_1 = D_2 \oplus \bigoplus_{C \in \mathcal{K}} C.$$

It remains to be shown that the family $\mathcal{K} \cup \{D_2\}$ is a linearly independent subset of $\mathcal{R}_\kappa$.

The basis cycles in $\mathcal{K}$ are certainly linearly independent. Further, since $C^* \notin \mathcal{K}$, not all cycles of the basis representation of D_2 are contained in $\mathcal{K}$. Hence no nonempty subset of $\{D_2\} \cup \mathcal{K}$ can add up to zero. Further, as the cycles in $\mathcal{K}$ belong to the basis representation of either D_1 or D_2, they have weight at most κ, and are relevant.

Case (II): Now we assume that the basis representations of D_1 and D_2 do not share a common basis cycle. Let us remark, first, that this case can actually occur, even if $D_1 \sim_k D_2$, see Fig. 11.

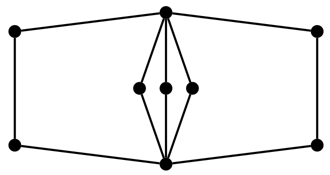


FIG. 11. *The graph generalizes the one from Fig. 10 and has 6 cycles of length 5 and three cycles of length 4; all of them are relevant. Every minimum cycle basis has to contain one of the left three 5-cycles, one of the right three 5-cycles and two of the three central* 4*-cycles. Let $\mathcal{B}$ be the minimum cycle basis of this unweighted graph that consists of the four bounded faces of this drawing. Again the left three* 5*-cycle lie in one pre-class and the right ones in another one. As before, all* 5*-cycles are $\sim_5$-equivalent. However, it is not true that the base-representation of every left and every right nonbasis cycle share an element. Consider, for instance, the left and right* 5*-gon that use the central of the three middle-paths. Their basis representation are the left 5- and 4-gon and the right 5- and 4-gon, respectively. So they have disjoint basis representations although they are $\sim_5$-equivalent.*

We will show in the next lemma, however, that if two cycles are interchangeable, there exists a *sequence* of pre-classes connecting the two, so that consecutive pre-classes either intersect, or have two representatives whose basis representations contain a shared shorter cycle.

Let, for $i \in \{1, \ldots, k\}$ in the following, $\mathcal{I}_i \subseteq \{1, \ldots, \mu\}$ be index sets such that $\mathcal{D}_i = \bigcup_{j \in \mathcal{I}_i} \mathcal{R}_{C_j}$ are disjoint pre-classes of cycles with weight κ. We define an auxiliary graph H_κ that has a vertex for each pre-class $\mathcal{D}_i$. Two vertices $\mathcal{D}_i$ and $\mathcal{D}_j$ of H_κ are joined by an edge if and only if there exist two cycles $D_i \in \mathcal{D}_i$ and $D_j \in \mathcal{D}_j$ whose basis representations share some cycle C^* with $w(C^*) < \kappa$.

By Case (Ib) and transitivity, it follows that if two sets, $\mathcal{D}_i$ and $\mathcal{D}_j$, are connected by a path in H_κ, they belong to the same $\sim_\kappa$ class. We now show the converse, namely that any two sets that belong to two different connected components of H_κ are not interchangeable. Let $\mathcal{K}_j$, $j = 1, \ldots, \mathrm{c}(H_\kappa)$ denote the connected components of H_κ. For simplicity (and in a slight abuse of notation) we write $C \in \mathcal{K}_j$ to indicate that C is a cycle that belongs to a vertex D of H_κ.

LEMMA 20. *Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be two different connected components of the graph H_κ defined above. Then for any two cycles $C_1 \in \mathcal{K}_1$ and $C_2 \in \mathcal{K}_2$ we have $C_1 \not\sim_\kappa C_2$.*

Proof. Let $C_1 \in \mathcal{K}_1$ and $C_2 \in \mathcal{K}_2$ be non-basis cycles of weight κ, and suppose for a contradiction that $C_1 \sim_\kappa C_2$.

Then, by Lemma 5 and Corollary 8, there is a representation

$$C_1 = C_2 \oplus \bigoplus_{Z \in \mathcal{L}} Z \oplus \bigoplus_{S \in \mathcal{S}} S,$$

where $\mathcal{L}$ consists of relevant cycles of (large) weight equal to κ and $\mathcal{S}$ is a subset of basis cycles with (small) weight strictly smaller than κ, such that the family $\mathcal{L} \cup \mathcal{S} \cup \{C_2\}$ is linearly independent. The set $\mathcal{L}$ can be partitioned into a set $\mathcal{L}_1 \subseteq \mathcal{K}_1$, a set $\mathcal{L}_2 \subseteq \mathcal{K}_2$ and a set $\mathcal{L}_3$ of other cycles of weight κ who belong to $\mathcal{K}_3, \dots, \mathcal{K}_p$. Then we can write the sum in the extended form

$$C_1 = C_2 \oplus \bigoplus_{Z \in \mathcal{L}_1} Z \oplus \bigoplus_{Z \in \mathcal{L}_2} Z \oplus \bigoplus_{Z \in \mathcal{L}_3} Z \oplus \bigoplus_{S \in \mathcal{S}} S. \tag{4.1}$$

Let us now consider the sets $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$ of basis cycles which appear in the basis representations of all cycles in the sets $\mathcal{L}_1, \mathcal{L}_2$ and $\mathcal{L}_3$, respectively. Since there is no edge in H_κ between different components, the sets $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$ are disjoint.

As $C_1 \in \mathcal{K}_1$, the basis representation of C_1 cannot contain cycles from $\mathcal{A}_2$ or $\mathcal{A}_3$. Hence C_1 can be obtained by adding only the cycles in $\mathcal{L}_1$ and possibly cycles from $\mathcal{S}$. Similarly, no cycle in $\mathcal{L}_1$ contributes to the basis representation of C_2. Further, by assumption, the basis representations of C_1 and C_2 do not share any cycle from $\mathcal{S}$.

But this implies that C_2 must be linearly dependent from $\mathcal{L} \cup \mathcal{S}$, a contradiction. Hence $C_1 \not\sim_\kappa C_2$. ∎

Consequently if $C_1 \sim_\kappa C_2$, then some cycle $D_2 \in \mathcal{L}_2 \cup \mathcal{L}_3$ is needed in the representation of C_1 that contributes with at least one of its basis cycles C^*. C^* is a basis cycle of C_1, or belongs to the basis representation of one of the cycles $D_1 \in \mathcal{L}_1$. Thus, $\mathcal{K}_1$ and $\mathcal{K}_2$, or $\mathcal{K}_1$ and a component $\mathcal{K}_i$, $i \in \{3, \dots, p\}$, respectively, are connected by an edge in H_κ. Hence, summarizing the arguments, we obtain the following corollary.

Corollary 21. *The connected components of H_κ form the different $\sim_\kappa$ equivalence classes.*

Here is another simple consequence.

Corollary 22. *If all cycles of $\mathcal{B}$ have the same weight κ, then disjoint pre-classes $\mathcal{D}_1$ and $\mathcal{D}_2$ cannot belong to the same $\sim_\kappa$ equivalence class.*

Proof. Using Lemma 20 and Case (Ib), two disjoint pre-classes $\mathcal{D}_1$ and $\mathcal{D}_2$ which belong to the same $\sim_\kappa$ equivalence class must contain two cycles C_1 and C_2 whose basis representations have a basis cycle of weight strictly smaller than κ, in common. But this is precluded by the assumption. ∎

4.3. An Algorithm for Computing the $\sim_\kappa$ Equivalence Classes. Corollary 21 leads to Algorithm 3 for computing the partition of $\mathcal{R}$ into $\sim_\kappa$ equivalence classes. We first generate the pre-classes by merging all sets $\mathcal{R}_{C_i}$ of cycles with the same weight with non-empty intersection by means of Lemma 19, and then determine the connected components of the graph H_κ from the previous subsection.

```
**Input:** Undirected edge-weighted graph $G$
**Output:** Partition of $\mathcal{R}$ into $\sim$ equivalence classes $\mathcal{W}_i, i = 1, \ldots, r$.
1: Apply Algorithm 1 to obtain a minimum cycle basis $\mathcal{B}$, kernel vectors $u_i, i = 1, \ldots, \mu$ and sets of feasible cycles $\mathcal{R}_{C_i}$ for all cycles $C_i \in \mathcal{B}$.
2: for each admissible weight $\kappa$ do
3:     Compute pre-classes $\mathcal{D}_\ell$, $\ell = 1, \ldots, p$ by merging all pairs, $\mathcal{R}_{C_i}, \mathcal{R}_{C_j}$ whenever $w(C_i) = w(C_j) = \kappa$ and $\mathcal{R}_{C_i} \cap \mathcal{R}_{C_j} \neq \emptyset$, using Lemma 19.
4:     Construct the graph $H_\kappa$ described in Subsection 4.2 and determine its connected components $\mathcal{K}_i$, $i = 1, \ldots, \mathrm{c}(H_\kappa)$.
5:     For $i = 1, \ldots, \mathrm{c}(H_\kappa)$ let $\mathcal{W}_i^\kappa$ denote the set of all cycles belonging to $\mathcal{K}_i$.
6: end for
7: Output $\mathcal{W}_i^\kappa$ for all $i$ and $\kappa$.
```

Algorithm 3: Assigning relevant cycles to $\sim_\kappa$ equivalence classes

The construction of H_κ can actually be carried out without computing the basis representation of every relevant cycle. Before we describe the correponding procedure in Subroutine 4 in detail, let us show first how Lemma 19 can be evoked again. In fact, with the shortest path computation of Subsection 4.1, we can check for every basis cycle C_k (of weight that is strictly smaller than κ) whether it belongs to the basis representation of some relevant cycle in different pre-classes as follows: Let $\mathcal{D}_i = \bigcup_{\ell \in I_i} \mathcal{R}_{C_\ell}$ for disjoint index sets I_i $(i = 1, 2)$ be pre-classes, and set $U_i := \bigcup_{\ell \in \{I_i\}} u_\ell$ for $i \in \{1, 2\}$. Now we compute the shortest $(v, (0, 0))$–$(v, (1, 1))$ paths for all $v \in G$ in the graphs $G_{u_k, u}$ and $G_{u_k, u'}$ for all $u \in U_1, u' \in U_2$. We obtain the following analogue to Remark 17.

Remark 23. *The following two statements are equivalent.*

(i) There exist $u_i \in U_1$, $u_j \in U_2$ and $v, w \in V$ such that there is a $(v, (0, 0))$–$(v, (1, 1))$ path P_v in G_{u_k, u_i} and a $(w, (0, 0))$–$(w, (1, 1))$ path P_w in G_{u_k, u_j}, both of weight $w(C_i)$.

(ii) The cycle C_k belongs to the basis representation of both, $W(P_v)$ and $W(P_w)$.

Remark 23 can be used in the following subroutine for constructing H_κ.

In conjunction with Subroutine 4, Algorithm 3 works correctly in time given in the following theorem.

Theorem 24. *Algorithm 3 (expanded by Subroutine 4) determines a partition of $\mathcal{R}$ into $\sim_\kappa$ equivalence classes. It requires time $O(m^4 n)$ after $\mathcal{R}$ is computed. The overall time complexity is in $O(m^4 n + |\mathcal{R}| m n^2)$.*

Proof. The correctness of Algorithm 3 follows from Lemmas 18, 19, Corollary 21, and Remark 23. The pre-classes can be computed in time

$$O(m^2(n(m + n \log n))) = O(\max\{m^3 n, m^2 n^2 \log n\}),$$

```
Input: $G$, $\mathcal{B}$, pre-classes $\mathcal{D}_\ell, \ell = 1, \dots, p$ of cycles with weight $\kappa$, kernel vectors $u_i, i = 1, \dots, \mu$.
Output: Graph $H_\kappa$
1:  Create a vertex for each pre-class $\mathcal{D}_\ell, \ell = 1, \dots, p$.
2:  for all pairs, $(i, j)$ with $1 \le i < j \le p$ do
3:    for each basis cycles $C_k$ of weight $w(C_k) < \kappa$ do
4:      Let $u_k$ be the kernel vector for $C_k$ and $U_i$ and $U_j$ the corresponding sets for $\mathcal{D}_i$ and
        $\mathcal{D}_j$.
5:      For each $v \in G$, compute shortest $(v,(0,0))$–$(v,(1,1))$ paths in $G_{u_i,u_k}$ for $u_i \in U_i$ and
        in $G_{u_j,u_k}$ for $u_j \in U_j$.
6:      if there exist two paths of weight $\kappa$ for some triple $(u_i, u_j, u_k)$ then
7:        add an edge in $H_\kappa$ between $\mathcal{D}_i$ and $\mathcal{D}_j$.
8:      end if
9:    end for
10: end for
11: Return $H_\kappa$.
```

Subroutine 4: Computing the graph H_κ

assuming that every pair of basis cycles $\{C_i, C_j\}$ has to be checked by computing n shortest paths in G_{u_i,u_j}.

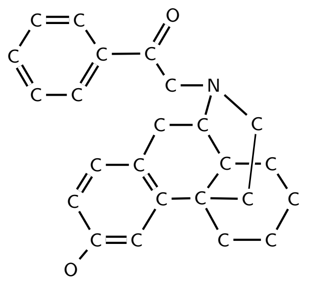
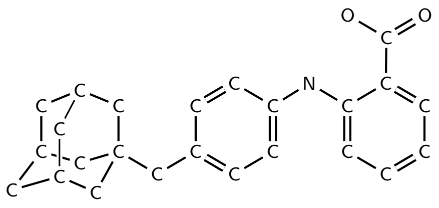
In the second step we examine disjoint pre-classes of cycles with the same weight following Lemma 23, again by determining for at most m^2 pairs of cycles up to $O(m)$ times n shortest paths, requiring time $O(m^3(n(m+n\log n)))$. This dominates the time necessary for computing and merging the connected components of H_κ, as H_κ has at most μ vertices. The overall time complexity is obtained by adding the running time of Algorithm 1 for computing $\mathcal{R} = \bigcup_{i=1}^{\mu} \mathcal{R}_{C_i}$; see Theorem 14. ∎

Note that $\mathcal{R}$ need not be explicitly available if we only want to assign the basis cycles to $\sim_\kappa$ equivalence classes. This will be used in the next section.

```
Input: Undirected edge-weighted graph $G$
Output: Invariant $\vec{\beta}(G)$.
1: Compute a minimum cycle basis $\mathcal{B}$ and kernel vectors, $u_i, i = 1, \dots, \mu$ with a preferred
   algorithm.
2: for each admissible weight $\kappa$ do
3:   Assign the basis cycles to pre-classes $\mathcal{D}_\ell$, $\ell = 1, \dots, p$ by generating for all pairs of cycles
     $C_i, C_j$ shortest paths in $G_{u_i,u_j}$ with Lemma 19.
4:   Construct the graph $H_\kappa$ from Subsection 4.2 with Subroutine 4.
5:   Determine, for each connected component of $H_\kappa$ their number of cycles from $\mathcal{B}$.
6: end for
7: Compute the size of each class and return $\vec{\beta}(G)$.
```

Algorithm 5: Computing the graph invariant $\vec{\beta}(G)$

Table 1
Different molecules $C_{24}H_{27}NO_2$ and their invariants

 1. Levophenacylmorphan. $(\|2_e\|2_e\|2_e\|2_e\|2_e\|2_e\|2_e\|6_e\|6_e\|6\|6\|6\|)$	 2.[1] $(2_e\|2_e\|2_e\|2_e\|2_e\|2_e\|2_e\|6\|6\|6,6,6\|)$

[1] N-(4-(1-adamantylmethyl)phenyl) Anthranilic acid

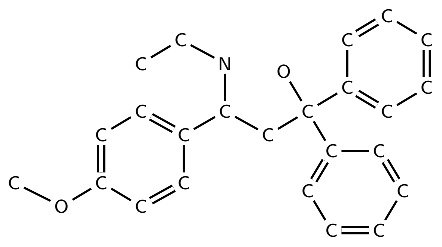
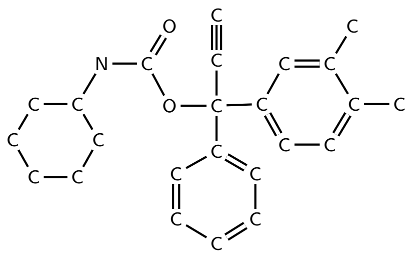
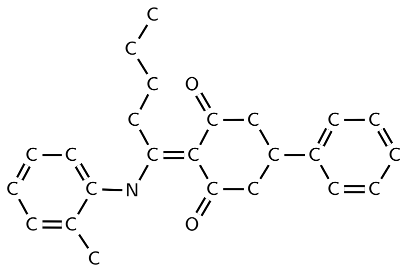
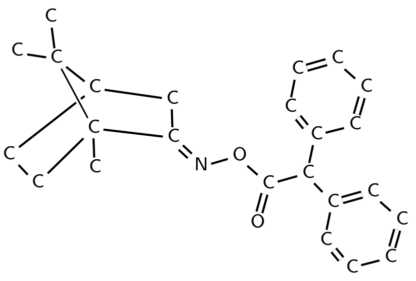
5. Computing the invariant $\vec{\beta}(G)$ efficiently. In the previous section, we have computed the $\sim_\kappa$ equivalence classes explicitly. The invariant $\vec{\beta}(G)$ can be derived from this partition. However, as mentioned before, the computation of $\vec{\beta}(G)$ does not rely on the explicit availability of the set $\mathcal{R}$ of all relevant cycles. In fact, with Lemmas 19 and 23, it is possible to assign only the basis cycles in $\mathcal{B}$ to $\sim_\kappa$ equivalence classes and to note their size. For the sake of completeness, we include the pseudo-code of the method in Algorithm 5.

Corollary 25. *$\vec{\beta}(G)$ can be computed in time $O(m^4 n)$.*

6. Molecular Examples. The assignment of minimum basis cycles to $\sim_\kappa$ equivalence classes contains more structural information than the cycle length vector of a minimum cycle basis alone. We illustrate this with a few examples of different molecules with the formula $C_{24}H_{27}NO_2$; see Tables 1 and 2. All information is concentrated in one vector explained below the corresponding molecular graph. It is based on $\vec{w}$, but the $\sim_\kappa$-equivalence classes are separated by vertical lines, and a subscript e indicates that the corresponding cycle is essential.

Note that, in Table 1, the Graphs 1 and 2 share the same vector $\vec{w}$. The different structure can, however, be detected by means of the information provided by $\vec{\epsilon}(G)$ and $\vec{\beta}(G)$. The same is true for the Graphs 3, 4 and 5 in Table 2.

Table 2
Different molecules $C_{24}H_{27}NO_2$ and their invariants

 3.[2] $(\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 6\vert 6\vert 6)$	 4. Carbamic acid.[3] $(2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2,2\vert 6_e\vert 6\vert 6\vert)$
 5.[4] $(\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 6_e\vert 6\vert 6\vert)$	 6.[5] $(2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 2_e\vert 5_e\vert 5_e\vert 6\vert 6\vert)$

[2] 1-,1-diphenyl-3-(ethylamino)-3-(p-methoxyphenyl)-1-Propanol
[3] N-cyclohexyl-,1-phenyl-1-(3,4-xylyl)-2-propynyl ester
[4] 2-[1-[(2-methylphenyl)amino]pentylidene]-5-phenyl-cyclohexane-1,3-dione
[5] [(1,7,7-trimethyl-2-bicyclo[2.2.1]heptylidene)amino] 2,2-diphenylacetate

REFERENCES

[Bab16] L. Babai. Graph isomorphism in quasipolynomial time. *arXiv:1512.03547v2*, 2016.

[Ber04] F. Berger. *Minimum Cycle Bases in Graphs.* PhD thesis, Zentrum Mathematik der Technischen Universität München, 2004.

[BFG$^+$04] F. Berger, C. Flamm, P.M. Gleiss, J. Leydold, and P.F. Stadler. Counterexamples in chemical ring perception. *J. Chem. Inf. Sci.*, 44:323–331, 2004.

[BGdV04] F. Berger, P. Gritzmann, and S. de Vries. Minimum cycle bases for network graphs. *Algorithmica*, 40:51–62, 2004.

[BGdV09] F. Berger, P. Gritzmann, and S. de Vries. Minimum cycle bases and their applications. In J. Lerner, D. Wagner, and K.A. Zweig, editors, *Algorithmics of Large and Complex Networks*, volume 5515 of *LNCS*, pages 34–49. Springer, 2009.

[BP94] R. Balducci and R.S. Pearlman. Efficient solution of the ring perception problem. *Chem. Inf. Comput. Sci.*, 34:822–831, 1994.

[CdV04] E. Cheng and S. de Vries. Separating multi-oddity constrained shortest circuits over the polytope of stable multisets. *OR-Letters*, 32(2):181–184, 2004.

[Coo71] S.A. Cook. The complexity of theorem-proving procedures. *Proc. 3rd ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[DGHL89a] Geoffrey M. Downs, Valerie J. Gillet, John D. Holliday, and Michael F. Lynch. Review of ring perception algorithms for chemical graphs. *J. Chem. Inf. Comput. Sci.*, 29:172–187, 1989.

[DGHL89b] G.M. Downs, V.J. Gillet, J.D. Holliday, and M.F. Lynch. Review of ring perception algorithms for chemical graphs. *J. Chem. Inf. Comput. Sci.*, 29:172–187, 1989.

[Die97] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics.* Springer Verlag New York Berlin Heidelberg, 1997.

[Epp98] D. Eppstein. Finding the k shortest paths. *SIAM J. Computing*, 28(2):652–673, 1998.

[Fuj87] S. Fujita. A new algorithm for selection of synthetically important rings: the Essential Set of Essential Rings for organic structures. *J. Chem. Inf. Comput. Sci.*, 28:78–82, 1987.

[Fuj88] S. Fujita. Logical perception of ring opening, ring closure, and rearrangement reactions based on imaginary transition structures. selection of the Essential Set of Essential Rings. *J. Chem. Inf. Comput. Sci.*, 28:1–9, 1988.

[Gal14] F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303. ACM, 2014.

[GJ79a] M.R. Garey and D.S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness.* Freeman, New York, 1979.

[GJ79b] J. Gasteiger and C. Jochum. An algorithm for the perception of synthetically important rings. *J. Chem. Inf. Comput. Sci.*, 19:43–48, 1979.

[Gle01] P.M. Gleiss. *Short cycles - Minimum cycle bases of graphs from Chemistry and Biochemistry.* PhD thesis, Institut für theoretische Chemie, Universität Wien, 2001.

[GLS00] P.M. Gleiss, J. Leydold, and P.F. Stadler. Interchangeability of relevant cycles in graphs. *Electronic J. Comb.*, 7:# R16, 2000.

[GLS03] P.M. Gleiss, J. Leydold, and P.F. Stadler. Circuit bases of strongly connected digraphs. *Discussiones Math. Graph Th.*, 23:241–260, 2003.

[HM93] D. Hartvigsen and R. Mardon. When do short cycles generate the cycle space? *J. Comb. Theory, Ser. B*, 57:88–99, 1993.

[JWD00] C.A James, D. Weininger, and J. Delaney. Daylight Theory Manual – Daylight 4.8. http://www.daylight.com, 2000.

[KMMP04] T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch. A faster algorithm for minimum cycle bases of graphs. In *Automata, Languages and Programming*, volume 3142 of *LNCS*, pages 846–857, 2004.

[KST93] J. Köbler, U. Schöning, and J. Torán. *The Graph Isomorphism Problem: Its Structural Complexity.* Springer Verlag Boston, New York, 1993.

[Kuc97] D. Kuck. Centrohexacyclic or K_5 molecules — development of a growing class of topologically nonplanar organic compounds. (6):1043–1057, 1997.

[LS98] J. Leydold and P. F. Stadler. Minimal cycle basis of outerplanar graphs. *Electronic J. Comb.*, 5:# R16, 1998.

[May14] John W. May. *Cheminformatics for genome-scale metabolic reconstructions*. PhD thesis, University of Cambridge, 2014.

[MM09] K. Mehlhorn and D. Michail. Minimum cycle bases: Faster and simpler. *ACM Trans. Alg.*, 6(1):8:1–8:13, 2009.

[MN99] K. Mehlhorn and S. Näher. *LEDA - A platform for combinatorial and geometric computing*. Cambridge University Press, 1999.

[MS14] J.W. May and C. Steinbeck. Efficient ring perception for the chemistry development kit. *J. Cheminformatics*, 6(1):3, 2014.

[Oxl92] J.G. Oxley. *Matroid Theory*. Oxford Graduate Texts in Mathematics. Oxford University Press, 1992.

[Plo71] M. Plotkin. Mathematical basis of ring-finding algorithms in CIDS. *J. Chem. Documentation*, 11:60–63, 1971.

[Tol14] J.M.C. Toledo. *An investigation into the structural basis for nucleic acid small molecule binding*. PhD thesis, Carleton University Ottawa, 2014.

[Vis97] P. Vismara. Union of all the minimum cycle bases of a graph. *Electronic J. Comb.*, 4:# R9, 1997.

[WBD98] P. Willet, J.M. Barnard, and G.M. Downs. Chemical similarity searching. *J. Chem. Inf. Comput. Sci.*, 38:983–996, 1998.