PARTINIONING ORBITOPES FOR PERMUTATION OF ROWS AND COLUMNS – TERM PAPER

MATTHIAS GÖRGENS

1. Introduction

Symmetry is a problem in integer programming. The introduction and analysis of orbitopes has been one recent step towards removing those symmetries. Orbitopes exploit the fact that one can introduce groups acting on the feasible solutions of a symmetric integer problem such that members of the same orbit share the same objective value. Obviously, any optimization algorithm that only considers at least one representative of each orbit will still find an optimal solution, if there is one. Orbitopes are the convex hulls of all representatives thus considered.

In this paper I will consider a certain kind of partitioning orbitopes. Partitioning orbitopes are polytopes that live in the set of 0/1-matrices with at most, resp. exactly, one 1-entry per row. Matrices that can be transformed into each other by permutation of columns are considered equal and belong to one orbit. The partitioning orbitopes are inclusion minimal polytopes that contain one and only one member from each orbit: the unique representative matrix with lexicographically sorted columns.

In [1] Faenza and Kaibel give compact extended formulations for the packing and partitioning orbitopes. In this paper I want to give extended formulations for symmetric partitioning orbitopes: In addition to column permutation, row permutations are allowed as well. More formally the group $S = S_R \times S_C$ acts on the solution, where S_R and S_C are the symmetric groups on rows and columns. The next section gives a description of the representatives contained in a symmetric partitioning orbitope.

2. Vertices of the Symmetric Partitioning Orbitopes

Let p be the number of columns and q be the number of rows of matrices in the set $M \subseteq \{0,1\}^{p\times q}$ of all 0/1-matrices with exactly one 1-entry in each row. Let $S = S_R \times S_C$ be the Cartesian product of the symmetric groups on rows and columns. S acts on M.

Obviously we can require representatives to be lexicographically sorted along rows and columns. Thus we get representative matrices with 1 in the top-left. In each row the 1-entry stays in the column it dwelled in the row above – or wanders off one column right. In fact sorting along the rows is sufficent for this normal form; sorting the columns does not add to the requirements.

Using a very simple flow network one can easily find the facettes of the resulting *simple symmetric partition orbitope* – see 4.

Unfortunately this way some orbits of S intersect the orbitope with more than one member. To guarantee exactly one representative per orbit we restrict the orbitope further:

Definition 1 (Exact Symmetric Partitioning Orbitope)**.** Given $p, q \in \mathbb{N}\setminus\{0\}$, define the exact symmetric partitioning orbitope $S^{=}_{p,q}$ as the convex hull of all matrices in $\{0{,}1\}^{p\times q}$ whose rows are in lexicographically decreasing order and whose columns are in decreasing order by Hamming weight.

Computer generated examples suggest that no polynomial in $p \cdot q$ limits the number of exact symmetric partioning orbitope's facettes.

But - I will show an extended formulation for $S^{=}_{p,q}$ by linear inequalities that is polynomially bounded in number of both variables and facettes in $p \cdot q$.

3. Facettes of the Simple Symmetric Partitioning Orbitope

The simple symmetric partitioning orbitope permits an efficient description by equalities and inequalities in the original space of variables.

4. Polynomial Optimization on the Exact Symmetric Partitioning Orbitope

We will reduce maximization over the orbitope to finding a longest s-t-path in an acyclic weighted digraph.

4.1. Setup.

Let $p, q \in \mathbb{N} \setminus \{0\}$. Given a matrix $M \in \mathbb{Q}^{p\times q}$ of objective values, consider the program:

$$\max \langle M, \mathbf{x}\rangle \text{ s.t. } x \in S^{=}_{p,q} \tag{1}$$

where

$$\langle M, \mathbf{x}\rangle := \sum_{(i,j)\in[p]\times[q]} x_{i,j} \cdot M_{i,j}.$$

We will construct a acyclic weighted graph $D := (V, A)$ to project each vertex of $S^{=}_{p,q}$ and its objective value to one s-t-path in D and its length.

Define the vertex set:

$$V := ([p] \times [q] \times [p]) \uplus \{s\} \uplus \{t\}.$$

Each *ordinary* vertex, that is each vertex besides s and t, encodes a row, a column and the maximum Hamming weight per column still allowed.

The arcs $A := \overline{A} \uplus A_t$ can be described as a union of ordinary arcs $\overline{A}$ and arcs A_t ending in t. Setting $s := (0, 0, q)$ allows the arcs from s to be formalized as ordinary arcs.

$$\overline{A} := \{(r, c, h) \to (r + h', c + 1, h') \in V \times V : h' \le h\} \tag{2}$$

$$A_t := ([p] \times q \times [p]) \times \{t\} \tag{3}$$

For parent and child nodes and arcs we will use the following notation:

$$\begin{aligned}
\vec{\mathrm{in}}\colon V &\to \mathcal{P}(A)\\
v &\mapsto (V \times \{v\}) \cap A\\
\vec{\mathrm{out}}\colon V &\to \mathcal{P}(A)\\
v &\mapsto (\{v\} \times V) \cap A\\
\dot{\mathrm{in}}\colon V &\to \mathcal{P}(V)\\
v &\mapsto \left\{u\colon\ (u,v) \in \vec{\mathrm{in}}(v)\right\}\\
\dot{\mathrm{out}}\colon V &\to \mathcal{P}(V)\\
v &\mapsto \left\{w\colon\ (v,w) \in \vec{\mathrm{out}}\,(v)\right\}
\end{aligned}$$

To link objective values over $S^{=}_{p,q}$ with s-t-path-lengths in D we introduce arc weights $m\colon A \to \mathbb{Q}$.

$$m\left((r,c,h) \to (r+h', c+1, h')\right) := \sum_{i=r+1}^{r+h'} M_{i,c+1} \tag{4}$$

$$m\left((r,c,h) \to t\right) := 0 \tag{5}$$

4.2. **Finding the Longest s-t-Path in D.** Call $P_s\colon V \to A^*$ the function that maps each node v to the longest path connecting s and v. Here A^* means the Kleene closure of A: I.e. the set of all strings of arcs $a \in A$ including the empty string. Also set

$$\begin{aligned}
m\colon A^* &\to \mathbb{R}\\
\omega &\mapsto \sum_{a\in\omega} m\,(a)
\end{aligned}$$

To find the longest s-t-path in D is to calculate $P_s(t)$. Dynamic programming lends itself to this task. Since D is acyclic we start knowing $P_s(s) = \lambda$.

For each $v \in V$ for which $P_s(v)$ is unknown but for whose every parent $w \in \dot{\mathrm{in}}(v)$ the path $P_s(w)$ is already known, we conclude

$$m\left(P_s\left(v\right)\right) = \max_{w\in\dot{\mathrm{in}}(v)} m\left(P_s(w) \cdot (w \to v)\right).$$

The graph D admits at least one trivial path from s to t. Together with the acyclic and finite nature of D this guarantees that we learn $P_s(t)$ eventually. More specifically we get to know $P_s(t)$ after looking at each arc of D at most a constant number of times.

5. Extended Formulation for the Exact Symmetric Partitioning Orbitope

The aforementioned Graph D can be used to derive an extended formulation for the exact symmetric partitioning orbitope. For this formulation we have to construct a linear description of finding the longest path in a digraph. Fortunately – as it is well known – network flows do the trick. The transformation from network flows back to the original variable space of $S^{=}_{p,q}$ can be done with a linear function, as well.

We introduce flow variables $f \in \mathbb{R}^{A}_{\geq 0}$.
For sets M we define

$$f(M) := \sum_{m \in M} f(m).$$

Now, we can begin with the extendend formulation for the exact symmetric partitioning orbitope. For each vertex $v \in V \setminus \{s,t\}$ we have:

$$f\left(\vec{\mathrm{in}}(v)\right) = f\left(\vec{\mathrm{out}}(v)\right) \tag{6}$$

and for s we get:

$$f\left(\vec{\mathrm{out}}(s)\right) = 1 \tag{7}$$

To link f with x of our original problem (1) we introduce the following conditions that are in analogue to 4:

$$x\left(r, c\right) = \sum_{r'-h' \leq r \leq r'; h' \leq h} f\left(\left(r'-h', c-1, h\right) \to \left(r', c, h'\right)\right) \tag{8}$$

6. Restricting the Symmetric Group

In this section we will have a look at some other groups $G = G_{Rows} \times S_{Columns}$ operating on the solutions. Always the permutation of all columns will be allowed. But we will look at G_{Rows} being the symmetric group on the last t rows and then on the first t rows in order.

6.1. **Permuting the last rows.**

6.2. **Permuting the first rows.**

References

[1] Yuri Faenza and Volker Kaibel. Extended formulations for packing and partitioning orbitopes, 2008.