Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

GUILLAUME SAGNOL, MARCO BLANCO, THIBAUT SAUVAGE

The Cone of Flow Matrices: Approximation Hierarchies and Applications

ZIB Report 17-32 (June 2017)

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

The Cone of Flow Matrices: Approximation Hierarchies and Applications

Guillaume Sagnol[12], Marco Blanco[1], Thibaut Sauvage[3]

July 3, 2017

Abstract

Let G be a directed acyclic graph with n arcs, a source s and a sink t. We introduce the cone $\mathcal{K}$ of flow matrices, which is a polyhedral cone generated by the matrices $\mathbf{1}_P\mathbf{1}_P^T \in \mathbb{R}^{n\times n}$, where $\mathbf{1}_P \in \mathbb{R}^n$ is the incidence vector of the $(s,t)-$path P. We show that several hard flow (or path) optimization problems, that cannot be solved by using the standard arc-representation of a flow, reduce to a linear optimization problem over $\mathcal{K}$. This cone is intractable: we prove that the membership problem associated to $\mathcal{K}$ is NP-complete. However, the affine hull of this cone admits a nice description, and we give an algorithm which computes in polynomial-time the decomposition of a matrix $X \in \operatorname{span}\mathcal{K}$ as a linear combination of some $\mathbf{1}_P\mathbf{1}_P^T$'s. Then, we provide two convergent approximation hierarchies, one of them based on a completely positive representation of $\mathcal{K}$. We illustrate this approach by computing bounds for the quadratic shortest path problem, as well as a maximum flow problem with pairwise arc-capacities.

Keywords Flows in graphs, Approximation hierarchies, Copositive programming

1 Introduction

Throughout this paper we denote by $\mathcal{S}_n$ the set of $n\times n-$symmetric matrices, and by $\mathcal{S}_n^+$ the set of $n\times n-$symmetric positive semidefinite matrices, that is,

$$\mathcal{S}_n^+ = \operatorname{conv}(\{\boldsymbol{x}\boldsymbol{x}^T : \boldsymbol{x}\in\mathbb{R}^n\}) = \operatorname{cone}(\{\boldsymbol{x}\boldsymbol{x}^T : \boldsymbol{x}\in\mathbb{R}^n,\ \|\boldsymbol{x}\|_2 = 1\}),$$

where $\operatorname{conv}(S)$ and $\operatorname{cone}(S)$ stand for the convex hull and the conic hull of a set S, respectively. We also introduce the notation $\mathcal{C}_n^*$ for the set of *completely positive matrices* of size $n\times n$:

$$\mathcal{C}_n^* := \operatorname{conv}(\{\boldsymbol{x}\boldsymbol{x}^T : \boldsymbol{x}\in\mathbb{R}_+^n\}),$$

where $\mathbb{R}_+$ is the set of nonnegative real numbers. The space $\mathcal{S}_n$ is equipped with the inner product $\langle A,B\rangle := \operatorname{trace} AB$, and the associated Frobenius norm $\|X\|_F = \sqrt{\langle X,X\rangle} = \left(\sum_{i,j} X_{i,j}^2\right)^{1/2}$. The ith vector of the canonical basis of $\mathbb{R}^n$ is denoted by $\boldsymbol{e}_i$. The cardinality of S is denoted by $|S|$.

Let $G=(V,E)$ be a directed acyclic graph (DAG) with n arcs and m vertices. Let $s\in V$ and $t\in V$ denote two designated vertices of G, respectively called *source* and *sink*. Throughout we assume that s has no incoming arcs, t has no outgoing arcs, and for all $v\notin\{s,t\}$ there exists an (s,t) path that goes through v. We call a graph satisfying the above properties *a proper DAG*.

[1]Zuse Institute Berlin, Berlin, Germany.
[2]Technical University Berlin, Berlin, Germany.
[3]École Polytechnique, Palaiseau, France.

We denote the set of $(s,t)-$paths in G by $\mathcal{P}$. Unless expressly stated otherwise, the word "path" is always used to denote an $(s,t)-$path $P \in \mathcal{P}$.

We say that $\boldsymbol{f} \in \mathbb{R}_+^{|\mathcal{P}|}$ is a *$P-$flow* (for *path-based flow*) of value $u \geq 0$ if $\sum_{P\in\mathcal{P}} f_P = u$. A vector $\boldsymbol{x} \in \mathbb{R}_+^n$ is called an *$A-$flow* (for *arc-based flow*) of value $u \geq 0$, or simply a *flow of value* u when there is no ambiguity, if it satisfies the following flow conservation equations:

$$\forall v \in V, \sum_{e\in\delta^+(v)} x_e - \sum_{e\in\delta^-(v)} x_e = \begin{cases} -u & \text{if } v = s \\ u & \text{if } v = t \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The set of all A-flows of value u is denoted by $\mathcal{A}(u)$, and we use the notation $\mathcal{A} := \cup_{u\geq 0}\mathcal{A}(u)$ for the set of all $A-$flows (of any nonnegative value). Since G is a DAG, it is well known that $\mathcal{A}(1) = \text{conv}(\{\mathbf{1}_P : P \in \mathcal{P}\})$ and $\mathcal{A} = \text{cone}(\{\mathbf{1}_P : P \in \mathcal{P}\})$, where $\mathbf{1}_P$ is the incidence vector of the path P, that is, the elements of $\mathbf{1}_P \in \{0,1\}^{|E|}$ satisfy $(\mathbf{1}_P)_e = 1$ if $e \in P$ and $(\mathbf{1}_P)_e = 0$ otherwise. This comes from the fact that every $A-$flow $\boldsymbol{x} \in \mathcal{A}$ can be decomposed as $\boldsymbol{x} = \sum_{P\in\mathcal{P}} f_P \mathbf{1}_P$ for some $\boldsymbol{f} \in \mathbb{R}_+^{|\mathcal{P}|}$; see, e.g. [AMO93]. Note that $x_e = \sum_{P\ni e} f_P$ represents the amount of flow that goes through arc e (the summation indexed by "$P \ni e$" goes over all paths $P \in \mathcal{P}$ that include arc e).

Many optimization problems over graphs –such as the maximum flow problem or the minimum cost flow problem– can be solved using the arc-representation of a flow, which has the great advantage of being compact, while the number of paths might grow exponentially with the size of the graph. However, for some problems, some interactions exist between the arcs visited by the same "particle of flow". Such problems cannot be solved by the arc-representation, which just counts the amount of flow on each arc without tracking the path followed by each infinitesimal particle. This article presents a new scheme for deriving relaxations of certain flow problems, when the path of each individual particle in the graph matters.

In this paper, we propose to study the following polyhedral cone:

$$\mathcal{K} := \text{cone}(\{\mathbf{1}_P\mathbf{1}_P^T : P \in \mathcal{P}\}),$$

which we call *the cone of flow matrices*. If $X \in \mathcal{K}$, it can be decomposed as $X = \sum_{P\in\mathcal{P}} f_P \mathbf{1}_P\mathbf{1}_P^T$ for some flow $\boldsymbol{f} \in (\mathbb{R}_+)^{|\mathcal{P}|}$. The motivation for studying the cone $\mathcal{K}$ is that it introduces a certain amount of coupling between the arcs of a path. Indeed, $X_{ij} = \sum_{P\supseteq\{i,j\}} f_P$, is the amount of flow going through both arcs i and j. In particular, the vector of diagonal elements of X is $\text{diag}\, X = \sum_{P\in\mathcal{P}} f_P \mathbf{1}_P$, and corresponds to the standard arc-representation of the flow. As before, we denote by $\mathcal{K}(u)$ the set of flow matrices of value $u \geq 0$, that is,

$$\mathcal{K}(u) := \{X \in \mathcal{K} : \sum_{i\in\delta^+(s)} X_{ii} = u\}. \tag{2}$$

It is easy to see that $X = \sum_{P\in\mathcal{P}} f_P \mathbf{1}_P\mathbf{1}_P^T \in \mathcal{K}(u)$ if and only if $\sum_P f_P = u$. In particular,

$$\mathcal{K}(1) = \text{conv}(\{\mathbf{1}_P\mathbf{1}_P^T : P \in \mathcal{P}\}).$$

Related work We are not aware of any article that studied the cone $\mathcal{K}$ before. However, this paper is related to a series of results that use lift-and-project methods (see, e.g. [Lau03]) to obtain relaxations of hard combinatorial problems over graphs, or to reformulate them as completely positive programs, see [Dür10]. Such reformulations often rely on geometrical arguments, such as characterization of the extreme points of some polytopes. A particularity of the present article is that the proof of our completely positive result is purely combinatorial.

This paper is also related to several articles that study some flow- or path-optimization problems on directed graphs, with interactions between the arcs of each path. This is the case of

the NP-hard Shortest Path Problem with Forbidden Pairs (SPPFP) or the quadratic shortest path problem (QSPP) [RCH$^+$16], which play a central role in the present study, see Sections 2 and 3. We also mention the work of Baier et al. [BEH$^+$10] who study the problem of sending the maximal amount of flow on paths of bounded lengths, and show (among many other results) that this problem is NP-complete in general, but can be solved in polynomial time when the arcs are of unit length. For this problem, an arc-based representation of the flow cannot be used. In fact, the authors also show that unless P=NP, there is no polynomial-time algorithm that can transform an arc-representation of a length-bounded flow to a path-representation that only sends particles on paths of bounded length. A slightly more general version of this result was also discovered independently in [CSSM07], in the context of congestion games. Here, the cost of an arc is interpreted as the travel time (also called *latency*), and depends on the number of users that take this arc. The problem of computing a fair flow, that is, a flow of users that minimizes the maximal latency over the network, is shown to be NP-complete.

We also mention some work about the set of $\{0,1\}-$completely positive matrices, that is, the set of integer-valued matrices $X \in \mathbb{Z}^{n\times n}$ that admit a factorization of the form $X = UU^T$, where $U \in \{0,1\}^{n\times r}$. The set of $\{0,1\}$-completely positive matrices was first studied in [BX05], and the problem of computing a decomposition $X = UU^T$ is studied in [BR06]. This problem is very close to the problem of decomposing an integer-valued flow matrix as $X = \sum_{k=1}^{r} \mathbf{1}_{P_k}\mathbf{1}_{P_k}^T$, except that we have an additional restriction that the columns of U must form an $(s,t)-$path in a given graph.

Organization and contribution We will see in this article that several hard combinatorial problems can be formulated as linear optimization problems over $\mathcal{K}$. A important example is the quadratic shortest path problem (cf. Section 2), which appears naturally in the definition of the dual cone $\mathcal{K}^*$, see Section 3. The main result of this section (Theorem (3.5)) states that the problem of deciding whether a matrix X belongs to the cone $\mathcal{K}$ is NP-complete. An intermediate result, which we need to project $\mathcal{K}$ onto its affine hull, is also interesting *per se*: A symmetric matrix $X \in \mathcal{S}_n$ is called a *signed flow matrix* if it satisfies certain equality constraints (for example, the columns of X must be flow vectors, but there is no nonnegativity constraints on the elements of X). We give a polytime algorithm that decomposes a signed flow matrix X as $X = \sum_P f_P \mathbf{1}_P\mathbf{1}_P^T$, where the f_P are real numbers of arbitrary signs.

Tractable approximations of $\mathcal{K}$ yield relaxations for linear optimization problems over $\mathcal{K}$. We propose two convergent approximation hierarchies in Section 4. The second one is based on a completely positive representation of $\mathcal{K}$, as the intersection of $\mathcal{C}_n^*$ and a compact polyhedral cone. An alternative completely positive representation of $\mathcal{K}$ could also have been derived from a general result of Burer [Bur09]. We compare the two representations in Section 4.3, and show that our characterization of $\mathcal{K}$ leads to tighter bounds. Then, we show in Section 5 that the results of Section 4 do not hold anymore if we remove the assumption that the graph G is acyclic. The quality of our relaxations is evaluated for instances of the quadratic shortest path problem (QSPP) and for the *maximum flow problem with pairwise arc-capacities* in Section 6. Finally, we mention some other potential applications of the proposed framework and further perspectives in Section 7.

2 The quadratic shortest path problem

Given a cost vector $\boldsymbol{c} \in \mathbb{R}^n$, where c_a is the cost of arc a, the shortest path problem is to find the path $P \in \mathcal{P}$ minimizing $\sum_{a\in P} c_a = \boldsymbol{c}^T\mathbf{1}_P$. Since the graph G is a DAG, it is well known that this problem can be solved efficiently by dynamic programming, even if $\boldsymbol{c}$ has some negative components.

Analogously, assume there is a cost $Q_{i,j}$ if one chooses a path going through both i and j. This is the quadratic shortest path problem (QSPP):

$$\mathrm{qspl}(Q) = \min_{P\in\mathcal{P}} \mathbf{1}_P^T Q \mathbf{1}_P, \tag{3}$$

where the symbol qspl(Q) stands for the *quadratic shortest path length* associated with the cost matrix Q. The QSPP was recently shown to be NP-hard to approximate, and APX-hard in the special case where the cost matrix Q is positive semidefinite [RCH$^+$16]. Here, we point out that the authors of [RCH$^+$16] study a slightly different version of the QSPP in which there is a separate linear cost $\boldsymbol{c}^T\mathbf{1}_P$ in the objective function. However, one can always assume that the QSPP is of the form (3), by putting all linear costs on the diagonal of the matrix Q. Now, we claim that the QSPP can be formulated as a linear optimization problem over $\mathcal{K}(1)$:

$$\text{qspl}(Q) = \min_{X \in \mathcal{K}(1)} \langle X, Q \rangle. \tag{4}$$

Indeed, for a feasible $X = \sum_{P \in \mathcal{P}} f_P \mathbf{1}_P \mathbf{1}_P^T$, with $\sum_P f_P = 1$, it is straightforward that $\langle X, Q \rangle = \sum_{P \in \mathcal{P}} f_P \mathbf{1}_P^T Q \mathbf{1}_P$, and so the optimal solution only gives a positive weight $f_P > 0$ to paths solving the QSPP.

The above example already shows that, unless P=NP, we have no hope of finding a compact representation of the cone $\mathcal{K}$ (i.e., a description of $\mathcal{K}$ relying on a polynomial number of linear inequalities). Otherwise, the QSPP could be solved in polynomial time by linear programming (LP). In the next section, we show an even stronger negative result: it is NP-hard to check whether a given symmetric matrix X belongs to $\mathcal{K}$.

3 The membership problem

We study the following question:

$$\text{MEM}(\mathcal{K}) : \textit{Given } X \in \mathcal{S}_n, \textit{ does } X \textit{ belong to } \mathcal{K}?$$

A certificate for the membership $X \in \mathcal{K}$ can be given as a decomposition of the form $X = \sum_P f_p \mathbf{1}_P \mathbf{1}_P^T$. Note that from Carathéodory's theorem (see, e.g. [HUL12]), there always exists such a decomposition involving no more than $\frac{n(n+1)}{2} + 1$ paths. This already shows that the membership problem for $\mathcal{K}$ is in NP.

Below, we will show that the problem MEM($\mathcal{K}$) is NP-complete, by reasoning on the dual cone $\mathcal{K}^*$ of $\mathcal{K}$. In fact, we also have a direct proof of this result not relying on $\mathcal{K}^*$, but the proof we present here is shorter, and we think it sheds more light on the problem. We first show that $\mathcal{K}^*$ is the set of cost matrices for which the quadratic shortest path length is nonnegative. By definition,

$$\begin{aligned}\mathcal{K}^* = \{Y \in \mathcal{S}_n \,|\, \forall X \in \mathcal{K},\ \langle X, Y \rangle \geq 0\} &= \{Y \in \mathcal{S}_n \,|\, \forall p \in \mathcal{P},\ \mathbf{1}_P^T Y \mathbf{1}_P \geq 0\} \\ &= \{Y \in \mathcal{S}_n \mid \text{qspl}(Y) \geq 0\}.\end{aligned}$$

We will now show that the weak membership problem for $\mathcal{K}^*$ is NP-hard. The weak membership problem WMEM(S), where $S \subset \mathbb{R}^N$ has nonempty interior, is defined as follows:

$$\begin{aligned}\text{WMEM}(S) : \ \textit{Given } \boldsymbol{x} \in \mathbb{R}^N \textit{ and } \epsilon > 0, \textit{assert either } (i)\ B(\boldsymbol{x}, \epsilon) \cap S \neq \emptyset; \\ \textit{or } (ii)\ B(\boldsymbol{x}, \epsilon) \not\subseteq S,\end{aligned}$$

where $B(\boldsymbol{x}, \epsilon) := \{\boldsymbol{z} \in \mathbb{R}^n : \|\boldsymbol{z} - \boldsymbol{x}\| \leq \epsilon\}$. Note that both conditions (i) and (ii) may be valid for points $\boldsymbol{x}$ that are close to the boundary of S. In particular, $\boldsymbol{x} \in S$ implies (i) and $\boldsymbol{x} \notin S$ implies (ii). It follows that any algorithm that solves MEM(S) also solves WMEM(S); cf. [DG14]. Therefore, the NP-hardness of WMEM(S) implies that MEM(S) is also NP-hard.

Proposition 3.1. *The weak membership problem* WMEM($\mathcal{K}^*$) *is NP-hard.*

Proof. The convex cone $\mathcal{K}$ is clearly closed, and pointed (it contains no line, i.e., $X \in \mathcal{K}, -X \in \mathcal{K} \implies X = 0$) because elements of $\mathcal{K}$ only have nonnegative components. Therefore, $\mathcal{K}^*$ has a nonempty interior (see e.g. [BV04]).

Now, consider the problem WMEM($\mathcal{K}^*$). Asserting that $(X, \epsilon) \in \mathcal{S}_n \times \mathbb{R}_{++}$ satisfies condition (i) means that there exists a matrix $Y \in \mathcal{S}_n$ such that $\|Y - X\|_F \leq \epsilon$ and $\forall P \in \mathcal{P},\ \mathbf{1}_P^T Y \mathbf{1}_P \geq 0$. Hence,

$$\forall P \in \mathcal{P}, \quad \mathbf{1}_P^T X \mathbf{1}_P \geq \mathbf{1}_P^T (X - Y) \mathbf{1}_P \geq -\sum_{i,j} |X_{ij} - Y_{ij}| \geq -n\|X - Y\|_F,$$

where the last inequality follows from Cauchy-Schwarz. So if (X, ϵ) satisfies condition (i), we must have $\mathrm{qspl}(X) \geq -n\epsilon$. Using an analogous reasoning, we can show that if (X, ϵ) satisfies condition (ii), we must have $\mathrm{qspl}(X) < n\epsilon$.

Now, we use a result from [RCH+16]. The authors give a polynomial reduction from the *path with forbidden pairs* problem (PFPP), which is known to be NP-complete [GMO76], to the QSPP: given an instance $\mathcal{I}$ of the PFPP, a matrix $Q_{\mathcal{I}}$ is constructed (in polynomial time) in such a manner that $\mathcal{I}$ is a yes-instance if and only if $\mathrm{qspl}(Q_{\mathcal{I}}) = 0$, and $\mathcal{I}$ is a no-instance if and only if $\mathrm{qspl}(Q_{\mathcal{I}}) \geq 2$. By adding an arc of cost -1 after t, we obtain an instance $Q'_{\mathcal{I}}$ of the QSPP such that $\mathcal{I}$ is a yes-instance (no-instance) if and only if $\mathrm{qspl}(Q'_{\mathcal{I}}) = -1$ (≥ 1). According to the discussion above, $(X, \epsilon) = (Q'_{\mathcal{I}}, \frac{1}{2n})$ satisfies condition (i) for WMEM($\mathcal{K}^*$) if and only if $\mathcal{I}$ is a no-instance, and it satisfies condition (ii) if and only if $\mathcal{I}$ is a yes-instance. This shows that WMEM($\mathcal{K}^*$) is NP-hard. □

It is shown in [FL16, Theorem 5.3] that if K is a proper cone (i.e., closed, convex, pointed, and with nonempty interior), then the WMEM problem for K^* is polynomial-time reducible to the WMEM problem for K. In our case, $\mathcal{K}$ is not proper, because it has an empty interior. This means that the problem WMEM($\mathcal{K}$) is ill-defined. Indeed, since $\mathcal{K}$ has an empty interior, every $X \in \mathcal{S}_n$ satisfies condition (*ii*): $\exists X' \in B(X, \epsilon) : X' \notin \mathcal{K}$. So we must reason *relatively to the affine hull* of $\mathcal{K}$, which, since $0 \in \mathcal{K}$, coincides with the linear envelope

$$\operatorname{span} \mathcal{K} = \{ \sum_{P \in \mathcal{P}} f_p \mathbf{1}_P \mathbf{1}_P^T :\ \forall P \in \mathcal{P}, f_P \in \mathbb{R}\}.$$

A variant of the WMEM problem for the case of a set $S \in \mathbb{R}^N$ contained in an affine space of dimension less than N is proposed in [FL16]: Let H denote the affine hull of S, that is, the smallest affine subspace of $\mathbb{R}^N$ that contains S. The *weak membership problem for S relatively to H* is

$$\begin{aligned}\mathrm{WMEM}(S; H):\ \textit{Given } \boldsymbol{x} \in \mathbb{R}^N \textit{ and } \epsilon > 0, \textit{assert either } (i)\ (\boldsymbol{x} \in H) \wedge \big(B(\boldsymbol{x}, \epsilon) \cap S \neq \emptyset\big);\\ \textit{or } (ii)\ (\boldsymbol{x} \notin H) \vee \big(B(\boldsymbol{x}, \epsilon) \cap H \nsubseteq S\big).\end{aligned}$$

We will now state an extension of the result of Friedland and Lim [FL16, Theorem 5.3], for the case in which K is not full dimensional, but a basis for the linear envelope of K is given. Then, we will give a compact description of the linear space $\operatorname{span} \mathcal{K}$, which will allow us to conclude that MEM($\mathcal{K}$) is NP-complete.

Theorem 3.2. *Let $K \subseteq \mathbb{R}^N$ be a closed, convex, pointed cone, and let H denote the linear envelope of K. If an orthonormal basis $U = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r] \in \mathbb{R}^{N \times r}$ of H is known, then* WMEM(K^*) *is polynomial-time reducible to* WMEM($K; H$).

Proof. We first observe that K^* has a nonempty interior (because K is pointed), so WMEM(K^*) is well-defined.

Let us define the following cone: $K' := \{\boldsymbol{x}' \in \mathbb{R}^r : U\boldsymbol{x}' \in K\}$. We will start by proving a few properties on K' and its dual cone K'^*:

(P1) K' is proper

(P2) $K' = \{U^T \boldsymbol{x} : \boldsymbol{x} \in K\}$

(P3) $K^* = \{\boldsymbol{y} : U^T \boldsymbol{y} \in K'^*\}$

(P4) $K^* = UK'^* \oplus \ker U^T = \{U\boldsymbol{y}' + \boldsymbol{z} : \quad \boldsymbol{y}' \in K'^*,\ U^T\boldsymbol{z} = \boldsymbol{0}\}$

For the property (P1), note that K' is closed, convex, and full-dimensional by construction. Moreover it is pointed. Indeed, if $\boldsymbol{x}' \in K'$ and $-\boldsymbol{x}' \in K'$,then $U\boldsymbol{x}' \in K$ and $-U\boldsymbol{x}' \in K$, so $U\boldsymbol{x}' = \boldsymbol{0}$ because K is pointed, and $\boldsymbol{x}' = \boldsymbol{0}$ because U has full-rank.

To prove Property (P2), let $\boldsymbol{x}' \in K'$. By definition, $\boldsymbol{x} := U\boldsymbol{x}' \in K$. So we have $\boldsymbol{x}' = U^TU\boldsymbol{x}' = U^T\boldsymbol{x}$. Conversely, if $\boldsymbol{x}' = U^T\boldsymbol{x}$ for some $\boldsymbol{x} \in K$, then $U\boldsymbol{x}' = UU^T\boldsymbol{x}$. But, UU^T is the orthogonal projector over H, and $\boldsymbol{x} \in H$, so $U\boldsymbol{x}' = UU^T\boldsymbol{x} = \boldsymbol{x}$.

For (P3), note that $\boldsymbol{y} \in K^*$ if and only if

$$\begin{aligned} &\forall \boldsymbol{x} \in K, \quad \boldsymbol{y}^T\boldsymbol{x} = 0 \\ \iff &\forall \boldsymbol{x} \in K, \quad \boldsymbol{y}^TUU^T\boldsymbol{x} = 0 \\ \iff &\forall \boldsymbol{x}' \in K', \quad \boldsymbol{y}^TU\boldsymbol{x}' = 0, \end{aligned}$$

where the last equivalence comes from Property (P2), and means that $U^T\boldsymbol{y}$ is in the dual cone of K'.

It remains to prove Property (P4), where the symbol $\oplus$ denotes a direct sum, which means that the decomposition of some $\boldsymbol{y} \in K^*$ as $U\boldsymbol{y}' + \boldsymbol{z}$ with $\boldsymbol{y}' \in K'^*$ and $\boldsymbol{z} \in \ker U^T$ is unique. Note that the unicity of the decomposition follows from the fact that the nullspace of U^T and the range of U are orthogonal.

Let $\boldsymbol{y} = U\boldsymbol{y}' + \boldsymbol{z}$, with $\boldsymbol{y}'$ and $\boldsymbol{z}$ as above, and let $\boldsymbol{x} = UU^T\boldsymbol{x} \in K$. We have:

$$\boldsymbol{x}^T\boldsymbol{y} = \boldsymbol{x}^TU\boldsymbol{y}' + \boldsymbol{x}^TU\underbrace{U^T\boldsymbol{z}}_{=0} = (\underbrace{U^T\boldsymbol{x}}_{\in K'})^T\ \underbrace{\boldsymbol{y}'}_{\in K'^*} = 0.$$

Conversely, assume that $\boldsymbol{y} \in K^*$. We can decompose $\boldsymbol{y}$ on H and its orthogonal complement, as follows:

$$\boldsymbol{y} = UU^T\boldsymbol{y} + (I - UU^T)\boldsymbol{y}.$$

The vector $\boldsymbol{z} := (I - UU^T)\boldsymbol{y}$ is in $\ker U^T$ because $(I - UU^T)$ is the orthogonal projector over $\ker U^T$, and the vector $\boldsymbol{y}' := U^T\boldsymbol{y}$ is in K'^* by Property (P3).

To prove the theorem, we will show that, if the matrix U is given, then we have

$$\mathrm{WMEM}(K^*) \overset{(R1)}{\Longrightarrow} \mathrm{WMEM}(K'^*) \overset{(R2)}{\Longrightarrow} \mathrm{WMEM}(K') \overset{(R3)}{\Longrightarrow} \mathrm{WMEM}(K; H),$$

where $A \implies B$ means that A is polynomial-time reducible to B.

(R1). To show this reduction, we will show that

$$(\boldsymbol{y}, \epsilon) \text{ satisfies } (i) \text{ for } \mathrm{WMEM}(K^*) \iff (U^T\boldsymbol{y}, \epsilon) \text{ satisfies } (i) \text{ for } \mathrm{WMEM}(K'^*)$$

and

$$(\boldsymbol{y}, \epsilon) \text{ satisfies } (ii) \text{ for } \mathrm{WMEM}(K^*) \iff (U^T\boldsymbol{y}, \epsilon) \text{ satisfies } (ii) \text{ for } \mathrm{WMEM}(K'^*)$$

Let $\boldsymbol{y}_0 \in \mathbb{R}^N$ satisfy $\|\boldsymbol{y} - \boldsymbol{y}_0\| \leq \epsilon$. This implies that $\|U^T(\boldsymbol{y} - \boldsymbol{y}_0)\| \leq \epsilon$, because since the columns of U are orthonormal, the spectral norm of U^T is 1. By Property (P3), we have

$$\boldsymbol{y}_0 \in K^* \iff U^T\boldsymbol{y}_0 \in K'^*, \tag{5}$$

which proves the implications in the direction ($\Rightarrow$) for both the conditions (i) and (ii).

Conversely, let $\boldsymbol{y}'_0 \in \mathbb{R}^r$ be a vector such that $\|U^T\boldsymbol{y} - \boldsymbol{y}'_0\| \leq \epsilon$. We define $\boldsymbol{y}_0 := U\boldsymbol{y}'_0 + (I - UU^T)\boldsymbol{y}$, so that $\|\boldsymbol{y} - \boldsymbol{y}_0\| = \|U(U^T\boldsymbol{y} - \boldsymbol{y}'_0)\| = \|U^T\boldsymbol{y} - \boldsymbol{y}'_0\| \leq \epsilon$. Now, we claim that

$$\boldsymbol{y}_0 \in K^* \iff \boldsymbol{y}'_0 \in K'^*, \tag{6}$$

which will prove the implications in the direction ($\Leftarrow$) for both the conditions (i) and (ii). The claim is a direct consequence of Property (P4), because $\boldsymbol{z} := (I - UU^T)\boldsymbol{y} \in \ker U^T$, and the decomposition of the form $\boldsymbol{y}_0 = U\boldsymbol{y}'_0 + \boldsymbol{z}$ is unique.

(R2). Since K' is a proper cone (Property (P1)), this reduction follows from the aforementioned result of Friedland and Lim [FL16, Theorem 5.3].

(R3). To show this reduction, we will show that

$$(\boldsymbol{x'}, \epsilon) \text{ satisfies } (i) \text{ for WMEM}(K') \iff (U\boldsymbol{x}', \epsilon) \text{ satisfies } (i) \text{ for WMEM}(K; H)$$

and

$$(\boldsymbol{x'}, \epsilon) \text{ satisfies } (ii) \text{ for WMEM}(K') \iff (U\boldsymbol{x}', \epsilon) \text{ satisfies } (ii) \text{ for WMEM}(K; H).$$

Let $\boldsymbol{x}_0 \in H$ be a vector satisfying $\|U\boldsymbol{x}' - \boldsymbol{x}_0\| \leq \epsilon$. We have $\boldsymbol{x}_0 = U\boldsymbol{x}'_0$ for some vector $\boldsymbol{x}'_0 \in \mathbb{R}^r$. So, $\|U\boldsymbol{x}' - \boldsymbol{x}_0\| = \|U(\boldsymbol{x}' - \boldsymbol{x}'_0)\| = \|\boldsymbol{x}' - \boldsymbol{x}'_0\| \leq \epsilon$. We now use the fact that

$$\boldsymbol{x}'_0 \in K' \iff \boldsymbol{x}_0 = U\boldsymbol{x}'_0 \in K, \tag{7}$$

which simply follows from the definition of K'. This proves the implications in the direction ($\Leftarrow$) for both the conditions (i) and (ii).

Conversely, let $\boldsymbol{x_0}' \in \mathbb{R}^r$ be a vector such that $\|\boldsymbol{x}' - \boldsymbol{x}'_0\| \leq \epsilon$. We have $U\boldsymbol{x}' \in H$, $U\boldsymbol{x_0}' \in H$, and $\|U\boldsymbol{x}' - U\boldsymbol{x}'_0\| \leq \epsilon$. So we use again the property of Equation (7), which proves the implications in the direction ($\Rightarrow$) for both the conditions (i) and (ii).

□

Now, we give a nice description of the subspace

$$\operatorname{span} \mathcal{K} = \{ \sum_{P \in \mathcal{P}} f_p \mathbf{1}_P \mathbf{1}_P^T : \quad \forall P \in \mathcal{P}, f_P \in \mathbb{R} \}.$$

This will allow us to construct a basis of this subspace, and the proof is also interesting *per se*: It is constructive and shows how to decompose a matrix from $\operatorname{span} \mathcal{K}$ as a *signed P-flow*, that is, a flow $\{f_P : P \in \mathcal{P}\}$ in which the flow on a path is not restricted to be nonnegative. This means that if an optimization problem can be formulated as a linear optimization problem over $\mathcal{K}$, the relaxation that one obtains by dropping the constraints $f_P \geq 0$ can be solved in polynomial time, by linear programming (LP).

Let $D \subseteq E \times E$ be the set of pairs of arcs (i, j) such that no path $P \in \mathcal{P}$ uses both i and j. Let $\mathcal{V}$ be the linear subspace defined by the set of equations

$$\mathcal{V} := \left\{ X \in \mathcal{S}_n : \begin{array}{ll} \forall v \notin \{s,t\}, & \sum_{e \in \delta^+(v)} X_{ee} - \sum_{e \in \delta^-(v)} X_{ee} = 0 \\ \forall v \notin \{s,t\}, \forall i \in E, & \sum_{e \in \delta^+(v)} X_{ie} - \sum_{e \in \delta^-(v)} X_{ie} = 0 \\ \forall i, j \in D, & X_{i,j} = 0. \end{array} \right\}.$$

Note that $\mathcal{V}$ is a linear space (in particular, the X_{ij}'s *are not* required to be nonnegative). Then we have:

Theorem 3.3. *The linear space spanned by the matrices $\mathbf{1}_P \mathbf{1}_P^T$'s ($\forall P \in \mathcal{P}$) is equal to $\mathcal{V}$:*

$$\operatorname{span} \mathcal{K} = \mathcal{V}.$$

Moreover, given a matrix $X \in \mathcal{V}$, a decomposition of the form $\sum_{P \in \hat{\mathcal{P}}} f_p \mathbf{1}_P \mathbf{1}_P^T$, where $\hat{\mathcal{P}}$ is a subset of $\mathcal{P}$, can be computed in polynomial time by Algorithm 1.

The inclusion $\operatorname{span} \mathcal{K} \subseteq \mathcal{V}$ is trivial, and we give a constructive proof of the reverse inclusion: It suffices to prove that Algorithm 1 terminates and returns the correct decomposition to prove the theorem.

To do so, we will need one technical lemma, which deals with the decomposition of a *signed flow vector*. It can be seen a variant of the flow decomposition theorem, when flow-values are allowed to be negative in a DAG. The lemma is proved in the appendix.

Lemma 3.4. *Let $\boldsymbol{x} \in \mathbb{R}^n$ be a* signed flow vector*, i.e., a vector satisfying*

$$\forall v \notin \{s,t\}, \quad \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = 0.$$

Then, we can compute in polynomial time a decomposition $\boldsymbol{x} = \sum_{i=1}^{k} f_i \mathbf{1}_{P_i}$, where $\forall i,\ P_i \in \mathcal{P}, f_i \in \mathbb{R}$.

We can now prove the correctness of Algorithm 1:

Proof of Theorem 3.3. By construction, the decomposition returned by the algorithm necessarily satisfies $X = \sum_i f_i \mathbf{1}_{P_i} \mathbf{1}_{P_i}^T$. So we must only show that the algorithm terminates.

We first admit that the iterations are well defined, and we claim that at the beginning of each iteration, the matrix X_0 and the graph G_0 satisfy the following three properties:

(a) $X_0 \in \mathcal{V}$.

(b) $j \notin E_0 \implies X_0 \boldsymbol{e}_j = \mathbf{0}$, where E_0 is the set of arcs of G_0: the jth column of X_0 is $\mathbf{0}$ whenever arc j has already been removed.

(c) If there exists no (s,t)-path in G_0 using both arcs i and j, then $(X_0)_{ij} = 0$.

These properties clearly hold at the beginning of the algorithm, by definition of $\mathcal{V}$. So we must show that the iterations –assuming they are well defined– produce an updated matrix $X_0^+ := X_0 - \sum_{k=1}^{r} f_k \mathbf{1}_{P_k} \mathbf{1}_{P_k}^T$ and an updated graph G_0^+ which still satisfy the properties (a)-(c). Property (a) follows from the fact that at each iteration, X_0 is updated by a linear combination of some $\mathbf{1}_P \mathbf{1}_P^T$'s, and it is clear that each matrix $\mathbf{1}_P \mathbf{1}_P^T$ is an element of $\mathcal{V}$.

For (b), note that the paths P_k's involved during some iteration only use the remaining arcs E_0 of G_0. So the jth column of $\mathbf{1}_{P_k} \mathbf{1}_{P_k}^T$ is $\mathbf{0}$ for all $j \notin E_0$, and the columns of X_0^+ corresponding to arcs already removed from G_0 will remain equal to $\mathbf{0}$ after the update. Now, we look at the value of the column a after the update (where a is the arc selected during the iteration):

$$X_0^+ \boldsymbol{e}_a = (X_0 - \sum_{k=1}^{r} f_k \mathbf{1}_{P_k} \mathbf{1}_{P_k}^T) \boldsymbol{e}_a = \boldsymbol{x} - \sum_{k=1}^{r} f_e \mathbf{1}_{P_k} (\mathbf{1}_{P_k}^T \boldsymbol{e}_a) = \boldsymbol{x} - \sum_{\substack{k \in \{1,\dots,r\} \\ P_k \ni a}} f_k \mathbf{1}_{P_k} = \mathbf{0}$$

where the last equality comes from the fact that each P_k uses arc a, and $\sum_{k=1}^{r} f_k \mathbf{1}_{P_k}$ is a decomposition of the signed flow vector $\boldsymbol{x} = X_0 \boldsymbol{e}_a$. Similarly the column e of X_0^+, where e is an arc such that all paths using a also use e, is equal to $\mathbf{0}$. Indeed, we know that $(X_0^+)_{ia} = 0$ for all $i \in E$, and the vector $\{(X_0^+)_{ij} | j \in E\}$ is a signed flow vector (by property (a)), so in particular $(X_0^+)_{ia} = (X_0^+)_{ie} = 0$. Hence, at the end of an iteration, all columns of $(X_0)^+$ corresponding to arcs not present in G_0^+ are equal to $\mathbf{0}$.

For (c), let v be the vertex returned by the BFS, and let u be one of its direct predecessors. By construction there is a single path from s to u, and either there are at least two paths from s to v, or $v = t$. Let i and j denote two arcs such that there is no path using both i and j in G_0^+. The updated value of $(X_0)_{ij}$ is

$$(X_0^+)_{ij} = (X_0)_{ij} - \sum_{\substack{k \in \{1,\dots,r\} \\ P_k \supseteq i,j}} f_k \tag{8}$$

If there was already no path using i and j in G_0, then we had $(X_0)_{ij} = 0$, and $(X_0^+)_{ij}$ remains equal to 0. Otherwise, all paths using the arcs i and j were also using arc a. Let us assume (without loss of generality) that $i \prec j$ for some topological ordering $\prec$ of the arcs of G_0. We claim that we must have $i \prec a$. Indeed, the case $a \prec i \prec j$ can be excluded. This is clear if $v = t$,

Algorithm 1 Signed flow matrix decomposition

Input: $X \in \mathcal{V}$

Output: A set of paths $P_1, \ldots, P_\ell \in \mathcal{P}$ and real numbers $f_1, \ldots, f_\ell$ such that $X = \sum_k f_k \mathbf{1}_{P_k} \mathbf{1}_{P_k}^T$.

1: $G_0 \leftarrow G$
2: $X_0 \leftarrow X$
3: Initialize an empty decomposition of X.
4: **while** $X_0 \neq 0$ **do**
5: ▷ Perform a BFS (Breadth First Search) of the graph G_0, starting at s, until a vertex v is found such that, either $v = t$, or v has already been visited by the BFS.
6: ▷ $a \leftarrow (u, v)$, where u is a direct predecessor of v
7: ▷ Let $\boldsymbol{x} := X_0 \boldsymbol{e}_a$ denote the ath column of X_0
8: ▷ Decompose the vector $\boldsymbol{x}$, which is a signed flow vector, as

$$\boldsymbol{x} = \sum_{k=1}^{r} f_k \mathbf{1}_{P_k},$$

where each P_k is an $(s,t)-$path in G_0 that goes through arc a.
9: ▷ Append (P_k, f_k) $(\forall k = 1, \ldots, r)$ to the decomposition of X
10: ▷ $X_0 \leftarrow X_0 - \sum_{k=1}^{r} f_k \mathbf{1}_{P_k} \mathbf{1}_{P_k}^T$
11: ▷ Remove from the graph G_0 all the arcs $e \in E$ such that $\forall P \in \mathcal{P}, (e \in P \iff a \in P)$
12: **end while**
13: **return** the decomposition of X

and otherwise there are at least two paths from s to v, so the removal of a would leave at least one path in G_0^+ using both i and j.

Now, $i \prec a$ implies that i is located on the single path from s to u, so each P_k must use the arc i. Thus we have

$$\sum_{\substack{k \in \{1,\ldots,r\} \\ P_k \supseteq i,j}} f_k = \sum_{\substack{k \in \{1,\ldots,r\} \\ P_k \ni j}} f_k = x_j = (X_0)_{ja}, \tag{9}$$

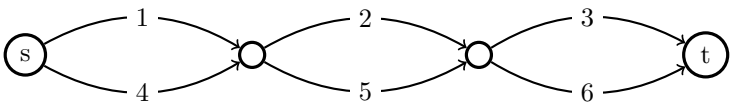
because the (f_k, P_k)'s are a decomposition of the signed flow $\boldsymbol{x} = X_0 \boldsymbol{e}_a$. Finally, we use the fact that $\{(X_0)_{je} | e \in E\}$ is a signed flow, supported by the paths of G_0 that use arc j. Since all paths of G_0 using both i and j also use the arc a, this implies that $(X_0)_{ja} = (X_0)_{ji} = (X_0)_{ij}$. Combining this with the equations (8) and (9), we get: $(X_0^+)_{ij} = 0$.

With the above properties, it is easy to show that the algorithm terminates. Indeed, at least one arc is removed at each iteration, so after at most n iterations the graph G_0 is empty, and by property (b) we must have $X_0 = 0$. It remains to show that the iterations are well defined. Indeed, we must show how to compute the decomposition of line 8.

This decomposition follows from Lemma 3.4, but we must show that it is possible to decompose the signed flow vector $\boldsymbol{x}$ by using paths of G_0 that all go through the selected arc a. Let G' be the directed graph obtained by keeping only arcs $e \in E' := \{e \in E : x_e = (X_0)_{ae} \neq 0\}$. By property (c), G' is the union of all $(s,t)-$paths of G_0 that use arc a. Hence, G' is a proper DAG, and it is easy to see that all (s,t)-paths in G' must use arc a. We can now apply Lemma 3.4 to get a decomposition of the signed flow vector $\boldsymbol{x}' := \{x_e | e \in E'\}$ over the graph G'. This decomposition has the required property: all paths of the decomposition go through a, and use edges of $E' \subseteq E_0$ only. □

We can finally prove the main result of this section:

Theorem 3.5. *The membership problem* MEM($\mathcal{K}$) *is NP-complete.*



$$X = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Figure 1: Example of a matrix $X \in \mathcal{K}_2 \setminus \mathcal{K}$, with the corresponding graph G. We have $X \notin \mathcal{K}$, because $X_{1,1} = X_{1,2} = X_{1,3} = 1$ implies that all the infinitesimal particles of flow going through arc 1 also pass through arcs 2 and 3, but this contradicts $X_{2,3} = 0$.

Proof. We already know that MEM($\mathcal{K}$) is in NP, so it suffices to prove that MEM($\mathcal{K}$) is NP-hard.

Theorem 3.3 tells us that the linear envelope $H = \operatorname{span} \mathcal{K}$ of $\mathcal{K}$ is equal to the set of solutions of a system of $\mathcal{O}(n^2)$ equations. We can hence construct (in polynomial time) an orthonormal basis $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$ for $H = \mathcal{V}$, which we rearrange to form a $n^2 \times r$ matrix U with orthonormal columns. Then, by Theorem 3.2, we know that WMEM($\mathcal{K}, H$) is NP-hard. Finally, MEM($\mathcal{K}$) is NP-hard, because for all $\epsilon > 0$, $X \in \mathcal{K}$ implies that (X, ϵ) satisfies condition (i) of WMEM($\mathcal{K}, H$), and $X \notin \mathcal{K}$ implies that (X, ϵ) satisfies condition (ii).

□

4 Approximation Hierarchies

4.1 A Tensor-based hierarchy

Let $X \in \mathcal{K}$. We already observed that $\operatorname{diag} X$ is a flow, i.e., $\operatorname{diag} X \in \mathcal{A}$. It is easy to see that columns of X are flows, too: if $X = \sum_{P \in \mathcal{P}} f_P \mathbf{1}_P \mathbf{1}_P^T$, then $X\boldsymbol{e}_i = [X_{1i}, X_{2i}, \ldots, X_{ni}]^T = \sum_{P \ni i} f_P \mathbf{1}_P$ is the *subflow of all particles that go through arc i*, which is a flow of value $\sum_{P \ni i} f_P = X_{ii}$. Hence,

$$\mathcal{K} \subseteq \mathcal{K}_2 := \{X \in \mathcal{S}_n \,:\, \operatorname{diag} X \in \mathcal{A}, \quad X\boldsymbol{e}_i \ \in \mathcal{A}(X_{ii}),\ \forall i \in \{1, \ldots, n\}\}\,.$$

In words, $\mathcal{K}_2$ is the polyhedron containing all symmetric matrices such that the diagonal is a flow, and the ith column is a flow of value X_{ii}. Of course, the converse inclusion does not hold, since we know that $\mathcal{K}$ does not admit a compact description (unless $P = NP$). A counter-example is depicted in Figure 1.

For $u \geq 0$ we also introduce the set $\mathcal{K}_2(u)$, which consists of all matrices $X \in \mathcal{K}_2$ such that the diagonal of X is a flow of value $u \geq 0$:

$$\mathcal{K}(u) \subseteq \mathcal{K}_2(u) := \left\{ X \in \mathcal{K}_2 : \sum_{i \in \delta^+(s)} X_{ii} = u \right\}.$$

The above inclusion already shows how to construct one relaxation of optimization problems over $\mathcal{K}$. For example, a lower bound for the QSPP (see Section 2) can be computed by replacing the constraint $X \in \mathcal{K}(1)$ (which is polyhedral but involves one variable for each path $P \in \mathcal{P}$) in Problem (4) by the compact constraint $X \in \mathcal{K}_2(1)$. It is also easy to see that a matrix X is of the form $\mathbf{1}_P \mathbf{1}_P^T$ for some path $P \in \mathcal{P}$ if and only if

$$X \in \mathcal{K}_2^{\text{MIP}} = \big\{X \in \mathcal{K}_2 : \ \ \forall i \in \{1, \ldots, n\},\ X_{ii} \in \{0, 1\}\big\}.$$

Therefore, the QSPP can be reformulated as the following mixed integer programming (MIP) problem:

$$\begin{aligned} \mathrm{qspl}(Q) = \min \quad & \langle Q, X \rangle \\ s.t. \quad & X \in \mathcal{K}_2^{\mathrm{MIP}}. \end{aligned} \tag{10}$$

More generally, we can extend this approach by considering the equalities that must be satisfied by the tensor $T^{\boldsymbol{f}} = \{T^{\boldsymbol{f}}_{i_1,\ldots,i_k}\}_{1\le i_1,\ldots,i_k\le n}$, where $T^{\boldsymbol{f}}_{i_1,\ldots,i_k} = \sum_{P\supseteq\{i_1,\ldots,i_k\}} f_P$ is the amount of flow using all arcs from the set $\{i_1, i_2, \ldots, i_k\}$. By definition, $T^{\boldsymbol{f}}$ belongs to the set $\mathbb{T}_{n^k} \subset \mathbb{R}^{n\times\cdots\times n}$ of *setwise symmetric tensors*, that satisfy the property that $T_{i_1,\ldots,i_k}$ only depends on its set of indices $\{i_1, \ldots, i_k\}$ (e.g., for $k = 3$ we have $T_{ijj} = T_{iij}$). For ease of notation, we can hence index the elements of $T \in \mathbb{T}_{n^k}$ as T_J, where J belongs to the set $\mathcal{I}_k$ of all nonempty subsets of $\{1, \ldots, n\}$ with at most k elements:

$$\mathcal{I}_k := \{J \subseteq \{1, \ldots, n\} : 1 \le |J| \le k\}.$$

For $T \in \mathbb{T}_{n^k}$ and $J \in \mathcal{I}_{k-1}$, we further introduce the notation

$$\begin{aligned} \operatorname{diag} T &= [T_{\{1\}}, \ldots, T_{\{n\}}]^T \in \mathbb{R}^n \\ \operatorname{mat} T &= \{T_{\{ij\}}\}_{1\le i,j\le n} \in \mathcal{S}_n \\ \operatorname{beam}_J(T) &= [T_{J\cup\{1\}}, \ldots, T_{J\cup\{n\}}]^T \in \mathbb{R}^n. \end{aligned}$$

Now, for $k = 1, 2, \ldots$, we construct the following sets:

$$\mathcal{T}_k = \{T \in \mathbb{T}_{n^k} : \operatorname{diag} T \in \mathcal{A}, \quad \forall J \in \mathcal{I}_{k-1},\ \operatorname{beam}_J(T) \in \mathcal{A}(T_J)\}.$$

Note that $\mathcal{T}_1$ is isomorphic to $\mathcal{A}$ and $\mathcal{T}_2$ is isomorphic to $\mathcal{K}_2$. By construction, if $X = \sum_{P\in\mathcal{P}} f_P \mathbf{1}_P \mathbf{1}_P^T \in \mathcal{K}$, then the tensor

$$T^{\boldsymbol{f}} = \sum_{P\in\mathcal{P}} f_P \underbrace{\mathbf{1}_P \otimes \cdots \otimes \mathbf{1}_P}_{k \text{ times}}$$

is such that $T^{\boldsymbol{f}} \in \mathcal{T}_k$, and $\operatorname{mat} T^{\boldsymbol{f}} = X$. Hence, for all $k \ge 2$ we have

$$\mathcal{K} \subseteq \mathcal{K}_k := \{\operatorname{mat} T : T \in \mathcal{T}_k\}.$$

For some $k \ge 2$, let $T \in \mathcal{T}_{k+1}$ and define $T' = \{T_J\}_{J\in\mathcal{I}_k} \in \mathbb{T}_{n^k}$. We clearly have $T' \in \mathcal{T}_k$ and $\operatorname{mat} T' = \operatorname{mat} T$, which shows $\mathcal{K}_{k+1} \subseteq \mathcal{K}_k$. Hence, we have:

$$\mathcal{K} \subseteq \cdots \subseteq \mathcal{K}_3 \subseteq \mathcal{K}_2. \tag{11}$$

In fact, we can even show that the hierarchy converges after a finite number of steps: It holds that $\mathcal{K} = \mathcal{K}_N$, where N is the length of the longest $(s,t)-$path in G. We point out that $\mathcal{K}_k$ is defined by a set of $|\mathcal{I}_k| + 1$ flows on a graph with m vertices; its description involves $\mathcal{O}(mn^{k-1})$ linear equations on $\mathcal{O}(n^k)$ variables.

Proposition 4.1. *Let $N = \max\{|P| : P \in \mathcal{P}\}$ denote the length of the longest $(s,t)-$path in G. Then, we have the following approximation hierarchy for $\mathcal{K}$:*

$$\mathcal{K} = \mathcal{K}_N \subseteq \mathcal{K}_{N-1} \subseteq \cdots \subseteq \mathcal{K}_3 \subseteq \mathcal{K}_2.$$

Proof. The inclusions $\mathcal{K} \subseteq \mathcal{K}_{i+1} \subseteq \mathcal{K}_i$ hold for all $i \ge 2$, by construction. So the only thing to show is the inclusion $\mathcal{K}_N \subseteq \mathcal{K}$. Let $X \in \mathcal{K}_N$. By definition, there exists a tensor $T \in \mathcal{T}_N$ such that $X = \operatorname{mat} T$. For all $P \in \mathcal{P}$, P is a nonempty set of arcs of cardinality bounded by N, so $P \in \mathcal{I}_N$ and T_P is well defined.

We first need to prove an intermediate result: we claim that if J is a subset of $\{1,\dots,n\}$ such that no path contains all arcs of J, then $T_J = 0$. Let J be such a set of arcs. Then, there exist two arcs $i, j \in J$ such that $\forall P \in \mathcal{P},\ (i \notin P \text{ or } j \notin P)$. Now, let $S = J \setminus \{j\}$, and consider the vector $\boldsymbol{x} = \text{beam}_S(T)$, which must be a flow of value T_S. Moreover, we have $T_S \geq x_i + x_j$ (since no path traverses both i and j), with $x_i = T_{S\cup\{i\}} = T_S$ and $x_j = T_{S\cup\{j\}} = T_J$. This means: $T_S \geq T_S + T_J$, from which we deduce $T_J = 0$.

We will now show by induction on $k = N, N-1, \dots, 1$ that the equality

$$T_J = \sum_{P\in\mathcal{P}:P\supseteq J} T_P \tag{12}$$

holds for all $J \in \mathcal{I}_N$ of cardinality k. For $k = 2$, this property shows that $X \in \mathcal{K}$. We initiate the induction at $k = N$. Let J be a subset of $\{1,\dots,n\}$ of cardinality N. Then either J is a path, or there is no path P such that $J \subseteq P$. In both cases, the equality (12) holds.

Now, assume that the property holds for $k = j+1,\dots,N$, and let J be a subset of $\{1,\dots,n\}$ of cardinality j. We distinguish three cases: (i) if J is a path, then the sum on the right-hand-side of (12) reduces to T_J, so the equality holds; (ii) if no path contains J, then we know that $T_J = 0$, and the sum the right-hand-side of (12) goes over the empty set; (iii) Otherwise, J can be augmented to form a path. So there exists a vertex $v \neq s$ with a single predecessor in J and no successor in J, or a vertex $v \neq t$ with a single successor in J and no predecessor in J. Let us assume the former case (the second case can be handled similarly), and let $\delta^-(v) \cap J = \{i\}$, $\delta^+(v) \cap J = \emptyset$. The vector $\boldsymbol{x} = \text{beam}_J(T)$ is a flow, so $\sum_{e\in\delta^-(v)} x_e = \sum_{e\in\delta^+(v)} x_e$. There is no path using two arcs from $\delta^-(v)$, so $T_{J\cup\{e\}} = 0$ for $e \in \delta^-(v) \setminus \{i\}$, and the left hand side reduces to $T_{J\cup\{i\}} = T_J$. For an arc $e \in \delta^+(v)$, $J \cup \{e\}$ is of cardinality $j+1$, so we can use the induction hypothesis: $x_e = T_{J\cup\{e\}} = \sum_{P\in\mathcal{P}:P\supseteq J\cup\{e\}} T_P$. Finally, since no path can use two arcs of $\delta^+(v)$, the sets $\{P \in \mathcal{P} : P \supseteq J \cup \{e\}\}$ form a partition of $\{P \in \mathcal{P} : P \supseteq J\}$, and the right-hand side of the flow conservation at v becomes $\sum_{P\in\mathcal{P}:P\supseteq J} T_P$. □

4.2 A completely positive representation

Observe that all flow matrices $X \in \mathcal{K}$ are positive semidefinite by construction, and even completely positive because $\mathbf{1}_P\mathbf{1}_P^T \in \mathcal{C}_n^* \subset \mathcal{S}_n^*$. It follows that

$$\mathcal{K} \subseteq \mathcal{K}_2^* \subseteq \mathcal{K}_2^+ \subseteq \mathcal{K}_2$$

where we defined $\mathcal{K}_2^* := \mathcal{K}_2 \cap \mathcal{C}_n^*$ and $\mathcal{K}_2^+ := \mathcal{K}_2 \cap \mathcal{S}_n^+$. In fact, we next show that the first inclusion holds with equality.

Optimization problems over $\mathcal{C}_n^*$ are in general intractable, but the set of completely positive matrices can be approximated by simpler cones. In particular, there exist several inner and outer nested approximation hierarchies converging to $\mathcal{C}_n^*$, see e.g. [Dür10, Las14].

Theorem 4.2. *A symmetric matrix $X \in \mathcal{S}_n$ is a flow matrix if and only if it belongs to $\mathcal{K}_2$ and is completely positive, i.e., $\mathcal{K} = \mathcal{K}_2^*$.*

Proof. Let $X \in \mathcal{K}_2$, and assume that $X \in \mathcal{C}_n^*$, that is, $X = \sum_{k=1}^q (\boldsymbol{x}^k)(\boldsymbol{x}^k)^T$ for some vectors $\boldsymbol{x}^1,\dots,\boldsymbol{x}^q \in \mathbb{R}_+^n$. We are going to prove that $X \in \mathcal{K}$, which shows $\mathcal{K}_2^* = \mathcal{K}_2 \cap \mathcal{C}_n^* \subseteq \mathcal{K}$.

For all $(i,k) \in \{1,\dots,n\} \times \{1,\dots,q\}$, denote by x_i^k the i^{th} element of $\boldsymbol{x}^k$, and denote by $\boldsymbol{x}_i$ the vector of dimension q with elements $(x_i^1,\dots,x_i^q)$. Observe that $X_{ij} = \boldsymbol{x}_i^T\boldsymbol{x}_j = \sum_k x_i^k x_j^k$.

Consider a vertex $v \in V \setminus \{s\}$, and let i be an arc incident to v. We have $X \in \mathcal{K}_2$, so the i^{th} column of X is a flow of value X_{ii}, and the amount of this flow passing through vertex v cannot exceed X_{ii}:

$$\sum_{e\in\delta^-(v)} X_{ei} \leq X_{ii}.$$

Since $i \in \delta^-(v)$ and the X_{ei}'s are nonnegative, the inequality above must be an equality. Hence, $\sum_{e\in\delta^-(v)\setminus i}\sum_k x_e^k x_i^k = 0$. All terms of this sum are nonnegative, which implies that $x_i^k x_j^k = 0$

whenever two distinct arcs i and j are incident to the same vertex v. To summarize, for all k and for all $v \in V \setminus \{s\}$, there exists at most one arc $e \in \delta^-(v)$ such that $x_e^k > 0$. Similarly, for all $v \in V \setminus \{t\}$ there is at most one arc $e \in \delta^+(v)$ satisfying $x_e^k > 0$.

Now, consider a vertex $v \in V \setminus \{s,t\}$. We define $K^- = \left\{k \in \{1,\dots,q\} : \exists e \in \delta^-(v) : x_e^k > 0\right\}$ and $K^+ = \left\{k \in \{1,\dots,q\} : \exists e \in \delta^+(v) : x_e^k > 0\right\}$. For $k \in K^-$ ($k \in K^+$) we denote by i_k^- (i_k^+) the unique arc $e \in \delta^-(v)$ ($e \in \delta^+(v)$) such that $x_e^k > 0$. Let us write the flow conservation equation at v, for the flow corresponding to the i^{th} column of X:

$$\forall i \in [N], \quad \sum_{e \in \delta^-(v)} \sum_k x_e^k x_i^k = \sum_{e \in \delta^+(v)} \sum_k x_e^k x_i^k.$$

For each k, each sum over $e \in \delta^-(v)$ and $e \in \delta^+(v)$ has at most one nonzero term, so the equation simplifies to:

$$\forall i \in [N], \quad \sum_{k \in K^-} x_{i_k^-}^k x_i^k = \sum_{k \in K^+} x_{i_k^+}^k x_i^k. \tag{13}$$

Summing Eq. (13) over all $i \in \delta^-(v)$ (respectively over $i \in \delta^+(v)$), we obtain:

$$\sum_{k \in K^-} (x_{i_k^-}^k)^2 = \sum_{k \in K^- \cap K^+} x_{i_k^+}^k x_{i_k^-}^k = \sum_{k \in K^+} (x_{i_k^+}^k)^2.$$

From the Cauchy-Schwarz inequality applied to the vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^q$, where $u_k = x_{i_k^-}^k$ if $k \in K^-$ and $u_k = 0$ otherwise, and $v_k = x_{i_k^+}^k$ if $k \in K^+$ and $v_k = 0$ otherwise, we see that $\boldsymbol{u} = \pm \boldsymbol{v}$. Since the x_i^k's are nonnegative, we have $x_{i_k^-}^k = x_{i_k^+}^k$ for all $k \in K^+ = K^-$. From this, we deduce that for all k the vector $\boldsymbol{x}^k$ is a flow that is supported by a single $(s,t)-$path P_k (because for each non-terminal vertex v, $\sum_{e \in \delta^-} x_e^k = u_k = v_k = \sum_{e \in \delta^+} x_e^k$). Finally, we have $\boldsymbol{x}_k = \alpha_k \mathbf{1}_{P_k}$ for some $\alpha_k \in \mathbb{R}$, so it holds that $X = \sum_k f_k \mathbf{1}_{P_k} \mathbf{1}_{P_k}^T \in \mathcal{K}$, where we have set $f_k = \alpha_k^2 \geq 0$. □

4.3 Relation with the completely positive representation of Burer

There is another way to obtain a completely positive representation of the cone $\mathcal{K}$. Indeed, Burer showed in [Bur09] that all binary quadratic optimization problems can be reformulated as a linear program over the cone of completely positive matrices. In particular, one result shown in the Section 3 of this article can be stated as follows: *Denote by L_0 the set of $\{0,1\}$ vectors satisfying $A\boldsymbol{x} = \boldsymbol{b}$, where $A \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, and assume that $(A\boldsymbol{x} = \boldsymbol{b} \implies \boldsymbol{x} \in [0,1]^n)$. We further assume that there exists a vector $\boldsymbol{y} \in \mathbb{R}^m$ such that $\boldsymbol{\alpha} := A^T\boldsymbol{y} \in \mathbb{R}_+^n$ and $\boldsymbol{b}^T\boldsymbol{y} = 1$. Then, we have*

$$\operatorname{conv}\left\{\boldsymbol{x}\boldsymbol{x}^T : \boldsymbol{x} \in L_0\right\} = \mathcal{C}_n^* \cap \mathcal{M},$$

where

$$\mathcal{M} = \left\{X \in \mathcal{S}_n : X\boldsymbol{\alpha} = \operatorname{diag} X, \quad AX\boldsymbol{\alpha} = \boldsymbol{b}, \quad \operatorname{diag}(AXA^T) = \boldsymbol{b} \circ \boldsymbol{b}, \quad \boldsymbol{\alpha}^T X \boldsymbol{\alpha} = 1\right\}. \tag{14}$$

In the above equation, the symbol $\circ$ denotes the Hadamard product, so $\boldsymbol{b} \circ \boldsymbol{b}$ is the $m-$dimensional vector with coordinates $(\boldsymbol{b} \circ \boldsymbol{b})_i = b_i^2$. A direct application of this result yields a completely positive representation of the cone $\mathcal{K}$:

Proposition 4.3. *Let $\bar{A} \in \mathbb{R}^{m-2 \times n}$ be the matrix such that the flow conservation equations (1) at all nodes $v \notin \{s,t\}$ can be written as $\bar{A}\boldsymbol{x} = \mathbf{0}$. Let $\boldsymbol{\alpha}$ be the $n-$dimensional vector with coordinates $\alpha_i = 1$ if $i \in \delta^+(s)$, and $\alpha_i = 0$ otherwise. Then, we have $\mathcal{K} = \mathcal{C}_n^* \cap \mathcal{K}_{\text{BURER}}$, where*

$$\mathcal{K}_{\text{BURER}} = \left\{X \in \mathcal{S}_n : X\boldsymbol{\alpha} = \operatorname{diag} X, \quad \bar{A}X\boldsymbol{\alpha} = \mathbf{0}, \quad \operatorname{diag}(\bar{A}X\bar{A}^T) = \mathbf{0}\right\}.$$

Proof. Let A be the incidence matrix of G, and let $\boldsymbol{b}$ be the vector of dimension m such that $b_s = -1$ and $b_t = 1$, so the flow conservation equations (1) can be written as $A\boldsymbol{x} = \boldsymbol{b}$ for a flow of value $u = 1$. It is well known that the (s,t)-paths $P \in \mathcal{P}$ are exactly the binary solutions of the equations $A\boldsymbol{x} = \boldsymbol{b}$, that is,

$$\exists P \in \mathcal{P} : \boldsymbol{x} = \mathbf{1}_P \iff \begin{cases} A\boldsymbol{x} = \boldsymbol{b} \\ \boldsymbol{x} \in \{0,1\}^n. \end{cases}$$

This means that we can use the result of Burer to obtain a completely positive representation of $\mathcal{K}(1) = \operatorname{conv}\{\mathbf{1}_P\mathbf{1}_P^T : P \in \mathcal{P}\}$. Indeed, let $\boldsymbol{y} \in \mathbb{R}^m$ be the vector such that $y_s = 0$ and $y_v = 1$ (for all $v \in V \setminus \{s\}$). Then, we have $A^T\boldsymbol{y} = \boldsymbol{\alpha} \geq \mathbf{0}$ and $\boldsymbol{b}^T\boldsymbol{y} = 1$. So we have $\mathcal{K}(1) = \mathcal{C}_n^* \cap \mathcal{M}$, where $\mathcal{M}$ is the polyhedral set defined as in (14). Finally, the result of the proof is obtained by removing the constraints that impose that the flow is of unit value. □

It can easily be seen that we have $\mathcal{K}_2 \subseteq \mathcal{K}_{\text{BURER}}$, while the converse inclusion does not hold (even if we include the inequalities $X_{ij} \geq 0$ in the definition of $\mathcal{K}_{\text{BURER}}$). As a result, relaxations relying on the set $\mathcal{K}_2$ will yield tighter bounds than the relaxations one would have obtained from the general result of [Bur09].

5 The cone of flow matrices in general graphs

So far, we have assumed that the underlying directed graph is acyclic. In this section, we will show that this condition is indeed necessary, as relaxing it invalidates the natural generalizations of Proposition 4.1 and Theorem 4.2. Nevertheless, we will see that it is still possible to use the hierarchy (11) to obtain lower bounds for optimization problems such as the QSPP.

The theory we presented is based on the fact that an (s,t)-flow on a DAG can be decomposed into a set of flows on (elementary) (s,t)-paths. In the presence of cycles, the equivalent property is given to us by the well-known *Flow Decomposition Theorem*, which states that an (s,t)-flow on a general directed graph can be decomposed into a set of flows on elementary (s,t)-paths and on cycles. That is, $x \in \mathbb{R}_+^n$ satisfies the flow conservation constraints if and only if

$$x = \sum_{P \in \mathcal{P}} f_P \mathbf{1}_P + \sum_{C \in \xi} f_C \mathbf{1}_C$$

for some $f_P, f_Q \geq 0$, where ξ is the set of cycles in G.

Following the approach in the introduction, the natural extension of $\mathcal{K}$ to the general case is to define

$$\mathcal{K} := \operatorname{cone}(\{\mathbf{1}_P\mathbf{1}_P^T : P \in \mathcal{P}\} \cup \{\mathbf{1}_C\mathbf{1}_C^T : C \in \xi\}) = \left\{ \sum_{P \in \mathcal{P}} f_P \mathbf{1}_P\mathbf{1}_P^T + \sum_{C \in \xi} f_C \mathbf{1}_C\mathbf{1}_C^T \Big| f_P, f_C \geq 0 \right\}.$$

Defining $\mathcal{K}(u)$ exactly as in (2), it is once again easy to see that

$$\mathcal{K}(u) = \left\{ \sum_{P \in \mathcal{P}} f_P \mathbf{1}_P\mathbf{1}_P^T + \sum_{C \in \xi} f_C \mathbf{1}_C\mathbf{1}_C^T \Big| f_P, f_C \geq 0, \sum_{P \in \mathcal{P}} f_P = u \right\}.$$

Furthermore, following an argument analogous to that used in Section 4.1, we can see that the inclusions

$$\mathcal{K} \subseteq \cdots \subseteq \mathcal{K}_3 \subseteq \mathcal{K}_2$$

still hold. In fact, it is easy to see that $\mathcal{K}_r = \mathcal{K}_n$ for each $r \geq n$, because the elements of a tensor $T \in \mathbb{T}_{n^k}$ are indexed by sets $J \in \mathcal{I}_k \subseteq \{1, \ldots, n\}$. So, to prove convergence as in Proposition 4.1, we would need $\mathcal{K} = \mathcal{K}_n$ to be satisfied. Unfortunately, this does not always hold. To see

this, consider the very simple example of a directed graph $G = (V, E)$ with $V = \{s, t\}$ and $E = \{(s, t), (t, s)\}$. We have $\mathcal{P} = \{\{(s, t)\}\}$ and $\xi = \{\{(s, t), (t, s)\}\}$, and so it holds that

$$\mathcal{K} = \text{cone}\left(\left\{\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\right\}\right) = \left\{\begin{pmatrix} a & b \\ b & b \end{pmatrix} \Big| a \geq b \geq 0\right\}.$$

On the other hand, by definition of $\mathcal{K}_2$, we have

$$\mathcal{K}_n = \mathcal{K}_2 = \left\{\begin{pmatrix} a & b \\ b & c \end{pmatrix} \Big| a \geq b \geq c \geq 0\right\},$$

which proves that the hierarchy does not converge.

Similarly, the generalization of Theorem 4.2 cannot be proven in this case. That is, the equality

$$\mathcal{K} = \mathcal{K}_2 \cap \mathcal{C}_n^*$$

does not hold. To see this, consider the same example as before, and note that $\mathcal{C}_2^* = \mathcal{S}_2^+ \cap \mathbb{R}_+^{n \times n}$ (it is known that the cone of completely positive matrices coincides with the cone of *doubly nonnegative matrices* for $n \leq 4$, see [BSM03, DA13]). So, we have

$$\mathcal{K}_2 \cap \mathcal{C}_n^* = \left\{\begin{pmatrix} a & b \\ b & c \end{pmatrix} \Big| \; a \geq b \geq c \geq 0, \; ac - b^2 \geq 0\right\},$$

which again does not equal $\mathcal{K}$. We observe however that the constraint $X \succeq 0$ improved the situation, since it reduced the interval in which the variable b must lie (now we have $a \geq \sqrt{ac} \geq b \geq c \geq 0$).

Fortunately, the study of $\mathcal{K}_2$ does give us insight on some optimization problems. For example, consider the quadratic shortest path problem (QSPP) as defined in Section 2:

$$\text{qspl}(Q) = \min_{p \in \mathcal{P}} \mathbf{1}_P^T Q \mathbf{1}_P.$$

As is done with various shortest path problems on graphs with cycles, we assume that there exist no cycles with negative (quadratic) length. That is,

$$\mathbf{1}_C^T Q \mathbf{1}_C \geq 0 \qquad \forall C \in \xi.$$

Then, we have

$$\begin{aligned}
\min_{P \in \mathcal{P}} \mathbf{1}_P^T Q \mathbf{1}_P &= \min_{\substack{f_P \geq 0 \\ \sum_P f_P = 1}} \sum_{P \in \mathcal{P}} f_P \mathbf{1}_P Q \mathbf{1}_P^T \\
&= \min_{\substack{f_P, f_C \geq 0 \\ \sum_P f_P = 1}} \sum_{P \in \mathcal{P}} f_P \mathbf{1}_P Q \mathbf{1}_P^T + \sum_{C \in \xi} f_C \mathbf{1}_C Q \mathbf{1}_C^T \\
&= \min_{\substack{f_P, f_C \geq 0 \\ \sum_P f_P = 1}} \Big\langle Q, \sum_{P \in \mathcal{P}} f_P \mathbf{1}_P \mathbf{1}_P^T + \sum_{C \in \xi} f_C \mathbf{1}_C \mathbf{1}_C^T \Big\rangle \\
&= \min_{X \in \mathcal{K}(1)} \langle Q, X \rangle \geq \min_{X \in \mathcal{K}_2(1)} \langle Q, X \rangle.
\end{aligned}$$

The last inequality follows from the inclusion $\mathcal{K} \subseteq \mathcal{K}_2$. Since the last expression is an LP, it can be used to obtain lower bounds for the QSPP.

6 Computational results

6.1 Bounds for the Quadratic Shortest Path Problem

In this section, we compute some lower bounds for the QSPP, of the form

$$\begin{aligned} \mathrm{qspl}(Q) = \min_{X \in \mathcal{K}(1)} \langle Q, X \rangle \geq \min_{X} \ & \langle Q, X \rangle \\ s.t. \ & \sum_{i \in \delta^+(s)} X_{ii} = 1 \\ & X \in K, \end{aligned}$$

where K is one of the following cones: $\mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_2^+$. This gives bounds that can be computed in polynomial time, by using either a Linear Programming (LP) or Semidefinite Programming (SDP) solver.

We will compare these bounds to the bound proposed in [RCH+16, Section 4.2], which we denote by RBGL for Reformulation-based Bound of the Gillmore-Lawler type. This bound can be computed very efficiently: it reduces to solving n min-cost flow problems.

We solved instances on square ($d = 2$) and cubic ($d = 3$) grids of size ℓ, that is, a graph $G_{d,\ell} = (V, E)$, where $V = \{1, \ldots, \ell\}^d$, and $E = \{(\boldsymbol{u}, \boldsymbol{v}) \in V^2 : \|\boldsymbol{u} - \boldsymbol{v}\| = 1, \boldsymbol{v} \geq \boldsymbol{u}\}$. The source of $G_{d,\ell}$ is $s = [1, \ldots, 1] \in \mathbb{R}^d$ and its sink is $t = [\ell, \ldots, \ell] \in \mathbb{R}^d$. We also consider instances on the bidirected grids $\bar{G}_{d,\ell} = (V, E)$, which we construct from $G_{d,\ell}$ by adding an arc (v, u) for each existing arc (u, v). The cost matrices are generated at random, with linear costs Q_{ii} drawn uniformly at random in the interval $[0, 4]$, and interaction costs $Q_{ij} = Q_{ji}$ $(i \neq j)$ drawn uniformly in $[0, 1]$. The nonnegativity of the costs ensures that there are no cycles of negative quadratic length, so as seen in the previous section, the relaxations over $\mathcal{K}_2, \mathcal{K}_3$, or $\mathcal{K}_2^+$ are still valid over the general directed graphs $\bar{G}_{d,\ell}$.

We solved the LPs with CPLEX 12.6 [CPL09] over a PC with 8 cores at 3.6 GHz and with 32 GB RAM under Linux. The relaxation over $\mathcal{K}_2^+$ is an SDP, but we solved these problems with the LP solver of CPLEX, too, by using a basic cutting plane strategy (see [KM06]): we start by computing the solution of the LP relaxation X^* over $\mathcal{K}_2$. Then, at each iteration, we compute the smallest eigenvalue $\lambda_{\min}$ of X^* and its associated eigenvector $\boldsymbol{u}$. If $\lambda_{\min} \geq 0$, X^* is a solution of the SDP relaxation over $\mathcal{K}_2^+$. Otherwise, we add the linear cut inequality $\boldsymbol{u}^T X \boldsymbol{u} \geq 0$ to the LP and we iterate. For the instances we considered, this cutting-plane strategy was more effective than using a commercial SDP solver.

Our results are displayed in Table 1. Each row of the table gives summary information for 20 instances on one particular graph, corresponding to 20 randomly generated cost matrices. The first three columns give a ratio of the form P/N, which means that we were able to solve N relaxations out of the 20 instances (all failures were due to memory overflow), and P of them have no gap. The next columns show the mean gap (averaged over the N instances where the relaxation could be solved) between the computed lower bound and the optimal solution. We computed the RBGL bound from [RCH+16], and its gap is also shown in the table. The last columns show the CPU time required to compute the bounds (in seconds), as well as the optimal solution by using the MIP (10) over $\mathcal{K}_2^{\mathrm{MIP}}$.

We observe that the bounds from the relaxations over $K = \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_2^+$ are of a much better quality than the bound RGBL of [RCH+16]. For example, the mean gap for the $\mathcal{K}_2^+$-relaxation is always less than 1%. However, the computing cost of our bounds are much higher, especially for large instances. We also already mentioned that the proposed relaxations are very demanding in terms of memory. Finally, we observe that the bounds from our relaxation over the bidirected graphs $\bar{G}_{d,\ell}$ are excellent, and even better than for the instances on the acyclic graphs $G_{d,\ell}$.

	Instances without gap			mean gap				CPU (s)				
Graph	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_2^+$	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_2^+$	RBGL	$\mathcal{K}_2^{MIP}$	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_2^+$	RBGL
$G_{2,6}$	20/20	17/17	20/20	0.00%	0.00%	0.00%	16.40%	0.03	0.01	0.07	0.01	0.09
$G_{2,8}$	18/20	20/20	19/20	0.17%	0.00%	0.02%	23.18%	0.27	0.10	6.08	0.11	0.31
$G_{2,10}$	12/20	0/0	15/20	0.46%	–	0.03%	27.28%	3.37	2.29	–	2.33	0.86
$G_{2,12}$	5/20	0/0	10/20	0.73%	–	0.10%	30.44%	142.20	33.03	–	179.12	1.95
$G_{2,14}$	0/20	0/0	0/19	1.90%	–	0.76%	31.97%	2263.48	697.28	–	1623.91	3.89
$G_{3,3}$	20/20	20/20	20/20	0.00%	0.00%	0.00%	15.10%	0.01	0.00	0.02	0.00	0.06
$G_{3,4}$	19/20	20/20	20/20	0.01%	0.00%	0.00%	24.91%	0.21	0.05	1.26	0.05	0.40
$G_{3,5}$	19/20	0/0	20/20	0.03%	–	0.00%	32.07%	17.95	14.62	–	14.85	1.76
$\bar{G}_{2,4}$	20/20	20/20	20/20	0.00%	0.00%	0.00%	8.59%	0.01	0.00	2.47	0.00	0.05
$\bar{G}_{2,6}$	20/20	5/5	20/20	0.00%	0.00%	0.00%	15.07%	0.20	0.16	9739.66	0.16	0.35
$\bar{G}_{2,8}$	15/20	0/0	20/20	0.11%	–	0.00%	23.32%	8.72	4.98	–	5.15	1.33
$\bar{G}_{2,10}$	12/20	0/0	19/20	0.22%	–	0.00%	27.47%	132.89	97.17	–	99.57	3.79
$\bar{G}_{2,12}$	6/20	0/0	8/9	0.24%	–	0.00%	30.47%	8232.74	869.54	–	6105.35	8.84
$\bar{G}_{3,3}$	20/20	20/20	20/20	0.00%	0.00%	0.00%	15.67%	0.08	0.02	1003.11	0.03	0.24
$\bar{G}_{3,4}$	19/20	0/0	20/20	0.01%	–	0.00%	28.87%	14.17	11.24	–	12.09	1.89
$\bar{G}_{3,5}$	19/20	0/0	20/20	0.01%	–	0.00%	33.37%	781.29	705.66	–	732.19	8.59

Table 1: Results for 20 random instances of the QSPP for each considered graph.

6.2 Maximum flow problem with pairwise arc-capacities

The relaxation scheme proposed in this paper is not restricted to problems in which a single path must be computed, such as the the QSPP. In this section, we study a special flow problem that can also be formulated as an optimization problem over $\mathcal{K}$. This is a variant of the maximum flow problem, in which, for all pairs $(i, j) \in E \times E$, there is a *paired capacity* C_{ij} that limits the amount of flow sent across both arcs i and j. Such a problem could arise in a telecommunication network with interference between the arcs i and j. The Maximum Flow problem with Paired Arc-Capacities (MFPAC) is to send the maximum amount of flow $\sum_{P\in\mathcal{P}} f_P$ from s to t, subject to the pairwise capacity constraints

$$\sum_{\{P\in\mathcal{P}: i\in P, j\in P\}} f_P \leq C_{ij} \quad (\forall i, j \in E \times E).$$

When all paths can be enumerated, MFPAC can be formulated as the following linear program (LP):

$$\begin{aligned} \max_{\boldsymbol{f}\in\mathbb{R}_+^{|\mathcal{P}|}} \quad & \sum_{P\in\mathcal{P}} f_P \\ \text{s.t.} \quad & X = \sum_{P\in\mathcal{P}} f_P \mathbf{1}_P \mathbf{1}_P^T \leq C, \end{aligned} \tag{15}$$

where the inequality $X \leq C$ is componentwise, i.e., $X_{ij} \leq C_{ij}$. If the enumeration of all paths is prohibitive, we can also design a column generation procedure to solve Problem (15), but it can be seen that the pricing problem reduces to solving a QSPP: At each iteration, we solve the restricted master problem, which corresponds to Problem (15), where $\mathcal{P}$ is replaced by a subset $\hat{\mathcal{P}} \subset \mathcal{P}$. Then, we compute the quadratic shortest path P^* for the cost matrix Y, where Y is the optimal dual variable of the constraint $X \leq C$. The column generation procedure ends if $\text{qspl}(Y) \geq 1$. Otherwise, we add P^* into $\hat{\mathcal{P}}$ and we iterate.

It is straightforward that MFPAC can be reformulated as a linear optimization problem over $\mathcal{K}$:

$$\begin{aligned} \max_{X\in\mathcal{S}_n} \quad & \sum_{e\in\delta^+(s)} X_{ee} \\ \text{s.t.} \quad & X \leq C \\ & X \in \mathcal{K}. \end{aligned}$$

In what follows, we solve the LP relaxations obtained by replacing the constraint $X \in \mathcal{K}$ by $X \in \mathcal{K}_2$ or $X \in \mathcal{K}_3$, and the semidefinite programming (SDP) relaxation obtained by using the

	Instances without gap			mean gap			worse gap		
Graph	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_2^+$	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_2^+$	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_2^+$
$G_{2,2}$	20/20	20/20	20/20	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
$G_{2,3}$	20/20	20/20	20/20	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
$G_{2,4}$	17/20	20/20	17/20	0.77%	0.00%	0.77%	7.75%	0.00%	7.75%
$G_{2,5}$	10/20	13/14	10/20	2.98%	0.00%	2.45%	15.67%	0.05%	11.62%
$G_{2,6}$	12/20	19/20	12/20	3.26%	0.01%	3.07%	11.94%	0.15%	11.94%
$G_{2,7}$	7/20	20/20	7/20	5.29%	0.00%	5.11%	16.27%	0.00%	16.26%
$G_{2,8}$	7/20	18/20	7/20	5.22%	0.08%	4.87%	27.15%	0.88%	20.58%
$G_{2,9}$	9/20	0/0	9/20	3.44%	–	3.43%	15.08%	–	15.06%
$G_{2,10}$	10/20	0/0	10/20	3.75%	–	3.71%	17.39%	–	16.81%
$G_{3,2}$	20/20	20/20	20/20	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
$G_{3,3}$	15/20	16/16	15/20	0.46%	0.00%	0.46%	3.79%	0.00%	3.78%
$G_{3,4}$	9/20	18/18	9/20	1.29%	0.00%	1.29%	8.96%	0.00%	8.95%
$G_{3,5}$	19/20	0/0	19/20	0.07%	–	0.07%	1.48%	–	1.47%
$H_{3,5}$	14/20	20/20	14/20	4.04%	0.00%	2.59%	27.26%	0.00%	14.19%
$H_{3,6}$	8/20	19/20	8/20	5.29%	0.14%	4.72%	18.45%	2.75%	14.01%
$H_{3,7}$	7/20	17/20	7/20	6.49%	0.39%	5.96%	28.87%	5.82%	25.85%
$H_{3,8}$	7/20	18/20	7/20	6.10%	0.11%	4.50%	28.70%	1.93%	14.01%
$H_{3,9}$	6/20	16/19	6/20	11.33%	0.31%	8.39%	37.79%	2.93%	23.21%
$H_{3,10}$	2/20	17/20	2/20	13.69%	0.15%	9.95%	40.69%	2.09%	24.28%
$H_{3,11}$	3/20	17/20	3/20	13.37%	0.36%	10.37%	41.17%	5.92%	28.41%
$H_{3,12}$	5/20	18/20	5/20	11.18%	0.22%	7.61%	48.37%	2.92%	26.23%
$H_{3,13}$	4/20	18/20	4/20	11.03%	0.19%	9.04%	40.88%	3.76%	28.20%

Table 2: Results for 20 instances of each considered graph. Each row gives the results for three relaxations of the MFPAC problem (15), with $K \in \mathcal{K}$ replaced by $K \in \mathcal{K}_2$, $K \in \mathcal{K}_3$, or $K \in \mathcal{K}_2^+$.

constraint $X \in \mathcal{K}_2^+$, that is, $X \in \mathcal{K}_2, X \succeq 0$. This gives upper bounds for Problem (15) that can be computed in polynomial time. In addition to instances defined over grid graphs $G_{d,\ell}$, we also computed bounds for the MFPAC on the series-parallel graph $H_{3,\ell}$, which is a graph with ℓ vertices and 3 parallel arcs from i to $i+1$ ($i = 1, \ldots, \ell - 1$). Obviously, for this graph we set $s = 1$ and $t = \ell$.

As for the QSPP, we generated 20 random instances for each considered graph. The capacity matrix $C \in \mathcal{S}_n$ have been generated with diagonal elements drawn from the uniform distribution $\mathcal{U}([0, 4])$, and off-diagonal elements follow $\mathcal{U}([0, 1])$. Table 2 shows, for each graph and each relaxation, the number of instances for which the relaxation yields the optimal solution. In the table, P/N means that we were able to solve N relaxations out of the 20 instances (within a time-limit of 15 minutes, all failures were due to memory overflow), and P of them have no gap. The other columns give the mean gap and the worse gap across the N solved relaxations, where the gap is defined as $\delta = \text{val(relaxation)}/\,\text{val(Problem (15))} - 1$ and val(P) denotes the optimal value of Problem P.

From the table, we see that $\mathcal{K}_2$ already yields pretty good upper bounds, although the gaps are not as good as for the QSPP. The approximation based on $\mathcal{K}_3$ gives excellent results, especially for grid graphs, where it almost always found the optimal solution. However, many attempts to solve instances over $\mathcal{K}_3$ failed because of memory issue. The non-polyhedral approximation $\mathcal{K}_2^+$ is better than $\mathcal{K}_2$, and does not have the disadvantage to require a huge amount of memory. Finally, we observe that the quality of the relaxations decreases as the graph grows, but the gaps seem to reach a plateau and stabilize for the instances we considered.

7 Perspectives

We have presented some approximation hierarchies for the cone of flow matrices $\mathcal{K}$, that can be used to obtain relaxations for several hard flow-optimization problems that can not be handled by the standard arc-representation of a flow. We think that the cone of flow matrices $\mathcal{K}$ could be used for a variety of other applications.

We mentioned in the introduction the problem of computing a maximum flow supported by

paths of length $\leq L$ [BEH+10]. Let c_i denote the length of arc i, so that the length of a path reads $\boldsymbol{c}^T\mathbf{1}_P$: Then, a feasible flow $\boldsymbol{x}$ necessarily satisfies $\boldsymbol{c}^T\boldsymbol{x} \leq L$, because $\boldsymbol{c}^T\boldsymbol{x}$ represents the average length of the path followed by an infinitesimal particle of flow:

$$\boldsymbol{c}^T\boldsymbol{x} = \sum_i c_i x_i = \sum_i c_i \sum_{P \ni i} f_P = \sum_{P\in\mathcal{P}} f_P \underbrace{\mathbf{1}_P^T\boldsymbol{c}}_{\leq L}.$$

Similarly, if we lift the problem over the space of flow matrices, the matrix $X = \sum_P f_P \mathbf{1}_P\mathbf{1}_P^T$ associated with the flow must satisfy

$$\boldsymbol{e}_i^T X \boldsymbol{c} \leq L \qquad \forall i \in \{1,\ldots,n\}, \tag{16}$$

because the *sub-flow* of particles using arc i is also supported by paths of length $\leq L$. Such inequalities can be used to obtain tighter bounds for the L-bounded maximum flow problem.

Another potential application is in the field of *optimal design of experiments* (DoE). Here, the goal is to decide which proportion w_i of a total number of experimental runs should be performed under the conditions $\boldsymbol{x}_i$, when the aim is to estimate an unknown vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^n$. It is assumed that an experimental run at $\boldsymbol{x}_i$ provides an observation of the form $y_i = f(\boldsymbol{x}_i)^T\boldsymbol{\theta} + \epsilon_i$, where the measurement errors are identically and independently distributed (i.i.d.). This leads to optimization problems of the form

$$\max_{\boldsymbol{w}\geq\mathbf{0},\sum_i w_i=1} \quad \Phi\left(\sum_i w_i f(\boldsymbol{x}_i) f(\boldsymbol{x}_i)^T\right),$$

where $\Phi : \mathbb{S}_n^+ \to \mathbb{R}$ is a concave function, such as the D-criterion $\Phi_D : M \mapsto (\det M)^{\frac{1}{n}}$, or the A-criterion $\Phi_A : M \mapsto (\frac{1}{n}\operatorname{trace} M^{-1})^{-1}$, see e.g. [Puk93]. In some applications, the $f(\boldsymbol{x}_i)$'s can correspond to incidence vectors of paths in a graph. Assume for example that we want to measure the delay θ_i of each link $e \in E$ in a telecommunication network. We assume that it is possible to measure the *ping time* on each route from s to t, that is, we can measure $y_P = \mathbf{1}_P^T\boldsymbol{\theta} + \epsilon$; A similar problem was studied in [TN05]. Then, the problem of selecting the optimal proportion w_P of *pings* to perform on route P can be formulated as a compact, convex optimization problem over $\mathcal{K}(1)$:

$$\max_{\boldsymbol{w}\geq\mathbf{0},\sum_P w_P=1} \quad \Phi\left(\sum_{P\in\mathcal{P}} w_P \mathbf{1}_P\mathbf{1}_P^T\right) = \max_{X\in\mathcal{K}(1)} \Phi(X).$$

Another possible application in the field of DoE is for the model of spring balance weighing designs [JN83, Gra11, FHK11]: The goal is to determine the weight θ_i of n items, and it is possible to measure the sum of the weights of any subset of the articles (by placing these items in the spring balance): $y_S = \sum_{i\in S}\theta_i + \epsilon_S$. If the measurement errors are independently and identically distributed, the DoE problem can be written as

$$\max_{\boldsymbol{w}\geq\mathbf{0},\sum_S w_S=1} \quad \Phi\left(\sum_S w_S \mathbf{1}_S\mathbf{1}_S^T\right),$$

where the sum goes over all subsets S of $\{1,\ldots,n\}$, and $\mathbf{1}_S$ is the incidence vector of S. Now, consider the graph $H_{2,n+1}$, which has $n+1$ vertices and 2 parallel arcs between i and $i+1$ $(i = 1,\ldots,n)$, and where we set $s = 1$ and $t = n+1$. For an appropriate order of the $2n$ arcs of this graph, the incidence vector of an $(s,t)-$path is of the form $[\mathbf{1}_S^T, \mathbf{1}_{\bar{S}}^T]^T$, where S is a subset of $\{1,\ldots,n\}$ and $\bar{S} = \{1,\ldots,n\} \setminus S$ is its complement. It follows that the DoE problem for the spring balance design model can be formulated as a convex optimization problem over $\mathcal{K}(1)$ for the graph $G = H_{2,n+1}$:

$$\begin{aligned} \max_{X_{11},X_{12},X_{22}} \quad & \Phi(X_{11}) \\ s.t. \quad & \begin{pmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{pmatrix} \in \mathcal{K}(1) \end{aligned}$$

where X_{11}, X_{12} and X_{22} are the $n \times n$ blocks of a flow matrix. This formulation has the advantage of being compact, and does not require the enumeration of all subsets of $\{1, \ldots, n\}$. We point out that analytic formulas for some optimal designs of the spring balance model are known [JN83], but the present approach could be used to compute optimal designs in the presence of additional restrictions. For example, we can impose a limit on the proportion of weighings that involve item i by using a constraint of the form $X_{ii} \leq \alpha$, or we can obtain bounds for the problem in which the number of items to use in each weighting is bounded, by adding constraints of the same form as (16).

So far, our relaxations can only be used to obtain bounds for some optimization problem. In particular, we have only worked with the outer approximation $\mathcal{C}_n^* \subseteq \mathcal{S}_n^+ \cap \mathbb{R}_+^{n \times n}$, but inner approximations of $\mathcal{C}_n^*$ also exist [Las14]. If we solve the relaxation obtained by replacing the constraint $X \in \mathcal{K}$ by the constraint $X \in \mathcal{K}^{\text{inner}}$, where $\mathcal{K}^{\text{inner}} \subset \mathcal{K}$, we will obtain a matrix $X \in \mathcal{K}$, but it is unclear how to obtain a decomposition of the form $X = \sum_P f_P \mathbf{1}_P \mathbf{1}_P^T$, with $f_P \geq 0$. A possibility would be to apply the decomposition algorithm 1, but there is no guarantee that the returned f_P are nonnegative. Maybe it is possible to design a procedure that transports some weight from a set of *backward paths* to a set of *forward paths*, in order to get a decomposition where each f_P is nonnegative.

Another open question is whether there exists a combinatorial algorithm to solve linear optimization problems over $\mathcal{K}_2, \mathcal{K}_3, \ldots$. Alternatively, we would like to investigate decomposition methods in order to solve problems over $\mathcal{K}_3$ or $\mathcal{K}_3$ without the need of considering $\mathcal{O}(n^2)$ or $\mathcal{O}(n^3)$ variables, respectively.

References

[AMO93] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice hall, 1993.

[BEH+10] G. Baier, T. Erlebach, A. Hall, E. Köhler, P. Kolman, O. Pangrác, H. Schilling, and M. Skutella. Length-bounded cuts and flows. *ACM Transactions on Algorithms (TALG)*, 7(1):4, 2010.

[BR06] A. Berman and U.G. Rothblum. A note on the computation of the cp-rank. *Linear Algebra and its Applications*, 419(1):1–7, 2006.

[BSM03] A. Berman and N. Shaked-Monderer. *Completely positive matrices*. World Scientific, 2003.

[Bur09] S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120(2):479–495, 2009.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BX05] A. Berman and C. Xu. {0, 1} completely positive matrices. *Linear algebra and its applications*, 399:35–51, 2005.

[CPL09] IBM ILOG CPLEX. V12. 1 users manual for cplex. Technical report, International Business Machines Corporation, 2009.

[CSSM07] J.R. Correa, A.S. Schulz, and N.E. Stier-Moses. Fast, fair, and efficient flows in networks. *Operations Research*, 55(2):215–225, 2007.

[DA13] H. Dong and K. Anstreicher. Separating doubly nonnegative and completely positive matrices. *Mathematical Programming*, pages 1–23, 2013.

[DG14] P.J.C. Dickinson and L. Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Computational optimization and applications*, 57(2):403–415, 2014.

[Dür10] M. Dür. Copositive programming–a survey. In *Recent advances in optimization and its applications in engineering*, pages 3–20. Springer, 2010.

[FHK11] L. Filová, R. Harman, and T. Klein. Approximate e-optimal designs for the model of spring balance weighing with a constant bias. *Journal of Statistical Planning and Inference*, 141(7):2480–2488, 2011.

[FL16] S. Friedland and L.-H. Lim. The computational complexity of duality. *arXiv preprint arXiv:1601.07629*, 2016.

[GMO76] H.N. Gabow, S.N. Maheshwari, and L.J. Osterweil. On two problems in the generation of program test paths. *IEEE Transactions on Software Engineering*, 2(3):227–231, 1976.

[Gra11] M. Graczyk. A-optimal biased spring balance weighing design. *Kybernetika*, 47(6):893–901, 2011.

[HUL12] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2012.

[JN83] M. Jacroux and W. Notz. On the optimality of spring balance weighing designs. *The Annals of Statistics*, 11(3):970–978, 1983.

[KM06] K. Krishnan and J.E. Mitchell. A unifying framework for several cutting plane methods for semidefinite programming. *Optimization methods and software*, 21(1):57–74, 2006.

[Las14] J.B. Lasserre. New approximations for the cone of copositive matrices and its dual. *Mathematical Programming*, 144(1-2):265–276, 2014.

[Lau03] M. Laurent. A comparison of the sherali-adams, lovász-schrijver, and lasserre relaxations for 0–1 programming. *Mathematics of Operations Research*, 28(3):470–496, 2003.

[Puk93] F. Pukelsheim. *Optimal Design of Experiments*. Wiley, 1993.

[RCH+16] B. Rostami, A. Chassein, M. Hopf, D. Frey, C. Buchheim, F. Malucelli, and M. Goerigk. The quadratic shortest path problem: complexity, approximability, and solution methods. Technical report, Technical Report available at www. optimization-online. org, 2016.

[TN05] Y. Tsang and R. Nowak. Optimal network tomography. In *IEEE INFOCOM 2005 Student Workshop*, Miami, FL, USA, 2005.

A Proof of Lemma 3.4

Proof. Let $\boldsymbol{x}$ be a signed flow on G. We can see $\boldsymbol{x}$ as a standard (nonnegative) flow over the directed graph $G' = (V', E')$, which has an arc from i to j if $x_{ij} > 0$, and an arc from j to i if $x_{ij} < 0$ (note that G' might have cycles). We denote by $|\boldsymbol{x}|$ the vector of absolute values of $\boldsymbol{x}$. By the flow decomposition theorem (see [AMO93]), we can compute (in polynomial time) a decomposition of the form

$$|\boldsymbol{x}| = \sum_{P \in \mathcal{P}'} f_P \mathbf{1}_P + \sum_{C \in \mathcal{C}'} f_C \mathbf{1}_C,$$

with $f_P \geq 0$, $f_C \geq 0$, $\mathcal{P}'$ is a set of $(s,t)-$paths in G', $\mathcal{C}'$ is a set of cycles in G', and $\mathbf{1}_C$ is the arc-incidence vector of cycle C. We can multiply (elementwise) the above equations by the vectors of signs of x_e. This yield an equality of the form

$$\boldsymbol{x} = \sum_{P \in \mathcal{P}'} f_P \tilde{\mathbf{1}}_P + \sum_{C \in \mathcal{C}'} f_C \tilde{\mathbf{1}}_C, \tag{17}$$

where the element e of $\tilde{\mathbf{1}}_P$ is 1 if $e \in P$ and $x_e > 0$, -1 if $e \in P$ and $x_e < 0$, and 0 otherwise.

Now, we shall see that the flow on each path and cycle of G' can be decomposed as a sum of positive and negative flows over paths of G.

We only handle the case of paths, the construction for cycles is similar. We recall that G is a proper graph, which means that there is a path $P(s,v)$ from s to v and a path $P(v,t)$ from v to t for all $v \in V$. Let P be a path in G'. The path P can decomposed as an alternating sequence of *forward subpaths*, where the arcs of P have the same direction as in G, and *backward subpaths*, where the arcs of P have the opposite direction as in G. We denote this sequence as $(P_1^+, P_1^-, P_2^+, \ldots, P_{k-1}^-, P_K^+)$, where each P_i^+ is a forward subpath and each P_i^- is a backward subpath. We denote the extremities of P_i^- by u_i and v_i, so the extremities of P_i^+ are v_{i-1} and u_i, and we have $s = v_0$, $t = u_k$.

Now we claim that we can write $\tilde{\mathbf{1}}_P = \sum_{i=1}^{k} \mathbf{1}_{Q_i^+} - \sum_{i=1}^{k-1} \mathbf{1}_{Q_i^-}$, where the Q_i^+ (Q_i^-) are paths in G, defined by

$$\begin{aligned} Q_i^+ &= P(s, v_{i-1}) \cup P_i^+ \cup P(u_i, t) \\ Q_i^- &= P(s, v_i) \cup P_i^\flat \cup P(u_i, t), \end{aligned}$$

where $P_i^\flat$ denotes the set of arcs (u,v) such that $(v,u) \in P_i^-$ (so $P_i^\flat \subseteq E$). Indeed,

$$\begin{aligned} \sum_{i=1}^{k} \mathbf{1}_{Q_i}^+ - \sum_{i=1}^{k-1} \mathbf{1}_{Q_i}^- =& \sum_{i=1}^{k} \left(\mathbf{1}_{P(s,v_{i-1})} + \mathbf{1}_{P_i^+} + \mathbf{1}_{P(u_i,t)} \right) - \sum_{i=1}^{k-1} \left(\mathbf{1}_{P(s,v_i)} + \mathbf{1}_{P_i^\flat} + \mathbf{1}_{P(u_i,t)} \right) \\ =& \mathbf{1}_{P(s,v_0)} + \sum_{i=1}^{k} \mathbf{1}_{P_i^+} - \sum_{i=1}^{k-1} \mathbf{1}_{P_i^\flat} + \mathbf{1}_{P(u_k,t)} \\ =& \underbrace{\mathbf{1}_{P(s,s)}}_{=0} + \tilde{\mathbf{1}}_P + \underbrace{\mathbf{1}_{P(t,t)}}_{=0} \end{aligned}$$

A similar construction can be done for cycles. Then, in (17) we can replace each $\tilde{\mathbf{1}}_P$ and each $\tilde{\mathbf{1}}_C$ by a sum of forward paths and backward paths in G, to obtain the desired decomposition.

□