A decomposition theory for vertex enumeration of convex polyhedra

Arne C. Reimers[1,2,3], Leen Stougie[4,5]

April 23, 2014

Abstract

In the last years the vertex enumeration problem of polyhedra has seen a revival in the study of metabolic networks, which increased the demand for efficient vertex enumeration algorithms for high-dimensional polyhedra given by inequalities. In this paper we apply the concept of branch-decomposition to the vertex enumeration problem of polyhedra $P = \{x : Sx = b, x \geq 0\}$. Therefore, we introduce the concept of k-module and show how it relates to the separators of the linear matroid generated by the columns of S. This then translates structural properties of the matroidal branch-decomposition to the context of polyhedra. We then use this to present a total polynomial time algorithm for polytopes P for which the branch-width of the linear matroid generated by S is bounded by a constant k.

1 Introduction

Given a polyhedron P described by inequalities, the vertex enumeration problem asks for the enumeration of the vertices of P. With the advent of genome scale metabolic networks in the field of computational systems biology, vertex enumeration and variants thereof became an important biological analysis tool. However, the size of genome scale metabolic networks (1000 or more reactions) implies high dimensional polytopes (≈ 1000 dimensions) and in turn computational challenges. Current methods for vertex enumeration in computational systems biology [26, 12] employ a variant of the double description method [15] and are only applicable for small or medium-scale networks. Pivoting based enumeration algorithms [7, 1, 23] on the other hand are unsuitable due to the large amount of degenerate vertices.

While it is possible to show that for non-degenerate polyhedra pivoting based algorithms exist that run in *total polynomial time* (i.e. the run time is bounded by a polynomial function of the input and output size) [1], it is a long standing and basic open question if there exists an algorithm that enumerates the vertices of polytopes (bounded polyhedra) given by inequalities in total polynomial time.

Indeed, the difficulty to obtain polynomial output-sensitive bounds has fundamental complexity theoretic causes. Khachiyan et al. showed that vertex enumeration of general polyhedra is NP-hard [13] and hence, it is not possible to enumerate the vertices of (unbounded) polyhedra in total polynomial time unless $\mathbf{P} = \mathbf{NP}$ [4].

In this paper we approach the vertex enumeration problem from a novel, decomposition based, perspective. Therefore, we extend the concept of *flux module*, previously introduced for specific enumeration problems on metabolic networks [16, 17] to k-modules for the decomposition of arbitrary polyhedra and even non-convex subsets of $\mathbb{R}^n$. In Sec. 2 we will show how k-modules correspond to k-separators from *matroid theory* [20]. This opens the connection to *branch-width* [10, 22], a concept related to the better known concept of tree-width [9, 21, 11]. Tree-width has been introduced on graphs and measure how close a graph is to a tree. If the tree-width is assumed to be a constant parameter, then many famous **NP**-hard graph optimization problems, like maximum independent set, can be solved in polynomial time.

[1] Department of Mathematics and Computer Science, Freie Universität Berlin , Arnimallee 6, 14195 Berlin, Germany
[2] International Max Planck Research School for Computational Biology and Scientific Computing (IMPRS-CBSC), Max Planck Institute for Molecular Genetics, Ihnestr 63-73, D-14195 Berlin, Germany
[3] Berlin Mathematical School (BMS), Berlin, Germany
[4] Centre for Mathematics and Computer Science (CWI), Science Park 123, 1098 XG Amsterdam, The Netherlands
[5] Operations Research, VU University, De Boelelaan 1085, 1081 HV, Amsterdam, The Netherlands

Branch-width, which we will define for independent reading in Section 3, has similar implications. The concept of branch-width extends to matroids in a straight-forward way [10]. Hence, by considering the linear matroid generated by the columns of S, we can apply it to decompose the coefficient matrix S of any polyhedron $P = \{x : Sx = b, x \geq 0\}$.

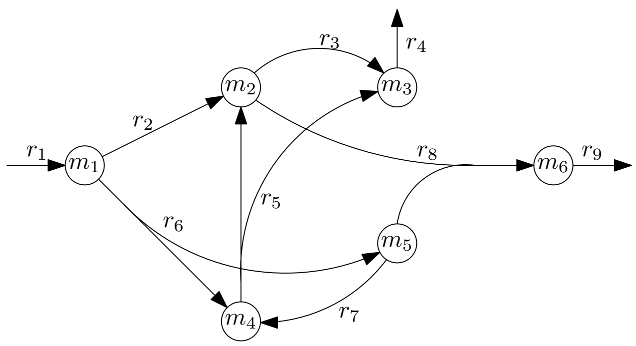
This relationship has already been used by Cunningham and Geelen [6] to show the tractability for a class of integer programs. In this paper, we show that branch-width is also a highly useful complexity measure for polytopes. Therefore, we present a total polynomial time algorithm for vertex enumeration if the branch-wdith of the underlying matroid is bounded by a constant k.

We close the introduction with a brief introduction to metabolic networks in Section 1.1 to motivate our terminology and definitions. Motivated by metabolic networks, we define k-modules in Sec. 2 and characterize them using matroid theory. These results are then used in Section 3 to translate branch-decompositions of matroids into polyhedral theory to derive the central result of this paper: a total polynomial time vertex enumeration algorithm for polytopes of which the coefficient matrix has low branch-width (bounded by a constant).

1.1 Metabolic Networks - a Biological Motivation

The work was motivated by the analysis of metabolic networks. We will briefly introduce metabolic networks as an illustration of a nice application area, but also to give sense to the definition of *modules* of polyhedra, which will become, in an extended form, crucial in the decomposition of polyhedra in Section 3.

A metabolic network is defined on a set of reactions $\mathcal{R}$ and metabolites $\mathcal{M}$. A metabolic network is used to model the chemical reactions that can happen inside a biological cell. It is defined by a stoichiometric matrix $S \in \mathbb{R}^{\mathcal{M} \times \mathcal{R}}$. Each reaction $r \in \mathcal{R}$ corresponds to a column S_r of S. The entry S_{mr} of S_r for $m \in \mathcal{M}$ denotes how much of the metabolite m is taken up ($S_{mr} < 0$) or produced ($S_{mr} > 0$) by r. Most entries are 0, signifying that the metabolite is not part of the reaction. We notice that $\mathcal{M}, \mathcal{R}$ should be understood as index sets that also name the entries of the matrix. In particular, even if sets $A, B \subseteq \mathcal{R}$ satisfy $|A| = |B|$ but $A \neq B$, we will distinguish the spaces $\mathbb{R}^A$ and $\mathbb{R}^B$. For simplicity, we assume here that all reactions are irreversible, i.e. all reaction speeds are positive.



$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

Figure 1: An example network and its corresponding stoichiometric matrix S. Note, that the stoichiometric coefficients are in general not only $-1, 0$ or 1, but can be arbitrary rational numbers.

The *flux space* of a metabolic network is then defined as the polyhedron

$$P = \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b, x \geq 0\}. \tag{1}$$

In general, in metabolic network studies almost all coefficients of b are assumed to be 0, modeling steady state of the network. Non-zero coefficients of b allow to model additional constraints like fixed nutrient uptake rate or a target level of yield. In that sense, the vectors $x \in P$ are a generalization of flows in directed graphs.

Note, that since we did not pose any constraints on the structure of S, every polyhedron can be formulated as the flux space of a metabolic network, possibly by splitting variables and adding slack variables.

One of the main computational challenges in metabolic network analysis is the enumeration of *elementary flux modes* (EFM) [25], which can be considered a variation of enumeration of the vertices and rays of the flux space polyhedron. In particular for flux spaces $\{x \in \mathbb{R}^{\mathcal{R}} : Sx = 0, x \geq 0\}$ EFM enumeration is equivalent to enumerating the vertices of the polytope $\{x \in \mathbb{R}^{\mathcal{R}} : Sx = 0, \sum_{i \in \mathcal{R}} x_i = 1, x \geq 0\}$.

For a certain class of flux spaces we observed that, in practice, the flux space often exhibits a structure that allows the decomposition of P into what we baptised as flux modules [16].

Before we repeat the definition of flux module, we introduce some additional notation. We use index notation to restrict to a subset of rows/columns. I.e. $x_i, i \in \mathcal{R}$ will denote the component of x at index i, x_A will denote the subvector of x obtained by retaining the components in $A \subseteq \mathcal{R}$. Observe that $x_i = (x_A)_i$ for all $i \in A \subseteq \mathcal{R}$ due to the naming property of $\mathcal{R}$. Also, we write S_A to denote the submatrix with all columns in A.

Definition 1 [Flux Module, [16]] $A \subseteq \mathcal{R}$ is a flux module if there exists a $d \in \mathbb{R}^{\mathcal{M}}$ s.t. $S_A x_A = d$ for all $x \in P$. We call d the interface flux of the module. □

If we are given a partition $\mathcal{X}$ of $\mathcal{R}$ into flux modules, then this reduces enumeration of $\mathcal{V}(P)$, the set of vertices of P, to vertex enumeration of the composing flux spaces $P^A = \{x \in \mathbb{R}^A : S_A x = d, x \geq 0\}$, $A \in \mathcal{X}$:

Theorem 1 ([16]) *Let* $P = \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b, x \geq 0\}$, $b \neq 0$. *Let* $\mathcal{X} = \{A_1, \dots, A_n\}$ *be a partition of* $\mathcal{R}$ *into flux modules, then*

$$\mathcal{V}(P) = \prod_{A \in \mathcal{X}} \mathcal{V}(P^A) := \left\{x \in \mathbb{R}^{\mathcal{R}} : x_A \in \mathcal{V}(P^A)\ \forall A \in \mathcal{X}\right\}.$$

In [17] we found that global properties of the space P play a minor role for the characterization of modules. To be more precise it suffices to look at a single point y inside P and consider a neighbourhood of it. This neighbourhood captures all the characteristics that are needed to analyse modularity of the whole space. To remove unnecessary dimensions, we define the set $\mathcal{Q}$ of variable elements

$$\begin{aligned} \mathcal{Q} &:= \{r \in \mathcal{R} : x_i^{\max} \neq x_i^{\min}\}, \text{ where} \\ x_i^{\max} &:= \sup\{x_i : x \in P\} \\ x_i^{\min} &:= \inf\{x_i : x \in P\} \end{aligned} \tag{2}$$

and obtained the following result, where we use $\ker(S)$ to denote the kernel of matrix S:

Theorem 2 ([17]) *If* $\bar{P} \subseteq \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b\}$, *is convex, it holds for all* $A \subseteq \mathcal{R}$

$$A \text{ is } \bar{P}\text{-module} \Leftrightarrow A \cap \mathcal{Q} \text{ is } \ker(S_{\mathcal{Q}})\text{-module.}$$

To ensure the existence of a point y inside $\bar{P}$, we assumed convexity of $\bar{P}$. We will later see that convexity is not necessary. Matroid theory then formed the key to understand $\ker(S_{\mathcal{Q}})$-modules and also delivered very efficient methods for computing flux modules.

Theorem 3 ([17]) $A \subseteq \mathcal{R}$ *is a* $\ker(S)$*-module if and only if* A *is a separator in the matroid represented by* S. □

Although, as we observed, these results work very well for a certain class of flux spaces of metabolic networks, the decomposition theorem becomes unpractical for more general cases, since no flux modules are found. Thus, also for decomposition of polyhedra in general this notion is too restricted to be of more general interest. Therefore, we introduce the notion of k-module in the next section as a generalization of flux module. It will allow us to generalize the above decomposition Theorem 1 in Section 3.

2 k-Modules

In this section we will use P not only to denote a polyhedron, but very often as a subset of a polyhedron (in the previous section we used the notation $\bar{P}$ for it). The precise meaning will always be made clear.

Definition 2 (*k*-module) Let $P \subseteq \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b\}$. $A \subseteq \mathcal{R}$ is a *k-module* of P if there exists a $d \in \mathbb{R}^{\mathcal{M}}$ and a $D \in \mathbb{R}^{\mathcal{M} \times k}$ s.t.

$$\forall x \in P\ \exists \alpha \in \mathbb{R}^k : S_A x_A = d + D\alpha.$$

We call d the constant interface vector of the module and D the variable interface matrix. If we can choose $d = 0$ we will say that the k-module is a *linear* k-module. □

Notice that the flux-modules of the previous section are equivalent to 0-modules in this definition. Biologically, we can understand k-modules as follows: In addition to the fixed function (the interface flux d), k-modules also allow additional variable functions (spanning a k-dimensional space). Since biological subsystems often have several side functions, this increases the applicability significantly. In other application areas, we can understand a k-module as a subsystem that only has few (k) interactions to the rest of the system. Without proof, we give some observations, which may help the reader to get some intuition for the notion.

Observation 1 *Let $P \subseteq \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b\}$.*

(i) $A \subseteq \mathcal{R}$ is a flux-module iff A is a 0-module;

(ii) Every set A with k elements is a (linear) k-module; in particular every $i \in \mathcal{R}$ is a (linear) 1-module;

(iii) A is a $(k-1)$-module $\Rightarrow$ A is a k-module;

(iv) The property of k-module is not closed under disjoint union, as is the case with the property of 0-module;

(v) Let $B \subseteq \mathcal{R}$ be a 0-module. It holds for all $A \subseteq \mathcal{R} \setminus B$ that A is a k-module if and only if $A \dot\cup B$ is a k-module;

(vi) $\emptyset$ is a k-module for all $k \geq 0$.

We observe that for a given k-module A the variable interface D is not unique (unless $k = 0$). However, the linear space spanned by D, defined by $\operatorname{span}(D) := \{D\alpha : \alpha \in \mathbb{R}^k\}$ is unique, if k is chosen to be minimal, in which case k is the dimension of $\operatorname{span}(D)$.

Proposition 1 *Let $P \subseteq \mathbb{R}^{\mathcal{R}}, A \subseteq \mathcal{R}$ and let k be minimal s.t. A is a k-module of P. Let D, D' be two different variable interfaces of A. Then $\operatorname{span}(D) = \operatorname{span}(D')$.*

Proof Assume $\operatorname{span}(D) \neq \operatorname{span}(D')$. By definition of variable interface, it follows that

$$\begin{aligned} &\forall x \in P : S_A x_A \in d + \operatorname{span}(D) \\ &\forall x \in P : S_A x_A \in d + \operatorname{span}(D') \\ \Rightarrow &\forall x \in P : S_A x_A \in d + \operatorname{span}(D) \cap \operatorname{span}(D'). \end{aligned} \tag{3}$$

Since $\operatorname{span}(D) \neq \operatorname{span}(D')$, it follows that $\operatorname{span}(D) \cap \operatorname{span}(D') \subset \operatorname{span}(D)$. Hence,

$$\dim(\operatorname{span}(D) \cap \operatorname{span}(D')) < \dim(\operatorname{span}(D)).$$

It follows that there exists a $D'' \in \mathbb{R}^{\mathcal{M} \times \ell}$ with $\ell < k$ and $\operatorname{span}(D'') = \operatorname{span}(D) \cap \operatorname{span}(D')$. By (3) it follows that D'' is a variable interface of A and hence, k was not minimal; a contradiction. ■

2.1 Restriction to Linear Vector Spaces

Similar to the case of flux modules, also for k-modules we can restrict ourselves to the analysis of linear vector spaces:

Theorem 4 *Let $P \subseteq \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b\}, b \in \mathbb{R}^{\mathcal{M}}$. Then holds for all $A \subseteq \mathcal{R}$ that*

$$A \text{ is a } P\text{-}k\text{-module} \Leftrightarrow A \cap \mathcal{Q} \text{ is } \ker(S_{\mathcal{Q}})\text{-}k\text{-module},$$

where $\mathcal{Q}$ is the set of variables with non-constant values, as defined in (2). □

We observe that this theorem simplifies the characterization of P-k-modules substantially. It is not only sufficient to analyze linear vector spaces, but we also only need to work with linear k-modules:

Observation 2 *Let $P \subseteq \{v \in \mathbb{R}^{\mathcal{R}} : Sx = b\}$. If $0 \in P$, then holds for all $A \subseteq \mathcal{R}$ that*

$$A \text{ is a linear } P\text{-}k\text{-module} \Leftrightarrow A \text{ is a } P\text{-}k\text{-module}.$$

PROOF $\Rightarrow$: By definition.

$\Leftarrow$: Since $0 \in P$ it follows that there exists a $\alpha \in \mathbb{R}^k$ with $0 = S_A 0_A = D\alpha + d$. Hence, $d = D(-\alpha)$, which completes the proof. ∎

We will derive lemmas from which the proof of Theorem 4 will follow. The first lemma, using $\operatorname{aff}(P)$ to denote the affine hull of P, shows why we do not need a convexity assumption on P:

Lemma 1 *Let $P \subseteq \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b\}$. It holds for all $A \subseteq \mathcal{R}$ that*

$$A \text{ is a } P\text{-}k\text{-module} \Leftrightarrow A \text{ is a } \operatorname{aff}(P)\text{-}k\text{-module.}$$

The variable and constant interfaces are the same.

PROOF We only show that if A is a P-k-module, then A is also an $\operatorname{aff}(P)$-k-module. The other direction is trivial since $P \subseteq \operatorname{aff}(P)$.

Let D be the variable interface matrix and d the constant interface vector of the P-k-module A. Let $x \in \operatorname{aff}(P)$ be arbitrary but fixed. Hence, there exist $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ and $x^1, \dots, x^n \in P$ for some $n \in \mathbb{N}$ s.t.

$$x = \sum_{j=1}^{n} \lambda_j x^j, \qquad 1 = \sum_{j=1}^{n} \lambda_j.$$

Since $x^j \in P$, there exists an $\alpha^j \in \mathbb{R}^k$ with $S_A x_A^j = D\alpha^j + d$ for every $j = 1, \dots, n$. It follows that

$$\begin{aligned} S_A x_A &= \sum_{j=1}^{n} \lambda_j S_A x_A^j = \sum_{j=1}^{n} \lambda_j (D\alpha^j + d) \\ &= D \sum_{j=1}^{n} \lambda_j \alpha^j + d \sum_{j=1}^{n} \lambda_j = D \sum_{j=1}^{n} \lambda_j \alpha^j + d. \end{aligned}$$

This concludes the proof. ∎

We can simplify the space that we have to analyze even further if the relative interior w.r.t. $\ker S$ is non-empty:

Lemma 2 *Let $P \subseteq \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b\}$. If there exists $y \in P$ and $\epsilon > 0$ s.t. $y + w \in P$ for all $w \in \ker S$ with $\|w\|_\infty < \varepsilon$ (i.e. the relative interior of P is non-empty), then for all $A \subseteq \mathcal{R}$*

$$A \text{ is a } P\text{-}k\text{-module} \Leftrightarrow A \text{ is a } \ker(S)\text{-}k\text{-module.}$$

PROOF $\Leftarrow$: Let $x^1 \in P$ be arbitrary but fixed. We define $d = S_A x_A^1$. Let $x^2 \in P$ be arbitrary. By definition of P, we have $x^1 - x^2 \in \ker S$. Since A is a $\ker(S)$-k-module, there exists by Obs. 2 an $\alpha \in \mathbb{R}^k$ such that $S_A(x_A^2 - x_A^1) = D\alpha$, where D is the variable interface of the $\ker(S)$-k-module A. Thus, $S_A x_A^2 = S_A x_A^1 + D\alpha = d + D\alpha$ and A is an affine P-k-module.

$\Rightarrow$: Assume A is a P-k-module. Hence, there exist $d \in \mathbb{R}^k, D \in R^{\mathcal{M} \times k}$ s.t. for all $x \in P$ there exists an $\alpha \in \mathbb{R}^k$ s.t. $S_A x_A = d + D\alpha$. In particular, there exists $\alpha^y \in \mathbb{R}^k$ s.t. $S_A y_A = d + D\alpha^y$.

For a proof by contradiction we assume that A is not a $\ker(S)$-k-module. Hence, there exists $w \in \ker(S)$ s.t. for all $\alpha \in \mathbb{R}^k$ we have $S_A w_A \neq D\alpha$. By definition of y, there exists an $\varepsilon > 0$ s.t. $y + \varepsilon w \in P$. We conclude that

$$\begin{aligned} & S_A(y_A + \varepsilon w_A) = S_A y_A + \varepsilon S_A w_A \neq d + D\alpha^y + D\alpha && \text{for all } \alpha \in \mathbb{R}^k. \\ \Rightarrow\ & S_A(y_A + \varepsilon w_A) \neq d + D\alpha && \text{for all } \alpha \in \mathbb{R}^k. \end{aligned}$$

This is a contradiction. ∎

To obtain a polyhedron with non-empty relative interior, we start with a preliminary simple lemma, which shows that modules stay invariant under the operation of projection onto the space $\mathcal{Q}$ of non-constant variables, as defined in (2). We write $\mathrm{pr}_A P$ to denote the projection of a set $P \subseteq \mathcal{R}$ on the variables in A, i.e. $\mathrm{pr}_A P := \{v_A : v \in P\}$.

Lemma 3 *It holds for all $A \subseteq \mathcal{Q}$, where $\mathcal{Q}$ is defined as in* (2)*:*

$$A \textit{ is (linear) } P\text{-}k\text{-}\textit{module} \Leftrightarrow A \textit{ is (linear) } \mathrm{pr}_{\mathcal{Q}} P\text{-}k\text{-}\textit{module}.$$

Proof By definition of projection, we have the following equivalence (for all $d \in \mathbb{R}^{\mathcal{M}}$, $D \in \mathbb{R}^{\mathcal{M}\times k}$):

$$\begin{aligned} S_A x_A &= d + D\alpha && \forall x \in P \exists \alpha \in \mathbb{R}^k \\ \Leftrightarrow S_A x_A &= d + D\alpha && \forall x \in \mathrm{pr}_{\mathcal{Q}} P \exists \alpha \in \mathbb{R}^k \end{aligned}$$

since $A \subseteq \mathcal{Q}$. ■

We use this lemma to prove the following stronger version:

Lemma 4 *Let $P \subseteq \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b\}$ and $\mathcal{Q}$ be defined as in* (2)*. It holds for all $A \subseteq \mathcal{R}$*

$$A \textit{ is } P\text{-}k\text{-}\textit{module} \Leftrightarrow A \cap \mathcal{Q} \textit{ is } \mathrm{pr}_{\mathcal{Q}} P\text{-}k\text{-}\textit{module}.$$

Proof Clearly, $A \setminus \mathcal{Q}$ is a 0-module. Hence, by Obs. 1(v) it follows that A is a P-k-module if and only if $A \cap \mathcal{Q}$ is a P-k-module. By Lemma 3 it follows that $A \cap \mathcal{Q}$ is a P-module iff $A \cap \mathcal{Q}$ is a $\mathrm{pr}_{\mathcal{Q}} P$-module and hence, the lemma follows. ■

Theorem 4 now follows directly from the previous lemmas.

Proof (Theorem 4) Let $x^{\max}, x^{\min}$ and $\mathcal{Q}$ be defined as in (2). Define $d := b - S_{\mathcal{R}\setminus\mathcal{Q}} x^{\max}_{\mathcal{R}\setminus\mathcal{Q}}$. By definition of $\mathcal{Q}$, it follows that $\mathrm{pr}_{\mathcal{Q}} P \subseteq \{x : S_{\mathcal{Q}} x = d\}$. By Lemma 1 it follows for $A \subseteq \mathcal{R}$ that $A \cap \mathcal{Q}$ is a $\mathrm{pr}_{\mathcal{Q}} P$-$k$-module iff $A \cap \mathcal{Q}$ is a $\mathrm{aff}(\mathrm{pr}_{\mathcal{Q}} P)$-$k$-module. We observe that $\mathrm{aff}(\mathrm{pr}_{\mathcal{Q}} P) = \{x : S_{\mathcal{Q}} x = d\}$ and hence $\mathrm{aff}(\mathrm{pr}_{\mathcal{Q}} P)$ has non-empty relative interior. By Lemma 2 it follows that $A \cap \mathcal{Q}$ is a $\mathrm{pr}_{\mathcal{Q}} P$-$k$-module iff $A \cap \mathcal{Q}$ is a $\ker(S_{\mathcal{Q}})$-k-module. By Lemma 4 it follows that A is a P-k-module iff $A \cap \mathcal{Q}$ is a $\ker(S_{\mathcal{Q}})$-k-module. ■

2.2 Matroid Theory for k-Modules

The previous subsection showed that we can restrict ourselves to the analysis of linear vector spaces of the form $P = \ker S$. We will show now that k-modules have a counterpart in matroid theory in the form of separators.

Definition 3 (k-separator, [20]) Let M be a matroid on the element set $\mathcal{R}$. A set $A \subseteq \mathcal{R}$ is a k-separator if and only if

$$\mathrm{rank}(A) + \mathrm{rank}(\mathcal{R} \setminus A) - \mathrm{rank}(\mathcal{R}) < k.$$

Theorem 5 *$A \subseteq \mathcal{R}$ is a* $(\ker S)$*-k-module if and only if A is a $k+1$-separator in the matroid M represented by S.* □

We will now prove Theorem 5, which connects k-modules to $(k+1)$-separators. The proof goes along the same lines as the proof we gave in [17] of Theorem 3. We reuse Lemma 5 that we already showed in [17] and repeat its proof for completeness sake.

Lemma 5 ([17]) *Let $A \subseteq \mathcal{R}$ and $S \in \mathbb{R}^{\mathcal{M}\times\mathcal{R}}$. Then holds*

$$\dim(S_A \mathrm{pr}_A \ker(S)) = \dim(\ker(S)) - \dim(\ker(S) \cap X^{\perp}) - \dim(\ker(S) \cap X),$$

where $X = \{x \in \mathbb{R}^{\mathcal{R}} : x_i = 0 \ \forall i \notin A\}$ and $X^{\perp} = \{x \in \mathbb{R}^{\mathcal{R}} : x_i = 0 \ \forall i \in A\}$.

Proof Define $L = \mathrm{pr}_A(\ker(S))$ and consider the map

$$\mathrm{pr}_A : \ker(S) \to L$$

By the fundamental theorem on homomorphisms it follows that $\dim(L) = \dim(\ker(S)) - \dim(\ker(S) \cap \ker(\mathrm{pr}_A))$. Observe that $\ker(\mathrm{pr}_A) = X^\perp$. Hence, we get

$$\dim(L) = \dim(\ker(S)) - \dim(\ker(S) \cap X^\perp).$$

We can identify $L \subseteq \mathbb{R}^A$ with $L \times 0^{\mathcal{R} \setminus A} \subseteq \mathbb{R}^{\mathcal{R}}$. Observe that

$$\dim(S(L)) = \dim(S_A(L)) = \dim(S_A \mathrm{pr}_A \ker(S)).$$

It follows again by the fundamental theorem on homomorphisms that

$$\dim(S(L)) = \dim(L) - \dim(L \cap \ker(S)).$$

With the identification, we also observe that $L = X \cap \ker(S)$. We conclude

$$\begin{aligned} \dim(S_A \mathrm{pr}_A \ker(S)) &= \dim(S(L)) = \dim(L) - \dim(L \cap \ker(S)) \\ &= \dim(\ker(S)) - \dim(\ker(S) \cap X^\perp) - \dim(L \cap \ker(S)) \\ &= \dim(\ker(S)) - \dim(\ker(S) \cap X^\perp) - \dim(X \cap \ker(S)), \end{aligned}$$

which concludes the proof. ■

Lemma 6 *$A \subseteq \mathcal{R}$ is a k-module if and only if $\dim(S_A \mathrm{pr}_A \ker S) \leq k$.*

Proof We have

$$\begin{aligned} & \dim(S_A \mathrm{pr}_A \ker S) \leq k \\ \Leftrightarrow\quad & \exists D \in \mathbb{R}^{\mathcal{M} \times k} : S_A \mathrm{pr}_A \ker S \subseteq \{D\alpha, \alpha \in \mathbb{R}^k\} \\ \Leftrightarrow\quad & \exists D \in \mathbb{R}^{\mathcal{M} \times k} : \{S_A x_A : x \in \ker S\} \subseteq \{D\alpha, \alpha \in \mathbb{R}^k\} \\ \Leftrightarrow\quad & \exists D \in \mathbb{R}^{\mathcal{M} \times k}\ \forall x \in \ker S\ \exists \alpha \in \mathbb{R}^k : S_A x_A = D\alpha, \end{aligned}$$

which concludes the proof. ■

Proof (Thm. 5) It follows from the definition of rank that

$$\begin{aligned} \dim(\ker S) &= |\mathcal{R}| - \mathrm{rank}(\mathcal{R}), \\ \dim(\ker S \cap X) &= |A| - \mathrm{rank}(A), \\ \dim(\ker S \cap X^\perp) &= |\mathcal{R} \setminus A| - \mathrm{rank}(\mathcal{R} \setminus A). \end{aligned}$$

By Lemma 6 a set $A \subseteq \mathcal{R}$ is a k-module if and only if

$$\begin{aligned} & k \geq \dim(S_A \mathrm{pr}_A \ker S) \\ \Leftrightarrow\ & k \geq \dim(\ker S) - \dim(\ker S \cap X^\perp) - \dim(\ker S \cap X) && \text{(by Lemma 5)} \\ \Leftrightarrow\ & k \geq (|\mathcal{R}| - \mathrm{rank}(\mathcal{R})) - (|A| - \mathrm{rank}(A)) - (|\mathcal{R} \setminus A| - \mathrm{rank}(\mathcal{R} \setminus A)) \\ \Leftrightarrow\ & k \geq \mathrm{rank}(A) + \mathrm{rank}(\mathcal{R} \setminus A) - \mathrm{rank}(\mathcal{R}) \end{aligned}$$

Hence, A is a $k+1$-separator if and only if A is a k-module. ■

2.3 Finding k-Modules

Summarizing the results from the previous two subsections, we conclude that a polyhedron $P = \{v \in \mathbb{R}^{\mathcal{R}} \mid Sv = b, v \geq 0\}$ contains a k-module if and only if the linear matroid with elements the columns $\mathcal{Q}$ of the matrix $S_{\mathcal{Q}}$ contains a $k+1$ separator. It follows that if we want to test whether a polyhedron contains a non-trivial

k-module, i.e a k-module containing more than k elements, we only have to check whether the corresponding matroid is $k+1$-connected. Algorithms for testing connectivity have been developed by Bixby and Cunningham [5, 14, 2, 3]. If k is assumed fixed, the connectivity test can be performed in polynomial time [3] using matroid intersection.

Algorithms that directly compute decompositions of matroids into separators, hence modules by the theory developed above, have been studied in the context of branch-decompositions [18, 19]. In the following we use that branch-decompositions can be understood to hierarchically divide the matroid, in order to develop a recursive algorithm for vertex enumeration.

3 Decomposition Theorems

In this section we will state several decomposition theorems.

3.1 Decompositions for polyhedra of bounded branch width

In this section we relate the existence of a certain decomposition of polyhedron $P = \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b, x \geq 0\} \neq \emptyset$ to the branchwidth of the linear matroid ([10, 22]) with elements $\mathcal{R}$, the columns of S.

Definition 4 (branch-width) Let M be a matroid on a set $\mathcal{R}$ of elements. Let $\rho(A) := \text{rank}(A) + \text{rank}(\mathcal{R} \setminus A) - \text{rank}(\mathcal{R}) + 1$ be the connectivity function.

- A *branch decomposition* (T, τ) consists of a tree T with nodes of degree 3 and 1 and a bijective map τ that maps the leaves of T onto $\mathcal{R}$. For short hand notation we write for sets A of leaves: $\tau(A) := \{\tau(a) : a \in A\}$.
- The *width* of an edge e of T is $\rho(\tau(A_e))$, where (A_e, B_e) is the partition of the leaves of T given by $T \setminus e$. Observe that deleting an edge of T splits T into two connected components, one with leaves A_e and the other with leaves B_e. This is also well defined, since $\rho(A) = \rho(\mathcal{R} \setminus A)$.
- The width of a branch decomposition is the maximal width of an edge $e \in T$.
- The *branch-width* of M is the minimal width of all possible branch-decompositions. □

Observe that $\rho(A) \leq k$ if and only if A is a k-separator of the matroid M (Def. 3), which is again the case if and only if A is a P-k-module.

We now interpret the branch-decomposition as a hierarchical structure of k-modules. This enables us to apply recursive divide-and-conquer algorithms, like the vertex enumeration algorithm that we are going to present in this section.

In what follows we use notation $A \dot\cup B$ to indicate that sets A and B are disjoint and to denote their union. We call a family W of subsets of $\mathcal{R}$ *binary rooted* if it satisfies the following properties:

(P1) For each $A \in W$, $A \neq \mathcal{R}$ there exists exactly one $B \in W$ with $A \dot\cup B \in W$.

(P2) For each $C \in W, |C| \geq 2$ there exist $A, B \in W$ with $A \dot\cup B = C$.

This describes a binary rooted tree with root $\mathcal{R}$ and leaves all the single element sets of $\mathcal{R}$.

To facilitate the exposition, we assume in this section that for polytope P the set of variables with constant value is empty; i.e, the set $\mathcal{Q}$ as defined in (2) is equal to $\mathcal{R}$.

Proposition 2 *Let $P = \{x \in \mathbb{R}^{\mathcal{R}} : Sx = b, x \geq 0\} \neq \emptyset$ and assume $\mathcal{Q} = \mathcal{R}$. Let M be the linear matroid generated by the columns of S. There exists a binary rooted family* `Mod` *of k-modules of P if and only if M has branch-width at most $k+1$.*

Proof **⇒:** Define the tree $T = (\texttt{Mod} \setminus \{\mathcal{R}\}, E)$ with vertex set the sets of `Mod` except for $\mathcal{R}$ and an edge (A, B) between two sets (vertices) A and B if

- there exists $C \in \texttt{Mod}$ with $A = B \dot\cup C$ or $B = A \dot\cup C$ or

- $A \dot{\cup} B = \mathcal{R}$.

Clearly, this tree defines a branch decomposition. The leaves of T are sets containing a single element and hence the map τ of the branch decomposition has $\tau(\{i\}) = i$ for all $i \in \mathcal{R}$. We observe that if we delete an edge $e = (A, B)$ from T that either A or B is the union of the leaves of its subtree. Let us assume w.l.o.g. that this is A. Since A is a k-module, it follows by Theorem 5 and Theorem 4 that A is a $k+1$ separator of M. Hence, (T, τ) has width at most $k+1$ and thus, the branch-width of M is at most $k+1$.

⇐: Let (T, τ) be a branch-decomposition of M with width at most $k+1$. We obtain a rooted binary tree T' from T by choosing an arbitrary edge $e = (a, b)$, removing e and adding a root c with children a, b, and direct all edges away from c. For each node a of T' we define the set $A(a) := \{\tau(i) : i \text{ is leaf under } a\}$. By definition of branch-decomposition, we observe for each node a of T' that $A(a)$ is a (k+1)-separator and hence, a k-module by Theorem 5 and Theorem 4. It follows that $\texttt{Mod} = \{A(a) : a \text{ node of } T'\}$ is a family of k-modules that satisfies properties (P1) and (P2). ■

As we have seen, the last part of the proof of this proposition is constructive, in the sense that it gives us a binary rooted family `Mod` of k-modules from a branch decomposition of width at most $k+1$ of the linear matroid defined by the columns of S. We will now show how such a family `Mod` can be used to develop a recursive algorithm for vertex enumeration.

For each module $A \in \texttt{Mod}$ let $D^A \in \mathbb{R}^{\mathcal{M} \times k}$ denote the variable interface and $d^A \in \mathbb{R}^{\mathcal{M}}$ the constant interface and define

$$P^A := \{x \in \mathbb{R}^A : S_A x = D^A \alpha + d, x \geq 0, \exists \alpha \in \mathbb{R}^k\}. \tag{4}$$

In what follows we will study faces of P; i.e., intersections of P with separating hyperplanes. Notice that every face F is completely characterized by the variables that have value 0; i.e., $x_F = 0$ for all $x \in F$. Therefore we take the liberty to use the name 'faces' for subsets of variables. Note that not for every subset F of variables there exists a face where only the variables in F have value 0. Other variables may indirectly be also forced to 0. To capture this issue, we speak about feasible and infeasible faces, even though by its proper definition every face is feasible.

Definition 5 (feasible A-face) For $A \subseteq \mathcal{R}$ a set $F \subseteq A$ is called a *feasible A-face* if there exists a $x \in P$ with $x_F = 0$ and $x_{A \setminus F} > 0$. □

Definition 6 (vertex feasible A-face) For $A \subseteq \mathcal{R}$ a set $F \subseteq A$ is called *vertex feasible A-face* if there exists a vertex v of P with $v_F = 0$ and $v_{A \setminus F} > 0$. □

We note, that whereas testing for $A \subseteq \mathcal{R}$ if a subset F defines a feasible A-face can be done easily by linear programming (see Prop. 7), testing however if $F \subseteq A$ is a vertex feasible A-face is NP-hard [8].

To approximate vertex feasible faces, we introduce the relaxed notion of minimal A-face:

Definition 7 (minimal A-face) For $A \in \texttt{Mod}$ a set $F \subseteq A$ is a minimal A-face if there exist no distinct $y, z \in \{x \in P^A : x_F = 0\}$ with $S_A y = S_A z$, i.e. S_A is injective on $\{x \in P^A : x_F = 0\}$. □

Algorithm 1 enumerates all minimal feasible C-faces for a given k-module $C \in \texttt{Mod}$, by recursively enumerating all minimal A-faces and all minimal B-faces for the two k-modules $A, B \in \texttt{Mod}$ that constitute C; i.e., $C = A \dot{\cup} B$.

That this is correct follows from the following theorem.

Theorem 6 *Algorithm 1 computes all the minimal feasible C-faces for a given $C \in \texttt{Mod}$.* □

To prove Thm. 6, we observe the following decomposition property of minimal faces:

Lemma 7 *Let $A, B, C \in \texttt{Mod}$ with $C = A \dot{\cup} B$. For every minimal feasible C-face F^C there exists a minimal feasible A-face F^A and a minimal feasible B-face F^B with $F^C = F^A \cup F^B$.*

Proof Let F^C be an arbitrary but fixed minimal C-face. Hence, there exists a $x \in P$ with $x_{F^C} = 0$ and $x_{C \setminus F^C} > 0$.

Algorithm 1 Algorithm to compute all minimal feasible C-faces for $C \in \texttt{Mod}$. For $C = \mathcal{R}$, this algorithm will compute all vertices.

```
function F = getMinimalFeasibleFaces(C)
if |C| = 1 then
    F := ∅.
    if ∅ feasible for C then
        F := F ∪ {∅}.
    end if
    if C minimal feasible for C then
        F := F ∪ {C}.
    end if
else
    Let A, B ∈ Mod with C = A ∪̇ B.
    F^A := getMinimalFeasibleFaces(A)
    F^B := getMinimalFeasibleFaces(B)
    F := {F^A ∪ F^B : F^A ∈ F^A, F^B ∈ F^B}.
    for F ∈ F do
        if F not minimal for C or F not feasible for C then
            F := F \ {F}.
        end if
    end for
end if
```

Define $F^A := F^C \cap A$ and $F^B := F^C \cap B$. Since $C = A\dot\cup B$, it follows that $F^C = F^A \cup F^B$. Furthermore, F^A is a feasible A-face and F^B is a feasible B-face, since it follows from $x_{F^C} = 0$ that $x_{F^A} = 0, x_{F^B} = 0$ and from $x_{C\setminus F^C} > 0$ that $x_{A\setminus F^A} > 0$ and $x_{B\setminus F^B} > 0$.

We only have to show that F^A and F^B are also minimal. For proof by contradiction assume that F^A is not minimal (the case F^B non minimal is analogous). It follows that there exist distinct $y, z \in P^A$ with $y_{F^A} = 0, z_{F^A} = 0$ and $S_A y = S_A z$. Define $w \in \mathbb{R}^C$ by $w_A = y - z$ and $w_B = 0$. We observe that $S_C w = S_A w_A = 0$ and $\mathrm{supp}(w) \subseteq \mathrm{supp}(x_C)$. It follows that there exists an $\alpha > 0$ such that $x_C + \alpha w \in P^C$ and $x_C - \alpha w \in P^C$. This is a contradiction to minimality of F^C, since $y \neq z$ and hence, $w \neq 0$. ■

Proof (Thm. 6) First notice that the only two possible faces of a k-module C with $|C| = 1$ are $\emptyset$, and C itself.

Any k-module $C \in \texttt{Mod}$ with $|C| \geq 2$ is constituted by two disjoint k-modules $A, B \in \texttt{Mod}$: $C = A\dot\cup B$. By Lemma. 7 we know that for any minimal feasible C-face F^C, there exist a minimal feasible A-face F^A and a minimal feasible B-face F^B, such that $F^C = F^A \cup F^B$. Since Algorithm 1 tests every possible pair consisting of a minimal A-face and a minimal B-face on feasibility and minimality for C, this implies the theorem for any set $C \in \texttt{Mod}$ with $|C| \geq 2$. ■

That Algorithm 1 eventually outputs all vertices of P is a corollary of the following theorem:

Theorem 7 *Let $A \in \texttt{Mod}$ be an affine 0-module. Then, a feasible A-face is minimal if and only if it is vertex feasible.* □

Corollary 1 *The minimal feasible $\mathcal{R}$-faces are the vertices of P and Algorithm 1 applied to $C = \mathcal{R}$ computes all vertices of P.*

Proof By Theorem 6 all computed $\mathcal{R}$-faces are minimal feasible and thus, by Theorem 7 vertex feasible, since $\mathcal{R}$ is an affine 0-module.

Since there are no further variables, every vertex feasible $\mathcal{R}$-face is a vertex. ■

Before we prove Thm. 7, we observe the following property that allows us to relate minimal feasible faces and vertex feasible faces.

Proposition 3 *Every vertex feasible A-face F for $A \in \texttt{Mod}$ is a minimal A-face.*

Proof By definition of vertex feasible A-face, there exists a vertex v of P with $v_F = 0$ and $v_{A\setminus F} > 0$. For proof by contradiction, we assume now that F is not a minimal A-face. It follows that there exist distinct $y, z \in \{x \in P^A : x_F = 0\}$ with $S_A y = S_A z$. We observe that $w' := y - z$ satisfies $S_A w' = S_A y - S_A z = 0$. Define $w \in \mathbb{R}^{\mathcal{R}}$ by $w_A = w'_A$ and $w_{\mathcal{R}\setminus A} = 0$. We observe that $\operatorname{supp}(w) \subseteq \operatorname{supp}(v)$. Hence, there exists an $\alpha > 0$ such that $v - \alpha w \in P$ and $v + \alpha w \in P$. This is a contradiction to the assumption that v is a vertex. ∎

Proof (Thm. 7) Proposition 3 shows that every vertex feasible A-face is a minimal A-face.

Let F be minimal feasible A-face. Since A is a 0-module it holds for all $x \in P^A$ that $S_A x_A = d$. It follows by definition of minimal A-face that $x \in P^A$ with $x_F = 0$ is unique. Since F is also feasible such x exists and satisfies $x_{A\setminus F} > 0$.

We observe that by construction P is pointed. Since F is a feasible A-face, it follows that $\hat{P} := \{x \in P : x_F = 0\}$ is a non-empty pointed polyhedron. Hence, there exists a vertex y of $\hat{P}$. Clearly, y is also a vertex of P and satisfies $y_A \in P^A$ and $y_F = 0$. Thus, $y_A = x_A$ and F is a vertex feasible A-face. ∎

Now we will show that in case P is a polytope (i.e. P bounded) the existence of a set `Mod` of k-modules makes Algorithm 1 run in polynomial time, for fixed k. Therefore, we will do a number of observations that allow us to bound the number of minimal feasible faces and finally obtain the runtime bound. The following proposition still also holds for unbounded polyhedra:

Proposition 4 *For every minimal A-face F for $A \in$ `Mod` holds that* $\dim f \leq k$ *with* $f = \{x \in P^A : x_F = 0\}$.

Proof Since A is a k-module, it follows that S_A maps every point in f into a k-dimensional space. If $\dim f > k$, it follows that S_A is not injective on f and hence, F would not be minimal. ∎

We observe the following corollary, which may give another intuition for the final complexity bound.

Corollary 2 *Every vertex feasible A-face F for $A \in$ `Mod` satisfies* $\dim\{x \in P^A : x_F = 0\} \leq k$.

Proof Directly from Prop. 3 and Prop. 4. ∎

Lemma 8 *Assume P is bounded. Suppose for $A \in$ `Mod` that F is a feasible A-face and let $h = \dim f$ with $f = \{x \in P^A : x_F = 0\}$. Then there exist a set of $\ell \leq h+1$ vertex feasible A-faces $F^1, \ldots, F^\ell$ such that $F = F^1 \cap F^2 \cap \ldots \cap F^\ell$.*

Proof Since F is a feasible A-face, there exists a $y \in P$ with $y_F = 0$ and $y_{A\setminus F} > 0$. Therefore, y lies in the face $\{x \in P : x_F = 0\}$ of P. Since P is bounded, there exist a set of vertices V^f of P such that $y \in \operatorname{conv}(V^f)$ and $w_F = 0$ for all $w \in V^f$.

It follows that $y_A \in \operatorname{conv}(\operatorname{pr}_A V^f)$ and $\operatorname{pr}_A V^f \subseteq P^A$. Since $\dim\{x \in P^A : x_F = 0\} = h$, there exist by Carathéodory's theorem [24] $\ell \leq h+1$ points $w^1, w^2, \ldots, w^\ell \in \operatorname{pr}_A V^f$ with $y_A \in \operatorname{conv}(w^1, \ldots, w^\ell)$.

Clearly $F^i = \{j \in \mathcal{R} : w^i_j = 0\}$ is a vertex feasible A-face and $F \subseteq F^i$ for each $i = 1, \ldots, \ell$. For every $j \in A \setminus F$ with $j \in F^i$ for all $i = 1, \ldots, \ell$ it follows that $y_j = 0$, since $y_A \in \operatorname{conv}(w^1_A, \ldots, w^\ell_A)$, and hence, $F \supseteq F^1 \cap F^2 \cap \ldots \cap F^\ell$. Thus, $F = F^1 \cap F^2 \cap \ldots \cap F^\ell$. ∎

Proposition 5 *If P is bounded then holds for all $A \in$ `Mod` that*

$$|\{F \subseteq A : F \textit{ minimal feasible A-face}\}| \leq |\{F \subseteq A : F \textit{ vertex feasible A-face}\}|^{k+1}.$$

Proof By Prop. 4 every minimal feasible A-face has dimension at most k. Hence, by Lemma 8 there exists an injective map that assigns to each minimal feasible A-face a non-empty set of at most $k+1$ vertex feasible A-faces. Let c_{vert} denote the number of vertex feasible A-faces. There are at most

$$\sum_{i=1}^{k+1} \binom{c_{\text{vert}}}{i}$$

non-empty subsets of at most $k+1$ elements. For $c_{\text{vert}} = 1$, we have $\sum_{i=1}^{k+1} \binom{c_{\text{vert}}}{i} = 1 = c_{\text{vert}}^{k+1}$ and for $c_{\text{vert}} \geq 2$, we can estimate

$$\sum_{i=1}^{k+1} \binom{c_{\text{vert}}}{i} \leq \sum_{i=1}^{k+1} \frac{c_{\text{vert}}^i}{i!} \leq \sum_{i=1}^{k+1} \frac{2^{k+1-i} c_{\text{vert}}^i}{k+1} \leq (k+1) \frac{c_{\text{vert}}^{k+1}}{k+1} = c_{\text{vert}}^{k+1},$$

since $2^{k+1-i} \geq \frac{k+1}{i!}$ for all $i \leq k \in \mathbb{N}$.

By injectivity it follows that this is also a bound on the number of minimal feasible A-faces. ∎

Proposition 6 *For $A \in \texttt{Mod}$ holds that*

$$|\{F \subseteq A : F \textit{ vertex feasible } A\textit{-face}\}| \leq |\{v \in \mathbb{R}^{\mathcal{R}} : v \textit{ is a vertex of } P\}|.$$

Proof Let F^1, F^2 be distinct vertex feasible A-faces and let v^1, v^2 be representing vertices. It follows that $\operatorname{supp}(v^1_A) = A \setminus F^1 \neq A \setminus F^2 = \operatorname{supp}(v^2_A)$. Hence, $v^1 \neq v^2$ and the result follows. ∎

Theorem 8 *Assume P is bounded. Let $A, B, C \in \texttt{Mod}$ with $C = A \dot\cup B$. Assume the set of minimal feasible A-faces $\mathcal{F}^A$ and the set of minimal feasible B-faces $\mathcal{F}^B$ are given. Then the set of minimal feasible C-faces $\mathcal{F}^C$ can be computed in time*

$$O\left(|\mathcal{V}|^{2k+2} t\right),$$

where $\mathcal{V}$ is the set of vertices of P and t is the time needed to check if a face is feasible and minimal.

Proof By Lemma 7 every minimal feasible C-face can be obtained from a combination of one minimal feasible A-face and a minimal feasible B-face. It follows that we have to consider at most $|\mathcal{F}^A| \cdot |\mathcal{F}^B|$ combinations. By Proposition 6 and Proposition 5 it follows that

$$|\mathcal{F}^A| \cdot |\mathcal{F}^B| \leq |\mathcal{V}|^{k+1} \cdot |\mathcal{V}|^{k+1} \leq |\mathcal{V}|^{2k+2}.$$

For each candidate we have to check if it is feasible and minimal, which gives the final runtime bound. ∎

We now observe that the time for checking if a face is minimal feasible can be done in polynomial time, which then leads us to the final result on the runtime of Alg. 1.

Proposition 7 *Given $A \in \texttt{Mod}$, it can be checked in input polynomial time if a A-face F is minimal feasible.*

Proof To check feasibility, we just have to solve the following LP:

$$\begin{aligned} \max\ & z \\ \text{s.t. } Sx &= b \\ x_F &= 0 \\ x_i - z &\geq 0 && \forall i \in A \setminus F \\ x &\geq 0 \end{aligned}$$

If the LP is unbounded or the optimal value is greater than 0 then we have found a solution $x^* \in P$ with $x^*_F = 0$ and $x^*_{A \setminus F} > 0$, which proves feasibility of F. If F is feasible, there exists a solution of the LP with $z > 0$ and hence, the optimal value of the LP has to be positive.

To check minimality, do the following: By minimizing / maximizing each $i \in A$ using linear programming, we compute the following set G:

$$\begin{aligned} G &:= \{i \in A : u_i > \ell_i\}, \text{ where} \\ u_i &:= \sup\{x_i : x \in P^A, x_F = 0\} && \forall i \in A \\ \ell_i &:= \inf\{x_i : x \in P^A, x_F = 0\} && \forall i \in A \end{aligned}$$

Observe that $G \cap F = \emptyset$.

Claim 3.1 *F is minimal if and only if $S_G : \mathbb{R}^G \to \mathbb{R}^{\mathcal{M}}$ is injective.*

PROOF $\Rightarrow$: Assume S_G is not injective, i.e. there exists a $x' \in \mathbb{R}^G \setminus \{0\}$ with $S_G x' = 0$. Define $x \in \mathbb{R}^A$ by $x_G = x'$ and $x_{A \setminus G} = 0$. By construction of G and convexity, there exists a $v \in P^A$ with $v_F = 0$ and $\ell_G < v < u_G$. Hence, there exists an $\alpha > 0$ such that $v + \alpha x \geq 0$ and $v - \alpha x \geq 0$. We observe further that $S_A(v \pm \alpha x) = S_A v \pm \alpha S_A x = S_A v$ and hence, $v \pm \alpha x \in P$ with $(v \pm \alpha x)_F = 0$. This is a contradiction to minimality of F.

$\Leftarrow$: Assume F is not minimal. Hence, there exist $x, y \in P^A$ with $x_F = y_F = 0$ and $S_A x = S_A y$. It follows that $w := x - y$ satisfies $S_A w = 0$ and by definition of G, we have $w_{A \setminus G} = 0$. Hence, $S_G w_G = S_A w = 0$, which is a contradiction to injectivity of S_G. ■

Since we can check injectivity of matrices by computing the nullspace matrix in polynomial time, we can also check minimality of F in polynomial time. ■

Theorem 9 *If P is a polytope and* `Mod` *is a family of k-modules satisfying Properties (P1) and (P2) for constant k, then Algorithm 1 runs in total polynomial time.*

PROOF As mentioned before, `Mod` can be represented as a binary tree, rooted at $\mathcal{R}$, with leaves the single element modules. We observe that the time spend for the leaves (modules $A \in$ `Mod` with $|A| = 1$) is $O(\mathcal{R}t)$, where t is the time needed to check if the corresponding face is minimal and feasible. Let $\mathcal{C}$ be the set of interior nodes, then the total time needed for determining the minimal feasible faces of all modules corresponding to interior nodes is, by Theorem 8,

$$O\left(|\mathcal{C}||\mathcal{V}|^{2k+1}t\right),$$

where t is the time needed to check if a given face is minimal and feasible. By Prop. 7 t grows polynomially in the input size. Since the number of internal nodes $|\mathcal{C}|$ of a binary tree is linear in the number of leaves $|\mathcal{R}|$, the result follows. ■

4 Conclusion

In this paper we introduced (affine) k-modules as a tool to decompose polyhedra given by inequalities in general and metabolic networks in particular.

We showed a strong connection between k-modules and $(k+1)$-separators in matroid theory. This allows the application of algorithms to find $(k+1)$-separators from matroid theory. This way we were able to extend the concept of a decomposition into flux modules [16] to a decomposition into k-modules by using branch-decompositions.

Branch-decompositions turned out to be a valuable structure for the vertex enumeration problem, because it allowed us to show that the vertex enumeration problem can be solved for polytopes in total polynomial time if the branch-width is bounded by a constant. If the vertices of polytopes in general can be enumerated in total polynomial time is still a major open problem in computational discrete geometry.

The presented vertex enumeration algorithm also works for unbounded polyhedra. However, we were not able to show a total polynomial run-time bound. This may be linked to the fact that vertex enumeration of polyhedra in general is NP-hard.

We think that the presented algorithm is a major step towards understanding the complexity of the vertex enumeration algorithm. Further studies are needed to investigate the practical applicability of the algorithm, for example in the context of Elementary Flux Mode enumeration of metabolic networks.

References

[1] David Avis and Komei Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*, 8(1):295–313, 1992.

[2] Robert E. Bixby and William H. Cunningham. Matroids, graphs, and 3-connectivity. In John Adrian Bondy and William Thomas Tutte, editors, *Graph Theory and Related Topics*. Acad. Press, 1979.

[3] Robert E. Bixby and William H. Cunningham. Matriod optimization and algorithms. Technical report, Rice University, 1990.

[4] Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Kazuhisa Makino. Generating vertices of polyhedra and related monotone generation problems. Technical report, DIMACS, 2007.

[5] W. H. Cunningham. *A combinatorial decomposition theory*. PhD thesis, University of Waterloo, Ontario, Canada, 1973.

[6] William H. Cunningham and Jim Geelen. *Integer Programming and Combinatorial Optimization*, chapter On Integer Programming and the Branch-Width of the Constraint Matrix, pages 158–166. Lecture Notes in Computer Science. Springer, 2007.

[7] M. E. Dyer. The complexity of vertex enumeration methods. *Mathematics of Operations Research*, 8(3):381–402, 1983.

[8] Komei Fukuda, Thomas M. Liebling, and François Margot. Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron. *Computational Geometry*, 8(1):1–12, 1997.

[9] Rudolf Halin. S-functions for graphs. *Journal of Geometry*, 8(1-2):171–186, 1976.

[10] Illya V. Hicks and Sang-Il Oum. *Wiley Encyclopedia of Operations Research and Management Science*, chapter Branch-Width and Tangles. Wiley, 2011.

[11] Petr Hliněný and Geoff Whittle. Matroid tree-width. *European Journal of Combinatorics*, 27(7):1117–1128, 2006.

[12] Kristopher A. Hunt, James P. Folsom, Reed L. Taffs, and Ross P. Carlson. Complete enumeration of elementary flux modes through scalable demand-based subnetwork definition. *Bioinformatics*, 2014. in press.

[13] Leonid Khachiyan, Endre Boros, Konrad Borys, Khaled Elbassioni, and Vladimir Gurvich. Generating all vertices of a polyhedron is hard. *Discrete Computational Geometry*, 39(1-3):174–190, 2008.

[14] Stein Krogdahl. The dependence graph for bases in matroids. *Discrete Mathematics*, 19:47–59, 1977.

[15] T. S. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall. *Contributions to the Theory of Games*, chapter The double description method. Princeton University Press, Princeton, RI, 1953.

[16] Arne C. Müller and Alexander Bockmayr. Flux modules in metabolic networks. *Journal of Mathematical Biology*, 2013. in press (AC Müller is now called AC Reimers).

[17] Arne C. Müller, Frank J. Bruggeman, Brett G. Olivier, and Leen Stougie. Fast flux module detection using matroid theory. In *Research in Computational Molecular Biology*, volume 8394 of *Lecture Notes in Computer Science*, pages 192–206, 2014. (AC Müller is now called AC Reimers).

[18] Sang-il Oum and Paul Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory*, Series B 96:514–528, 2006.

[19] Sang-il Oum and Paul Seymour. Testing branch-width. *Journal of Combinatorial Theory*, Series B 97:385–393, 2007.

[20] James Oxley. *Matroid Theory*. Oxford Graduate Texts in Mathematics. Oxford University Press, New York, second edition edition, 2011.

[21] Neil Robertson and Paul D. Seymour. Graph minors. iii. planar tree-width. *Journal of Combinatorial Theory*, Series B 36:49–64, 1984.

[22] Neil Robertson and P.D. Seymour. Graph minors. x. obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153190, 1991.

[23] Gnter Rote. Degenerate convex hulls in high dimensions without extra storage. In *SCG '92 Proceedings of the eighth annual symposium on Computational geometry*, pages 26–32.

[24] Alexander Schrijver. *Theory of Linear and Integer Rpogramming.* Wiley, Chichester, 1986.

[25] S. Schuster and C. Hilgetag. On elementary flux modes in biochemical systems at steady state. *J. Biol. Systems*, 2:165–182, 1994.

[26] Marco Terzer and Jörg Stelling. Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics*, 24(19):2229–2235, 2008.