THE LONDON SCHOOL
OF ECONOMICS AND
POLITICAL SCIENCE

Abstract tropical linear programming

Article:

Abstract Tropical Linear Programming

Georg Loho

Department of Mathematics
London School of Economics
United Kingdom

`g.loho@lse.ac.uk`

Abstract

In this paper we develop a combinatorial abstraction of tropical linear programming. This generalizes the search for a feasible point of a system of min-plus-inequalities. We obtain an algorithm based on an axiomatic approach to this generalization. It builds on the introduction of signed tropical matroids based on the polyhedral properties of triangulations of the product of two simplices and the combinatorics of the associated set of bipartite graphs with an additional sign information. Finally, we establish an upper bound for our feasibility algorithm applied to a system of min-plus-inequalities in terms of the secondary fan of a product of two simplices. The appropriate complexity measure is a shortest integer vector in a cone of the secondary fan associated to the system.

Mathematics Subject Classifications: 14T05, 90C05 (52C40, 91A50, 05E45)

1 Introduction

Tropical linear programming is a method to determine a feasible solution of a linear inequality system, where addition is replaced by minimum and multiplication is replaced by usual addition. It is intimately connected to the classical version of linear programming as tropical polyhedra are essentially projections of classical polyhedra [21]. Formulating linear programming over an appropriate valued field, one obtains tropical linear programming as a shadow through the valuation map. Even more, there is a tropical simplex method for which the sequence of bases is in bijection with the sequence of bases in a run of the classical simplex method [5].

The study of tropical linear programming is motivated by the connection with the following two major open problems. The first is Smale's 9th problem which asks for a strongly polynomial algorithm in linear programming. The connection of tropical and classical linear programming already resulted in the disproval of the continuous Hirsch-conjecture for the central path and a proof that log-barrier interior point methods are

not strongly polynomial, both in [6]. Secondly, tropical linear programming is equivalent to mean payoff games, see [2] and Section A.3. These games are of special interest in computer science as their complexity lies in NP ∩ co-NP but no polynomial algorithm is known [69]. This includes the subclass of parity games which lie at the heart of several hard instances for the classical simplex method [29, 39] but which have recently shown to be at least quasipolynomial solvable [17].

In the history of linear programming, it was a conceptual breakthrough to formulate the simplex method in the more abstract language of oriented matroids. After the development of the simplex method by Dantzig [18], the sign vectors occurring in the pivoting steps were studied in a more axiomatic way. This abstraction was initiated by Rockafellar [59] and it led to the work of Bland [12], Fukuda [30] and Todd [65, 66] on oriented matroid programming. Furthermore, it motivated the development of crisscross methods [64].

In this paper we formulate an abstract version of tropical linear programming. This is based on *signed tropical matroids*, a tropical analogue of oriented matroids. An axiomatic study of "tropical oriented matroids" originated in the work by Ardila and Develin [8] to describe and generalize the combinatorics of tropical point configurations. It was further developed by Oh and Yoo [54] and Horn [43]. The latter established a realizability result with "tropical pseudohyperplanes", that also proved the bijection of this concept of tropical oriented matroids with not necessarily regular subdivisions of the product of two simplices $\Delta_{n-1} \times \Delta_{d-1}$. Recall from [19, §2.2.3] that a subdivision is regular if it is induced by a height function but not all subdivisions are of this form.

1.1 Our results

The abstraction of a tropical linear inequality system by a *signed tropical matroid* is described in terms of a well-structured set of bipartite graphs on the node set $[d] \sqcup [n]$ where each edge is labeled by '+' or '−'. Building on an axiomatic way for defining the set of these *covector graphs*, we extend the notion of *feasibility* and *infeasibility* to signed tropical matroids using the signs on the edges. These graphs are the analogue of the sign vectors of an oriented matroid, but contain primal and dual information.

The overall structure of our algorithm is motivated by the simplex method, which we recall in Appendix A.1. We adapt the scheme of iterating over bases with local exchanges. The role of basic solutions is taken by the important concept of a *basic covector (graph)*, see Section 3.3. We iterate as long as the current basic covector contains a negative edge connecting a leaf in $[n]$ and a certain subset of $[d]$, which is motivated by the greedy approach of successively satisfying violated inequality.

While the correctness and termination of the simplex method can be shown by the increase in an objective function, we mimic this by distinguishing one particular node in $[d]$ of each basic covector and consider paths emerging from it. The arguments leading to the combinatorial analogue of an increase rely on the parity, even or odd, of the distance of an edge from the distinguished node. Together with the particular structure of matchings in basic covectors, this allows us to deduce a process which is guaranteed to terminate with a certificate of feasibility or infeasibility (Theorem 41). Note that our scheme has a

lot of freedom in the actual choice of the pivot element. The abstract pivoting allows us to deduce a generalized tropical duality theorem (Theorem 44) and a new algorithm for tropical inequality systems (Algorithm 6).

Our definition of signed tropical matroids is motivated by the correspondence with subdivisions of $\Delta_{n-1} \times \Delta_{d-1}$. We use the axiomatic description of polyhedral subdivisions [19] and enrich it with an additional sign information. Using the equivalence shown in [43] of tropical oriented matroids and subdivisions of $\Delta_{n-1} \times \Delta_{d-1}$ one could also use the axioms in [8] and equip it with an additional sign information. However, as our arguments mainly rely on the characterization of triangulations in [7], going back to [60], we restrict ourselves to these purely polyhedral notions.

To keep the exposition of the algorithm as simple as possible, we start with a generic subclass of signed tropical matroids given by a short list of axioms in Definition 12. To make the algorithm applicable to general signed tropical matroids, we employ the concepts of extension and refinement from polyhedral geometry in our setting. We mainly translate the polyhedral constructions to operations on the set of bipartite graphs. Similar techniques have been used in related works by Allamigeon et al. [4] and Horn [43].

We use global geometric properties of the occurring polyhedral subdivisions related to generalized permutohedra [56] to ensure the existence of basic covectors by an abstract Cramer theorem in Section 3.2. This can be seen as a polyhedral generalization of [1, Theorem 6.1], as well as [58, Corollary 5.4], and it is related to the linkage trees in [63, Theorem 2.4].

The running time of our algorithm applied to tropical linear inequality systems (Algorithm 6) is related to the minimal length of integer vectors in the secondary fan of $\Delta_{n-1} \times \Delta_{d-1}$. This follows as we show that the number of iterations can be bounded by a polynomial in the coefficients of the inequality system. Furthermore, the behaviour of the algorithm only depends on the triangulation. Hence, a minimal representative of a coefficient matrix in the containing cone of the secondary fan of $\Delta_{n-1} \times \Delta_{d-1}$ yields an upper bound.

Through this result one obtains a connection between the complexity of a simplex-like algorithm for (tropical) linear programming and the length of a minimal lattice vector in a cone. In particular, this establishes the length of a minimal lattice vector as a natural complexity measure of a subdivision and of an algorithm. The secondary fan of $\Delta_{n-1} \times \Delta_{d-1}$ can be seen as the tropical analogue of the realization space of polytopes for fixed parameters, cf. [57]. In this vein, finding a minimal realization of the combinatorial type of a tropical point configuration is related to the determination of a shortest non-zero lattice vector which was tackled in the pioneering work [50].

The exposition is complemented by the formal relations between tropical linear programming and some other algorithmic problems. We formulate a tropical inequality system which is equivalent to a given AND-OR-network [52]. Furthermore, we show how our results tie in with the equivalence of the feasibility problem for tropical linear inequality systems and finding winning states of a mean payoff game.

1.2 Organization of the paper

Section 2 is dedicated to the introduction of the main concepts for describing the combinatorics of tropical linear inequality systems. Our main algorithm is presented in Section 3. Note that the algorithms are rather simple in the required terminology, however, we need the technical tools to prove the correctness. We move on to define general signed tropical matroids, the abstraction of tropical linear inequality systems, in Section 4. Building on polyhedral methods, we show in Section 5 how one can reduce signed tropical matroids to the subclass for which we formulate our algorithm. This is followed by an application to the special case of tropical linear inequality systems in Section 6. For this, we can drop some requirements on the input and deduce upper bounds on the number of iterations.

We give an overview of related algorithmic problems, the simplex method, AND-OR-networks and mean payoff games, in Appendix A. The main polyhedral prerequisites are collected in Appendix B. We finish in Appendix C with applications of our algorithm to analyze mean payoff games and tropical linear inequality systems, in particular for finding the maximal support of a feasible point.

1.3 Related algorithms

The tropical linear feasibility problem has connections to several other problems as further described in Section A. Therefore, algorithms for scheduling with AND-OR-networks [52], mean payoff games [24, 69, 38] and classical linear programming [5, 4, 10] are applicable to this problem. Furthermore, beside the algorithms for tropical inequality systems [14, 15], one can also use algorithms for tropical equality systems [35, 16] which are equivalent via the reformulation $a \leqslant b \Leftrightarrow a = \min(a, b)$. The algorithms with the currently fastest runtime for mean payoff games are [23] and [26]. We consider mean payoff games on a bipartite game graph with $(n + d)$ nodes, m edges and maximal weight ω. The algorithm presented in [23] has a provable runtime of $O(\min\left(m(n + d)\omega, m(n + d)2^{(n+d)/2} \log \omega\right)$ and [26] takes $O\left(m(n + d)((n + d)\omega)^{1-1/(n+d)}\right)$. Theorem 75 gives a rough upper bound for our algorithm in the realizable case of $O(d^3(n + d^2)\omega)$. However, as discussed above, the runtime depends on a potentially smaller pseudo-polynomial parameter than ω. We note that, in the realizable case, our algorithm is similar to the method developed in [11] as the latter is also combinatorial and pseudopolynomial. The precise relation is subject to further work.

The presented results are part of the dissertation of the author [51].

2 Basic definitions for tropical linear inequality systems

We start with the definitions for a *tropical semiring* and introduce *covector graphs* in different flavors as they will be our main tool. They were first defined by Develin and Sturmfels under the name of *types* in [20] and further studied as *covectors* in [27], as well as in [45].

2.1 Covector graphs for signed systems

The *tropical numbers* consist of the set $\mathbb{T}_{\min} = \mathbb{R} \cup \{\infty\}$. Equipped with the two operations $\oplus$ and $\odot$, where $x \oplus y := \min(x, y)$ and $x \odot y := x + y$ for $x, y \in \mathbb{T}_{\min}$, they form the *tropical semiring*. Just as well, we could consider $\oplus = \max$ as tropical addition. The operations extend to vectors and matrices componentwise and we can define a matrix product analogously to the classical case.

We use the notation $[d] = \{1, \ldots, d\}$ and define the sum over an empty set to be ∞. Furthermore, the symbol $\sqcup$ denotes the disjoint union of the two (color) classes of nodes of a bipartite graph.

We define a *(tropical) signed system* as a pair (A, Σ) with $(a_{ji}) = A \in \mathbb{T}_{\min}^{n \times d}$ and $(\sigma_{ji}) = \Sigma \in \{+, -, \bullet\}^{n \times d}$, where $a_{ji} = \infty \Leftrightarrow \sigma_{ji} = \bullet$. It defines a homogeneous tropical linear inequality system by

$$\bigoplus_{i \in [d],\ \sigma_{ji} = +} a_{ji} \odot x_i \leqslant \bigoplus_{i \in [d],\ \sigma_{ji} = -} a_{ji} \odot x_i \qquad \text{for } j \in [n] \ . \tag{1}$$

A point $x \in \mathbb{T}_{\min}^d$ is *feasible* for (A, Σ) if it fulfills all the inequalities, otherwise we call it *infeasible*. A signed system is *feasible* if there is a feasible point in $\mathbb{T}\mathbb{A}^d = \mathbb{T}_{\min}^d \setminus \{(\infty, \ldots, \infty)\}$; otherwise it is *infeasible*. The set of feasible points in $\mathbb{T}\mathbb{A}^d$ is the *feasible region*. Such a feasible region is a *tropical cone*, which means that it is closed under tropical addition and scalar multiplication. A *tropical halfspace* is the feasible region of a single tropical linear inequality.

Note that the sign information which we encode in the sign matrix Σ occurs in the patchworking method of Viro [67] and is, alternatively, added to the tropical semiring to form the "symmetrized tropical semiring" [1].

Definition 1. The *(tropical) covector (graph)* $G_A(x)$ of a finite point $x \in \mathbb{R}^d$ is the bipartite graph on the node set $[d] \sqcup [n]$ containing an edge $(i, j) \in [d] \times [n]$ if and only if $a_{ji} + x_i = \min\{a_{jk} + x_k \mid k \in [d], a_{jk} \neq \infty\}$. This means that the covector graph encodes those entries in each row of the product $A \odot x$ where the minimum is attained.

Note that we label the entries of A by pairs $(j, i) \in [n] \times [d]$ and choose the reverse order to denote the edges $(i, j) \in [d] \times [n]$ of a covector graph. We will write pairs for the edges even if we consider it as an undirected graph. Often, we will call tropical covector graphs just covectors.

The nodes in $[d]$ are *coordinate nodes* and in $[n]$ are the *apex nodes*. Coordinate nodes correspond to the variables and are visualized by square nodes. Apex nodes correspond to the rows and the inequalities, respectively. They are depicted by circle nodes. Depending on the sign given by Σ, we call an edge in a covector graph *negative* or *positive*.

Example 2. Consider the signed system $(A, \Sigma) = ((0, 0, 0), (+, -, +))$. For each point $x \in \mathbb{R}^3$ with pairwise distinct coordinates, the decomposition in Figure 1 shows where the minimum is attained in the product $(0, 0, 0) \odot x = \min(x_1, x_2, x_3)$.

On the left of Figure 1, we put the plain covector graphs whereas, on the right, we add the sign information given by Σ.
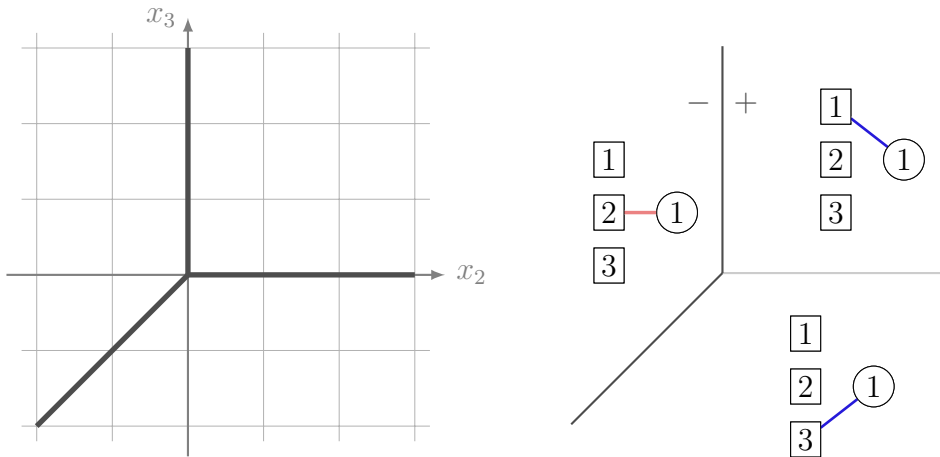
Figure 1: We dehomogenize by setting $x_1 = 0$. We depict the covector graphs of the points, where the minimum is attained only once, for $A = (0, 0, 0)$ and $\Sigma = (+, -, +)$, see Example 2. Negative edges are red, positive edges are blue.

Directly from the definition, we obtain a characterization of finite feasible points.

Proposition 3. *A point $x \in \mathbb{R}^d$ is feasible for the signed system (A, Σ) if and only if no apex node is only incident with negative edges in $G_A(x)$.*

Proof. By definition, a point is infeasible if and only if there is a $j \in [n]$ with

$$\bigoplus_{\sigma_{ji}=+, i \in [d]} a_{ji} \odot x_i > \bigoplus_{\sigma_{ji}=-, i \in [d]} a_{ji} \odot x_i \ .$$

This means that the minimum is attained only for entries with a minus sign. From this follows the claim with Definition 1. $\square$

The cells $\left\{ x \in \mathbb{R}^d \mid G_A(x) \text{ const} \right\}$ define a *covector decomposition* of $\mathbb{R}^d$. This is the same polyhedral subdivision of $\mathbb{R}^d$ as in [45] if we replace max by min.

Notice that the covector graphs are homogeneous in the sense that adding an element of $\mathbb{R} \cdot \mathbf{1} = \mathbb{R} \cdot (1, \dots, 1)$ to a cell yields the same covector graph and the cells in the covector decomposition all contain $\mathbb{R} \cdot \mathbf{1}$ as lineality.

We fix a matrix $A \in \mathbb{T}_{\min}^{n \times d}$, for which every row contains a finite entry, and denote by Γ the complete bipartite graph $K_{d,n}$ on the node set $[d] \sqcup [n]$ with the entries of A as weights on its edges. A *matching* on $D \sqcup N$ with $D \subseteq [d]$ and $N \subseteq [n]$ is a subgraph of $K_{d,n}$ in which each node has degree 1. The *value* of a matching μ with respect to a matrix A is the sum $\sum_{(i,j) \in \mu} a_{ji}$. A matching is *minimal* if all the other matchings in the induced subgraph of $K_{d,n}$ on $D \sqcup N$ have a bigger value.

Combining [45, Proposition 30] and [45, Proposition 38] yields the following characterization which is similar to [44, Theorem 6.1].

Proposition 4. *A bipartite graph G on $[d] \sqcup [n]$ is a covector graph of a point $x \in \mathbb{R}^d$ with respect to A if and only if the following three conditions hold:*

1. *No apex node $j \in [n]$ is isolated in G.*

2. *Let μ be a matching in G on a subset $D \sqcup N$ of the nodes with $D \subseteq [d]$, $N \subseteq [n]$ and $|D| = |N|$. Then μ is a minimal matching in Γ.*

3. *Let μ and η be minimal matchings in Γ. If μ is contained in G, so is η.*

2.2 Generalized covector graphs

To make use of covector graphs also for points in $\mathbb{T}^d_{\min}$ with ∞ coordinates, we introduce a generalized notion that is slightly different from the approach chosen in [45, §3.5].

Definition 5. *The* support $\mathrm{supp}(x)$ *of a point $x \in \mathbb{T}^d_{\min}$ is the set $\{i \in [d] \mid x_i \neq \infty\}$. Furthermore, the* generalized covector graph *of x is the bipartite graph on the node set $[d] \sqcup [n]$ containing an edge $(i, j) \in [d] \times [n]$ if and only if*

$$a_{ji} + x_i = \min \{ a_{jk} + x_k \mid k \in \mathrm{supp}(x), a_{jk} \neq \infty \} \neq \infty .$$

We denote it by $G_A(x)$, like the covector graphs from Definition 1. In contrast to covector graphs of points in $\mathbb{R}^d$ the generalized covector graphs possibly have isolated apex nodes. A (generalized) covector graph without an isolated apex node is called *proper*.

Note that a generalized covector graph can also be the empty graph and the corresponding point is feasible. The empty graph is the covector graph of $(\infty, \ldots, \infty)$ but also for $(0, \infty, \infty)$ with respect to $(\infty, 0, 0)$. This happens, if the support of all the rows is contained in a common proper subset of $[d]$.

Definition 6. *A (generalized) covector graph G is* infeasible *if there is an apex node which is only incident with negative edges. If G is not infeasible we call it* feasible.

We obtain the following more general version of Proposition 3. It assures that the two notions of feasibility agree for points with finite components and it is the suitable formulation for defining the feasibility in signed tropical matroids, see Section 4.

Proposition 7. *A point $x \in \mathbb{T}^d_{\min}$ is feasible for the signed system (A, Σ) if and only if no apex node is only incident with negative edges in the generalized covector graph $G_A(x)$.*

Proof. Fix $j \in [n]$ and consider the corresponding inequality Equation 1. If j is only incident to negative edges the right hand side is surely smaller and the inequality is not fulfilled. If j has no neighbors in $G_A(x)$ then both sides of the inequality are ∞ and the inequality is fulfilled. Otherwise, it is also a valid inequality. $\qquad\square$

This allows us to examine the feasibility of general tropical inequality systems via generalized covector graphs.
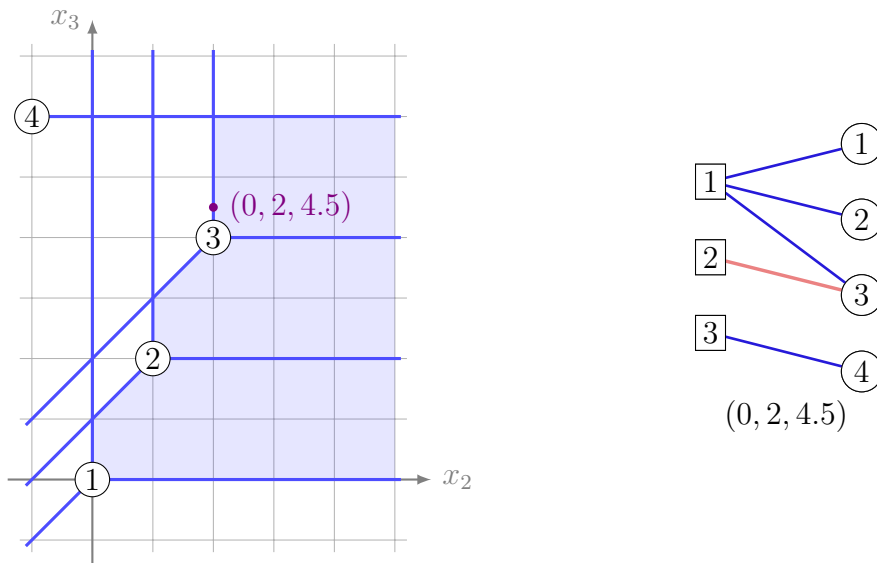
Figure 2: As always, we set $x_1 = 0$ to cancel out the lineality $\mathbb{R} \cdot \mathbf{1}$. The shaded area is the *feasible* region of a signed system formed by the four inequalities from Example 8. The crooked lines are the boundaries of the *tropical halfspaces*. The bipartite graph is the covector graph of $(0, 2, 4.5)$, where the negative edge is red.

Example 8. The left part of Figure 2 depicts the feasible region of the signed system (A, Σ) with

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & -2 \\ 0 & -2 & -4 \\ 0 & \infty & -6 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} + & - & - \\ + & - & + \\ + & - & + \\ - & \bullet & + \end{pmatrix} .$$

This gives rise to the inequality system

$$
\begin{aligned}
0 + x_1 &\leqslant \min(0 + x_2, 0 + x_3) \\
\min(0 + x_1, x_3 - 2) &\leqslant x_2 - 1 \\
\min(0 + x_1, x_3 - 4) &\leqslant x_2 - 2 \\
x_3 - 6 &\leqslant 0 + x_1 \ .
\end{aligned}
$$

The covector graph of the point $(0, 2, 4.5)$ is shown in the right part of Figure 2. It is feasible since each apex node is incident with a positive edge.

The covector graph of the point $(\infty, 0, \infty)$ has the edges $(2, 1)$, $(2, 2)$ and $(2, 3)$. It is not proper and infeasible.

2.3 Computations for realizable covector graphs

Starting from a proper covector graph, the next lemma allows us to compute a point with given covector graph.

Let G be a connected covector graph with respect to $A \in \mathbb{T}_{\min}^{n \times d}$ and $\delta \in [d]$ a coordinate node. For any other coordinate $i \in [d]$, let $\delta = i_1, j_1, i_2, \ldots, i_s, j_s, i_{s+1} = i$ be any path from δ to i in G. By the definition of a covector graph, we obtain the sequence of equations $a_{j_t i_t} + x_{i_t} = a_{j_t i_{t+1}} + x_{i_{t+1}}$ for all the tuples (i_t, j_t, i_{t+1}) with $t \in [s]$. Summing up these equations yields $\sum_{t=1}^{s}(a_{j_t i_t} + x_{i_t}) = \sum_{t=1}^{s}(a_{j_t i_{t+1}} + x_{i_{t+1}})$. Equivalently, we obtain

$$\sum_{t=1}^{s} x_{i_{t+1}} - \sum_{t=1}^{s} x_{i_t} = \sum_{t=1}^{s} a_{j_t i_t} - \sum_{t=1}^{s} a_{j_t i_{t+1}}$$

and hence, $x_i - x_\delta = x_{i_{s+1}} - x_{i_1} = \sum_{t=1}^{s} a_{j_t i_t} - \sum_{t=1}^{s} a_{j_t i_{t+1}}$. These equations define x uniquely up to adding multiples of the all ones vector. Since we assumed G to be a covector graph, these necessary conditions are also sufficient. This construction is visualized in Figure 3. It proves the following.

Lemma 9. *The covector graph of x with respect to A is G.*
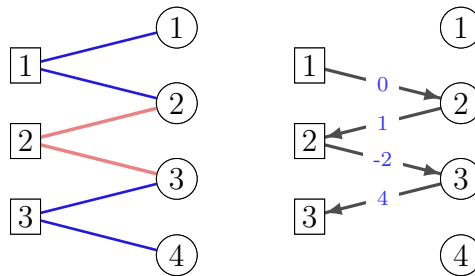

Figure 3: The computation of the point $(0, 1, 3)$ for a prescribed covector graph from Example 8.

For subsets $I \subseteq [d]$ and $J \subseteq [n]$ with $|J| = |I| - 1$ we define the *tropical Cramer solution* $A[J|I] \in \mathbb{T}^d$ by

$$A[J|I]_i = \begin{cases} \text{tdet}(A_{J, I \setminus \{i\}}) & \text{for each } i \in I \\ \infty & \text{else} \end{cases} \tag{2}$$

To cover the case $J = \varnothing$, we set $\text{tdet}(A_{\varnothing, \varnothing}) = 0$. These vectors occur as solutions to homogeneous tropical equality systems, see, e.g., [31, Theorem 18], [58, Corollary 5.4], in analogy to Cramer's rule in linear algebra. That means, a Cramer solution has the property that the minimum in the computation of each entry of $A_{J,I} \odot A[J|I]$ is attained at least twice. We mention the computational problem in Section 6, for an extensive study see [3].

We denote the generalized covector graph of $A[J|I]$ by $\mathcal{C}_A(J, I)$.

Example 10. Consider again the matrix A from Example 8. The point $(0, 1, 3)$ has the covector graph depicted on the left of Figure 3. On the right is the auxiliary weighted directed graph for computing the point from the covector graph.

It is the Cramer solution $\mathcal{C}_A(\{2, 3\}, \{1, 2, 3\})$.

Lemma 11. *Let $A \in \mathbb{T}^{(d-1) \times d}$ with $d \in \mathbb{N}$ and x the Cramer solution for this matrix. Then $|x_i - x_h| \leqslant 2 \cdot d \cdot \max\{|a_{ij}| \mid a_{ij} \neq \infty, (i,j) \in [d] \times [n]\}$ for any $i, k \in [d]$ with $x_i \neq \infty \neq x_k$.*

Proof. This follows from the definition of Cramer solution with the triangle inequality. $\square$

3 Abstract tropical linear programming

The generalization of the simplex method to oriented matroids in [12, 30, 66, 64], was a powerful step in the understanding of linear programming. The basic idea of the simplex method, considered as a feasibility algorithm, is taking a finite walk along edges and vertices in an arrangements of halfspaces as depicted in Figure 4. This is further explained in the Appendix A.1.
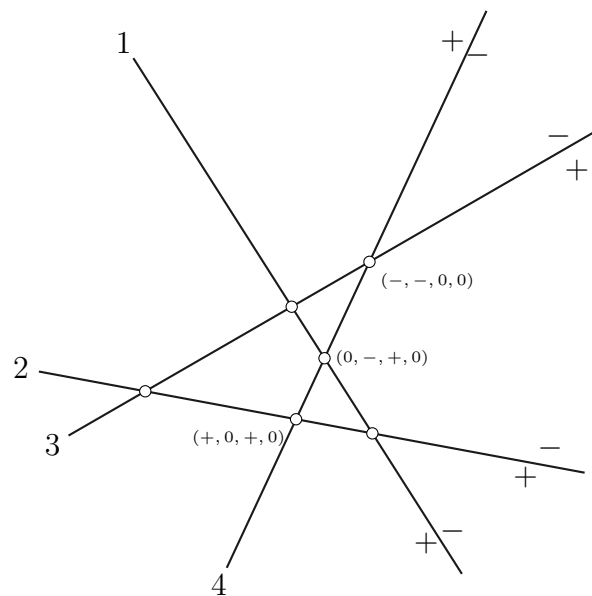

Figure 4: An affine halfspace arrangement in $\mathbb{R}^2$. The sign vectors denote in which halfspace of $1, 2, 3, 4$ the vertex of the arrangement lies. These signs form the sets J, K^+ and K^- in the explanation before Theorem 77.

In this section, we present an algorithm which finds a feasible cell in a tropical analogue of an oriented matroid and does not cycle. This is an abstraction of the feasibility problem for signed systems. We already saw in Proposition 3 and Proposition 7 that the feasibility of a point can be characterized by its covector graph with the signs on its edges. Hence, we will use an abstract version of covector graphs.

A purely axiomatic approach to grasp the crucial properties of the collection of covector graphs was started by Ardila and Develin in [8]. They introduced the name *tropical oriented matroid*. This approach was further developed in [54] and [43]. Finally, Horn proved in [43] that tropical oriented matroids encode exactly all subdivisions of $\Delta_{n-1} \times \Delta_{d-1}$, not only regular ones, and also the so called *tropical pseudo-hyperplane arrangements*.

In this section, we introduce a special class of signed tropical matroids and postpone their general introduction to Section 4. The class of signed tropical matroids treated here is the generalization of the set of covector graphs arising from generic matrices with only finite entries. In Section 5, we describe how one can reduce the general case to this. It follows from Lemma 54, Proposition 57 and Proposition 62.

3.1 Generic full signed tropical matroids

The following definition is based on the axioms for the maximal simplices in a triangulation of a product of two simplices given in [7, Proposition 7.2]. This is discussed further in Section 4.1.

Definition 12. A generic full signed tropical matroid (GFSTM) is a pair $(\mathcal{T}, \Xi)$. Here, Ξ is a $(d \times n)$-matrix in $\{+, -\}^{d \times n}$. Moreover, $\mathcal{T}$ is a set of spanning trees of the complete bipartite graph $K_{d,n}$ on $[d] \sqcup [n]$ together with all subgraphs of the trees without an isolated node in $[n]$, where the trees fulfill the following two properties.

1. For each tree G and each edge e of G either $G - e$ has an isolated node or there is another tree G containing $G - e$.

2. There do not exist two distinct trees G and H, and a cycle of $K_{d,n}$ which alternates between edges of G and H,

One can equivalently just consider the collection of trees without their subgraphs. We will refer to the last condition as *comparability*. Equivalently, one could require, that for all $D \subseteq [d]$ and $N \subseteq [n]$ with $|D| = |N|$ there is at most one matching on $D \sqcup N$ which is contained in a tree in $\mathcal{T}$.

Remark 13. The covector graphs of a signed system with a generic coefficient matrix and finite entries gives rise to a GFSTM. Such a GFSTM is *realizable*.

Extending the terminology for signed systems, we call the elements of $\mathcal{T}$ *covector graphs*. The nodes in $[d]$ are *coordinate nodes* and the nodes in $[n]$ are the *apex nodes*. Depending on the corresponding sign in the sign matrix, we say that an edge of a covector graph is *negative* or *positive*.

Example 14. Let

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -2 & -1 \\ 0 & -1 & -2 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} - & + & + \\ + & - & + \\ + & + & - \end{pmatrix}$$

That signed system (A, Σ) gives rise to the GFSTM depicted in Figure 5. The bipartite trees are just the maximal covector graphs as defined in Definition 1.

We give examples of GFSTMs, which do not come from signed systems, at the end of the section in Examples 45 and 46.

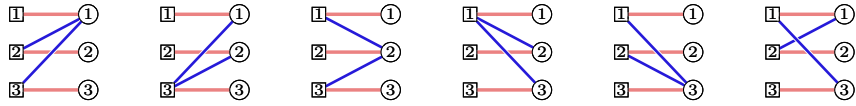Proposition 7 justifies the following definition.

Figure 5: A realizable GFSTM for the signed system in Example 14. Negative edges are red, positive edges are blue.

Definition 15. A covector graph G is *infeasible* if and only if there is an apex node in G which is only incident with negative edges. If G is not infeasible we call it *feasible*.

A covector graph G is *totally infeasible*, if it is infeasible and every coordinate node is incident with a negative edge.

A GFSTM is *feasible*, if it contains a feasible covector graph; otherwise we call it *infeasible*.

The property 'totally infeasible' is far stronger than just infeasible. In Lemma 17, we see that it forms a certificate that there is no feasible covector graph in the GFSTM and, hence, that the GFSTM is infeasible. It is in some sense dual to the notion of a feasible covector graph via the duality between the two parts of the node set of the bipartite graphs. In the generic case, the infeasibility of a totally infeasible covector graph already follows from the second condition.

The following operation corresponds to the treatment of points with infinite entries. We will treat this in more generality in Section 4.2.

Definition 16 (Contraction)**.** For a coordinate node $i \in [d]$, the *contraction* $\mathcal{S}_{/i}$ is the set of graphs which arise from those graphs of $\mathcal{S}$, for which i is isolated, by deleting the node i. We delete the ith column in the sign matrix.

For the contraction $\mathcal{S}_{/([d] \setminus D)}$, where $\mathcal{S}$ is defined on $[d]$ and $D \neq \varnothing$, we will also write $\mathcal{S}|_D$. In the realizable case, these are the covectors of the points with support D. We only consider points in $\mathbb{TA}^d = \mathbb{T}^d_{\min} \setminus \{(\infty, \ldots, \infty)\}$ which corresponds to $D \neq \varnothing$.

If $(\mathcal{S}, \Sigma)$ is induced by a signed system (A, Σ) then the operation corresponds to deleting the ith column of A. By construction, a contraction of a GFSTM is again a GFSTM.

With the latter notion we can now formulate an important consequence of the existence of a totally infeasible covector in a GFSTM. This is visualized in Figure 6.

Lemma 17 (Infeasibility certificate)**.** *If a covector graph G in a generic full STM $(\mathcal{T}, \Xi)$ is totally infeasible, then in every covector graph H of any contraction of $(\mathcal{T}, \Xi)$ there is a node in $[n]$ which is only incident with a negative edge.*

Proof. Notice that each covector graph in a contraction is constructed from a covector graph of $(\mathcal{T}, \Xi)$. Since one only removes isolated coordinate nodes to obtain the contraction, all covectors in every contraction is infeasible if so are all covector graphs in the original GFSTM.

By definition, G is infeasible and there is a set μ of negative edges on $[d] \sqcup N$ for some subset $N \subseteq [n]$, such that each node in $[d]$ is incident with exactly one edge in μ and each node in N is incident with at least one edge in μ.

Now, let H be any covector graph in $(\mathcal{T}, \Xi)$. Assume H is feasible. This implies that each apex node $j \in N$ is incident with a positive edge, which therefore does not lie in μ. Pick for each node in N one incident positive edge from H. This forms a cover η of N. Moreover, let D be the subset of the coordinate nodes $[d]$ which is covered by η. Then the graph on $D \sqcup N$ with edge set $\mu|_D \cup \eta$, where $\mu|_D$ are those edges in μ incident with D, has $|D| + |N|$ nodes and $|\mu|_D| + |\eta| \geqslant |D| + |N|$ edges. This implies that it contains a cycle C. Since every node in D is only incident with one edge from $\mu|_D$ and every node in N is only incident with one edge from η, the cycle C has to be alternating between μ and η. However, this contradicts the comparability in Definition 12. $\qquad\square$
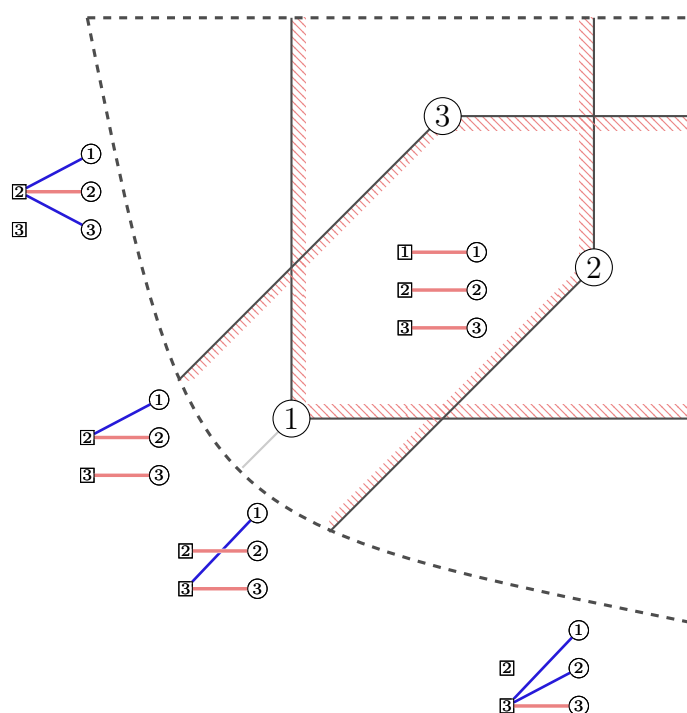
Figure 6: The configuration for the signed system from Example 14. The corresponding GFSTM contains a totally infeasible covector. The shaded bars indicate the *infeasible regions*. The dashed lines denote the boundary strata of the tropical projective space. The covectors on the boundary stratum corresponding to the contraction $\mathcal{T}|_{\{2,3\}}$ are also depicted and infeasible.

3.2 Existence of particular covector graphs

We start with a Menger-type lemma; see [13, §3] for similar results. It is purely graph theoretic but contains an important property for covector graphs.

Lemma 18. *Let G be a bipartite tree on the node set $D \sqcup N$ for arbitrary sets D and N with $|D| = k + 1$ and $|N| = k$ with a positive integer k. If the nodes in N all have degree 2 then, for each $i \in D$, the graph G with i deleted contains a perfect matching. Furthermore, G is the union of these matchings.*

Proof. Fix an arbitrary $i_0 \in D$. Since G is a tree, it has at least two leafs. In particular, there is an $i \in D \setminus \{i_0\}$ which is a leaf in G. Let $j \in N$ be the node adjacent to i. Deleting i and j yields a graph H on $(D \setminus \{i\}) \sqcup (N \setminus \{j\})$ for which each node in $N \setminus \{j\}$ has degree 2.

Proceeding by induction implies the claim about the containment of the matchings.

Furthermore, each edge is contained in such a perfect matching. For this, pick an arbitrary edge $(i, j) \in G$. Let $\ell \in D$ be the node distinct from i which is adjacent to j. Then (i, j) is contained in the perfect matching on $(D \setminus \{\ell\}) \sqcup N$. $\square$

The following result guarantees the existence of covector graphs with specific degree conditions. It can be seen as the theoretical justification of an oracle which provides us with the next covector graph in the iteration later on. Let $\mathcal{T}$ be a collection of spanning trees on $[d] \sqcup [n]$ fulfilling the two properties listed in Definition 12.

Proposition 19 ([54, Proposition 2.5]). *Let $(d_1, \ldots, d_n) \in [d]^n$ with $\sum_{j=1}^{n} d_j = n + d - 1$. There is exactly one tree in $\mathcal{T}$ for which each node $j \in [n]$ has degree d_j.*

The origin and a proof of the last statements are further discussed in the Appendix.

Our algorithm presented in Section 3.3 relies on the idea of pivoting between basic points similar to the simplex method. As further elaborated in Section A.1 the simplex method iterates over basic points. These are solutions of linear equality systems which can be computed by Cramer's rule. The next definition is motivated by the properties of the covector graph of a tropical Cramer solution as defined in (2).

We define *Cramer covectors* $\mathcal{C}(N, D \cup \{\delta\})$, where $\delta \in [d]$, $D \subseteq [d] \setminus \{\delta\}$ and $N \subseteq [n]$ with $|D| = |N|$, as the covector graphs in the contraction $\mathcal{T}|_{\{D \cup \delta\}}$ for which each node in N has degree 2. The former lemma guarantees the existence of Cramer covectors in a GFSTM. Note that it is also valid for $D = N = \varnothing$.

Cramer covectors are similar to linkage trees in the sense of [63] which were defined for the study of matching fields. Linkage trees are spanning trees on $k + 1$ nodes for which the k edges are bijectively labeled by the numbers in $[k]$. We replace each edge connecting j_0 with j_1 for $j_0, j_1 \in [k + 1]$ with label i for $i \in [k]$ by a new node with label i and two edges connecting j_0 with i, respectively j_1 with i. This yields a bipartite graph as in Lemma 18 which is essentially a Cramer covector.

Remark 20. [3, Theorem 4.18] implies that the covector graph of $A[J|I]$ for a generic, finite A is just the Cramer covector $\mathcal{C}(J, I)$ since there is a unique covector graph with the prescribed degree sequence. We will determine the covector graph for the non-generic case in Lemma 56.

We saw already in Lemma 19 and Lemma 18 that Cramer covectors have a particularly useful structure. We exploit this to construct Cramer covectors in a fixed GFSTM inductively.

Example 21 (Example 14 continued)**.** The sequence of degrees of the nodes on the right of each tree in Figure 5 has the sum $3 + 3 - 1 = 5$. We have exactly the degree sequences $(3, 1, 1)$, $(2, 2, 1)$, $(1, 3, 1)$, $(1, 2, 2)$, $(1, 1, 3)$, $(2, 1, 2)$. This demonstrates Proposition 19. The second, fourth and sixth covector graph in Figure 5 is a Cramer covector.

3.3 Description of the algorithm

We introduce an algorithm which either finds a feasible or a totally infeasible covector graph in a GFSTM. By Lemma 17, a totally infeasible covector is a certificate that such a GFSTM does not contain a feasible covector.

Like the variant of the simplex method presented in Subsection A.1, the algorithm constructs a sequence of subsets (a basis) of apex nodes (which correspond to inequalities). In each step, we consider a covector which is defined by this sequence and check if it is feasible. Here, we assume that we have an oracle which gives us a basic covector defined by a basis formed of apex nodes. If it is not feasible yet, there is an apex node which is only incident with negative edges (corresponding to a violated inequality). This determines which apex (variable) will enter the basis. For classical oriented matroid programming, this is described in, e.g., [12, Theorem 4.5].

Now, our approach diverges. While in the simplex method, one has to compute which variable leaves the basis, we deduce from Lemma 23 with the properties of a *basic covector* which apex leaves the basis. This can already be seen in Figure 7. To arrive at this insight, we will prove in Subsection 3.4 that moving along abstract tropical lines yields a basic covector if we start from one.

Furthermore, the termination of the simplex method is guaranteed by the increase of a linear functional. As we are working in a setting without weights such an argument is not at hand. However, again the special structure, in particular the preservation of the *distinguished direction*, of the basic covectors yields a purely combinatorial tool to measure the progress of the algorithm. The distinguished direction corresponds to the coordinate with respect to which one would dehomogenize the tropical linear inequality system.

The powerful definition of a basic covector comes with the additional difficulty to find one. We will solve this in Subsection 3.5 by an inductive construction via contractions of a GFSTM.

To emphasize that covector graphs take the role of vectors in the classical simplex method we denote them by y.

A *basic covector (graph)* y with *distinguished direction* δ and *support* $(D \cup \{\delta\}) \subseteq [d]$ with $D \subseteq [d] \setminus \{\delta\}$ is a covector graph on $[d] \sqcup [n]$ such that

1. it is a spanning tree on $(D \cup \{\delta\}) \sqcup N$,

2. each coordinate node in $[d] \setminus (D \cup \{\delta\})$ is isolated,

3. there is a $|D|$-set of apex nodes $N \subseteq [n]$, called *basis*, so that each node in N has degree 2 in y,

4. δ is not adjacent to an apex node in N via a negative edge,

5. each apex node in N is incident with a positive and a negative edge,

6. no two negative edges, each of which is incident with some node in N, are adjacent.

The apex nodes in the *basis* are called *basic apices*, the others *non-basic apices*. If Σ has a '$-$' at position $i \in [d]$ in row $j \in [n]$, we say that the apex node j has *shape i* resp. it is *i-shaped*.

Later on, we will construct a sequence of basic covectors. If there are apex nodes $p \neq q \in [n]$ so that N and $N \setminus \{p\} \cup \{q\}$ are bases, we say that p is the *leaving apex* and q is the *entering apex*.
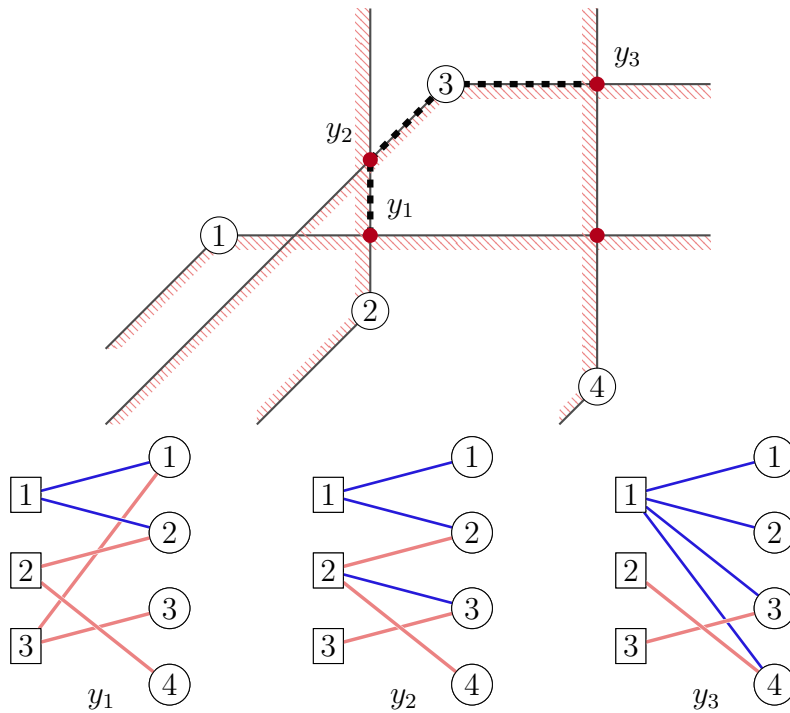
Figure 7: A path (dashed) along points with basic covectors (the four red points). The infeasible region is marked. In each step, a negative edge is removed from the covector graph. The bases are $\{1, 2\}$, $\{2, 3\}$ and $\{3, 4\}$.

Example 22. The graphs at the bottom of Figure 7 are the covector graphs of the points P_1, P_2 and P_3 in the top part. They are all basic covectors. The distinguished direction is $\delta = 1$. The corresponding bases are $\{1, 2\}$, $\{2, 3\}$ and $\{3, 4\}$. The apices 2 and 4 are 2-shaped, the apices 1 and 3 are 3-shaped.

We start with the nice structural property of basic covectors which connects the sign structure with the matching structure.

Lemma 23. *The negative edges which are incident with a basic apex form a perfect matching on $D \sqcup N$ in y. Furthermore, the edges in a path emerging from δ to another coordinate node are alternatingly positive and negative.*

Proof. Consider the induced subgraph $\tilde{y}$ of y on $(D \cup \{\delta\}) \sqcup N$. Each apex node is incident with a negative edge. By (5) and (6) in the definition, no two negative edges are incident, and by (4), δ is not incident with a negative edge. Hence, the negative edges define an injective function from N to D. Because of $|N| = |D|$, this function is also bijective. This yields the required matching.

Since each node in N has degree 2 and the nodes in $[d] \setminus (D \cup \{\delta\})$ are isolated, $\tilde{y}$ is a tree. Fix an arbitrary $i \in D$ and let $\rho = (e^0, e^1, \ldots, e^k)$ be the edge sequence from δ to i in $\tilde{y}$. Since e^0 is positive and incident with the same apex node as e^1 we conclude that e^1 is negative. Therefore, e^2 has to be positive again as it is incident with the same coordinate node as e^1. Iterating this argument, we obtain that the edges in ρ are alternatingly positive and negative. $\qquad\square$

The former lemma tells us that there is exactly one i-shaped apex node for each $i \in D$ in the basis N. From Proposition 19, we know that there is at most one basic covector defined by $(D \cup \{\delta\})$ and N. If the Cramer covector $\mathcal{C}(N, D \cup \{\delta\})$ fulfills the conditions 4, 5 and 6, it is the basic covector with these parameters and we denote it by $\mathcal{B}(N, D, \delta)$.

Corollary 24. *The Cramer covector $\mathcal{C}(N, (D \cup \{\delta\}))$ is the basic covector $\mathcal{B}(N, D, \delta)$ if and only if the negative edges, which are incident with the basic apices, form a perfect matching on $D \sqcup N$.*

3.4 Pivoting between basic covectors

The crucial piece for our feasibility algorithm is a method to find a new basic covector which is 'in the right direction' and 'similar to the old one'. In particular, the new basic covector should have the same distinguished direction. We present two variants for this in Algorithm 1 and Algorithm 2. The second one will evolve as an iteration over the first one. We need the first one for technical reasons in the proofs.

Assumption 1. The GFSTM is *trimmed* which means that Ξ has exactly one '$-$' entry in each row.

We discuss in Section 5.3 how one can make sure that this assumption is fulfilled.

Now, the idea for the pivoting is the following. If we remove a negative edge e which is incident to a basic apex p in a basic covector y with basis N then we obtain the covector graph $y - e$ having two trees as connected components and p leaves the basis. In this context, $-$ denotes set difference of the edge sets. We know from Definition 12 that there is exactly one other tree w containing this graph. Hence, there is an edge f such that $w = y - e + f$ where $+$ denotes union.

Now, three cases can occur. If w is again a basic covector graph with distinguished direction δ, we are done. Otherwise, either an apex node in N has degree 3 or another apex node has degree 2. We continue the iteration by removing an edge. This edge is

chosen such that no node becomes isolated and all nodes in $N \setminus \{p\}$ have degree $\geqslant 2$ as well as one negative incident edge. This ensures that δ remains the distinguished direction and yields the case distinction of Algorithm 1. A closer inspection reveals that we do not need to iterate over all these covectors to find another basic covector but can construct it directly which results in Algorithm 2. For the proof of this, we assigned the variable *completed* in Line 11 of Algorithm 1. The latter algorithm is merely a technical tool to show that the other algorithms building on it behave correctly.

Remark 25. The iteration in Algorithm 1 moves along an abstract version of a "tropical line". A tropical line is a sequence of ordinary lines as explained in [20, Proposition 3]. A more refined version for this is given in [5, §4]. Note that their description in terms of the "tangent digraph" is essentially the same as in terms of covector graphs in the realizable case. However, our approach also works in the non-realizable case.

Algorithm 1 Finding the next basic covector; see also Algorithm 2

Input: Basic covector graph $y = \mathcal{B}(N, D, \delta)$ and a non-basic apex r that is adjacent to D via a negative edge in y

Output: Basic covector graph with support $D \cup \delta$ and distinguished direction δ

  1: **procedure** NEXTBASICCOVECTOR(y,r)
  2:      $i \leftarrow$ coordinate node adjacent to r
  3:      $p \leftarrow$ basic apex adjacent to i via a negative edge   ▷ *the i-shaped basic apex of the basis N. It leaves the basis.*
  4:      $e \leftarrow$ edge connecting i and p
  5:      **do**
  6:         $w \leftarrow$ unique covector $\neq y$ in $\mathcal{T}|_{D \cup \{\delta\}}$ containing $y - e$ ▷ *see Def. 12*
  7:         $f \leftarrow w - (y - e)$
  8:         $q \leftarrow$ the apex node incident with f
  9:         **if** q is adjacent to i via a negative edge **then**
10:                              ▷ *w is the basic covector $\mathcal{B}(N \setminus p \cup q, D, \delta)$.*
11:            completed$\leftarrow (q = r)$
12:         **else if** q has degree 3 in w **then**
13:            $e \leftarrow$ the positive edge incident with q in $y - e = w - f$.
14:         **else**   ▷ *In this case, q is incident with two edges.*
15:            $e \leftarrow$ the edge incident with q in $y - e = w - f$.
16:         **end if**
17:         $y \leftarrow w$
18:      **while** y is no basic covector
19:      **return** y
20: **end procedure**

We build our arguments for the correctness of the algorithms on properties of the paths in basic covectors. Let the *length of a path* in a graph be the number of nodes contained in the path. Define the *δ-distance of an edge e* in the covector graph y as the minimum of the two lengths of the paths from a fixed coordinate node δ to the nodes

which are incident with e. Note that the path between two nodes in a tree is unique. We call the edge e *even* in y if the distance to the coordinate node δ is even, otherwise *odd*. We call this property the δ-*parity of an edge* in y.

3.4.1 Finding the next basic covector

Let y^0 be the input covector, r the input basic apex and p the leaving basic apex of shape i. We consider the sequence $y^1, y^2, \ldots$ of covectors which arise in Algorithm 1 in Line 6. Such a sequence is depicted in Figure 8. Then we can write $y^1 = y^0 - e^0 + f^1, y^2 = y^1 - e^1 + f^2, \ldots$ for appropriate edges e^ℓ and f^ℓ with $\ell \in \mathbb{N}$. Furthermore, let q^ℓ be the apex node, which is incident with f^ℓ in y^ℓ.
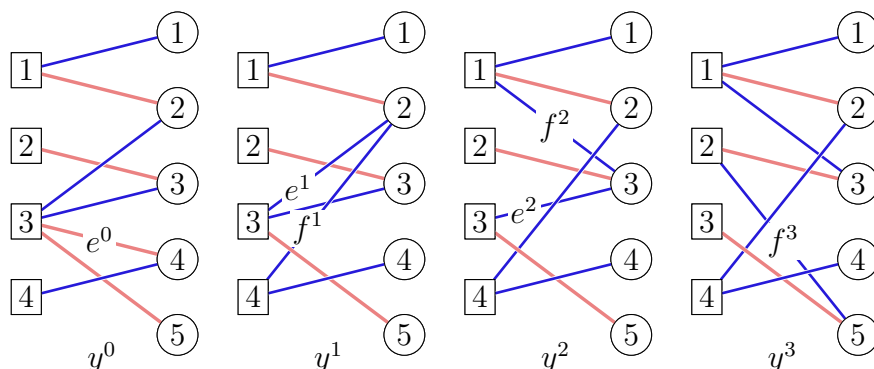

Figure 8: A possible sequence of covector graphs starting with an infeasible and ending with a feasible basic covector. Negative edges are light red, coordinate nodes left, apex nodes right, $\delta = 4$. The intermediate covectors are not basic.

Example 26. Figure 8 depicts a possible sequence of covectors arising in Algorithm 1 Line 6. The first and the last covector are basic with basis $\{2, 3, 4\}$ resp. $\{2, 3, 5\}$. The distinguished direction is $\delta = 4$.

In the realizable case, the two apices 2 and 3 would define a tropical line which eventually has to hit the halfspace defined by the apex node 5.

Lemma 27. *The covector graph $y^\ell - e^\ell$ has two connected components for all $\ell \geqslant 0$. Each node in $N \setminus \{p\}$ has degree 2 and is incident with a positive and a negative edge. All other apex nodes have degree 1. The negative edges, which are incident with a node in $N \setminus \{p\}$, are pairwise not adjacent.*

Proof. By construction, y^ℓ is always a tree, hence $y^\ell - e^\ell$ has two connected components. Line 13 ensures the properties of the nodes in $N \setminus \{p\}$. Line 15 guarantees that the other apex nodes have degree 1. The last claim follows as the negative edges, which are incident with a node in $N \setminus \{p\}$, are the same as in y^0. □

Since we started the iteration with a basic covector, we obtain a nice invariant which is fulfilled by the edges which are removed and added.

Lemma 28. *Let y^ℓ and $y^{\ell+1} = y^\ell - e^\ell + f^{\ell+1}$ be two consecutive covector graphs for $\ell \geqslant 0$. Then e^ℓ is even in y^ℓ and $f^{\ell+1}$ is odd in $y^{\ell+1}$.*

Proof. We proceed by induction. The first covector graph y^0 in the iteration is a basic covector.

From Lemma 23, we know that the paths from δ to another coordinate node are alternatingly positive and negative. We conclude that all the negative edges which are incident with a basic apex are even. Hence, line 4 in Algorithm 1 yields that e^0 is even as it is negative.

Now fix an $\ell \geqslant 1$ and consider the union $Y^\ell := y^{\ell-1} + f^\ell = y^\ell + e^{\ell-1}$ of $y^{\ell-1}$ and y^ℓ. There is a unique fundamental cycle in Y^ℓ which contains f^ℓ and $e^{\ell-1}$. An example for this is depicted in Figure 9. Consider the path ρ in Y^ℓ that contains $e^{\ell-1}$ and goes from δ to the first node incident with f^ℓ. By the induction hypothesis, $e^{\ell-1}$ is even in $y^{\ell-1}$. By the comparability condition in Definition 12, the fundamental cycle must not be alternating between edges of $y^{\ell-1}$ and y^ℓ. Therefore, with the evenness of $e^{\ell-1}$, the number of nodes in ρ must be even as well. Since the number of edges forming a cycle in a bipartite graph is even, this implies that the other path from δ to the first node incident with f^ℓ in Y^ℓ contains an odd number of nodes. This is exactly the path defining the δ-distance of f^ℓ in y^ℓ, hence, this δ-distance is odd.

To show that f^ℓ and e^ℓ have different parity in y^ℓ we consider the two cases in Algorithm 1 lines 13 and 15. The first case occurs if q^ℓ is a basic apex. Consider the path from δ to q^ℓ. By Lemma 27, the apex nodes along this path are only nodes in $N \setminus \{p\}$ and analogously to Lemma 23, we get that the path is alternatingly positive and negative. In particular, the path to the positive edge incident with q^ℓ with the higher δ-distance contains the other positive edge. Therefore, these two edges have different parity.

The second case occurs if q^ℓ is an apex node in $[n] \setminus (N \setminus \{p\})$ which has degree 2 in y^ℓ but is not of shape i. In this case, f^ℓ and e^ℓ are again incident with the same apex node q^ℓ. There is a unique path from δ to q^ℓ. Since it has to contain one of the two edges the claim follows. $\qquad\square$

Now, we have the tools to prove a first lemma which guarantees termination.

Lemma 29. *For $\ell \geqslant 1$, let $C^{\ell-1}$ be the set of nodes in the connected component of the distinguished direction δ in $y^{\ell-1} - e^{\ell-1}$. Then $q^\ell \notin C^{\ell-1}, q^\ell \in C^\ell$ and $C_1 \subsetneq C_2 \subsetneq \ldots$.*

Proof. Fix an arbitrary $\ell \geqslant 1$ indexing an element of the sequence (q^ℓ).

Not both endpoints of f^ℓ can be contained in $C^{\ell-1}$ as f^ℓ connects the two components of $y^{\ell-1} - e^{\ell-1}$. The path from δ to the endpoint of f^ℓ in y^ℓ has to be odd, by Lemma 28. Since such a path has to alternate between coordinate and apex nodes, this endpoint has to be a coordinate node. Hence, q^ℓ is not contained in $C^{\ell-1}$.

By the choice of e^ℓ in Line 13 or Line 15 of Algorithm 1, e^ℓ is incident with q^ℓ. Since e^ℓ is contained in $y^{\ell-1} - e^{\ell-1}$, the endpoint of e^ℓ different from q^ℓ must not lie in $C^{\ell-1}$, otherwise q^ℓ would lie in $C^{\ell-1}$. Subsuming, no endpoint of e^ℓ lies in $C^{\ell-1}$. Therefore, q^ℓ and the nodes in $C^{\ell-1}$ cannot be disconnected from δ in $y^\ell - e^\ell$. Hence, $q^\ell \in C^\ell$ and $C^{\ell-1} \subsetneq C^\ell$. $\qquad\square$
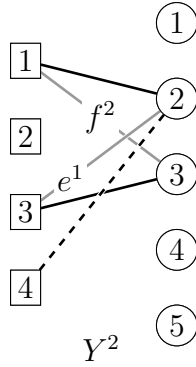
Figure 9: The fundamental cycle for y^1 and y^2 in Figure 8. The two graphs coincide in the black edges and differ in the green edges. The dashed edge connects the cycle with $\delta = 4$.

Example 30. The connected components of δ in the covector graphs in Figure 8 are $\{4, \overline{4}\}$, $\{1, 4, \overline{1}, \overline{2}, \overline{4}\}$, $\{1, 2, 4, \overline{1}, \overline{2}, \overline{3}, \overline{4}\}$, where the numbers with the line on top denote apex nodes.

Theorem 31. *Algorithm 1 does not cycle and yields a new basic covector with distinguished direction δ and support $(D \cup \delta)$ after less than n iterations.*

Proof. Note that the condition in Line 9 is fulfilled if q equals r. By Lemma 29, the set C^{ℓ} is increased by at least one apex node. Since there are only n apex nodes and the set fulfilling the condition in Line 9 is not empty, the algorithm terminates after less than n iterations.

Furthermore, the condition that $w \in \mathcal{T}|_{D \cup \{\delta\}}$ ensures that each coordinate node in $[d] \setminus (D \cup \{\delta\})$ is isolated. The condition in Line 9 together with Lemma 27 yields that the resulting covector graph is indeed a basic covector with distinguished direction δ. $\square$

If r does not enter the basis to form the new basic covector in Algorithm 1, it is still a non-basic apex, which is incident with a negative edge. Therefore, the following block yields the basic covector $y = \mathcal{B}(N \setminus p \cup r, D, \delta)$ where p is the leaving basic variable which has the same shape as r.

---

   completed← FALSE
   while not completed **do**
      NEXTBASICCOVECTOR(y,r)   ▷ *see Algorithm 1*
   end while   ▷ *If r does not become a basic apex it can be used again.*

This implies that $\mathcal{C}(N \setminus p \cup r, D \cup \{\delta\})$ is indeed a basic covector.

The former observations imply the following.

Corollary 32. *Algorithm 2 is correct and has the same result as an iterative application of Algorithm 1.*

Algorithm 2 Simplified variant of Algorithm 1 for finding the next basic covector

---

Input: Basic covector graph $y = \mathcal{B}(N, D, \delta)$ and a non-basic apex r that is adjacent to D via a negative edge in y

Output: The basic covector graph $\mathcal{B}(N \setminus p \cup r, D, \delta)$ where p is of the same shape as r

  1: **procedure** NextBasicCovector(y,r)
  2:     $i \leftarrow$ coordinate node adjacent to r
  3:     $p \leftarrow i$-shaped basic apex of the basis N
  4:     **return** $\mathcal{C}(N \setminus p \cup r, D \cup \{\delta\})$
  5: **end procedure**

Example 33. Observe that y^0 is the basic covector $\mathcal{B}(\{2,3,4\}, \{1,2,3\}, 4)$ and y^3 is the basic covector $\mathcal{B}(\{2,3,5\}, \{1,2,3\}, 4)$ in Figure 8. That illustrates Corollary 32 as the apex nodes 4 and 5 are both 3-shaped and 5 is a non-basic apex node incident with a negative edge in y^0.

3.4.2   Finding an extreme basic covector

Eventually, we want to determine a feasible or totally infeasible basic covector. A feasible covector cannot have an apex node of degree one which is incident with a negative edge. Therefore, we want to construct a new basic covector if there is such an edge. We know from the former section how this can be achieved. Iterating this approach yields Algorithm 4. To check if we reached a feasible or totally infeasible basic covector we need the subroutine CheckFeasible from Algorithm 3. It is just the algorithmic manifestation of Definition 15.

Remark 34. We are left with some freedom of choice for the entering apex at each basic covector. We do not specify a rule to choose the apex, the algorithms work for any choice. For an implementation we suggest to use the smallest index, like in Bland's rule for the simplex method.

Lemma 35. *Algorithm 3 correctly determines if $y = \mathcal{B}(N, D, \delta)$ is feasible, infeasible or totally infeasible in the sense of Definition 15.*

Proof. If the condition in Line 2 is fulfilled, the covector y is surely infeasible. Since, in a basic covector graph, all the coordinate nodes in D are incident to a basic apex via a negative edge, the condition in Line 3 implies that y is totally infeasible. The claim follows as feasible is the opposite of infeasible. $\qquad\square$

Algorithm 4 successively constructs basic covector graphs with Algorithm 2 until the result is feasible or totally infeasible.

At first, it is not clear that this terminates. We consider a run of this algorithm starting with the arbitrary basic covector y^0. Let y^k be a basic covector which is assigned in Line 5 of Algorithm 4 during this run. By Corollary 32, there is a sequence of covectors $y^0, y^1, \ldots, y^k$ (most of them not basic) which would occur as intermediate results by using Algorithm 1 instead of Algorithm 2.

Algorithm 3 Checking feasibility of a basic covector

Input: Basic covector graph $y = \mathcal{B}(N, D, \delta)$

Output: A classification of y based on the signs of the edges

1: **procedure** CHECKFEASIBLE(y,δ)
2:     **if** there is a non-basic apex node only incident with a negative edge **then**
3:         **if** there is a negative edge incident with δ **then**
4:             **return** TOTALLY–INFEASIBLE
5:         **else**
6:             **return** INFEASIBLE
7:         **end if**
8:     **else**
9:         **return** FEASIBLE
10:     **end if**
11: **end procedure**

Algorithm 4 Iterating over basic covectors

Input: Basic covector graph $y = \mathcal{B}(N, D, \delta)$

Output: A basic covector with support $(D \cup \delta)$ and distinguished direction δ which is either totally infeasible or feasible

1: **procedure** FINDEXTREMECOVECTOR(y)
2:     **while** (CHECKFEASIBLE(y, δ) = INFEASIBLE) **do**
3:         $r \leftarrow$ non-basic apex in y which is incident to D via a negative edge $\triangleright$ *such an r exists if y is infeasible, see Algorithm 3 Line 2 and 3*
4:         $p \leftarrow$ basic apex of y of the same shape as r
5:         $y \leftarrow \mathcal{C}(N \setminus p \cup r, D \cup \{\delta\})$
6:     **end while**
7:     **return** y
8: **end procedure**

Let $\mathcal{E}$ be the graph on $(D \cup \{\delta\}) \sqcup [n]$ whose set of edges are exactly those which are contained in all the graphs $y^0, \ldots, y^k$. Denote by $\mathcal{E}(\delta)$ the connected component in $\mathcal{E}$ containing δ and by $I(\delta)$ the subset of the coordinate nodes in $\mathcal{E}(\delta)$.

Proposition 36. *There is an apex node $j \in [n]$ and an $h \in [k]$ such that j has degree 2 in y^0 and degree 1 in y^ℓ for $\ell \geqslant h$. In particular, $y^k \neq y^0$.*

Proof. Since y^0 is connected there is an apex node j in y^0 which is connected to $I(\delta)$ and to $(D \cup \{\delta\}) \setminus I(\delta)$. The covector y^0 is basic and j has degree 2. Therefore, j is a basic apex.

If both edges incident with j are contained in $\mathcal{E}$ this would contradict the definition of $I(\delta)$. Therefore, there is an h so that the edge e^h, which is removed in step h, is incident with j. Since the edges of $\mathcal{E}$ are contained in all the graphs $y^0, \ldots, y^k$, the edge e^h has the same δ-distance in y^h as in y^0. With Lemma 23 and 28, the edge e^h is even and negative in y^h. Furthermore, the positive edge incident with j is incident with $I(\delta)$.

For $\ell \geqslant h$, no edge in $\mathcal{E}(\delta)$ is removed. Assume there would be an $\ell_0 \geqslant h$ so that f^{ℓ_0} is incident with j. Then f^{ℓ_0} would be even in y^{ℓ_0}. However, this contradicts Lemma 28. Subsuming, j has degree 1 in y^ℓ for $\ell \geqslant h$. $\qquad\square$

Remark 37. Geometrically, for the realizable case the set $\mathcal{E}(\delta)$ defines a lower dimensional tropical hyperplane, which contains all the points $y_1, \ldots, y_{k+1}$. It is given by the intersection of the boundaries of the tropical halfspaces which correspond to the apex nodes which are internal nodes of $\mathcal{E}(\delta)$.

For the non-realizable case, we only give the following rough upper bound. It is just the number of $|D|$-tuples analogously to the number of possible bases for the classical simplex method. We will give a better upper bound for the realizable case in Theorem 75.

Theorem 38. *Algorithm 4 terminates after less than $\binom{n}{|D|}$ iterations.*

Proof. By Proposition 36, any two basic covectors arising in Line 5 are distinct. Furthermore, the assignment of y as Cramer covector in that line yields an injective function from the $|D|$-subsets of $[n]$ to the basic covectors. This implies the claim. $\qquad\square$

Remark 39. In Algorithm 4, we could continue the iteration until only δ is incident with non-basic apices via negative edges. For other basic covectors, one still can apply Algorithm 2 to construct a new basic covector.

3.5 Finding a basic covector and even more

Until now, we assumed a basic covector to be given. Indeed, one easily finds a basic covector for each $\delta \in [d]$, namely the Cramer covector $\mathcal{C}(\varnothing, \{\delta\})$. Algorithm 4 allows us to determine a feasible or totally infeasible covector, which is even a basic covector. This covector lives in $\mathcal{T}|_{(D \cup \{\delta\}}$. If it is feasible then we are finished as we are only looking for a feasible covector in a contraction. However, a totally infeasible covector in $\mathcal{T}|_{(D \cup \{\delta\}}$ is not enough to guarantee the infeasibility of $\mathcal{T}$. On the other hand, we demonstrate how one

can construct a new basic covector in a contraction with a bigger support from a totally infeasible basic covector $y = \mathcal{B}(N, D, \delta)$.

This is based on the following proposition.

Proposition 40. *Let $D \subseteq [d]$, $\delta \in [d] \setminus D$ and $N \subseteq [n]$ with $|N| = |D|$. Furthermore, let y be a covector graph in the contraction $\mathcal{T}|_D$ containing a perfect matching μ on $D \sqcup N$. Then $\mathcal{C}(N, D \cup \{\delta\})$ contains μ.*

Proof. Applying Proposition 19 to $\mathcal{T}|_{(D \cup \{\delta\})}$ yields the existence of the covector graph $\mathcal{C}(N, D \cup \{\delta\})$ which has degree 2 for every node in N and degree 1 for the nodes in $[n] \setminus N$. By Lemma 18, the induced subgraph of $\mathcal{C}(N, D \cup \{\delta\})$ on $(D \cup \{\delta\}) \sqcup N$ contains a matching on $D' \sqcup N$ for every $|D|$-element subset D' of $(D \cup \{\delta\})$. Especially, it contains a perfect matching on $D \sqcup N$.

By the definition of the contraction $\mathcal{T}|_D$, there is a covector graph $\overline{y}$ in $\mathcal{T}|_{(D \cup \{\delta\})}$ extending y. The comparability condition yields that the two graphs $\overline{y}$ and $\mathcal{C}(N, D \cup \{\delta\})$ must contain the same matching μ on $D \sqcup N$. $\qquad\square$

By Definition 15 resp. Algorithm 3, there is a non-basic apex j in y which is incident to δ via a negative edge. Therefore, y contains a perfect matching μ on $(D \cup \{\delta\}) \sqcup (N \cup \{j\})$ which consists of negative edges. Consider an additional element $\delta' \in [n] \setminus (D \cup \{\delta\})$. By Proposition 40, the covector $y' = \mathcal{C}((N \cup \{j\}), (D \cup \{\delta\} \cup \{\delta'\}))$ also contains μ. With Corollary 24, we conclude that y' is the basic covector $\mathcal{B}((N \cup \{j\}), (D \cup \{\delta\}), \delta')$. Note that this argument works for any covector y which contains a matching of negative edges on $(D \cup \{\delta\}) \sqcup (N \cup \{j\})$.

Theorem 41. *Algorithm 5 correctly determines a totally infeasible basic covector in $\mathcal{T}$ or a feasible covector in a contraction of $\mathcal{T}$ in at most $d - 1$ iterations of Algorithm 4.*

Proof. From the discussion above the theorem, we know that the covector in Line 20 is indeed a basic covector. By Theorem 38, y is a feasible or totally infeasible basic covector after Line 9, and Lemma 35 shows that CHECKFEASIBLE correctly determines the feasibility status of a basic covector. In each iteration of the while-loop in Line 4, the algorithm either terminates or D is increased by one element.

Since D is a subset of $[d]$ with at most $d - 1$ elements, the claim follows. $\qquad\square$

Remark 42. The only passages in the algorithm where the data of the GFSTM is needed are the assignments of the Cramer covectors. In the realizable case, the input for Algorithm 5 is supposed to be given as a signed system (A, Σ). We discuss this further in Section 6.1.

In the non-realizable case, we assume to have an oracle which returns a Cramer covector for each fixed δ, D and N. Recall their guaranteed existence by Proposition 19. The requirements on this oracle should be further investigated in the context of *matching ensembles* [55].

Corollary 43. *Algorithm 5 needs at most $\sum_{k=1}^{d} \binom{n}{k}$ calls to the oracle that encodes $(\mathcal{T}, \Sigma)$ and returns Cramer covectors.*

Algorithm 5 Finding a feasible or totally infeasible covector graph

---

Input: A full generic trimmed STM $(\mathcal{T}, \Sigma)$

Output: A totally infeasible basic covector or a feasible covector in a contraction of $\mathcal{T}$

1:   $\delta \leftarrow$ an element of $[d]$
2:   $D \leftarrow \varnothing,\ N \leftarrow \varnothing$
3:   $y \leftarrow \mathcal{C}(\varnothing, \{\delta\})$
4:   **while** TRUE **do**
5:      $check \leftarrow \text{CHECKFEASIBLE}(y, \delta)$ $\triangleright$ *see Algorithm 3*
6:      **if** $check = $ INFEASIBLE **then**
7:          $y \leftarrow \text{FINDEXTREMECOVECTOR}(y)$ $\triangleright$ *see Algorithm 4*
8:          $check \leftarrow \text{CHECKFEASIBLE}(y, \delta)$
9:      **end if** $\triangleright$ *at this point y is guaranteed to be feasible or totally infeasible*
10:     **if** $check = $ FEASIBLE **then**
11:        **return** "feasible",y
12:     **end if** $\triangleright$ *at this point y is guaranteed to be totally infeasible*
13:     **if** $D \cup \{\delta\} = [d]$ **then**
14:        **return** "infeasible",y
15:     **else**
16:        $j \leftarrow$ non-basic apex incident with δ via a negative edge $\triangleright$ *exists by Algorithm 3 Line 3*
17:        $D \leftarrow D \cup \{\delta\}$
18:        $\delta \leftarrow$ node in $[d] \setminus D$.
19:        $N \leftarrow N \cup \{j\}$
20:        $y \leftarrow \mathcal{C}(N, D \cup \{\delta\})$
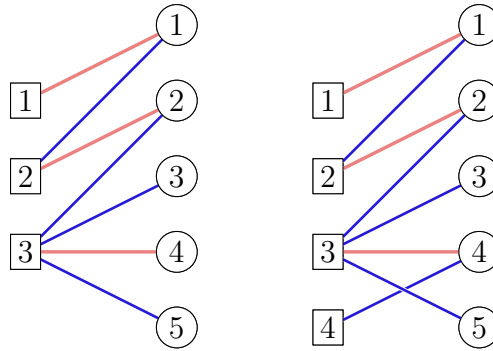21:   **end if**
22: **end while**

Figure 10: Constructing a basic covector with bigger support from a totally infeasible basic covector

Furthermore, the algorithm yields a partial generalization of [37, Lemma 11]. It is a theorem of alternatives for the feasibility of an STM. It covers a slightly different aspect than the "Tropical Farkas Lemma" [20, Proposition 9].

Theorem 44 (Tropical Farkas Lemma for GFSTM). *A full generic STM contains*

- *either a feasible covector in a contraction,*

- *or a totally infeasible covector,*

but not both.

Proof. By Theorem 41, Algorithm 5 returns a feasible or a totally infeasible covector. If the result is totally infeasible, Lemma 17 implies that the STM does not contain a feasible covector. This implies the claim. □

We demonstrate the course of the algorithms on two Examples from [42, 19] which are listed in Table 1. They are derived from two non-regular triangulations of $\Delta_5 \times \Delta_2$ and $\Delta_3 \times \Delta_3$; the connection between the covector graphs and triangulations is further described in Section 4.

The rows contain the covectors corresponding to the maximal simplices. The jth entry of a tuple contains the coordinate nodes which are adjacent to the apex node j. This is the compact form to write a covector, which was also used in, e.g., [20, 8].

Example 45. Figure 11 shows a sequence of basic covector graphs from the GFSTM given by the non-regular triangulation on the left of Table 1 and the sign matrix

$$
\Sigma = \begin{pmatrix} + & + & - \\ + & - & + \\ + & + & - \\ + & - & + \\ + & + & - \\ + & - & + \end{pmatrix} .
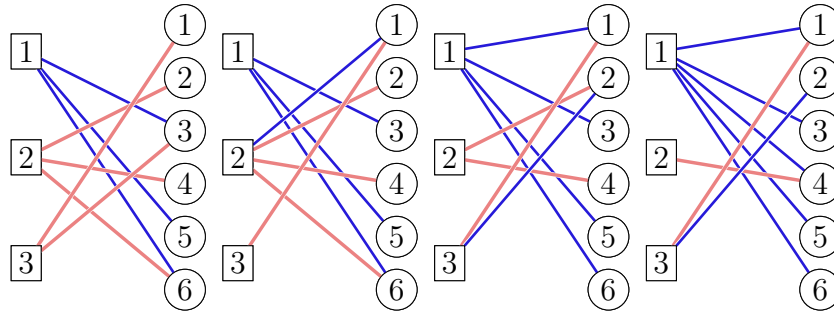$$

Figure 11: A sequence of basic covector graphs produced by a run of Algorithm 4, see Example 45. The first one is infeasible, the last one is feasible.

If we start Algorithm 5 with $\delta = 2$ then a possible sequence is given by the following table.

δ	Cramer covector	label	possible entering apex
2	$\mathcal{C}(\varnothing, \{2\}) = (2, 2, 2, 2, 2, 2)$	y^1	$2, 4, 6$
3	$\mathcal{C}(\{6\}, \{2, 3\}) = (3, 3, 3, 3, 3, 123)$	y^2	$1, 3, 5$
1	$\mathcal{C}(\{3, 6\}, \{1, 2, 3\}) = (3, 3, 13, 2, 1, 12)$	y^3	$1, 4$
	$\mathcal{C}(\{1, 6\}, \{1, 2, 3\}) = (23, 2, 1, 2, 1, 12)$	y^4	$2, 4$
	$\mathcal{C}(\{1, 2\}, \{1, 2, 3\}) = (13, 23, 1, 2, 1, 1)$	y^5	4
	$\mathcal{C}(\{1, 4\}, \{1, 2, 3\}) = (13, 3, 1, 12, 1, 1)$	y^6	

The last four covectors are depicted in Figure 11.

The non-regular subdivision is visualized in Figure 12 as a mixed subdivision via the Cayley trick. The black lines form "tropical pseudohyperplanes" in the sense of [8, §5] and [43, Theorem 4.2] which are dual to the mixed subdivision. The red points mark the cells which correspond to the basic covector graphs shown in Figure 11.

Example 46. Furthermore, we demonstrate a run of Algorithm 5 on the GFSTM given by the non-regular triangulation $\mathcal{T}$ on the right of Table 1 and the sign matrix

$$
\Sigma = \begin{pmatrix} - & + & + & + \\ + & - & + & + \\ + & + & - & + \\ + & + & + & - \end{pmatrix} .
$$

We start the algorithm with $\delta = 1$. The maximal covectors in the contractions are found by removing the nodes in $[d] \setminus (D \cup \{\delta\})$ and taking only those resulting graphs without isolated apex nodes.

The only covector in $\mathcal{T}|_{\{1\}}$ is $(1, 1, 1, 1)$. It is a totally infeasible basic covector and, with the new $\delta = 2$, we construct the basic covector $\mathcal{C}(\{1\}, \{1, 2\})$. The list of maximal
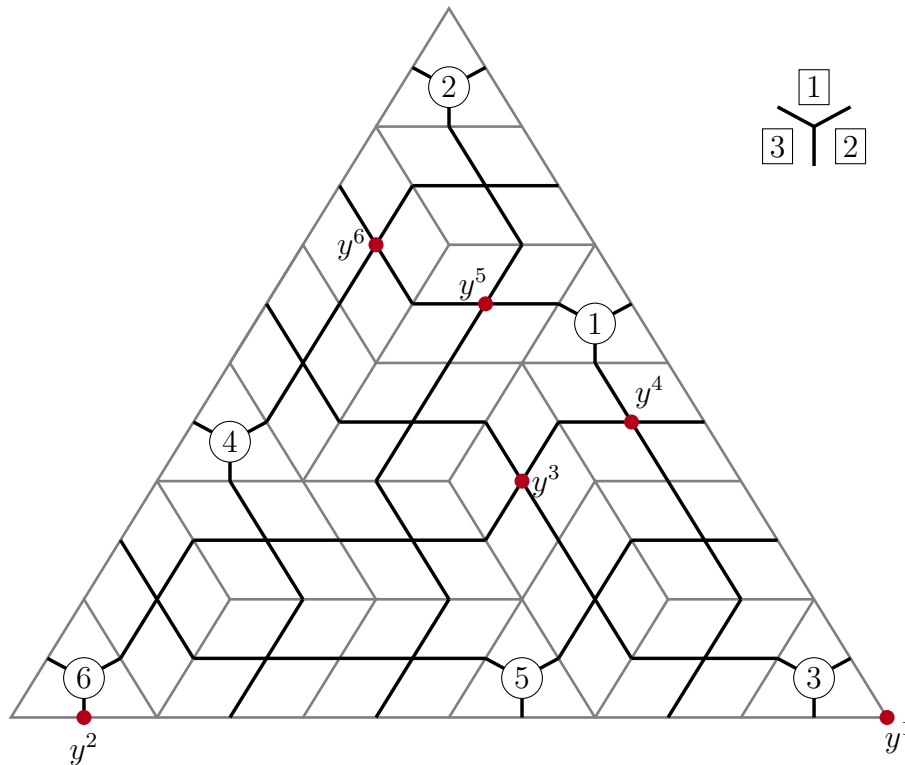
Figure 12: The non-regular subdivision from Example 45 represented as mixed subdivision of $6 \cdot \Delta_2$ which is possible through the Cayley trick. The black lines are tropical pseudo-hyperplanes in the sense of [43, Theorem 4.2]. The red intersection points correspond to basic covectors. This figure is basically the same as [40, Figure 3].

covectors in the contraction $\mathcal{T}|_{(\{1\} \cup \{2\})}$ is

$$(1, 12, 1, 1), (12, 2, 2, 2), (1, 2, 12, 1), (1, 2, 2, 12) \ .$$

So, the next basic covector is $(12, 2, 2, 2)$. It is already totally infeasible and no call to FINDEXTREME is necessary. With the new $\delta = 4$, we get $\mathcal{C}(\{1, 2\}, \{1, 2, 4\})$, which yields the covector $(14, 24, 4, 4)$.

Finally, the algorithm results in the totally infeasible basic covector $\mathcal{C}(\{1, 2, 4\}, [4])$. The just constructed sequence of basic covector graphs is depicted in Figure 13.

4   Signed tropical matroids

We considered a special class of signed tropical matroids to describe the algorithm in the previous section. Now, we introduce general signed tropical matroids. This relies on the notion of *polyhedral subdivisions*. We give a short introduction to the necessary polyhedral notions in the Appendix B.1 and refer to [68, 19] for further reading.

```
(1,    123,  1,    1,    1,    1)
(1,    23,   1,    12,   1,    1)
(123,  2,    1,    2,    1,    1)
(23,   2,    1,    2,    1,    12)
(23,   2,    1,    2,    12,   2)
(13,   23,   1,    2,    1,    1)
(13,   3,    1,    12,   1,    1)
(23,   2,    13,   2,    2,    2)
(2,    2,    123,  2,    2,    2)
(3,    2,    13,   2,    12,   2)
(3,    2,    13,   2,    1,    12)
(3,    23,   13,   2,    1,    1)
(3,    3,    13,   12,   1,    1)
(3,    3,    3,    123,  1,    1)
(3,    3,    3,    23,   1,    12)
(3,    3,    3,    23,   13,   2)
(3,    23,   3,    2,    1,    12)
(3,    23,   3,    2,    13,   2)
(3,    2,    3,    2,    123,  2)
(3,    3,    3,    3,    3,    123)
(3,    3,    3,    3,    13,   12)
```

```
(1234, 2,    3,    4)
(1,    1234, 3,    4)
(1,    2,    1234, 4)
(1,    2,    3,    1234)
(1,    12,   13,   14)
(12,   2,    23,   24)
(13,   23,   3,    34)
(14,   24,   34,   4)
(123,  2,    3,    24)
(13,   2,    3,    234)
(134,  23,   3,    4)
(14,   234,  3,    4)
(1,    123,  3,    34)
(1,    12,   3,    134)
(1,    124,  13,   4)
(1,    24,   134,  4)
(1,    2,    123,  14)
(1,    2,    23,   124)
(12,   2,    234,  4)
(124,  2,    34,   4)
```

Table 1: Non-regular triangulations of $\Delta_5 \times \Delta_2$ and $\Delta_3 \times \Delta_3$ from [42, 19]. The rows contain the covectors of the maximal simplices. The jth entry of a tuple contains the coordinate nodes which are adjacent to the apex node j.

For a matrix $A \in \mathbb{R}^{n \times d}$, it was shown in [20, Theorem 1] that the collection of covectors is in bijection with the cells in the regular subdivision of $\Delta_{n-1} \times \Delta_{d-1}$ with height function A. This was generalized in [27] and in [45] to matrices with ∞ entries. For those, the collection of covectors defines a regular subdivision of a subpolytope of $\Delta_{n-1} \times \Delta_{d-1}$, see [45, Corollary 34].

While regular subdivisions are not characterized by purely combinatorial axioms, one can use the defining properties of a polyhedral complex to describe not necessarily regular subdivisions. Hence, we start with a not necessarily regular subdivision of a subpolytope of $\Delta_{n-1} \times \Delta_{d-1}$ and derive a *signed tropical matroid* from this. Note that non-regular triangulations of $\Delta_{n-1} \times \Delta_{d-1}$ exist if and only if $(n-2)(d-2) \geqslant 4$, see [19, Theorem 6.2.19].

4.1 Axiom systems

Let $\mathcal{R}$ be a subdivision of a subpolytope $\mathcal{F}$ of $\Delta_{n-1} \times \Delta_{d-1}$. We identify subpolytopes of $\Delta_{n-1} \times \Delta_{d-1}$ and therefore the cells in $\mathcal{R}$ with subgraphs of the complete bipartite graph $K_{d,n}$ via the identification of the vertex (e_j, e_i) with the edge $(i, j) \in [d] \times [n]$. In this spirit, we define $\text{conv}(G) = \text{conv}\left\{(e_j, e_i) \mid (i, j) \in G\right\}$ for each subgraph G of $K_{d,n}$. Since all these graphs share the same node set $[d] \sqcup [n]$, we will often even identify them
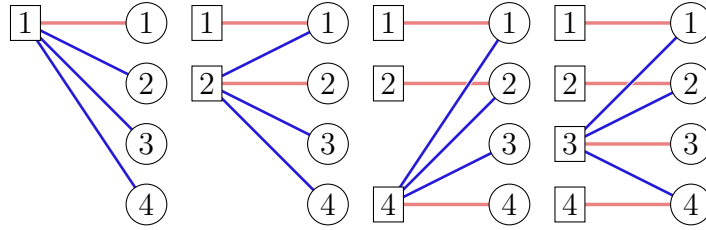
Figure 13: A sequence of basic covector graphs produced by a run of Algorithm 5, see Example 46.

with their set of edges.

Let Σ be a sign matrix $(\sigma_{ji}) \in \{+, -, \bullet\}^{n \times d}$ for which $\sigma_{ji} = \bullet$ if and only if $(i, j) \notin \mathcal{F}$. Moreover, let $\mathcal{S}$ be the set of bipartite graphs without isolated nodes in $[n]$, which correspond to cells in $\mathcal{R}$.

We summarize the required properties which mostly are just adaptions of the definition of a polyhedral subdivision, see [19, Definition 2.3.1].

Definition 47. A signed tropical matroid (STM) is a pair $(\mathcal{S}, \Sigma)$ where $\mathcal{S}$ is a set of subgraphs of $K_{d,n}$ and $(\sigma_{ji}) = \Sigma$ is a matrix in $\{+, -, \bullet\}^{n \times d}$. It has an associated *finity graph* $\mathcal{F} = \bigcup_{G \in \mathcal{S}} G$, which represents the union over all the edges occurring in the graphs in $\mathcal{S}$. Additionally, Σ fulfills $\sigma_{ji} = \bullet \Leftrightarrow (i, j) \notin \mathcal{F}$. We require:

1. No graph in $\mathcal{S}$ has an isolated node in $[n]$.

2. If H is contained in $\mathcal{S}$ then so are all the subgraphs G of H that do not have an isolated node in $[n]$ and for which $\operatorname{conv}(G)$ is a face of $\operatorname{conv}(H)$.

3. For each point $x \in \operatorname{conv}(\mathcal{F})$ there is an $H \in \mathcal{S}$ such that $x \in \operatorname{conv}(H)$.

4. For all H and G in $\mathcal{S}$ with $H \neq G$, the intersection $\operatorname{conv}(H) \cap \operatorname{conv}(G)$ is a face of $\operatorname{conv}(H)$ and $\operatorname{conv}(G)$ or empty.

To emphasize the dependence on n and d we also say that $(\mathcal{S}, \Sigma)$ is a signed tropical (n, d)-matroid. We will often identify $\mathcal{S}$ with the subdivision corresponding to the set of bipartite graphs. The bipartite graphs are the *covector graphs* or just *covectors* in analogy with classical oriented matroids. An STM is *realizable* if it is induced by a matrix A, which means that the covector graphs are generalized covector graphs in the sense of Definition 5 or, equivalently, that the polyhedral subdivision corresponding to $\mathcal{S}$ is regular. In this case, we will also use the notation $\mathcal{S}(A)$. Note that the collection of generalized covectors graphs in the realizable case fulfills all the properties which are listed in the last definition.

As in the realizable case, we consider the entries of Σ as signs on the edges; we call an edge with $+$ a *positive edge* and with $-$ a *negative edge*. *Apex nodes* are the nodes in $[n]$ and *coordinate nodes* are those in $[d]$.

Remark 48. For each apex node $j \in [n]$, the set of covector graphs, in which j is only incident with negative edges, and the set of covector graphs, in which j is only incident with positive edges, form complementary *pseudohalfspaces* in the sense of [42, Definition 5.5.8].

Example 49. The three full-dimensional simplices in the regular subdivision of $\Delta_1 \times \Delta_2$ in Figure 14 correspond to the three trees on $[2] \sqcup [3]$ with edge sets

$$\{(1,1),(1,2),(1,3),(2,3)\}, \{(1,1),(1,2),(2,2),(2,3)\}, \{(1,1),(2,1),(2,2),(2,3)\} \ .$$

The vertex of $\Delta_1 \times \Delta_2$ with label $(2,1)$ is hidden in the figure.

On the other hand, Figure 15 depicts a non-regular mixed subdivision of $4 \cdot \Delta_3$. By the Cayley trick ([61, §1]), triangulations of $\Delta_{n-1} \times \Delta_{d-1}$ are in bijection with fine mixed subdivisions of $n\Delta_{d-1}$. In particular, the full-dimensional cells in the subdivision in Figure 15 are in bijection with the full-dimensional cells in a subdivision of $\Delta_3 \times \Delta_3$ and furthermore, the trees in an STM on $[4] \sqcup [4]$.
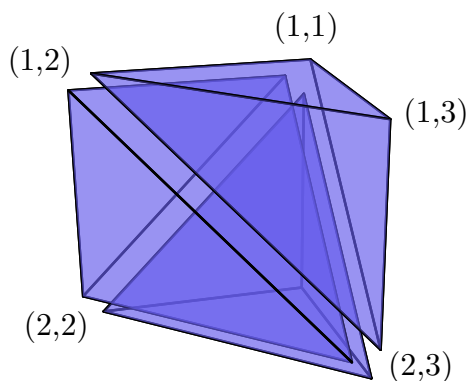

Figure 14: A regular subdivision of $\Delta_1 \times \Delta_2$. The vertices are labeled by the corresponding edges in $K_{3,2}$. This picture was created with `polymake` [32].

Definition 50. An STM $(\mathcal{S}, \Sigma)$ is *full* if the finity graph is $K_{d,n}$. In this case, Σ contains only $-$ and $+$. For the realizable case, the definition means that all the entries of the coefficient matrix are finite. The STM is *generic* if the subdivision is a triangulation or equivalently by [19, Lemma 6.2.8], all the graphs are forests.

In Section 5, we describe a way to modify a given signed tropical matroid $(\mathcal{S}, \Sigma)$ to obtain a generic full signed tropical matroid $(\mathcal{T}, \Xi)$ which is trimmed in the sense of Assumption 1.

In the generic full case, we have a particularly nice characterization of the bipartite graphs which are trees and correspond to the maximal cells in the subdivision. Then Definition 47 boils down to the conditions of [7, Proposition 7.2]. This was used for introducing a GFSTM in Definition 12 and justifies the terminology.
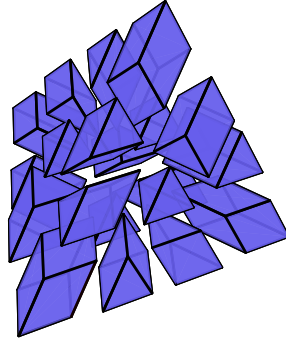
Figure 15: A non-regular subdivision of $\Delta_3 \times \Delta_3$. It is visualized as a non-regular mixed subdivision of $4\Delta_3$. This picture was created with the `polymake` extension `tropmat` by Silke Horn [41].

4.2 Matroid operations and feasibility

The following operations are useful for inductive arguments and yield the polyhedral methods to examine the boundary strata of the tropical projective space.

Analogously to classical oriented matroids one can define a tropical variant of the operations *deletion* and *contraction*, like in [8]. In the following, let $(\mathcal{S}, \Sigma)$ be a signed tropical (n, d)-matroid

For an apex node $j \in [n]$, the *deletion* $\mathcal{S}_{\setminus j}$ is the set of graphs which arise from the graphs of $\mathcal{S}$ by deleting the node j and the incident edges. These graphs describe the cells on the face $\{e_\ell \mid \ell \in [n] \setminus j\} \times \Delta_{d-1}$ of $\Delta_{n-1} \times \Delta_{d-1}$. We delete the jth row in the sign matrix. If $(\mathcal{S}, \Sigma)$ is induced by a signed system (A, Σ) then the operation corresponds to deleting the jth row of A.

For a coordinate node $i \in [d]$, the *contraction* $\mathcal{S}_{/i}$ is the set of graphs which arise from those graphs of $\mathcal{S}$ for which i is isolated by deleting the node i. These graphs describe the cells on the face $\Delta_{n-1} \times \{e_\ell \mid \ell \in [d] \setminus i\}$ of $\Delta_{n-1} \times \Delta_{d-1}$. We delete the ith column in the sign matrix. If $(\mathcal{S}, \Sigma)$ is induced by a signed system (A, Σ) then the operation corresponds to deleting the ith column of A.

By construction, a deletion and a contraction of an STM is again an STM.

Remark 51. Note that the formerly described operations are also related to classical matroid operations since products of simplices are matroid polytopes in the classical sense; see [33]. However, there is no direct translation and one should be careful not to confuse the tropical with the classical operation.

For the contraction $\mathcal{S}_{/([d] \setminus D)}$, where $\mathcal{S}$ is defined on $[d]$ and $D \neq \varnothing$, we will also write $\mathcal{S}|_D$. In the realizable case, these are the covectors of the points with support D. We only consider points in $\mathbb{TA}^d = \mathbb{T}^d_{\min} \setminus \{(\infty, \dots, \infty)\}$ which corresponds to $D \neq \varnothing$.

Lemma 52. *For the finite matrix $A \in \mathbb{R}^{n \times d}$, the covector graphs in the contraction $\mathcal{S}(A)|_D$ for any non-empty $D \subseteq [d]$ are exactly the generalized covectors of the points with support D.*

Proof. Fix a point $x \in \mathbb{T}\mathbb{A}^d$ with support $D \subseteq [d]$ and let

$$\omega > 2 \cdot \max\left(\max\left\{x_\ell \mid \ell \in \operatorname{supp}(x)\right\}, \max\left\{|a_{ji}| \mid (i,j) \in [d] \times [n]\right\}\right) \ .$$

Then the generalized covector graph of x is the same as the proper covector graph of the point $z \in \mathbb{R}^d$ with

$$z_i = \begin{cases} x_i & \text{for } i \in \operatorname{supp}(x) \\ \omega & \text{else} \end{cases} \ .$$

The other inclusion follows by setting the coordinates of isolated coordinate nodes to ∞. $\square$

Recall that a signed system (A, Σ), with coefficient matrix $A \in \mathbb{T}_{\min}^{n \times d}$, is feasible if and only if there is a point $x \in \mathbb{T}\mathbb{A}^d$ which fulfills the corresponding homogeneous tropical inequality system. Otherwise, we call it infeasible.

With Lemma 52, this translates to the following for systems with finite coefficients.

Corollary 53. *A signed system (A, Σ), with finite coefficients $A \in \mathbb{R}^{n \times d}$, is infeasible if and only if every covector graph in every contraction is infeasible.*

This motivates the definition of the feasibility of a full STM as generalization of the feasibility of a tropical linear inequality system. A full STM $(\mathcal{T}, \Sigma)$ is *feasible* if there is a contraction which contains a feasible covector graph, otherwise we call it *infeasible*.

We do not give the definition of feasibility for a general non-full STM, as a more axiomatic approach for collections of generalized covectors would be necessary. Our suggestion is the following: An STM is feasible if there is an extension that is feasible. For this, it would be nice to show that this is indeed the case if and only if all extensions are feasible.

5 Polyhedral constructions

In the last section, we introduced the concept of signed tropical matroids as a natural generalization of tropical linear inequality systems. To make the algorithm developed in Section 3 applicable to all signed tropical matroids, we provide the three concepts *refinement*, *extension* and *splitting*. As infinite coordinates and genericity often impose additional obstacles in tropical geometry, our techniques resemble those used in [4] and [43]

5.1 Refinement

The graphs in an STM $(\mathcal{S}, \Sigma)$ have a particularly simple form if $\mathcal{S}$ is a triangulation. Recall from Definition 50 that, in this case, we call the STM *generic* and [19, Lemma 6.2.8] tells us that all the graphs are forests and, especially, that the maximal polytopes in the subdivision are represented by trees. A method to construct a generic STM is by *refining* our subdivision $\mathcal{S}$. This means that we construct a triangulation $\mathcal{T}$ such that each polytope in $\mathcal{S}$ is the union of simplices in $\mathcal{T}$. Hence, every covector graph of $\mathcal{T}$ is a

forest and contained in a covector graph of $\mathcal{S}$. This idea is implicitly used in [4] in the perturbation of tropical linear inequality systems.

Since we want to preserve the feasibility of our system, we choose to refine our subdivision with heights defining a *lexicographic triangulation*. By [19, Definition 4.3.8], the lexicographic triangulation for a point configuration with $k \in \mathbb{N}$ points is the regular triangulation with heights $\psi_i \cdot c^i$ for $i \in [k]$ where $(\psi_1, \ldots, \psi_k) \in \{-, +\}^k$ is a sign vector and c is a sufficiently big positive number.

Now, let the matrix $(m_{ji}) = M \in \mathbb{R}^{n \times d}$ contain the heights for a lexicographic triangulation of $\Delta_{n-1} \times \Delta_{d-1}$ for which we only require that the sign pattern of M is the negative of the sign pattern of Σ and that $m_{ji} = \infty \Leftrightarrow \sigma_{ji} = \bullet$.

By [19, Lemma 2.3.16 & Corollary 2.3.18], we obtain a refinement of $\mathcal{S}$ with respect to M by taking the union of the subdivisions arising by restricting M to the cells of $\mathcal{S}$. Formally this means: Restricting M to the vertices of a cell C in $\mathcal{S}$ induces a regular subdivision of C which we denote by $C|_M$. The union $\bigcup_{C \in \mathcal{S}} C|_M$ of the simplices in each triangulation $C|_M$ is a triangulation of $\mathcal{F}$ which refines $\mathcal{S}$.

In the realizable case, [19, Lemma 2.3.16] implies that the height matrix corresponding to the refined subdivision is obtained by adding a small multiple of the perturbation matrix M.

The refinement $\mathcal{T}$ of the subdivision $\mathcal{S}$ with the matrix M fulfills the following:

Lemma 54. *Let G be a maximal covector graph of $\mathcal{S}$ and $G_1, \ldots, G_k$ the maximal covector graphs of $\mathcal{T}$ contained in G. Then G is infeasible if and only if G_ℓ is infeasible for every $\ell \in [k]$.*

Proof. If G is infeasible, there is an apex node which is only incident with negative edges. Since each G_ℓ is a connected subgraph of G without isolated nodes it also contains an apex node which is only incident to negative edges. Hence, it is infeasible.

Now, let G be feasible. For the covector graph G we define the matrix $M|_G$ by replacing every entry m_{ji} of M by ∞ for which (i, j) is not an edge of G. By construction, the polytope in the subdivision $\mathcal{S}$ corresponding to the covector graph G is split up in those polytopes whose corresponding graphs occur as maximal covector graphs in the covector decomposition with respect to $M|_G$. Since no apex node in G is only incident with negative arcs, the signed system $(M|_G, \Sigma)$ has the feasible point $\mathbf{0}$ by the choice of M. Then the maximal covector graphs which contain the covector of $\mathbf{0}$ are feasible. This implies the existence of a feasible covector. $\square$
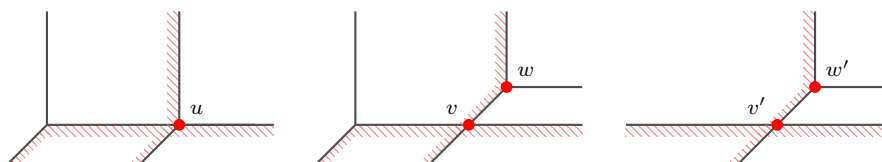

Figure 16: The perturbation of the signed system for the left picture yields the middle one which locally looks like the right one. See Example 55.

Example 55. Consider the signed system (A, Σ) with

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} + & + & - \\ + & - & + \end{pmatrix}.$$

For a sufficiently big $c \gg 1$ we construct the matrix

$$M = \begin{pmatrix} -c^1 & -c^2 & c^3 \\ -c^4 & c^5 & -c^6 \end{pmatrix}$$

with the negative of the sign pattern of Σ. For the covector graph G of the point $(0, 2, 0)$ on the left of Figure 17 this yields (with $M|_G$ as in the proof of Lemma 54)

$$A + \varepsilon \cdot M = \begin{pmatrix} -\varepsilon c^1 & -\varepsilon c^2 & \varepsilon c^3 \\ -\varepsilon c^4 & -2 + \varepsilon c^5 & -\varepsilon c^6 \end{pmatrix} \quad \text{and} \quad M|_G = \begin{pmatrix} -c^1 & \infty & c^3 \\ -c^4 & c^5 & -c^6 \end{pmatrix},$$

where $\varepsilon > 0$ is sufficiently small. Figure 16 shows the original configuration for A, the perturbed configuration for $A + \varepsilon \cdot M$ and the local configuration for $M|_G$. The points are $u = (0, 2, 0)$, $v = (\varepsilon(c^3 + c^5), 2 - \varepsilon(c^1 + c^6), \varepsilon(-c^1 + c^5))$, $w = (\varepsilon c^4, 2 - \varepsilon c^5, \varepsilon c^6)$, $v' = (c^3 + c^5, -c^1 - c^6, -c^1 + c^5)$ and $w' = (c^4, -c^5, c^6)$. Figure 17 depicts their covector graphs. The left one is the covector graph of u, the middle one of v and v', the right one of w and w'.
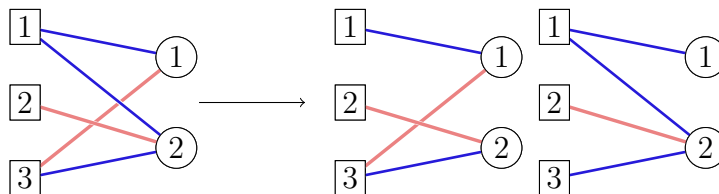

Figure 17: The covector graph is replaced by two trees in the refinement.

We also apply the perturbation technique to get a description of a Cramer solution in the non-generic case.

Lemma 56. *For a finite matrix $A \in \mathbb{R}^{n \times d}$, which is not necessarily generic, the Cramer covector $\mathcal{C}_A(J, I)$ is the union of all minimal matchings on $(I \setminus \{i\}) \sqcup J$ for all $i \in I$ and on $I \sqcup (J \cup \{j\})$ for all $j \in [n] \setminus J$.*

Proof. Let $\widehat{A}$ be any matrix which induces a triangulation that refines the subdivision induced by A in the sense of [19, Lemma 2.3.16]. Then there is a covector graph H with respect to A, which contains $G = \mathcal{C}_{\widehat{A}}(J, I)$.

By Proposition 4, each matching in H is a minimal matching. Since H contains matchings on $(I \setminus \{i\}) \sqcup J$ for all $i \in I$ and $I \sqcup (J \cup \{j\})$ for all $j \in [n] \setminus J$, it contains all minimal matchings on these vertex sets by the same Proposition. Therefore, we have to show that $H = \mathcal{C}_A(J, I)$.

Since G is connected, so is H, and we can apply Lemma 9 to construct a point $x \in \mathbb{R}^d$ which has H as covector graph with respect to A.

Fix a coordinate node $\delta \in I$. For any $i \in I \setminus \{\delta\}$, the path from δ to i is the symmetric sum of the perfect matchings in G on $(I \setminus \{\delta\}) \sqcup J$ and $(I \setminus \{i\}) \sqcup J$. With Lemma 9, we obtain that $x_i - x_\delta$ is the difference of the values of the two matchings. As these are minimal matchings, the values equal the determinants. This implies $x_i - x_\delta = \operatorname{tdet}(A_{J,(I \setminus \{i\})}) - \operatorname{tdet}(A_{J,(I \setminus \{\delta\})})$. As x is defined by its covector graph only up to addition of multiples of $\mathbf{1}$, the claim follows. $\qquad\square$

5.2 Extension from a subpolytope to $\Delta_{n-1} \times \Delta_{d-1}$

We introduce a construction which allows us to reduce the general case, where the finity graph is a subgraph of $K_{d,n}$, to the complete bipartite graph. This is particularly important as we define the algorithms in Section 3 only for a full STM. We give the justification for why we do not lose generality, and provide technical details for later reductions. We achieve this again by polyhedral means. The following technique was also applied to tropical oriented matroids in [43].

Let $\mathcal{F}$ be a subpolytope of $\Delta_{n-1} \times \Delta_{d-1}$ and $\mathcal{S}$ a subdivision of $\mathcal{F}$. An *extension* of $\mathcal{S}$ is a subdivision $\mathcal{T}$ of $\Delta_{n-1} \times \Delta_{d-1}$ which coincides with $\mathcal{S}$ on $\mathcal{F}$.

Placing triangulations provide a tool to construct an extension of a subdivision, see [19, Lemma 4.3.2]. In particular, for each subdivision of a subpolytope of $\Delta_{n-1} \times \Delta_{d-1}$ there is always an extension. To resolve the $\bullet$ entries of the sign matrix, we just replace them by $+$. We denote the modified sign matrix by Ξ. Note that the (in)feasibility of the covector graphs in $\mathcal{S}$ is preserved in $\mathcal{T}$.

We summarize these considerations.

Proposition 57. *The set of covectors in the STM defined by $(\mathcal{S}, \Sigma)$ is contained in the set of covectors defined by $(\mathcal{T}, \Xi)$.*

We study in more detail how an extension can be produced in the realizable case.

[19, Lemma 4.3.4] shows that a placing triangulation can be obtained by taking a rapidly increasing height function. Namely, if there are $k < n \cdot d$ entries with ∞ in $A \in \mathbb{T}_{\min}^{n \times d}$, let $\Omega = (\Omega_1, \ldots, \Omega_k)$ be a vector of "big" numbers. We require that

$$\Omega_1 > \sum_{a_{ji} \neq \infty} |a_{ji}| \quad \text{and} \quad \Omega_{\ell+1} > \sum_{a_{ji} \neq \infty} |a_{ji}| + \sum_{h=1}^{\ell} \Omega_h \quad \text{for all } \ell \in [k-1] \ . \tag{3}$$

We will calculate with the entries of Ω just formally and denote the resulting matrix by $A(\Omega)$.

Remark 58. One can think of these Ω_ℓ as artificial infinities. One approach to formalize this is by successively adjoining elements to $\mathbb{R}$. Here, the order extends the natural order on $\mathbb{R}$ such that Ω_ℓ is the biggest element in each extension step. In [4, §3.2], a similar technique with "infinitely small" values is used to reduce the case with $-\infty$ to the finite case.

To show that the matrix $A(\Omega)$ induces an extension of the subdivision of $\mathcal{F}$ by A, we iteratively replace the ∞ entries by the entries of Ω. Let A^1 be obtained from A by replacing one ∞ entry, which belongs to the edge e, with a positive number Ω_1 which is bigger than the sum of the absolute values of the finite entries of A. Consider an arbitrary maximal covector graph G with respect to A and let μ be a perfect matching on $D \sqcup N \subseteq [d] \sqcup [n]$ in G. By Proposition 4, the matching μ is minimal with respect to the coefficients of A^1. Hence, by definition of Ω_1, the edge e cannot be contained in μ. Since this is true for any matching in G, again by Proposition 4, the graph G is also a covector with respect to A^1. By iteratively inserting $\Omega_1, \Omega_2, \ldots, \Omega_k$ for the ∞ entries, this implies that the subdivision induced by $A(\Omega)$ extends the subdivision induced by A, since a polyhedral complex is given by its maximal cells. Furthermore, if A induces a triangulation, so does $A(\Omega)$.

We say that the signed system $(A(\Omega), \Xi)$ *extends* the signed system (A, Σ).

Lemma 59. *For the matrix $A \in \mathbb{T}^{n \times d}_{\min}$, let $(A(\Omega), \Xi)$ be an extension of the signed system (A, Σ). For any $x \in \mathbb{TA}^d$, the generalized covector graph $G_A(x)$ is infeasible, if the generalized covector graph $G_{A(\Omega)}(x)$ is infeasible.*

Proof. Within the proof, we denote $A(\Omega)$ by $(\widetilde{a_{ji}}) = \widetilde{A}$. Fix an arbitrary $x \in \mathbb{TA}^d$. If the generalized covector graph $G_{\widetilde{A}}(x)$ is infeasible, there is a $j_0 \in [n]$, which is only incident with negative edges in $G_{\widetilde{A}}(x)$. Let I be the set of coordinate nodes adjacent to j_0. Since the entries of $\widetilde{A}$ are finite, $G_{\widetilde{A}}(x)$ is a proper covector graph on the support of x. Hence, using the definition of the covector graph, we see that x fulfills the inequalities

$$\widetilde{a_{j_0 i}} + x_i < \widetilde{a_{j_0 \ell}} + x_\ell \qquad \text{for all } i \in I \text{ and } \ell \in \operatorname{supp}(x) \setminus I .$$

Each entry $\widetilde{a_{j_0 i}}$ with $i \in I$ equals $a_{j_0 i} \neq \infty$ because (j_0, i) is negative. With $\widetilde{a_{j_0 \ell}} \leqslant a_{j_0 \ell}$ for $\ell \in \operatorname{supp}(x) \setminus I$, we obtain

$$a_{j_0 i} + x_i < a_{j_0 \ell} + x_\ell \qquad \text{for all } \ell \in \operatorname{supp}(x) \setminus \{i\} .$$
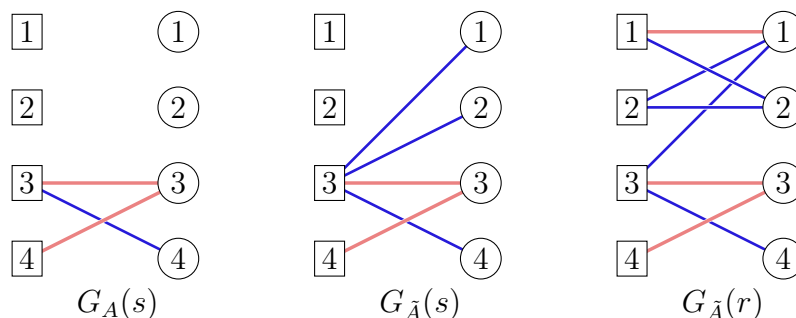
This implies that $G_A(x)$ is infeasible. $\qquad\square$

Figure 18: Three covector graphs for Example 60.

Example 60. Consider the signed systems (A, Σ) and $(\widetilde{A}, \Xi)$ with

$$A = \begin{pmatrix} 0 & 0 & \infty & \infty \\ 1 & 1 & \infty & \infty \\ \infty & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \Sigma = \begin{pmatrix} - & + & \bullet & \bullet \\ + & - & \bullet & \bullet \\ \bullet & + & - & - \\ + & - & + & + \end{pmatrix}$$

$$\widetilde{A} = \begin{pmatrix} 0 & 0 & \Omega_1 & \Omega_2 \\ 1 & 1 & \Omega_3 & \Omega_4 \\ \Omega_5 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \Xi = \begin{pmatrix} - & + & + & + \\ + & - & + & + \\ + & + & - & - \\ + & - & + & + \end{pmatrix}.$$

They yield the Cramer solutions $s = \mathcal{C}_A([3], [4]) = (\infty, \infty, 1, 1)$ and $r = \mathcal{C}_{\widetilde{A}}([3], [4]) = (\Omega_1 + 1, \Omega_1 + 1, 1, 1)$. The corresponding covector graphs are left and right in Figure 18. The relation between the left and middle covector illustrates Lemma 59.

5.3 Splitting apex nodes

To apply the algorithms that will be presented in Section 3 and 6 to an STM (A, Σ), we require that each row of Σ contains at most one negative entry. We call this property *trimmed*.

In the realizable case, this can be obtained very easily. Through the conversion

$$c_0 \leqslant \bigoplus_{\ell \in [m]} c_\ell \quad \Leftrightarrow \quad \left(c_0 \leqslant c_\ell \quad \forall \ell \in [m] \right) , \tag{4}$$

for arbitrary $c_0, c_1, \ldots, c_m \in \mathbb{T}_{\min}$ each tropical inequality system is equivalent to a system for which the minimum on the bigger side of the new inequalities contains only one term. Here, the number of inequalities is increased by a factor which is at most the number of coordinates, see Figure 19.
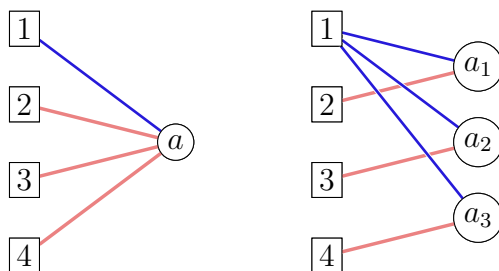

Figure 19: An apex node whose corresponding row in the sign matrix has three negative entries is replaced by three apex nodes.

This splitting of apex nodes was similarly used in [52, §7.4].

For the non-realizable case, we use a more complicated polyhedral construction, which uses local changes. In two steps, we obtain a bigger STM which mimics a splitting of the

inequalities in its covector graphs. A similar technique was used in [43, §7.2]. We know how to extend a non-full STM, by Subsection 5.2, and can assume that the STM is full.

Let $k > 1$ entries of the nth row of Σ be $-$.

Define the projection $\pi \colon \mathbb{R}^{n-1+k} \times \mathbb{R}^d \to \mathbb{R}^n \times \mathbb{R}^d$ as

$$(y_1, \ldots, y_{n-1}, y_n, \ldots, y_{n+k-1}, z_1, \ldots, z_d) \mapsto \left(y_1, \ldots, y_{n-1}, \sum_{\ell=0}^{k-1} y_{n+\ell}, z_1, \ldots, z_d \right) \ .$$

This defines a surjective mapping from $\Delta_{n-1+k-1} \times \Delta_{d-1}$ onto $\Delta_{n-1} \times \Delta_{d-1}$ and furthermore, a surjective mapping from the subgraphs of $K_{d,n+k-1}$ to the subgraphs of $K_{d,n}$.

Lemma 61. *The preimage under π of a simplex in $\Delta_{n-1} \times \Delta_{d-1}$, given by the bipartite graph G, is $G \cup \{ (i, n + \ell) \mid (i, n) \in G, \ell \in [k-1] \}$.*

Proof. Let H be any spanning subgraph of $K_{d,n+k-1}$. This defines a subpolytope of $\Delta_{n-1+k-1} \times \Delta_{d-1}$. A convex combination of its vertices is given by $\sum_{(i,j) \in H} \lambda_{i,j}(e_j, e_i)$ with $\sum_{(i,j) \in H} \lambda_{ij} = 1$. With the linearity of π, the projection of this point is

$$\sum_{(i,j) \in H, j \leqslant n-1} \lambda_{i,j} \pi((e_j, e_i)) + \sum_{(i,j) \in H, j \geqslant n} \lambda_{i,j} \pi((e_j, e_i))$$

which evaluates to

$$\sum_{(i,j) \in H, j \leqslant n-1} \lambda_{i,j}(e_j, e_i) + \sum_{(i,j) \in H, j \geqslant n} \lambda_{i,j}(e_n, e_i) \ .$$

Such a point lies in conv $\{ (e_j, e_i) \mid (i,j) \in G \}$ if and only if, for $\lambda_{ij} \neq 0$,

$$(i,j) \in H \Leftrightarrow \begin{cases} (i,j) \in G & \text{for } j \leqslant n-1 \\ (i,n) \in G & \text{for } j > n-1 \ . \end{cases}$$

With the linearity of π, the claim follows. $\qquad\square$

Fix an arbitrary $\varepsilon > 0$ and let $i_1, \ldots, i_k$ be the indices where the nth row of Σ is '$-$'. We define the matrix $(m_{ji}) = M \in \mathbb{R}^{(n+k-1) \times d}$ by

$$m_{ji} = \begin{cases} \varepsilon & \text{for } j \geqslant n, i = i_j \\ 0 & \text{else} \ . \end{cases}$$

We refine the subdivision of $\Delta_{n-1+k-1} \times \Delta_{d-1}$, which we just constructed, with this matrix M to obtain a subdivision $\widehat{\mathcal{S}}$.

Additionally, we replace the nth row of Σ with k copies of this row, where we replace all the $-$ entries in every row j for $j > n-1$ by $+$ except for $(j, i_{j-(n-1)})$, where we keep the $-$.

Finally, the following is similar to Lemma 54 and justifies the construction. Let $(\mathcal{S}, \Sigma)$ be the original and $(\widehat{\mathcal{S}}, \widehat{\Sigma})$ the modified STM.

Proposition 62. *Let G be a maximal covector graph of $\mathcal{S}$ and $G_1, \ldots, G_m$ the maximal covector graphs of $\widehat{\mathcal{S}}$ which is mapped to G by π. Then G is infeasible if and only if G_ℓ is infeasible for every $\ell \in [k]$.*

Proof. Let $\widehat{G}$ be the covector graph from Lemma 61 which is obtained by adding k copies of the apex node n. We define the matrix $M|_{\widehat{G}}$ by replacing every entry m_{ji} of M by ∞ for which (i, j) is not an edge of $\widehat{G}$.

By construction, $G_1, \ldots, G_m$ are exactly the maximal covector graphs with respect to $M|_{\widehat{G}}$.

Since feasibility is a property which can be checked independently for all apex nodes, it suffices to consider the apex node n in G resp. $n, \ldots, n + k - 1$ in $G_1, \ldots, G_m$.

Hence, the rows $n, \ldots, n + k - 1$ of $M|_{\widehat{G}}$ are, up to reordering of columns, of the form

$$
\begin{pmatrix}
0 & \varepsilon & \cdots & \varepsilon & 0 & \cdots & 0 \\
\varepsilon & \ddots & & \varepsilon & \vdots & 0 & \cdots & 0 \\
\vdots & \varepsilon & \ddots & \varepsilon & 0 & \cdots & 0 \\
\varepsilon & \cdots & \varepsilon & 0 & 0 & \cdots & 0
\end{pmatrix}
$$

where each 0 entry in the left part of the matrix is assigned a $-$ in $\widehat{\Sigma}$.

If G is infeasible, the right part of the matrix does not contain any columns and the corresponding inequality system is infeasible.

Otherwise, 0 is a feasible point. Therefore, at least one of the covectors $G_1, \ldots, G_m$ is feasible. $\square$

In this way, we can construct a signed tropical matroid $(\widehat{\mathcal{S}}, \widehat{\Sigma})$ such that the number of apex nodes is bounded by $n \cdot d$ and every row of $\widehat{\Sigma}$ contains exactly one negative entry.

In the realizable case, this translates to the following.

Corollary 63. *Let $I \subseteq \mathbb{N}$ be a finite index set, $b_0, b_i \in \mathbb{T}_{\min}$ for $i \in I$ and $\varepsilon > 0$ an arbitrary positive number. Then $b_0 \leqslant \bigoplus_{i \in I} b_i$ if and only if $b_0 \oplus \bigoplus_{i \in I \setminus \{\ell\}} (b_i + \varepsilon) \leqslant b_\ell$ for all $\ell \in I$.*

Example 64. The left picture of Figure 20 visualizes the inequality $x_1 \leqslant x_2 \oplus x_3$ where again the infeasible region is marked. The middle one depicts the replacement by the two inequalities $x_1 \oplus (\varepsilon \odot x_2) \leqslant x_3$ and $x_1 \oplus (\varepsilon \odot x_3) \leqslant x_2$ as in Corollary 63. Finally, the right one illustrates the conversion from Equation 4. The resulting inequalities are $x_1 \leqslant x_2$ and $x_1 \leqslant x_3$.

6   Feasibility of signed systems

We developed an algorithm to examine if a signed tropical matroid contains a feasible covector. The clearest form of the algorithm is given in Section 3 for a trimmed GFSTM.
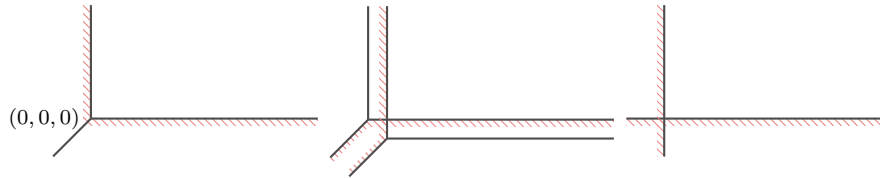
Figure 20: Starting from the left depiction, the middle one illustrates the construction of Corollary 63 and the right one illustrates Equation 4 applied to the left configuration, see Example 64.

We saw in Section 5 that each STM can be transformed into a trimmed GFSTM with the same feasibility status.

In this section, we demonstrate that the modifications are not necessary under the assumption of realizability. Furthermore, we derive a stronger upper bound on the runtime of the algorithm. As the algorithm terminates with an arbitrary solution, not necessarily the one with maximal support, we discuss how this can be determined in Appendix C. There, we also relate the outcome of the algorithm to the associated mean payoff game as explained in Appendix A.3.

In this section we assume that (A, Σ) is a trimmed signed system. We can always transform a general signed system to a trimmed one with Equation 4. Note that this is not a restriction on the corresponding inequality system but merely a requirement on the representation.

6.1   Solving general signed systems

The part in Algorithm 5, where the data of the STM is invoked, is the computation of a Cramer covector. For a general non-full STM, the Cramer covectors can be quite degenerated as one can see in Figure 18. However, Proposition 70 will ensure that it carries all the necessary information. While we invoked an oracle based on the existence of a Cramer covector following from Proposition 19 in the non-realizable case, we can explicitly determine a Cramer covector as the covector graph of a Cramer solution (2) in the realizable case.

Remark 65. For the realizable case, it is interesting to know the complexity of the computation of the Cramer covectors. The Cramer solution can be computed in $\mathcal{O}(d^3)$ by [3, Remark 8.2]. One derives the covector by evaluating the minimum in each row which needs $\mathcal{O}(dn)$ steps. Note that not all the edges of the covector graph are needed and therefore, this computation could be reduced. Subsuming, a Cramer covector $\mathcal{C}_A(N, (D \cup \{\delta\}))$ can be computed in $\mathcal{O}(d^3 + dn)$.

Furthermore, the role of a "totally infeasible" covector is not so clear as Lemma 17 shows the infeasibility implication only under the condition, that the STM is generic and full. However, we will see that this termination criterion can be replaced by a similar condition. Note that we modify the iteration slightly in the spirit of Remark 39. We continue the iteration until no node in D is incident with a negative leaf.

Again, we start with an element $\delta \in [d]$, $D = \varnothing$ and $N = \varnothing$. As long as there is an apex node in $[n] \setminus N$ of degree 1 in y incident to D via a negative edge this apex enters the basis N and the apex of the same shape is removed from N. Note that in the non-generic case there can be non-basic apex nodes of degree $\geqslant 2$. However, since we assume that the STM is trimmed they cannot be incident with more than one negative edge. After this iteration two cases can occur. If the result is already feasible, we terminate and return this feasible point. Otherwise, there is still an apex node of degree 1 incident with a negative edge. By construction, it cannot be adjacent to D and hence it is adjacent to δ. If the Cramer covector is already defined on the whole of $[d]$ this yields a point which certifies infeasibility. If this is not the case, we can add δ to D and obtain a covector graph which is defined on a bigger set of coordinates. Due to infinite entries of A, its coordinates in $D \cup \{\delta\}$ can be infinite, though.

Remark 66. An alternative way to start the algorithm in the realizable case is by adding d inequalities of the form $x_i \geqslant M_i$ for sufficiently small numbers $M_1, \ldots, M_d \in \mathbb{R}^d$. Here, one can pick numbers with are smaller than d times the minimal entries of A. The intersection if these special inequalities also yield a starting basis.

To deduce the correctness of Algorithm 6, we relate the sequence of points in the iteration in the non-generic non-full situation with a run of Algorithm 5.

6.2 Correctness and implications of the algorithm

To show the correctness of Algorithm 6, we reduce it to the correctness for full generic signed systems by exploiting the techniques established in Section 5. For this, fix an arbitrary trimmed signed system (A, Σ) and subsets $J \subseteq [n]$ and $I \subseteq [d]$ with $|J| = |I| - 1$.

Let $(A(\Omega), \Xi)$ be an extension of (A, Σ) in the sense of Subsection 5.2.

Lemma 67. *Each apex node of degree 1 in $\mathcal{C}_A(J, I)$ also has degree 1 in $\mathcal{C}_{A(\Omega)}(J, I)$ and is incident with the same coordinate node.*

Proof. Let (i, j) be an edge in $\mathcal{C}_A(N, D \cup \{\delta\})$ so that j has degree 1. For all $\ell \in I$, the choice of Ω in Equation 3 implies that $\mathrm{tdet}(A(\Omega)_{J, (I \setminus \{\ell\})})$ either equals $\mathrm{tdet}(A_{J, (I \setminus \{\ell\})})$ or it contains an Ω summand and $\mathrm{tdet}(A_{J, (I \setminus \{\ell\})}) = \infty$. The definition of a generalized covector graph yields $a_{ji} < \infty$ and $\mathrm{tdet}(A_{J, (I \setminus \{i\})}) < \infty$. Hence, $a_{ji} + \mathrm{tdet}(A(\Omega)_{J, (I \setminus \{i\})})$ is the minimum in row j and $(i, j) \in \mathcal{C}_{A(\Omega)}(N, D \cup \{\delta\})$. $\square$

Example 68. Lemma 67 is illustrated in Figure 18. The covectors on the left and on the right both contain the edge $(3, 4)$.

Let $(\widehat{A(\Omega)}, \Xi)$ by a refinement of the signed system $(A(\Omega), \Xi)$ in the sense of Subsection 5.1.

Lemma 69. *The covector graph $\mathcal{C}_{A(\Omega)}(J, I)$ contains the covector graph $\mathcal{C}_{\widehat{A(\Omega)}}(J, I)$. Furthermore, each apex node of degree 1 in $\mathcal{C}_{A(\Omega)}(J, I)$ also has degree 1 in $\mathcal{C}_{\widehat{A(\Omega)}}(J, I)$ and is incident with the same coordinate node.*

Algorithm 6 Determine the feasibility of a signed system

Input: A signed system (A, Σ) so that each row of Σ contains at most one $-$ entry.

Output: A feasible point or a point which guarantees the infeasibility of the signed system.

1: **procedure** FINDWITNESS$((A, \Sigma))$
2:     $\delta \leftarrow$ an element of $[d]$
3:     $D \leftarrow \varnothing$, $N \leftarrow \varnothing$
4:     $y \leftarrow \mathcal{C}_A(N, (D \cup \{\delta\}))$
5:     **while** TRUE **do**
6:         **while** there is a non-basic apex node of degree 1 in y incident to D via a negative edge **do**
7:             $r \leftarrow$ the apex fulfilling the while-condition
8:             $p \leftarrow$ basic apex of y of the same shape as r $\triangleright$ *p is an element of N.*
9:             $N \leftarrow N \setminus \{p\} \cup \{r\}$
10:             $y \leftarrow \mathcal{C}_A(N, (D \cup \{\delta\}))$
11:         **end while** $\triangleright$ *at this point, δ is the only coordinate node which can be incident with an apex node of degree 1 via a negative edge*
12:         **if** δ is incident with an apex node of degree 1 via a negative edge **then**
13:             **if** $|D| = d - 1$ **then**
14:                 **return** "infeasible", $A[N|(D \cup \{\delta\})]$
15:             **else**
16:                 $j \leftarrow$ non-basic apex of degree 1 incident with δ via a negative edge
17:                 $N \leftarrow N \cup \{j\}$
18:                 $D \leftarrow D \cup \{\delta\}$
19:                 $\delta \leftarrow$ node in $[d] \setminus D$.
20:                 $y \leftarrow \mathcal{C}_A(N, (D \cup \{\delta\}))$
21:             **end if**
22:         **else**
23:             **return** "feasible", $A[N|(D \cup \{\delta\})]$
24:         **end if**
25:     **end while**
26: **end procedure**

Proof. The containment follows from Lemma 56. The fact that $\mathcal{C}_{\widehat{A(\Omega)}}(J, I)$ is a spanning tree implies the second claim. □

Combining these two lemmata yields the desired relation between the covector in the original and the modified signed system.

Proposition 70. *Each apex node of degree 1 in* $\mathcal{C}_A(J, I)$ *also has degree 1 in* $\mathcal{C}_{\widehat{A(\Omega)}}(J, I)$ *and is incident with the same coordinate node.*

Proof. By Lemma 67, the edge is also an edge of $\mathcal{C}_{A(\Omega)}(J, I)$. Furthermore, by Lemma 69, it is an edge of $\mathcal{C}_{\widehat{A(\Omega)}}(J, I)$. □

Now, we gathered the necessary tools to prove termination and correctness.

Theorem 71. *Algorithm 6 computes a covector graph, which certifies the feasibility or infeasibility of the signed system* (A, Σ).

Proof. Fix a $\delta \in [d]$ and a subset $D \subseteq [d] \setminus \{\delta\}$. Assume that $N^1 \subseteq [n]$ is a subset of the apex nodes for which $\mathcal{C}_{\widehat{A(\Omega)}}(N^1, D \cup \{\delta\})$ is a basic covector (which is the case for $D = N = \varnothing$). Let $k \in \mathbb{N}$ so that $N^1, N^2, \ldots, N^k$ is the sequence of the set N in Line 9 for the first k iterations of the while-loop starting in Line 6, beginning with N^1. Then for all $\ell \in [k-1]$ there are $r^\ell, p^\ell \in [n]$ so that $N^{\ell+1} = N^\ell \setminus \{p^\ell\} \cup \{r^\ell\}$.

By the iteration condition of the while-loop in Line 6, the apex node r^ℓ is not in N^ℓ, it is of degree 1, and it is incident with a negative edge (i^ℓ, r^ℓ) in $\mathcal{C}_A(N^\ell, D \cup \{\delta\})$. Proposition 70 implies that r^ℓ also has degree 1 and is incident with (i^ℓ, r^ℓ) in $\mathcal{C}_{\widehat{A(\Omega)}}(N^{\ell+1}, D \cup \{\delta\})$. Now, Corollary 32 implies that $\mathcal{C}_{\widehat{A(\Omega)}}(N^{\ell+1}, D \cup \{\delta\})$ is a basic covector if so is $\mathcal{C}_{\widehat{A(\Omega)}}(N^\ell, D \cup \{\delta\})$, since p^ℓ is chosen just to match the shape of r^ℓ, independent of the covector graph.

Hence by induction, $\mathcal{C}_{\widehat{A(\Omega)}}(N^\ell, D \cup \{\delta\})$ is a basic covector for all $\ell \in [k]$. By Theorem 38 and Remark 39, there is an $h \in \mathbb{N}$ so that in $\mathcal{C}_{\widehat{A(\Omega)}}(N^h, D \cup \{\delta\})$ no non-basic apex node is incident with D via a negative edge. Proposition 70 yields that no non-basic apex node of degree 1 is of degree 1 in $\mathcal{C}_A(N^h, D \cup \{\delta\})$. Therefore, this covector graph is either feasible, which means that we are finished, or δ is incident with an apex node j of degree 1 via a negative edge. In the latter case, again with Proposition 70, this also holds in $\mathcal{C}_{\widehat{A(\Omega)}}(N^h, D \cup \{\delta\})$.

If $D \cup \{\delta\} = [d]$, then $\mathcal{C}_{\widehat{A(\Omega)}}(N^h, D \cup \{\delta\})$ is totally infeasible. By Lemma 17, all covector graphs in all contractions for $(\widehat{A(\Omega)}, \Xi)$ are infeasible. Combining Lemma 54 and Lemma 59 implies the infeasibility of the signed system (A, Σ).

Otherwise, for any $\delta' \in [d] \setminus (D \cup \{\delta\})$, the Cramer covector $\mathcal{C}_{\widehat{A(\Omega)}}(N^h \cup \{j\}, D \cup \{\delta\} \cup \{\delta'\})$ is the basic covector $\mathcal{B}_{\widehat{A(\Omega)}}(N^h \cup \{j\}, D \cup \{\delta\}, \delta')$ and we can continue the iteration of the while-loop in Line 5. The termination is guaranteed as D grows in each iteration of this while-loop. □

Corollary 72. *The following statements are equivalent:*

a) (A, Σ) *is feasible.*

b) $(A(\Omega), \Xi)$ *is feasible.*

c) $(\widehat{A}, \Sigma)$ *is feasible.*

d) $(\widehat{A(\Omega)}, \Xi)$ *is feasible.*

Proof. The equivalence of all statements follows from the equivalence of the feasibility of (A, Σ) and $(\widehat{A(\Omega)}, \Xi)$. Now, if (A, Σ) is feasible, there is a point $x \in \mathbb{TA}^d$ so that $G_A(x)$ is feasible. By Lemma 59, $G_{A(\Omega)}(x)$ is feasible as well. Lemma 54 implies that $(\widehat{A(\Omega)}, \Xi)$ contains a feasible covector. If (A, Σ) is infeasible, Algorithm 6 implicitly computes a totally infeasible covector graph in $(\widehat{A(\Omega)}, \Xi)$. That ensures the infeasibility of $(\widehat{A(\Omega)}, \Xi)$. □

Example 73. Consider the signed system (A, Σ) with

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ \infty & 0 & 0 & \infty \\ \infty & 4 & 2 & \infty \\ 1 & -5 & \infty & 0 \\ 4 & 0 & -7 & 3 \\ 0 & \infty & -9 & \infty \\ 0 & \infty & \infty & 3 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} + & - & + & + \\ \bullet & + & - & \bullet \\ \bullet & - & + & \bullet \\ - & + & \bullet & + \\ + & + & + & - \\ + & \bullet & - & \bullet \\ + & \bullet & \bullet & - \end{pmatrix}.$$

Note that the last two rows are obtained by splitting the inequality $x_1 \leqslant (-9) \odot x_3 \oplus 3 \odot x_4$ into $x_1 \leqslant (-9) \odot x_3$ and $x_1 \leqslant 3 \odot x_4$.

We want to execute Algorithm 6 for (A, Σ) and start with $\delta = 2$. The iterations are shown in the table. We choose $j = 1$ as first entering apex.

δ	Cramer solution	violated inequalities
2	$A[\varnothing\|\{2\}] = (\infty, 0, \infty, \infty)$	$j = 1, 3$
1	$A[\{1\}\|\{1, 2\}] = (1, 0, \infty, \infty)$	$r = 3$
	$A[\{3\}\|\{1, 2\}] = (4, \infty, \infty, \infty)$	$j = 4$
3	$A[\{3, 4\}\|\{1, 2, 3\}] = (-3, 3, 5, \infty)$	$j = 6$
4	$A[\{3, 4, 6\}\|\{1, 2, 3, 4\}] = (-5, 2, 4, -4)$	

The final result $(-5, 2, 4, -4)$ is a feasible point for the signed system.

6.3 Refined analysis of the runtime

For the abstract setting in Section 3, we gave only a rough upper bound on the number of iterations. For the realizable case, we obtain a better bound by combining Lemma 11 with the following Lemma. We show that Algorithm 6 is pseudopolynomial and only depends on the combinatorial structure of a triangulation of $\Delta_{n-1} \times \Delta_{d-1}$.

Lemma 74. *Let $x^\ell \in \mathbb{T}_{\min}^d$ such that y^ℓ is the covector graph of x^ℓ, which can be constructed from A by Lemma 9. For each $\ell \in [k]$, we get the inequalities*

$$x_i^{\ell-1} - x_\delta^{\ell-1} \leqslant x_i^\ell - x_\delta^\ell \qquad \text{for all } i \in (D \cup \{\delta\}) \ .$$

Proof. Lemma 9 allows us to express $x_i^{\ell-1} - x_\delta^{\ell-1}$ resp. $x_i^\ell - x_\delta^\ell$ as a sum along the path from δ to i in $y^{\ell-1}$ resp. y^ℓ, with the weights given by A.

For each i in the connected component $C^{\ell-1}$ of δ in $y^{\ell-1} - e^{\ell-1}$, there is exactly one path from δ to i and it is the same in $y^{\ell-1}$ and y^ℓ. Therefore, we obtain $x_i^{\ell-1} - x_\delta^{\ell-1} \leqslant x_i^\ell - x_\delta^\ell$.

Now, let i be a node in $[d] \setminus C^{\ell-1}$. Then the path from δ to i in $y^{\ell-1}$ contains $e^{\ell-1}$ and the one in y^ℓ contains f^ℓ. Denote the paths by $\rho^{\ell-1}$ and ρ^ℓ. Their symmetric sum is a subgraph of $y^{\ell-1} + f^\ell$ and is a union of cycles. Since $y^{\ell-1}$ is a tree, $y^{\ell-1} + f^\ell$ contains only the elementary cycle formed by f^ℓ. It decomposes into two matchings μ_0 and μ_1 where one of them, without loss of generality μ_0, contains both the edges $e^{\ell-1}$ and f^ℓ by the comparability condition in Definition 12.

In the formula for Lemma 9, odd edges get a positive sign and even edges a negative sign. Furthermore, we see that $(x_i^\ell - x_\delta^\ell) - (x_i^{\ell-1} - x_\delta^{\ell-1})$ is given by the difference of the sums over the two matchings μ_0 and μ_1. By Lemma 28, f^ℓ is odd in y^ℓ. This implies

$$(x_i^\ell - x_\delta^\ell) - (x_i^{\ell-1} - x_\delta^{\ell-1}) = \sum_{(j,i)\in\mu_0} a_{ji} - \sum_{(j,i)\in\mu_1} a_{ji} \ .$$

Finally, Proposition 4 yields that the difference $\sum_{(j,i)\in\mu_0} a_{ji} - \sum_{(j,i)\in\mu_1} a_{ji}$ is positive, since μ_1 is contained in the covector graph y^ℓ and hence minimal. $\qquad\square$

Let $\widetilde{A}$ be any matrix which induces the same triangulation as $\widehat{A(\Omega)}$. Recall the sequence $N^1, N^2, \ldots, N^h$ from the proof of Theorem 71. Then $\mathcal{C}_{\widetilde{A}}(N^1, D \cup \{\delta\}), \ldots,$ $\mathcal{C}_{\widetilde{A}}(N^h, D \cup \{\delta\})$ is a sequence of basic covector graphs. With Corollary 32, we can apply Lemma 74 to the associated points $\widetilde{A}[N^1 | D \cup \{\delta\}], \ldots, \widetilde{A}[N^h | D \cup \{\delta\}]$. Let $z^1, \ldots, z^h$ be the representatives of this sequence modulo $\mathbb{R}\cdot\mathbf{1}$ with $z_\delta^\ell = 0$. In this way, for all $\ell \in [h-1]$ this yields the inequalities

$$z_i^\ell \leqslant z_i^{\ell+1} \qquad \text{for all } i \in (D \cup \{\delta\}) \ ,$$

where at least one inequality is strict for each ℓ.

If $\widetilde{A}$ is an integer matrix, then the points z^ℓ have only integer entries. Hence, for all $\ell \in [h-1]$, the difference $z^{\ell+1} - z^\ell$ is a non-negative integer vector with at least one non-zero entry. We deduce $\sum_{i\in(D\cup\{\delta\})}(z_i^h - z_i^1) \geqslant h$.

Furthermore, defining $\omega = \max\{|\widetilde{a_{ij}}| \mid (i,j) \in [d] \times [n]\}$, Lemma 11 yields the inequality

$$\left| \sum_{i\in(D\cup\{\delta\})} (z_i^h - z_i^1) \right| = \left| \sum_{i\in(D\cup\{\delta\})} (z_i^h - z_\delta^h + z_\delta^h - z_i^1) \right| \leqslant$$

$$\sum_{i\in(D\cup\{\delta\})} |z_i^h - z_\delta^h| + \sum_{i\in(D\cup\{\delta\})} |z_\delta^h - z_i^1| \leqslant \qquad 2 \cdot d \cdot 2 \cdot \omega = 4d\omega \ .$$

We conclude the following.

Theorem 75. *The maximal number of iterations h of the while loop in Line 6 of Algorithm 6 fulfills $h \leqslant 4d\omega$.*

We deduce the following by combining the theorem with Remark 65 and the observation that D is increased by one element at each step of the while-loop in Line 65 of Algorithm 6.

Corollary 76. *Algorithm 6 determines a certificate of feasibility or infeasibility of a tropical linear inequality system in $O(d^3(n + d^2)\omega)$ iterations.*

Note that a similar idea is used to give bounds on the runtime in [10, §5.2] by using [28, Theorem 3.3].

To examine the parameter ω further, recall that each matrix in $\mathbb{R}^{n \times d}$ defines a height function for a regular subdivision of $\Delta_{n-1} \times \Delta_{d-1}$.

The set of all matrices which induce the same regular subdivision defines an open polyhedral cone. The collection of these cones is a complete fan, the *secondary fan*. For an introduction to secondary fans see [19, §5].

Since the secondary fan is the normal fan of the secondary polytope, see [34, §7] or [19, §5], which is a rational polytope for $\Delta_{n-1} \times \Delta_{d-1}$, every cone contains a rational and, hence, an integer vector.

Inspired by Theorem 75, we leave it as future work to give bounds on the minimal integer vectors in the cones of the secondary fan of $\Delta_{n-1} \times \Delta_{d-1}$. This might reveal either a good upper bound on the runtime of Algorithm 6 or special classes of instances which are particularly hard. Furthermore, it is interesting to consider the cones in the secondary fan which contain the weight functions describing parity games, see Subsection A.3.

6.4 Further questions

In the last section, we came up with an upper bound for the number of iterations in terms of integer vectors in the secondary fan of $\Delta_{n-1} \times \Delta_{d-1}$. This raises the question to determine lower or upper bounds on the maximal entry of minimal representatives in each cone of the secondary fan; such a study was started in [9]. Upper bounds that are polynomial in n and d would imply a polynomial runtime of Algorithm 6. In general, we propose the length of such a shortest vector as complexity measure for subdivisions. In this sense, our analysis connects the geometric complexity with the algorithmic complexity. Considering the secondary fan of $\Delta_{n-1} \times \Delta_{d-1}$ as an analogue of the space of all realizations of a tropical oriented matroids, cf. [57], it is an interesting question to examine the complexity to determine a minimal realization. The latter problem could be solved through an adaptation of the famous algorithm in [50] for the shortest vector problem.

To justify this complexity measure, one should be able to show that cones without 'short' integer vectors correspond to hard instances for parity games, respectively the simplex method provided e.g. in [29].

It needs to be clarified how the complexity of Algorithm 5 can be measured for non-regular subdivisions. Considering non-regular subdivisions which can be constructed from

a regular subdivision by a finite sequence of flips, one can ask for the running time of the algorithm in terms of a weight matrix of the regular subdivision and the encoding of the flips. This gives a notion of the complexity of a subdivision in the non-regular case.

Furthermore, we remark that the algorithm defines a direction on the cells of the subdivision of a product of two simplices. This orientation is independent of an objective vector and also defined for non-regular subdivisions. It would be interesting to see how the orientation provides a tool to study the topological and geometric properties of this polyhedral complex in a similar vein as a discrete Morse function.

Finally, one should study the similarity between the presented algorithm and the tropicalized simplex method in [5]. This allows one to exploit the combinatorics of a product of two simplices to study the complexity of the classical simplex method. In this vein, we wonder if there is a direct connection between the diameter of a polytope and the length of integer vectors in cones of the secondary fan of $\Delta_{n-1} \times \Delta_{d-1}$.

Acknowledgements

References

[1] M. Akian, G. Cohen, S. Gaubert, R. Nikoukhah, and J. P. Quadrat. Linear systems in (max, +) algebra. In *29th IEEE Conference on Decision and Control*, pages 151–156 vol.1, Dec 1990.

[2] Marianne Akian, Stéphane Gaubert, and Alexander Guterman. Tropical polyhedra are equivalent to mean payoff games. *Internat. J. Algebra Comput.*, 22(1):1250001, 43, 2012.

[3] Marianne Akian, Stéphane Gaubert, and Alexander Guterman. Tropical Cramer determinants revisited. In *Tropical and idempotent mathematics and applications*, volume 616 of *Contemp. Math.*, pages 1–45. Amer. Math. Soc., Providence, RI, 2014.

[4] Xavier Allamigeon, Pascal Benchimol, Stéphane Gaubert, and Michael Joswig. Combinatorial simplex algorithms can solve mean payoff games. *SIAM J. Optim.*, 24(4):2096–2117, 2014.

[5] Xavier Allamigeon, Pascal Benchimol, Stéphane Gaubert, and Michael Joswig. Tropicalizing the simplex algorithm. *SIAM J. Discrete Math.*, 29(2):751–795, 2015.

[6] Xavier Allamigeon, Pascal Benchimol, Stéphane Gaubert, and Michael Joswig. Log-barrier interior point methods are not strongly polynomial. *SIAM J. Appl. Algebra Geom.*, 2(1):140–178, 2018.

[7] Federico Ardila and Sara Billey. Flag arrangements and triangulations of products of simplices. *Adv. Math.*, 214(2):495–524, 2007.

[8] Federico Ardila and Mike Develin. Tropical hyperplane arrangements and oriented matroids. *Math. Z.*, 262(4):795–816, 2009.

[9] E. K. Babson and L. J. Billera. The geometry of products of minors. *Discrete Comput. Geom.*, 20(2):231–249, 1998.

[10] Pascal Benchimol. *Tropical aspects of linear programming*. Theses, École Polytechnique, December 2014.

[11] Henrik Björklund and Sergei Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. *Discrete Appl. Math.*, 155(2):210–229, 2007.

[12] Robert G. Bland. A combinatorial abstraction of linear programming. *J. Combinatorial Theory Ser. B*, 23(1):33–57, 1977.

[13] Béla Bollobás. *Modern graph theory*, volume 184 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1998.

[14] P. Butkovic and A. Aminu. Introduction to max-linear programming. *IMA J. Manag. Math.*, 20(3):233–249, 2009.

[15] Peter Butkovič. *Max-linear systems: theory and algorithms*. Springer Monographs in Mathematics. Springer-Verlag London, Ltd., London, 2010.

[16] Peter Butkovič and Karel Zimmermann. A strongly polynomial algorithm for solving two-sided linear systems in max-algebra. *Discrete Appl. Math.*, 154(3):437–446, 2006.

[17] Cristian Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time.

[18] George B. Dantzig. *Linear programming and extensions*. Princeton University Press, Princeton, N.J., 1963.

[19] Jesús A. De Loera, Jörg Rambau, and Francisco Santos. *Triangulations*, volume 25 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2010. Structures for algorithms and applications.

[20] Mike Develin and Bernd Sturmfels. Tropical convexity. *Doc. Math.*, 9:1–27 (electronic), 2004. erratum *ibid.*, pp. 205–206.

[21] Mike Develin and Josephine Yu. Tropical polytopes and cellular resolutions. *Experiment. Math.*, 16(3):277–291, 2007.

[22] Anton Dochtermann, Michael Joswig, and Raman Sanyal. Tropical types and associated cellular resolutions. *J. Algebra*, 356:304–324, 2012.

[23] Dani Dorfman, Haim Kaplan, and Uri Zwick. A Faster Deterministic Exponential Time Algorithm for Energy Games and Mean Payoff Games. In Christel Baier,

Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 114:1–114:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[24] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Internat. J. Game Theory*, 8(2):109–113, 1979.

[25] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 00:368–377, 1991.

[26] Nathanaël Fijalkow, Paweł Gawrychowski, and Pierre Ohlmann. The complexity of mean payoff games using universal graphs, 2018.

[27] Alex Fink and Felipe Rincón. Stiefel tropical linear spaces. *J. Combin. Theory Ser. A*, 135:291–331, 2015.

[28] András Frank and Éva Tardos. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.

[29] Oliver Friedmann. *Exponential Lower Bounds for Solving Infinitary Payoff Games and Linear Programs*. PhD thesis, University of Munich, 2011.

[30] Komei Fukuda. *Oriented matroid programming*. ProQuest LLC, Ann Arbor, MI, 1982. Thesis (Ph.D.)–University of Waterloo (Canada).

[31] Stéphane Gaubert and Max Plus. Methods and applications of (max, +) linear algebra. In *STACS 97 (Lübeck)*, volume 1200 of *Lecture Notes in Comput. Sci.*, pages 261–282. Springer, Berlin, 1997.

[32] Ewgenij Gawrilow and Michael Joswig. `polymake`: a framework for analyzing convex polytopes. In *Polytopes—combinatorics and computation (Oberwolfach, 1997)*, volume 29 of *DMV Sem.*, pages 43–73. Birkhäuser, Basel, 2000.

[33] I. M. Gel′fand, R. M. Goresky, R. D. MacPherson, and V. V. Serganova. Combinatorial geometries, convex polyhedra, and Schubert cells. *Adv. in Math.*, 63(3):301–316, 1987.

[34] I. M. Gel′fand, M. M. Kapranov, and A. V. Zelevinsky. *Discriminants, resultants, and multidimensional determinants*. Mathematics: Theory & Applications. Birkhäuser Boston, Inc., Boston, MA, 1994.

[35] Dima Grigoriev. Complexity of solving tropical linear systems. *Comput. Complexity*, 22(1):71–88, 2013.

[36] Dima Grigoriev and Vladimir V. Podolskii. Tropical Effective Primary and Dual Nullstellensätze, 2014. preprint `arXiv:1409.6215`.

[37] Dima Grigoriev and Vladimir V. Podolskii. Tropical effective primary and dual Nullstellensätze. In *32nd International Symposium on Theoretical Aspects of Computer Science*, volume 30 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 379–391. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2015.

[38] V. A. Gurvich, A. V. Karzanov, and L. G. Khachiyan. Cyclic games and finding minimax mean cycles in digraphs. *Zh. Vychisl. Mat. i Mat. Fiz.*, 28(9):1407–1417, 1439, 1988.

[39] Thomas Dueholm Hansen. *Worst-case Analysis of Strategy Iteration and the Simplex Method.* PhD thesis, Aarhus University, 2012.

[40] Sven Herrmann, Anders Jensen, Michael Joswig, and Bernd Sturmfels. How to draw tropical planes. *Electron. J. Combin.*, 16(2, Special volume in honor of Anders Björner):#R6, 26, 2009.

[41] Silke Horn. The polymake extension tropmat. http://solros.de/polymake/tropmat/.

[42] Silke Horn. *Tropical Oriented Matroids and Cubical Complexes.* PhD thesis, Technische Universität Darmstadt, 2012.

[43] Silke Horn. A topological representation theorem for tropical oriented matroids. *J. Combin. Theory Ser. A*, 142:77–112, 2016.

[44] Marianne Johnson and Mark Kambites. Face monoid actions and tropical hyperplane arrangements. *Int. J. Algebra Comput.*, 27(8):1001–1025, 2017.

[45] Michael Joswig and Georg Loho. Weighted digraphs and tropical cones. *Linear Algebra Appl.*, 501:304–343, 2016.

[46] Marcin Jurdziński. Deciding the winner in parity games is in UP ∩ *co*-UP. *Inform. Process. Lett.*, 68(3):119–124, 1998.

[47] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[48] Leonid Khachiyan. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR*, 244(5):1093–1096, 1979.

[49] Victor Klee and George J. Minty. How good is the simplex algorithm? In *Inequalities, III (Proc. Third Sympos., Univ. California, Los Angeles, Calif., 1969; dedicated to the memory of Theodore S. Motzkin)*, pages 159–175. Academic Press, New York, 1972.

[50] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

[51] Georg Loho. *Combinatorics of tropical linear programming.* PhD thesis, Technische Universität Berlin, 2017.

[52] Rolf H. Möhring, Martin Skutella, and Frederik Stork. Scheduling with AND/OR precedence constraints. *SIAM J. Comput.*, 33(2):393–415, 2004.

[53] Katta G. Murty. *Linear and combinatorial programming.* John Wiley & Sons, Inc., New York-London-Sydney, 1976.

[54] Suho Oh and Hwanchul Yoo. Triangulations of $\Delta_{n-1} \times \Delta_{d-1}$ and tropical oriented matroids. In *23rd International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2011)*, Discrete Math. Theor. Comput. Sci. Proc., AO, pages 717–728. Assoc. Discrete Math. Theor. Comput. Sci., Nancy, 2011.

[55] Suho Oh and Hwanchul Yoo. Triangulations of $\Delta_{n-1} \times \Delta_{d-1}$ and Matching Ensembles, 2013. preprint `arXiv:1311.6772`.

[56] Alexander Postnikov. Permutohedra, associahedra, and beyond. *Int. Math. Res. Not. IMRN*, 2009(6):1026–1106, 2009.

[57] Jürgen Richter-Gebert. *Realization spaces of polytopes*, volume 1643 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1996.

[58] Jürgen Richter-Gebert, Bernd Sturmfels, and Thorsten Theobald. First steps in tropical geometry. In *Idempotent mathematics and mathematical physics*, volume 377 of *Contemp. Math.*, pages 289–317. Amer. Math. Soc., Providence, RI, 2005.

[59] R. T. Rockafellar. The elementary vectors of a subspace of R^N. In *Combinatorial Mathematics and its Applications (Proc. Conf., Univ. North Carolina, Chapel Hill, N.C., 1967)*, pages 104–127. Univ. North Carolina Press, Chapel Hill, N.C., 1969.

[60] Francisco Santos. Triangulations of oriented matroids. *Mem. Amer. Math. Soc.*, 156(741):viii+80, 2002.

[61] Francisco Santos. The Cayley trick and triangulations of products of simplices. In *Integer points in polyhedra—geometry, number theory, algebra, optimization*, volume 374 of *Contemp. Math.*, pages 151–177. Amer. Math. Soc., Providence, RI, 2005.

[62] Sven Schewe. From parity and payoff games to linear programming. In *Mathematical foundations of computer science 2009*, volume 5734 of *Lecture Notes in Comput. Sci.*, pages 675–686. Springer, Berlin, 2009.

[63] Bernd Sturmfels and Andrei Zelevinsky. Maximal minors and their leading terms. *Adv. Math.*, 98(1):65–112, 1993.

[64] Tamás Terlaky. A finite criss-cross method for oriented matroids. *Alkalmaz. Mat. Lapok*, 11(3-4):385–398, 1985.

[65] Michael J. Todd. Complementarity in oriented matroids. *SIAM J. Algebraic Discrete Methods*, 5(4):467–485, 1984.

[66] Michael J. Todd. Linear and quadratic programming in oriented matroids. *J. Combin. Theory Ser. B*, 39(2):105–133, 1985.

[67] Oleg Viro. Dequantization of real algebraic geometry on logarithmic paper. In *European Congress of Mathematics, Vol. I (Barcelona, 2000)*, volume 201 of *Progr. Math.*, pages 135–146. Birkhäuser, Basel, 2001.

[68] Günter M. Ziegler. *Lectures on polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995.

[69] Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoret. Comput. Sci.*, 158(1-2):343–359, 1996.

A   Related algorithmic problems

The feasibility problem for tropical linear inequality systems is the problem of finding a feasible point of the system. We highlight the relation of this problem to scheduling, mean payoff games and classical linear programming.

The complexity of the decision problems for scheduling AND-OR-networks with arbitrary coefficients and mean payoff games is known to be in NP ∩ co-NP and even more in UP ∩ co-UP, see [46, 69, 24, 52], but there is no polynomial time algorithm known. This was also unclear for classical linear programming while the containment in the complexity class NP ∩ co-NP follows easily from linear programming duality. Finally, Khachiyan [48] and, not long after, also Karmarkar [47] provided polynomial-time algorithms. However, it is still unclear if there is a pivoting rule for the simplex method for which it runs in weakly or even strongly polynomial time, see, e. g., [18, 12, 49]. The close relations between tropical linear programming, mean payoff games and classical linear programming, in particular the simplex method, are demonstrated in [62, 2, 5, 4].

A.1   The simplex method

In [5], it was shown how a run of the classical simplex method translates to a run of a tropical simplex method under some technical assumptions on the input and the requirement that the pivoting rule is combinatorial. This led to a new algorithm for solving mean payoff games presented in [4] which is polynomial time equivalent to the simplex method with the given pivoting rule. A reduction from mean payoff games to linear programming was already given in [62]. However, this approach requires exponentially large coefficients which results in a pseudopolynomial running time due to cost of the arithmetic operations. This is resolved in the approach in [5] by considering only the signs determining the pivoting which can be computed directly from the input data.

We give a short introduction to the classical simplex method [18]. We present it as an algorithm to determine the feasibility of a classical linear inequality system. Our exposition is inspired by [53, §4.5].

It is important to observe the similarity between this variant of the simplex method and the algorithms in Section 3, in particular Algorithm 2. To obtain that algorithm as a tropicalization of the following variant of the simplex method, one would have to ensure that $x \geqslant 0$.

The feasibility problem is the task to find an $x \in \mathbb{R}^d$ which fulfills the system

$$A \cdot x \leqslant b \qquad \text{for } A \in \mathbb{R}^{n \times d}, b \in \mathbb{R}^n \ .$$

The following is meant to highlight that we can consider it as a method which traverses the vertex-edge graph of the affine hyperplane arrangement given by the equations $a_j \cdot x = b_j$ for $j \in [n]$. Here, a_j is the jth row of A. At each vertex, one is given a rule for choosing the consecutive vertex in a way that guarantees termination.

We assume that the system $(A|b)$ is generic by which we mean that the d-sets $J \subseteq [n]$ are in bijection with the points z which fulfill the subsystem $A_J z = b_J$ with row

indices in J. Start with an arbitrary d-set J_0 from $[n]$ and define $x^0 := A_{J_0}^{-1} b_{J_0}$. Then $[n]$ is partitioned into three sets, namely J_0, $K_0^+ := \{ j \in [n] \mid a_j x^0 < b_j \}$ and $K_0^- := \{ j \in [n] \mid a_j x^0 > b_j \}$. The set J_0 denotes the *basic* variables and $[n] \setminus J_0 = K_0^+ \cup K_0^-$ the *non-basic* variables.

Fix an arbitrary vector $y^0 \in \mathbb{R}^n$ with $y^0 \geqslant 0$ whose support is J_0, e.g. the characteristic vector of J_0 and define

$$c = A^\top y^0 \in \mathbb{R}^d .$$

In this way, we obtain a *primal linear program* (P) and its *dual linear program* (D)

$$(P) \quad \begin{array}{c} \max c^\top x \\ Ax \leqslant b \end{array} \qquad (D) \quad \begin{array}{c} \min b^\top y \\ A^\top y = c, \quad y \geqslant 0 \end{array} .$$

By construction, y^0 is a feasible point of the dual linear program. Therefore, we can apply "Phase II" of the simplex method as we are already equipped with a feasible point. We want to consider it as a feasibility algorithm for (P). In particular, we want to reach a point x_ℓ where $K_\ell^- = \varnothing$.

First, pick an index $r_0 \in K_0^-$. We want to change x^0 such that the index r_0 of the violated inequality *enters* the *basis*. This means that r_0 becomes a basic variable.

Define

$$i_0 = \arg\min \left\{ \frac{((A_{J_0}{}^\top)^{-1} c)_i}{((A_{J_0}{}^\top)^{-1} a_{r_0}{}^\top)_i} \geqslant 0 \ \middle| \ i \in [d] \right\} ,$$

and λ_0 as the value of this minimum. In the generic case, this minimum is attained at most once. If this minimum does not exist, the inequality system of (P) is infeasible. Note that the existence of this minimum is independent of the choice of c since the occurring numerators are the positive components of y^0. Let j_0 be the i_0-th element of J_0 considered as an ordered index tuple for the rows of A_{J_0}. Then j_0 is the *leaving* variable and $J_1 = J_0 \setminus \{j_0\} \cup \{r_0\}$ becomes the new basis. Now, we can restart the iteration. However, we keep c fixed and for $\ell \geqslant 1$ choose y^ℓ iteratively in the following way:

$$y_j^\ell = \begin{cases} ((A_{J_\ell}{}^\top)^{-1} c)_j & \text{for } j \in J_\ell \\ 0 & \text{for } j \in [n] \setminus J_\ell . \end{cases} \tag{5}$$

Theorem 77. *The vector $y^1 \in \mathbb{R}^n$ fulfills $y^1 \geqslant 0$, $c = A^\top y^1$ and $b^\top y^1 < b^\top y^0$.*

Proof. Consider the linear equality system

$$c = A_{J_0 \cup r_0}{}^\top z .$$

For $z_{d+1} = 0$ we get the solution $y_{J_0}^0 = (A_{J_0}{}^\top)^{-1} c$ and for $z_{i_0} = 0$ we obtain the solution $y_{J_1}^1 = (A_{J_1}{}^\top)^{-1} c$ (up to relabeling of the coordinates).

Furthermore, by multiplying both sides with $A_{J_0}^{-1}$ from the left, we obtain

$$
\left(
\begin{array}{cccc|c}
1 & 0 & \cdots & 0 & \\
0 & \ddots & 0 & 0 & \\
\vdots & 0 & \ddots & 0 & (A_{J_0}{}^\top)^{-1}a_{r_0}{}^\top \\
0 & \cdots & 0 & 1 &
\end{array}
\right) \cdot z = (A_{J_0}{}^\top)^{-1}c \ .
$$

This is equivalent to

$$
z_{[d]} = \left((A_{J_0}{}^\top)^{-1}c\right) - z_{d+1}\left((A_{J_0}{}^\top)^{-1}a_{r_0}{}^\top\right) \ . \tag{6}
$$

Choosing z_{d+1} as λ_0, we obtain $z_{i_0} = 0$ and hence, $y_{J_1}^1 = z_{[d]\setminus i_0}$. Moreover, Equation 6 implies $y^1 \geqslant 0$ and $c = A^\top y^1$. Finally, we obtain the difference

$$
b^\top \cdot y^1 - b^\top \cdot y^0 = b_{J_0 \cup r_0}{}^\top \left(\begin{pmatrix} y_{J_0}^0 - \lambda_0\left((A_{J_0}{}^\top)^{-1} \cdot a_{r_0}{}^\top\right) \\ \lambda_0 \end{pmatrix} - \begin{pmatrix} y_{J_0}^0 \\ 0 \end{pmatrix} \right) \ .
$$

This simplifies to

$$
\lambda_0 \left(b_{r_0} - b_{J_0}{}^\top (A_{J_0}{}^\top)^{-1} \cdot a_{r_0}{}^\top \right) \ .
$$

With $x^0 = A_{J_0}^{-1}b_{J_0}$, $a_{r_0} \cdot x^0 > b_{r_0}$ and $\lambda_0 \geqslant 0$, the claim follows. $\qquad\square$

If we continue the iteration with y^1 we obtain a sequence of d-subsets $J_0, J_1, \ldots, J_m$ of $[n]$. The sets in this sequence are pairwise disjoint since the sequence of the values $b^\top \cdot y^\ell$, which is defined by J_ℓ via Equation 5, is strictly decreasing. This implies the termination of the iteration as there are only finitely many subsets of $[n]$.

Remark 78. We could change y^ℓ after each iteration in a way that preserves the objective function value $b^\top \cdot y^\ell$ and the support. This would require a new computation of c. All the statements, in particular the ones concerning the termination of the algorithm, would remain valid.

A.2 Scheduling with AND-OR-networks

Scheduling is concerned with the task of putting several jobs into an order in which they are worked through such that certain constraints are fulfilled. We give a short introduction to a special class of scheduling problems, namely *AND-OR-networks*. They occur in project management with particular temporal dependencies and can be used to model resource constraints. They were extensively studied in, e.g., [52]. In particular, that work contains a formulation of the precedence relations for the starting times with min- and max-inequalities. It also shows the polynomial time equivalence with a decision problem associated to a mean payoff game. We display a tropical geometric relation between the formulation of the set of vectors of starting times and the feasible region of a suitable tropical signed system. For other instances of scheduling problems which can be expressed in terms of tropical inequalities or equations see, e. g., [14, §1].

To explain an AND-OR-network we consider the planning of a project. The single jobs depend on each other and are in some precedence relation. We assume that a started

job may not be interrupted. If a job can only start if all its predecessor jobs are finished, we call this an *AND-constraint*. If a job can start if at least one of its predecessors is finished, we call this an *OR-constraint*.

In Figure 21, one can see the Gantt chart of an AND-constraint and of an OR-constraint visualizing the dependence of the start and finish dates of jobs in these predecessor relations. Here, the dashed line denotes the starting time of the next job which is represented by the bottom bar, its predecessors forming the top three. The lengths of the bars illustrate the processing times.

Figure 21: Two types of constraints, OR left, AND right.

Notice that usually one requires the special *starting condition* that every job has to begin after some given point in time. In our model, this is covered by the fact that the expressions are additively homogeneous and hence, one can just mark one node and dehomogenize with respect to this coordinate.

For a broader introduction of scheduling with AND-OR-constraints see [52]. We give a formal definition to work with.

Definition 79. An *AND-OR-network* is given by a set of states V and a set of waiting conditions U. The waiting conditions are pairs (X, J) with $J \subseteq V$ and $X \subseteq V \setminus \{J\}$.

The pair (V, U) can be construed to be a directed bipartite graph $\mathcal{B}$ with node set $V \sqcup U$. Each waiting condition (X, J) is expressed by the arcs $(x, (X, J))$ for $x \in X$ and $((X, J), j)$ for $j \in J$. Because of $X \subseteq V \setminus \{J\}$, for each pair $v \in V$ and $u \in U$ there exists at most one of the arcs (u, v) or (v, u). We denote the arc set by $\mathcal{A}$.

Furthermore, we have a weight function $\omega \colon \mathcal{A} \to \mathbb{Q}$ on the arcs to encode processing times, or time lags if the weight is negative.

Then we can describe the precedence constraints for the vector of starting times $t \in \mathbb{T}_{\min}^{V \sqcup U}$ by the inequalities

$$
\begin{aligned}
t_v &\geqslant \max_{(u,v) \in \mathcal{A}} (t_u + \omega(u, v)) && \text{for all } v \in V && \text{(AND)} \\
t_u &\geqslant \min_{(v,u) \in \mathcal{A}} (t_v + \omega(v, u)) && \text{for all } u \in U && \text{(OR)} .
\end{aligned}
\tag{7}
$$

The max-inequalities correspond to AND-constraints and the min-inequalities to OR-constraints.

We can reformulate the first inequality in (7) by splitting the maximization into several inequalities to obtain

$$
\begin{aligned}
t_v &\geqslant (t_u + \omega(u, v)) && \text{for all } (u, v) \in \mathcal{A} \text{ with } u \in U, v \in V && \text{(8a)} \\
t_u &\geqslant \min_{(v,u) \in \mathcal{A}} (t_v + \omega(v, u)) && \text{for all } u \in U . && \text{(8b)}
\end{aligned}
$$

Observe that this already yields a signed system.

We can transform the first kind of inequalities (8a) further into

$$t_v - \omega(u,v) \geqslant \qquad t_u \quad \text{for all } (u,v) \in \mathcal{A} \text{ with } u \in U, v \in V \Leftrightarrow \qquad \text{(8a')}$$

$$\min_{(u,v)\in\mathcal{A}} (t_v - \omega(u,v)) \geqslant \qquad t_u \quad \text{for all } u \in U \ . \qquad \text{(8a'')}$$

Combining the two kinds of inequalities (8b) and (8a'') yields

$$\min_{(u,v)\in\mathcal{A}} (t_v - \omega(u,v)) \geqslant t_u \geqslant \min_{(v,u)\in\mathcal{A}} (t_v + \omega(v,u)) \qquad \forall u \in U \ .$$

Let $|V| = d$ and $|U| = n$. Then we define matrices $(a_{ji}) = A \in \mathbb{T}_{\min}^{n\times d}$ and $(\sigma_{ji}) = \Sigma \in \{+,-,\bullet\}^{n\times d}$ by identifying each node in V resp. U with indices in $[d]$ resp. $[n]$ and setting

$$(\ a(u,v) \ , \ \sigma(u,v) \ ) = \begin{cases} (\omega(v,u), +) & (v,u) \in \mathcal{A} \\ (-\omega(u,v), -) & (u,v) \in \mathcal{A} \\ (\infty, \bullet) & \text{else} \end{cases}$$

for $v \in V$ and $u \in U$. This defines a signed system (A, Σ) whose associated inequality system is

$$\min_{\sigma(u,v)=-} (t_v + a(u,v)) \geqslant \min_{\sigma(u,v)=+} (t_v + a(u,v)) \quad \text{for all} \quad u \in U \ . \qquad (9)$$

Conversely, if we are given a feasible solution $(t_v)_{v \in V}$ of (9) we can define starting times t_u for $u \in U$ by

$$t_u = \min_{\sigma(u,v)=-} (t_v + a(u,v)) \qquad (10)$$

such that $(t_k)_{k \in U \sqcup V}$ fulfills (7). We summarize our considerations in the following theorem.

Theorem 80. *The set of feasible points for* (9) *is the projection of the set of feasible starting times for* (7) *on the coordinates labeled by* V. *Furthermore, for every feasible point of* (9) *we find a feasible point of* (7).

Example 81. Figure 22 depicts the AND-OR-network for the signed system from Example 8. For this signed system, we know that $(0, 2, 4.5)$ is a feasible point. This translates to possible start times for the AND-nodes. With Equation 10, we calculate $(2, 1, 0, 0)$ as possible starting times for the OR-nodes.

With the dehomogenization $x_1 = 0$, the coordinatewise minimal point of the feasible region amounts to the point $(0, 0, 0)$. This yields $(0, -1, -2, 0)$ for the resulting start times of the OR-nodes.

Remark 82. The pseudopolynomial algorithm in [52, §7.2.2] uses the basic idea to make a violated inequality an equality. If a starting time t_j violates the inequality $t_j \geqslant \min_{i \in X}(t_i + d_{iw})$ for a waiting condition $w = (X, j)$, one assigns the new value $\min_{i \in X}(t_i + d_{iw})$ to t_j. This yields a pseudopolynomial algorithm as the iteratively computed starting times only increase and can be bounded from above. Similar ideas will come up later on in subsection 6.3.
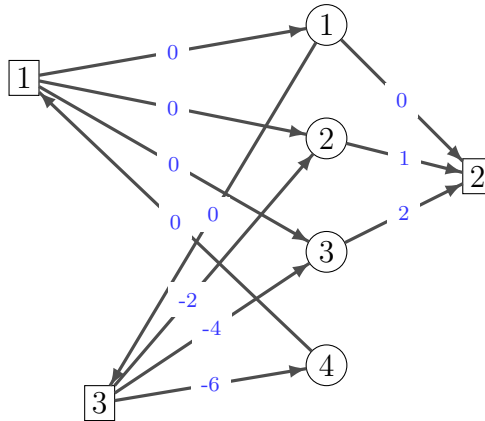
Figure 22: The scheduling network derived from the signed system of Example 8.

A.3 Mean payoff games

The connection between mean payoff games and tropical linear inequality systems, which we describe below, was established in [2]. A similar result implicitly occurs in [52, Lemma 7.5] and [62, Lemma 2].

A.3.1 Introduction to mean payoff games

We briefly introduce mean payoff games. Let $\mathcal{G}$ be a finite directed bipartite graph with node set $V_0 \sqcup V_1$, arc set $\mathcal{A}$ and a weight function $\omega \colon \mathcal{A} \to \mathbb{Q}$ on the arcs. Without loss of generality, we can assume that $V_0 = [d]$ and $V_1 = [n]$.

We define a finite two-player game with full information on $\mathcal{G}$, following [69]. At a node in V_p, it is the turn of player p, for $p \in \{0, 1\}$. Starting from a fixed node $k \in V_0 \sqcup V_1$, the players alternatingly choose an outgoing arc of the current node and move to the tip of the arc. If a player cannot move because there is no outgoing are, she looses. As soon as the directed path formed in this way produces a cycle, the game finishes. The *outcome* of the game with starting point k is the mean weight of the arcs in that cycle. One player tries to maximize, while the other player tries to minimize the outcome of the game.

A positional *strategy* for player $p \in \{0, 1\}$ is a subset τ_p of the arcs $\mathcal{A}$, such that each vertex in V_p is either isolated or incident to exactly one outgoing arc in τ_p. By [24], a mean payoff game has an optimal positional strategy.

Following [36, §7], we say that a position $i \in V_0$ is *non-losing* for player 1 if there is a strategy for player 1 such that the outcome of the game starting with i is non-negative. As we are only interested in the distinction between losing or non-losing this essentially amounts to considering *energy games* instead, see [23].

We construct a signed system from the bipartite graph $\mathcal{G}$ with the weights ω similar to Section A.2, but with switched signs.

Let $|V_0| = d$ and $|V_1| = n$. Then we define matrices $(a_{ji}) = A \in \mathbb{T}_{\min}^{n \times d}$ and $(\sigma_{ji}) = \Sigma \in \{+, -, \bullet\}^{n \times d}$ by identifying each node in V_0 resp. V_1 with indices in $[d]$ resp. $[n]$ and
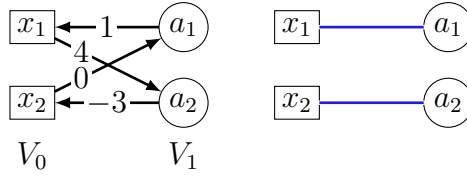
Figure 23: A bipartite graph $\mathcal{G}$ depicting the mean payoff game from Example 84 and a non-losing strategy τ for player 1 owning the circle nodes.

setting

$$(a(v_1, v_0), \sigma(v_1, v_0)) = \begin{cases} (\omega(v_0, v_1), -) & (v_0, v_1) \in \mathcal{A} \\ (-\omega(v_1, v_0), +) & (v_1, v_0) \in \mathcal{A} \\ (\infty, \bullet) & \text{else} \end{cases} \tag{11}$$

for $v_0 \in V_0$ and $v_1 \in V_1$.

Note that the former construction is reversible.

We state the main theorem connecting tropical linear inequality systems and mean payoff games, see [2, Theorem 3.2].

Theorem 83. *The set of non-losing states in V_0 for player 1 equals the set of those $i \in [d]$ for which there is a feasible point x for (A, Σ) with $x_i \neq \infty$.*

We sketch one direction of an independent proof to demonstrate how this ties in with the properties of covector graphs. Let $x \in \mathbb{T}_{\min}^d$ be a feasible point for (A, Σ) with support $D \neq \varnothing$. Since its covector graph G is feasible, each node in $[n]$ is either isolated or incident with a positive edge in G. If an apex node $j \in [n]$ is isolated in G, there is no arc between D and j in $\mathcal{G}$ either. For an isolated node, we pick no edge and for a non-isolated apex node, we pick one incident positive edge in G. This yields a strategy τ for player 1.

If a run of the game with starting node in D and fixed strategy τ for player 1 produces a cycle, it can only be a non-negative cycle by [45, Proposition 38]. This implies the claim.

Example 84. The signed system for the graph $\mathcal{G}$ from Figure 23 is given by

$$\begin{matrix} & \begin{matrix} x_1 & x_2 \end{matrix} & & \begin{matrix} x_1 & x_2 \end{matrix} \\ \begin{matrix} a_1 \\ a_2 \end{matrix} & \begin{pmatrix} -1 & 0 \\ 4 & 3 \end{pmatrix} & \begin{matrix} a_1 \\ a_2 \end{matrix} & \begin{pmatrix} + & - \\ - & + \end{pmatrix} \end{matrix} .$$

The corresponding inequality system is $x_1 - 1 \leqslant x_2, \quad x_2 + 3 \leqslant x_1 + 4$. The non-losing strategy is obtained from the positive edges of the feasible point $(0, -1)$.

We also relate the example for AND-OR-networks with the corresponding mean payoff game.

Example 85. By reversing the arcs and negating the weights in Figure 22, we obtain the game graph corresponding to the inequality system from Example 8. The blue edges in the covector graph of the feasible point $(0, 2, 4.5)$ yield the non-losing strategy formed by $(1, 1), (2, 1), (3, 1), (4, 3)$ (which are directed from circle to square nodes). This, for example, yields the positive cycle ④, ③, ③, ①.

A.3.2 Parity games as special mean payoff games

Parity games [25, 46] also are two player games with perfect information. However, we have no weights on the edges but on the vertices of the game graph. The vertices are assigned to the two players, *even* and *odd*. Even vertices are labeled by an even integer weight, odd vertices by an odd integer weight. Player even wins if the maximal number in the terminating cycle is even, otherwise odd wins.

Let $M = d + n$ be the number of vertices in the two classes. We can consider a parity game as a special mean payoff game where the outgoing edges of a vertex with label $k \in \mathbb{Z}$ get the weight $(-M)^k$. Then the winning states of the so constructed mean payoff game for player 0 resp. 1 are exactly the winning states of player even resp. odd in the parity game. For more details see, e.g., [46].

Recently, it was shown in [17] that parity games can be solved in quasipolynomial time. Parity games have served as suitable instances to demonstrate the worst-case complexity of many algorithms, see, e.g., [29, 39].

B Polyhedral prerequisities

B.1 Polytopes and polyhedral subdivisions

The *convex hull* of a subset $V = \{v_1, \dots, v_k\} \subset \mathbb{R}^d$ is defined as

$$\text{conv}(V) = \left\{ \sum_{i=1}^{k} \lambda_i \cdot v_i \ \middle| \ \sum_{i=1}^{k} \lambda_i = 1, \lambda_i \in \mathbb{R}_{\geqslant 0} \right\} \ .$$

A *polytope* is the convex hull of finitely many points. A *(affine) halfspace* is a set of the form

$$\left\{ x \in \mathbb{R}^d \ \middle| \ a \cdot x \leqslant b \right\} \ ,$$

for some $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$. A *polyhedron* is the intersection of finitely many halfspaces. By the Minkowski-Weyl theorem, polytopes are exactly the bounded polyhedra. The *face* of a polyhedron P is the intersection of P with a halfspace that does not contain an interior point of P.

A *subpolytope* of a polytope P is the convex hull of a subset of the vertices of P. The convex hull of k affinely independent points, for $k \in \mathbb{N}$, is a $(k-1)$-*simplex* and is denoted by Δ_{k-1}. In the following, Δ_{k-1} stands for the convex hull of the k *standard basis vectors* $e_1, e_2, \dots, e_k$ in $\mathbb{R}^k$, which is an instance of a $(k-1)$-simplex. The *product* of two polytopes $P \subseteq \mathbb{R}^d$ and $Q \subseteq \mathbb{R}^n$ is the convex hull of the pairs $(p, q) \in \mathbb{R}^{d+n}$ where p resp. q ranges over all the vertices of P resp. Q.

A *polyhedral complex* is a finite set $\mathcal{C}$ of polyhedra such that

1. each face of a polyhedron is also contained in $\mathcal{C}$,

2. the intersection of two polyhedra in $\mathcal{C}$ is empty or a face of both.

A polyhedral complex $\mathcal{C}$ is a *(polyhedral) subdivision* of a polyhedron P if the union of the polyhedra in $\mathcal{C}$ is P. If all polyhedra in a subdivision of a polytope P are simplices, than this comprises a *triangulation* of P. For constructing A subdivision of a polytope $P \subset \mathbb{R}^d$ is *regular* if it is the orthogonal projection, omitting the last coordinate, of the bounded cells of the polyhedron

$$\text{conv}\left\{ (x, h(x) \mid x \text{ vertex of } P \right\} + \mathbb{R}_{\geqslant 0} \cdot e_{d+1}$$

for some height function $h \colon \mathbb{R}^d \to \mathbb{R}$. Note that not all subdivisions of a polytope are regular as Example 45 shows.

B.2   An application

The deduction of Proposition 19 from [54, Proposition 2.5] relies on the Cayley trick ([19, §9]). The latter states that triangulations of $\Delta_{n-1} \times \Delta_{d-1}$ are in bijection with fine mixed subdivisions of $n\Delta_{d-1}$. The claim follows as the collection of bipartite graphs corresponds to the full-dimensional simplices in a triangulation of $\Delta_{n-1} \times \Delta_{d-1}$. Note that a similar statement was proven in [22, Proposition 4.2]. Because of the importance to us, we give a proof independently of [54], highlighting the connection between geometry and combinatorics.

Proof of Proposition 19. Let the *right degree sequence (RDS)* be the sequence of degrees of the apex nodes.

By [19, Theorem 6.2.13], which uses the unimodularity, respectively the equidecomposability, of $\Delta_{n-1} \times \Delta_{d-1}$, the number of full-dimensional simplices in a triangulation is $\binom{n+d-2}{n-1}$.

Furthermore, the number of compositions of $n + d - 1$ in n parts is $\binom{n+d-2}{n-1}$.

Hence, it suffices to prove that each sequence $(d_1, \ldots, d_n) \in [d]^n$ with $\sum_{j=1}^{n} d_j = n + d - 1$ occurs at most once as an RDS. We describe a construction to find a canonical form for a covector graph with a given RDS which will imply the claim. This approach is depicted in Figure 24.

Next, note that we can omit apex nodes of degree 1 as the graph remaining after this removal is still a tree. So, consider two distinct trees t_0 and t_1 with the same RDS $(d_1, \ldots, d_n)$ for which each degree is bigger than 1. From these trees, we construct trees s_0 and s_1 for which each apex node has degree 2. For this, we replace each apex node $j \in [n]$ of degree $d_j > 2$ with $d_j - 1$ nodes $k_1^j, \ldots, k_{d_j-1}^j$. Furthermore, if $i_{j_1} \leqslant \ldots \leqslant i_{j_{d_k}}$ are the neighbors of j, we add the edges

$$(i_{j_1}, k_1^j), (i_{j_2}, k_1^j), (i_{j_2}, k_2^j), \ldots, (i_{j_{d_k-1}}, k_{d_j-1}^j), (i_{j_{d_k}}, k_{d_j-1}^j) \ .$$

Hence, s_0 and s_1 are trees on the vertices $[d] \sqcup R$, where R is the d-set formed by the old apex nodes of degree 2 and the new apex nodes which arose from replacing apex nodes of degree > 2. By Lemma 18, these trees are the union of $(d-1) \times (d-1)$-matchings on $[d] \setminus \{i\} \sqcup R$ for all $i \in [d]$. From the uniqueness of the construction of s_0 resp. s_1 from t_0 resp. t_1 we deduce that s_0 and s_1 are also distinct. Therefore, there is an $i \in [d]$ for which the perfect matching μ_0 in s_0 on $[d] \setminus \{i\} \sqcup R$ and the perfect matching μ_1 in s_1 on $[d] \setminus \{i\} \sqcup R$ disagree. We conclude that their symmetric difference contains a non-trivial simple cycle C. If we contract the nodes $k_1^j, \ldots, k_{d_j - 1}^j$ back to the single node j for each apex node $j \in [n]$ of degree $d_j > 2$, then C becomes a cycle (where a node can appear multiple times). Since t_0 and t_1 are distinct, the cycle has to contain more than 1 apex node. Such a cycle is an alternating cycle in the sense of the comparability. This implies that t_0 and t_1 cannot both occur in the same triangulation. $\qquad\square$
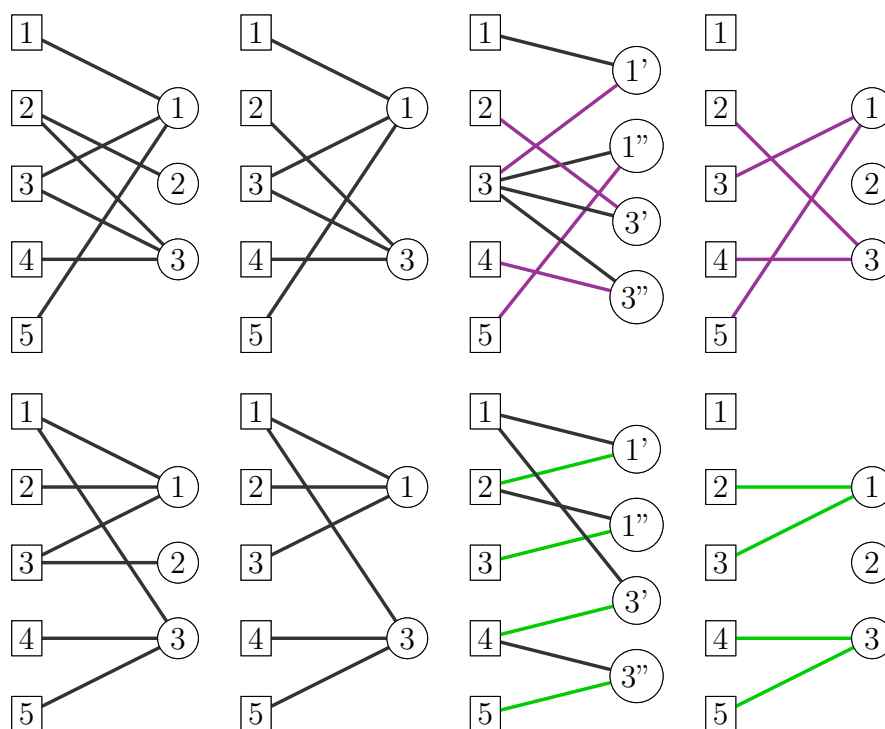

Figure 24: The construction to find an alternating cycle from the proof of Proposition 19.

C   Maximal support and partial strategies

Since the tropical sum of two feasible vectors is feasible again, the union of the supports of the feasible points is the support of a feasible point, see also [2, Theorem 3.2]. We call this the *feasible support*.

Algorithm 6 determines a feasible point of a signed system or certifies that there is none. However, a resulting feasible point does not need to have the full feasible support. We show how one can use Algorithm 6 to determine the feasible support. The interest to determine this is motivated by the interpretation of the feasible points as vectors of feasible starting times or winning positions in a mean payoff game presented in Section A.

We need some technical observations to achieve this.

Lemma 86. *Let (A, Σ) be a signed system for which the ith column of Σ only contains '+' entries. Then for any point $(z_1, \ldots, z_d) \in \mathbb{T}^d_{\min}$ there is a number $\xi \in \mathbb{R}$ for which $(z_1, \ldots, z_{i-1}, \xi, z_{i+1}, \ldots, z_d)$ is feasible.*

Proof. We can assume, without loss of generality, that $i = 1$. Now, let $k_j \in [d]$ be the index in row $j \in [n]$ for which $\sigma_{jk_j} = -$. For $\xi \leqslant \min \left\{ z_{k_j} + a_{jk_j} - a_{j1} \mid j \in [n] \right\}$ we obtain

$$(a_{j1} \odot \xi) \oplus \bigoplus_{\ell \in [d], \ell \neq k_j, 1} a_{j\ell} \odot z_\ell \leqslant \xi + a_{j1} \leqslant z_{k_j} + a_{jk_j} \quad \text{for all } j \in [n] .$$

Hence, $(\xi, z_2, \ldots, z_d)$ is feasible. $\qquad\square$

Observation 87. *If w and z are feasible solutions with $\mathrm{supp}(w) \cap \mathrm{supp}(z) = \{k\}$ for some $k \in [d]$, then the point $v = (-w_k) \odot w \oplus (-z_k) \odot z$ has the same pairwise coordinate differences as w and z on its support. By this we mean that $v_i - v_\ell = w_i - w_\ell$ for $i, \ell \in \mathrm{supp}(w)$ and $v_i - v_\ell = z_i - z_\ell$ for $i, \ell \in \mathrm{supp}(z)$.*

Observation 88. *The inequality $x_2 \oplus x_1 \leqslant (x_1 \odot a)$ is tautological for $a \geqslant 0$ and equivalent to $x_2 \leqslant x_1 \odot a$ for $a < 0$. Furthermore, $x_1 \oplus (x_1 \odot a)$ equals x_1 for $a \geqslant 0$ and $x_1 \odot a$ for $a < 0$.*

To determine the feasible support, we run Algorithm 6 several times with a successively reduced input. As long as the algorithm terminates with a feasible point z we modify the system and restart with the reduced system.

If the support of z consists only of one element i, we omit all the inequalities which contain the variable x_i. By Lemma 86, these inequalities are fulfilled for every point for which the ith component is sufficiently small.

Now, assume that the support of z consists of the indices $i_1, \ldots, i_k$ with $k \geqslant 2$. We replace x_{i_ℓ} in each inequality of the signed system by means of the equation

$$x_{i_\ell} = x_{i_k} + z_{i_\ell} - z_{i_k} . \tag{12}$$

With Observation 88, we can restore the property that each variable occurs on at most one side of each inequality. Furthermore, the reduced system has a feasible solution if and only if the original system has one since we can construct a solution, which fulfills all the Equations 12, by Observation 87.

As soon as we reach a totally infeasible point in a reduced system we can deduce that the complement of the current coordinate nodes forms the feasible support of the original system.

Example 89. Algorithm 6 behaves pairwise differently on the examples depicted in Figure 25 concerning the determination of the feasible support.

For the top left one, it finds a feasible point with support $\{3\}$ but needs a second run to find the certificate that this is already the feasible support.

For the top right one, it finds a feasible point whose support has 2 elements and needs a second run to determine the feasible support $\{1, 2, 3\}$.

For the bottom left one, it needs only one run to determine that the support is just the empty set.

For the bottom right one, starting with $\delta = 1$ and continuing with $\delta = 2$ or $\delta = 3$ yields feasible points with different supports. In the former case, we arrive at a basic point with support $\{1, 2, 3\}$. For the latter, the resulting basic point only has support $\{1, 3\}$.
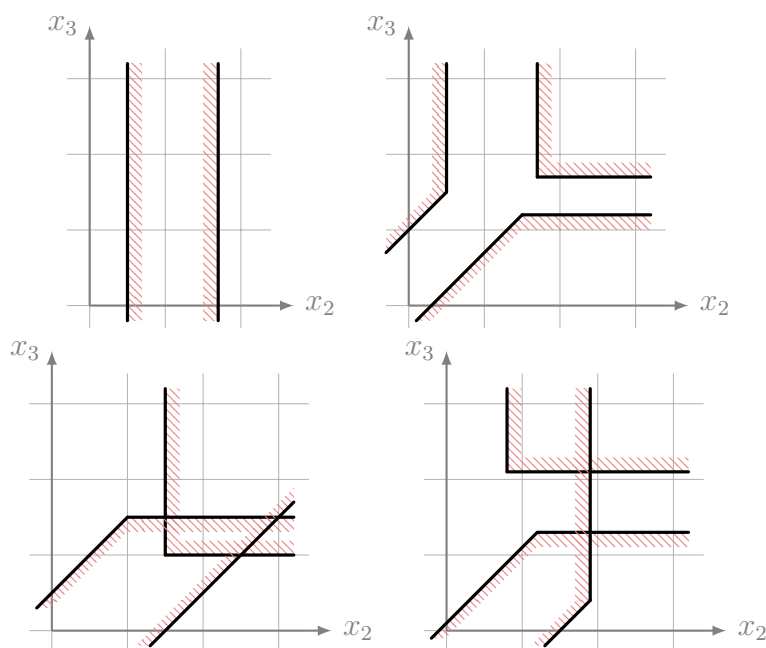

Figure 25: The bars indicate the *infeasible* regions. The supports of the feasible sets defined by the tropical halfspaces are different.

Moreover, we can use the former considerations to find a point, whose support is the feasible support, and a point which certifies that the feasible support cannot be bigger.

Definition 90. A covector graph G in $(\mathcal{S}(A), \Sigma)$ is *sufficiently infeasible* and *negatively covers* $D \subseteq [d]$ if there is a subset $N \subseteq [n]$ with $|N| = |D|$, for which $D = \bigcup_{j \in N} \operatorname{supp}(a_{j\cdot})$ and the induced subgraph of G on $D \sqcup N$ is a perfect matching consisting of negative edges.

The sufficiently infeasible covector graphs correspond to the generalized cycles with negative weight in [52]. We show how one can construct a sufficiently infeasible covector graph for a signed system.

Theorem 91. *If F is the feasible support of the signed system (A, Σ) then there is a sufficiently infeasible covector graph G which negatively covers $([d] \setminus F)$.*

Proof. We discussed how an iterative application of Algorithm 6 can be used to determine the feasible support of a signed system. For $F = [d]$ there is nothing to show. Otherwise, let (R, Υ) be last reduced system in the sequence of successively reduced signed systems; by construction, it is infeasible. Furthermore, let $(\widehat{R(\Omega)}, \Xi)$ be a refinement of an extension of (R, Υ).

Since (R, Υ) is infeasible, there is a totally infeasible covector graph H for $(\widehat{R(\Omega)}, \Xi)$. By the genericity of this system, there is a point x whose covector graph is contained in H and for which each basic apex of H is only incident with the negative edge.

We embed x into $\mathbb{T}^d_{\min}$ by setting the coordinates in F to ∞. Then by the construction of the extension and the refinement, x has the same covector graph with respect to A. Its covector graph G is sufficiently infeasible.

$\square$

Example 92. Consider the following four matrices:

$$
A = \begin{pmatrix} 0 & 0 & \infty & \infty \\ 0 & 2 & 11 & \infty \\ \infty & \infty & 0 & 0 \\ \infty & \infty & 2 & 0 \end{pmatrix} \quad \text{and} \quad \Sigma_1 = \begin{pmatrix} - & + & \bullet & \bullet \\ + & - & + & \bullet \\ \bullet & \bullet & - & + \\ \bullet & \bullet & + & - \end{pmatrix},
$$

$$
A(\Omega) = \begin{pmatrix} 0 & 0 & \Omega_1 & \Omega_2 \\ 0 & 2 & 11 & \Omega_3 \\ \Omega_4 & \Omega_5 & 0 & 0 \\ \Omega_6 & \Omega_7 & 2 & 0 \end{pmatrix} \quad \text{and} \quad \Sigma_2 = \begin{pmatrix} + & - & \bullet & \bullet \\ - & + & \bullet & \bullet \\ \bullet & \bullet & - & + \\ \bullet & \bullet & + & - \end{pmatrix}.
$$

At first, we examine the signed system (A, Σ_1). Starting with $\delta = 1$ we obtain:

δ	Cramer solution	violated inequalities
1	$A[\varnothing \mid \{1\}] = (0, \infty, \infty, \infty)$	$j = 1$
2	$A[\{1\} \mid \{1, 2\}] = (0, 0, \infty, \infty)$	

The point $(0, 0, \infty, \infty)$ is feasible and the algorithm stops. We reduce the system by replacing x_1 with x_2 and, by using Observation 88, arrive at the system

$$
(A', \Sigma_1') = \left(\begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix}, \begin{pmatrix} - & + \\ + & - \end{pmatrix} \right).
$$

The Cramer solution $C_{A'}(\{3\}, \{3, 4\}) = (0, 0)$ certifies the infeasibility of this reduced system. The point $x = (\infty, \infty, 0, 1)$ has a sufficiently infeasible covector graph.

As a second example, we consider the signed system (A, Σ_2). We construct Ξ by replacing the $\bullet$ entries in Σ_2 by $+$. Then $(A(\Omega), \Xi)$ is a generic extension of (A, Σ_2). The

Cramer solution $\mathcal{C}_{A(\Omega)}(\{1,3,4\},[4]) = (0,0,\Omega_4,\Omega_4+2)$ has a totally infeasible covector graph. From this, we can obtain the point $x = (1,0,\Omega_4,\Omega_4+1)$ which has a sufficiently infeasible covector graph. This point also yields a sufficiently infeasible covector graph for the signed system (A,Σ_2).
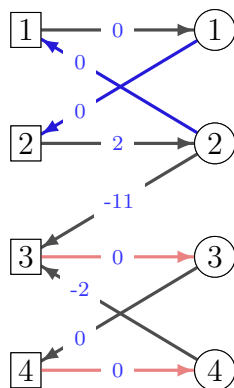

Figure 26: Graph for the mean payoff game corresponding to (A,Σ_1) from Example 92.

We conclude by interpreting a sufficiently infeasible covector graph in terms of mean payoff games. Recall the connection from Theorem 83. Extending the notions from Subsection A.3, we say that a coordinate node or an apex node is *winning* for the player on the coordinate nodes if there is a *winning strategy* meaning that the value of the game is negative when we start from such a position and this strategy is used on the coordinate nodes.

Let H be a sufficiently infeasible covector graph for the signed system (A,Σ) which negatively covers $D \subseteq [d]$.

Theorem 93. *The coordinate nodes in D and the apex nodes, whose support is contained in D, are winning positions for the strategy formed by the perfect matching μ consisting of negative edges contained in H.*

Proof. Let N be the set of the apex nodes, whose support is contained in D. Then the player on the apex nodes is forced to go back to D on N. Furthermore, the arcs formed from μ only go to N by the properties of H. Since H is a covector graph, Proposition 4 implies with the construction of the mean payoff graph in Equation 11 that all cycles reachable from N and from D through μ are negative. $\qquad\square$

With Theorem 83, we deduce an extension of Lemma 17 for the realizable case.

Corollary 94. *If $(\mathcal{S}(A),\Sigma)$ contains a sufficiently infeasible covector graph G which negatively covers D, then $\mathrm{supp}(z) \subseteq [d] \setminus D$ for every feasible point z of (A,Σ).*

Proof. Theorem 93 implies that the player on the coordinate nodes has a winning strategy which secures a negative value. Therefore, there cannot be a feasible point z with $\mathrm{supp}(z) \cap D \neq \varnothing$ since this would imply a non-losing strategy for the player on the apex nodes with starting positions $\mathrm{supp}(z)$ by Theorem 83. $\qquad\square$

Example 95. Figure 26 shows winning strategies in the mean payoff game corresponding to the signed system (A, Σ_1) from Example 92. The blue arcs form a non-loosing strategy for the player on the circle nodes. They are the positive edges in the covector graph of the feasible point $(0, 0, \infty, \infty)$. The purple arcs form a winning strategy for the player on the square nodes. They are the edges in the sufficiently infeasible covector graph of the point $(\infty, \infty, 0, 1)$.