Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

GUILLAUME SAGNOL, MARCO BLANCO,
THIBAUT SAUVAGE

Approximation Hierarchies for the cone of flow matrices

This paper was presented at the INOC 2017, and published in Electric Notes in Discrete Mathematics (doi: 10.1016/j.endm.2018.02.002).
An extended version of this work was also accepted for publication in Networks (doi: 10.1002/net.21820).

ZIB Report 18-20 (April 2018)

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

Approximation Hierarchies for the cone of flow matrices

Guillaume Sagnol[1,2], Marco Blanco[2], Thibaut Sauvage[3]

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. Several combinatorial problems reduce to a linear optimization problem over $\mathcal{K}$. This cone is intractable, but 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 a maximum flow problem with pairwise arc-capacities.

Keywords. Flows in graphs, Approximation hierarchies, Copositive programming

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,

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

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 $n \times n$: $\mathcal{C}_n^* := \operatorname{conv}(\{\boldsymbol{x}\boldsymbol{x}^T : \boldsymbol{x} \in \mathbb{R}_+^n\})$. The space $\mathcal{S}_n$ is equipped with the inner product $\langle A, B\rangle := \operatorname{trace} AB$, and the associated Frobenius norm $\|X\|_F = \sqrt{\langle X, X\rangle} = \big(\sum_{i,j} X_{i,j}^2\big)^{1/2}$. 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*. For simplicity we assume that s has no incoming arcs, and t has no outgoing arcs. We denote the set of $(s,t)-$paths in G by $\mathcal{P}$. 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

[1] Technische Universitt Berlin, Fakultt II, Institut fr Mathematik, MA 5-2, Strae des 17. Juni 136, 10623 Berlin, Germany

[2] Zuse Institute Berlin Takustr. 7, 14195 Berlin, Germany

[3] thibaut.sauvage@polytechnique.edu

flow conservation equation:

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

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 $\mathcal{A}(1) = \operatorname{conv}(\{\mathbf{1}_P : P \in \mathcal{P}\})$ and $\mathcal{A} = \operatorname{cone}(\{\mathbf{1}_P : P \in \mathcal{P}\})$, 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. [1]. 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 can be solved using the arc-representation of a flow, which has the great advantage to be compact, while the number of paths might grow exponentially with the size of the graph. However, for some problems there is a kind of coupling 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 particle.

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

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

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 $\operatorname{diag} X = \sum_{P\in\mathcal{P}} f_P \mathbf{1}_P$, 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,

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

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

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 reformulate hard combinatorial problems over graphs as completely positive programs, see [5]. In particular, we could have used a general result of Burer [3] to obtain a completely positive representation of $\mathcal{K}$. The representation we obtain in Theorem 3.2 is different, though, and our proof is purely combinatorial.

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 1), which appears naturally in the definition of the dual cone $\mathcal{K}^*$, see Section 2. Tractable approximations of $\mathcal{K}$ might yield approximation algorithms for these combinatorial optimization problems. We propose two convergent approximation hierarchies in Section 3, and we discuss an application for the *maximum flow problem with pairwise arc-capacities* in Section 4.

1 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):

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

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 [9]. The QSPP can be formulated as a linear optimization problem over $\mathcal{K}(1)$:

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

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 to find 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 $\mathcal{K}$.

2 The membership problem

We study the following question:

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

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 theorem, 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,

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

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:

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

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. It follows that any algorithm that solves MEM(S) also solves WMEM(S); cf. [4]. Therefore, the NP-hardness of WMEM(S) implies that of MEM(S).

Proposition 2.1. *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. [2]).

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 [9]. The authors give a polynomial reduction from the *path with forbidden pairs* problem (PFPP), which is known to be NP-complete [7], 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. $\square$

It is shown in [6, 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. However, we can prove the NP-hardness of MEM($\mathcal{K}$) by reasoning *relatively to the linear envelope of* $\mathcal{K}$. To do so, we first need to show that this linear envelope, that is,

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

can be computed in polynomial time.

Theorem 2.2. *The membership problem* MEM($\mathcal{K}$) *is NP-complete.*

3 Approximation Hierarchies

3.1 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}.

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 $k \geq 2$ we have

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

For some $k \geq 2$, 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 shown the following:

Proposition 3.1. *We have the following approximation hierarchy for $\mathcal{K}$:*

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

In fact, we can even show that $\mathcal{K} = \mathcal{K}_N$, where $N = \max\{|P| : P \in \mathcal{P}\}$ denote the length of the longest $(s,t)-$path in G. The proof is lengthy, and we reserve it for a journal version of this article. 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.

3.2 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 exists several inner and outer nested approximation hierarchies converging to $\mathcal{C}_n^*$, see e.g. [5, 8].

Theorem 3.2. *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 $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^q \in \mathbb{R}_+^n$. 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 $(i,k) \in \{1, \ldots, n\} \times \{1, \ldots, q\}$, denote by x_i^k the ith element of $\boldsymbol{x}^k$, and denote by $\boldsymbol{x}_i$ the vector of dimension q with elements $(x_i^1, \ldots, x_i^q)$. 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 ith 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 $K^- = \big\{k \in \{1,\ldots,q\} : \exists e \in \delta^-(v) : x_e^k > 0\big\}$ and $K^+ = \big\{k \in \{1,\ldots,q\} : \exists e \in \delta^+(v) : x_e^k > 0\big\}$. 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 ith column of X:

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

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:

$$\forall i \in [N], \qquad \sum_{k\in K^-} x_{i_k^-}^k x_i^k = \sum_{k\in K^+} x_{i_k^+}^k x_i^k. \tag{1}$$

Summing Eq. (1) 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 $\boldsymbol{x}_k = f_k \mathbf{1}_{P_k}$ for some $f_k \geq 0$, and $X = \sum_k f_k^2\, \mathbf{1}_{P_k}\mathbf{1}_{P_k}^T \in \mathcal{K}$. □

4 Maximum flow problem with pairwise arc-capacities

In this section, we present numerical results for 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

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

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

$$\begin{aligned}\max_{\boldsymbol{f}\in\mathbb{R}_+^{|\mathcal{P}|}} \quad & \sum_{P\in\mathcal{P}} f_P \\ \text{s.t.} \quad & X = \sum_{P\in\mathcal{P}} f_P \mathbf{1}_P \mathbf{1}_P^T \le C,\end{aligned} \tag{2}$$

where the inequality $X \le C$ is componentwise, i.e., $X_{ij} \le C_{ij}$. We can design a column generation procedure to avoid the enumeration of all paths, but it can be seen that the pricing problem reduces to solving a quadratic shortest path problem. It is straightforward that MFPAC can be reformulated as a linear optimization problem over $\mathcal{K}$:

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

We will use MFPAC as an example to test the quality of our approximation hierarchies. 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 constraint $X \in \mathcal{K}_2^+$, that is, $X \in \mathcal{K}_2, X \succeq 0$. This gives upper bounds for Problem (2) that can be computed in polynomial time. 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 $E = \{(\boldsymbol{u}, \boldsymbol{v}) \in V^2 : \|\boldsymbol{u} - \boldsymbol{v}\| = 1, \boldsymbol{v} \ge \boldsymbol{u}\}$, as well as on the series-parallel graphs $H_{3,\ell}$, which are graphs with ℓ vertices and 3 parallel arcs from i to $i+1$ ($i = 1, \ldots, \ell-1$). For $G_{d,\ell}$ we set $s = [1, \ldots, 1] \in \mathbb{R}^d$ and $t = [\ell, \ldots, \ell] \in \mathbb{R}^d$, and for $H_{3,\ell}$, the source is $s = 1$ and the sink $t = \ell$.

For each graph, we generated 20 random instances, corresponding to a capacity matrix $C \in \mathcal{S}_n$ whose diagonal elements are drawn from the uniform distribution $\mathcal{U}([0, 4])$, and off-diagonal elements are drawn from $\mathcal{U}([0, 1])$. Table 1 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 $\delta = \mathrm{val}(\text{relaxation})/\mathrm{val}(\text{Problem (2)}) - 1$, where $\mathrm{val}(P)$ denotes the optimal value of Problem P.

From the table, we see that $\mathcal{K}_2$ already yields pretty good upper bounds, although the quality of the relaxation decreases as the graph grows. The approximation based on $\mathcal{K}_3$ gives excellent results, especially for grid graphs, where it almost always found the optimal solution. The non-polyhedral approximation $\mathcal{K}_2^+$ is better than $\mathcal{K}_2$, but it is also much harder to solve, and it seems that the approximation based on $\mathcal{K}_3$ is both superior and easier to solve. In future work, we want to investigate decomposition methods to solve problems over $\mathcal{K}_3$ without the need of considering $\mathcal{O}(n^3)$ variables.

Graph	Instances without gap $\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_2^+$	mean gap $\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_2^+$	worse gap $\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_2^+$
$G_{2,2}$	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	7/13	2.98%	0.00%	1.81%	15.67%	0.05%	7.16%
$G_{2,6}$	12/20	19/20	12/20	3.26%	0.01%	2.79%	11.94%	0.15%	10.88%
$G_{2,7}$	7/20	20/20	7/20	5.29%	0.00%	4.75%	16.27%	0.00%	16.19%
$G_{2,8}$	7/20	18/20	0/0	5.22%	0.08%	–	27.15%	0.88%	–
$G_{2,9}$	9/20	0/0	0/0	3.44%	–	–	15.08%	–	–
$G_{2,10}$	10/20	0/0	0/0	3.75%	–	–	17.39%	–	–
$G_{3,2}$	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	13/16	0.46%	0.00%	0.49%	3.79%	0.00%	3.79%
$G_{3,4}$	9/20	18/18	0/0	1.29%	0.00%	–	8.96%	0.00%	–
$G_{3,5}$	19/20	0/0	0/0	0.07%	–	–	1.48%	–	–
$H_{3,5}$	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	6/19	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 1: Results for 20 instances of each considered graph. Each row gives the results for three relaxations of the MFPAC problem (2), with $K \in \mathcal{K}$ replaced by $K \in \mathcal{K}_2$, $\mathcal{K}_3$, or $\mathcal{K}_2^+$.

References

[1] Ahuja, R., T. Magnanti and J. Orlin, *Network flows: theory, algorithms, and applications* (1993).

[2] Boyd, S. and L. Vandenberghe, "Convex Optimization," Cambridge University Press, 2004.

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

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

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

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

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

[8] Lasserre, J., *New approximations for the cone of copositive matrices and its dual*, Mathematical Programming **144** (2014), pp. 265–276.

[9] Rostami, B., 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).