EXTENDED FORMULATIONS FOR COLUMN CONSTRAINED ORBITOPES

CHRISTOPHER HOJNY, MARC E. PFETSCH, AND ANDREAS SCHMITT

Abstract. In the literature, packing and partitioning orbitopes were discussed to handle symmetries that act on variable matrices in certain binary programs. In this paper, we extend this concept by restrictions on the number of 1-entries in each column. We develop extended formulations of the resulting polytopes and present numerical results that show their effect on the LP relaxation of a graph partitioning problem.

1. Introduction

Modeling combinatorial optimization problems as mixed integer optimization problems often leads to formulations which contain symmetry, i.e., solutions can be permuted to obtain new solutions with the same objective value. This slows solving via branch-and-bound down, since equivalent solutions are repeatedly inspected in different nodes of the branching scheme.

We illustrate this setting with the aid of the so-called *balanced partitioning problem* (BPP), a variant of the graph partitioning problem. Given an undirected graph (V, E) with $m = |V|$ nodes and a positive divisor n of m, the task is to find a partition of the nodes into exactly n equal sized parts that minimizes the number of edges between different parts. Applications include solving sparse matrix-vector multiplications with iterative-parallel algorithms, see Karypis and Kumar [9]. The problem can be formulated using binary variables $X_{v,i}$ to model whether node v belongs to part i and binary variables y_e to model whether edge e connects different parts. The problem exhibits obvious symmetry: Permuting the labels of the parts of a feasible solution yields another feasible solution. If the X-variables are considered as an $m \times n$-matrix, this permutation can be expressed by permuting columns of X.

A class of techniques to handle symmetry considers the removal of all but one lexicographically maximal representative of the set of equivalent solutions from the problem by cutting planes. In the above case, these hyperplanes would cut off all solutions whose columns in X are not sorted lexicographically non-increasing. To understand the polyhedral properties of these cutting planes, the notion of *full orbitopes* $O_{m,n}$ has been introduced in [8]. The polytope $O_{m,n}$ is the convex hull of all binary $m \times n$-matrices whose columns are sorted lexicographically non-increasing. Adding its facets defining inequalities to the problem at hand destroys the symmetry.

Utilizing further problem structure of orbitopes can give stronger cuts. In (BPP), for example, each row of X contains exactly one 1-entry. This additional structure can be handled by *partitioning orbitopes*, see [8] or Faenza

Date: June 2017.

and Kaibel [3]. Besides the row constraints, (BPP) consists of further bounds on X, namely exactly $\frac{m}{n}$ 1-entries are allowed in each column. For this reason, we consider the *column constrained (partitioning) orbitope* $C^{=k}_{m,n}$ with bound $k \in [m] := \{1, \ldots, m\}$. This polytope is the convex hull of all binary $m \times n$-matrices contained in $O_{m,n}$ that have exactly k 1-entries in every column, i.e.,

$$C^{=k}_{m,n} := \operatorname{conv}\Big(\Big\{X \in O_{m,n} \cap \{0,1\}^{m\times n} \;:\; \sum_{i=1}^{m} X_{i,j} = k, j \in [n]\Big\}\Big).$$

In this paper, we will shortly sketch an efficient algorithm to optimize over $C^{=k}_{m,n}$. Afterwards, an extended formulation for the case of two columns is derived via a disjunctive programming approach. A numerical comparison of symmetry handling with this extended formulation and the partitioning orbitope concludes our work.

2. Column Constrained Orbitopes

Loos [10] presented a polynomial time algorithm to optimize over full orbitopes. It uses dynamic programming and relies on the vertex structure of orbitopes. A binary matrix $X \in \{0,1\}^{m\times n}$ is a vertex of $O_{m,n}$ if and only if there exists a column index $t \in [n] \cup \{0\}$ such that the first row of X consists of t 1-entries followed by $n-t$ 0-entries and the two submatrices spanning all rows below the first row and columns 1 to t and $t+1$ to n, respectively, are vertices of orbitopes with suitable dimensions. Thus, an optimal vertex can be computed recursively and a Bellman equation for a dynamic programming algorithm can be derived.

The vertices of $C^{=k}_{m,n}$ have a similar structure. The first row of a vertex again divides into a 1-part of size t and a 0-part of size $n-t$. The two submatrices below the first row must belong to $C^{=k-1}_{m-1,t}$ and $C^{=k}_{m-1,n-t}$, respectively. By adjusting Loos' algorithm to regard the different bounds k and $k-1$, optimization over column constrained orbitopes is possible in time $\mathcal{O}(mn^3k)$. Thus, there are no obstacles from complexity theory preventing us to find a compact linear description of $C^{=k}_{m,n}$. In fact, for the bound $k=1$ the description of $C^{=k}_{m,n}$ is provided by the following system:

$$\sum_{i=1}^{m} X_{i,j} = 1, \qquad j \in [n], \tag{1a}$$

$$\sum_{i=1}^{s} X_{i,j+1} - \sum_{i=1}^{s} X_{i,j} \leq 0, \qquad (s,j) \in [m] \times [n-1], \tag{1b}$$

$$X_{i,j} \geq 0, \qquad (i,j) \in [m] \times [n]. \tag{1c}$$

These inequalities force feasible binary matrices to contain exactly one 1-entry in each column and their columns to be lexicographically non-increasing. Consequently, these matrices are exactly the vertices of $C^{=1}_{m,n}$. In fact, (1) is a complete linear description of $C^{=1}_{m,n}$, because the coefficient matrix of (1) is a network matrix, and thus totally unimodular.

For general bounds k, computational experiments with `polymake` [5], however, show that $C^{=k}_{m,n}$ has a much more complicated facet structure. But since handling symmetry associated with $C_{m,n}^{=k}$ is completely possible by forcing each pair of adjacent columns of X to belong to $C_{m,2}^{=k}$, cf. [6, Prop. 29], it suffices to only consider the column constrained orbitope $C_{m,2}^{=k}$ with two columns. The latter polytope has an easier vertex structure than $C_{m,n}^{=k}$: The first column of a vertex X of $C_{m,2}^{=k}$ has to be lexicographically not smaller than the second column by definition. Thus, either the first column coincides with the second or there exists a row $\ell \in [m]$ such that the entries coincide in each row above ℓ and $(X_{\ell,1}, X_{\ell,2}) = (1, 0)$. This leads to the definition of the critical row of a vertex X of $C_{m,2}^{=k}$, cf. Kaibel and Loos [7]:

$$\operatorname{crit}(X) \coloneqq \min(\{\ell \in [m] \,:\, (X_{\ell,1}, X_{\ell,2}) = (1, 0)\} \cup \{m + 1\}).$$

We can partition the vertices of $C_{m,2}^{=k}$ based on their critical row and define the polytopes P_ℓ by

$$P_\ell \coloneqq \operatorname{conv}\big(\{X \in C_{m,2}^{=k} \cap \{0,1\}^{m\times 2} \,:\, \operatorname{crit}(X) = \ell\}\big),\ \ell \in [m + 1].$$

We show that these polytopes are completely described by the following constraints

$$\sum_{i=1}^{m} X_{i,j} = k, \qquad j \in [2], \tag{2a}$$

$$X_{i,1} - X_{i,2} = 0, \qquad i \in [\ell - 1], \tag{2b}$$

$$(X_{\ell,1}, X_{\ell,2}) = (1, 0), \tag{2c}$$

$$0 \le X_{i,j} \le 1, \qquad (i, j) \in [m] \times [2]. \tag{2d}$$

Lemma 1. *For $\ell \in [m]$ the polytope P_ℓ consists of all those matrices $X \in \mathbb{R}^{m\times 2}$ which satisfy constraints (2a) to (2d). The polytope P_{m+1} is given by the constraints (2a), (2b) and (2d) with $\ell = m + 1$.*

Proof. Our arguments above show that every vertex of $C_{m,2}^{=k}$ with critical row ℓ satisfies the constraints. Also every integer point satisfying them is a vertex of $C_{m,2}^{=k}$, since there are exactly k 1-entries in each column by (2a) and the second column is lexicographically smaller or equal to the first by (2b) and (2c). Via elementary operations, we can transform the coefficient matrix of constraints (2a) and (2b) into a node-arc incidence matrix of a directed graph. Thus, the matrix is totally unimodular and system (2a) to (2d) defines an integral polytope. □

With the descriptions of the P_ℓ, we derive an extended formulation of $C_{m,2}^{=k}$.

Theorem 2. *Let $m \in \mathbb{N}$ and $k \in [m]$. A matrix $X \in \mathbb{R}^{m\times 2}$ lies within the column constrained orbitope $C^{=k}_{m,2}$ if and only if there exist matrices $Y^\ell \in \mathbb{R}^{m\times 2}$ and scalars $\lambda^\ell \in \mathbb{R}$ for $\ell \in [m+1]$ satisfying the constraints*

$$X_{i,j} = \sum_{\ell=1}^{m+1} Y^\ell_{i,j}, \qquad (i,j) \in [m]\times[2], \tag{3a}$$

$$\sum_{i=1}^{m} Y^\ell_{i,j} = k\lambda^\ell, \qquad (\ell,j) \in [m+1]\times[2], \tag{3b}$$

$$Y^\ell_{i,1} - Y^\ell_{i,2} = 0, \qquad \ell \in [m+1],\ \ i \in [\ell-1], \tag{3c}$$

$$(Y^\ell_{\ell,1}, Y^\ell_{\ell,2}) = (\lambda^\ell, 0), \qquad \ell \in [m], \tag{3d}$$

$$0 \le Y^\ell_{i,j} \le \lambda^\ell, \qquad (\ell,i,j) \in [m+1]\times[m]\times[2], \tag{3e}$$

$$\sum_{\ell=1}^{m+1} \lambda^\ell = 1. \tag{3f}$$

Proof. Since $C^{=k}_{m,2} = \operatorname{conv}\left(\cup_{\ell=1}^{m+1} P_\ell\right)$ and we know a complete linear description of each P_ℓ, we can apply disjunctive programming, see Balas [1, Theorem 2.1], to derive an extended formulation of $C^{=k}_{m,2}$ which is given by (3). □

Theorem 2 shows that $C^{=k}_{m,2}$ admits a compact extended formulation with $\mathcal{O}(m^2)$ variables and constraints. Note, however, that a complete linear description of $C^{=k}_{m,2}$ in the original space is unknown.

The results can be modified for the *column constraint packing orbitope* with two columns

$$C^{\le k}_{m,2} \coloneqq \operatorname{conv}\Big(\Big\{X \in O_{m,2} \cap \{0,1\}^{m\times 2} \,:\, \sum_{i=1}^{m} X_{i,j} \le k, j \in [2]\Big\}\Big).$$

Replacing equality in Constraint (2a) by inequality preserves totally unimodularity. Thus, changing equality in Constraint (3b) to inequality yields an extended formulation for $C^{\le k}_{m,2}$. A similar modification results in an extended formulation for the *column constrained covering orbitope* with two columns $C^{\ge k}_{m,2}$, which has at least k 1-entries in each column.

Coming back to the initial example (BPP), we observe that not only the number of 1-entries in each column of a solution is bounded, but additionally exactly one entry in each row must be 1. We can incorporate this further information by considering the polytope

$$\operatorname{conv}\Big(\Big\{X \in C^{=k}_{m,2} \cap \{0,1\}^{m\times 2} \,:\, X_{i,1} + X_{i,2} \le 1,\, i \in [m]\Big\}\Big) \tag{4}$$

to handle symmetry. Adding this polytope for each pair of adjacent columns, the number of 1-entries in each row is forced to be at most one. By a slight modification we can adjust the extended formulation from Theorem 2 to comply with Polytope (4): For each vertex X, the rows $(X_{i,1}, X_{i,2})$ with $i < \operatorname{crit}(X)$ must be zero, since these rows are either $(0,0)$ or $(1,1)$, but are only allowed to contain at most one 1-entry. These rows can thus be neglected in P_ℓ. Also Constraint (2b) is obsolete; instead $X_{i,1} + X_{i,2} \le 1$ for $i \in \{\ell+1,\dots,m\}$ needs to be added. Furthermore, the vertices can only have a critical row $\ell \le m - 2k$, otherwise there would not be enough 1-entries in each column. Thus, we do not use P_ℓ for $\ell > m - 2k$ in the extended formulation.

3. Numerical Results

We investigated the practical use of the derived extended formulation of $C^{=k}_{m,2}$ to handle the symmetry of (BPP) using the framework SCIP 3.2.1 [4] on a Linux cluster with Intel Xeon E5 CPUs with 3.50GHz, 10MB cache, and 32GB memory. The adjacency matrices of the graphs of our instances are from the University of Florida Sparse Matrix Collection [2]. We considered matrices with at least 6 and less than 100 rows, which are non-diagonal and pattern symmetric, i.e., only square matrices A with $A_{i,j} \neq 0$ if and only if $A_{j,i} \neq 0$. This results in 40 instances, which are to be partitioned into $n \in \{5, 10, 20\}$ sets. We added isolated nodes to the graph if n did not divide the number of nodes and disregard calculation when n exceeds the number of nodes. We compared two settings to estimate the strength of the extended formulation. The first setting restricts the variables to lie inside the partitioning orbitope via an already existing SCIP plugin. This plugin fixes the variables in the upper right triangle of X to zero and separates the facets of the partitioning orbitope. In the second setting, we added for each pair of adjacent columns of X the variables and constraints of the extended formulation for the Polytope (4).

We are most interested in the improvement of the LP relaxation by adding (constrained) orbitopes. Thus, we compared for each setting the value of the first solved LP relaxation.

To evaluate the improvement of adding the extended formulation, we calculated for each instance the quotient of the value of the extended formulation's LP relaxation d_{ef} and the value of the LP relaxation obtained by the orbitope setting d_{orbi}, i.e., the value $d_{\text{ef}}/d_{\text{orbi}}$. Since we consider a minimization problem a value greater than 1 indicates a better LP relaxation of the extended formulation.

Table 1 summarizes our results. Column "#instances" shows the number of considered instances for a fixed value of n (recall that some instances are disregarded), whereas the last three columns collect the arithmetic mean together with the min and max over the relative values for instances with the same partition size. We see that on average the LP relaxation of the extended formulation setting performs about 2% better than the orbitope. We note that, except for one instance with $n = 20$, the values are always at least 1. With an increasing number of parts the advantage of the extended formulation decreases. Nevertheless, improvements of nearly 20% for $n = 5$ to 11% for $n = 20$ can be observed for some instances.

Table 1. Statistics on the improvement ratio $d_{\text{ef}}/d_{\text{orbi}}$ for partition sizes $m \in \{5, 10, 20\}$.

n	#instances	arithmetic mean	min	max
5	40	1.0276	1.00	1.1931
10	37	1.0270	1.00	1.1382
20	34	1.0218	0.85	1.1149

4. Conclusions

In this paper, we introduced orbitopes with additional requirements and incorporated these properties to orbitopes via an extended formulation. In a computational study, we have seen that this extended formulation improves the first LP relaxation, on some instances even substantially. However, this effect decreases if we consider orbitopes with more columns. Furthermore the blow up of the variable space makes adding the whole extended formulation impractical. To benefit from the positive effect on the LP relaxation, schemes to separate the extended formulation as well as descriptions of $C_{m,n}^{=k}$ with fewer variables should be considered. In particular, a complete description in the original space would be desirable, but could not be achieved so far.

Acknowledgments. The research of the second and third author was supported by the German Research Foundation (DFG) as part of the Collaborative Research Centre 666 and 805, respectively.

References

[1] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1):3–44, 1998.

[2] T. A. Davis and Y. Hu. The University of Florida Sparse Matrix Collection. *ACM Transactions on Mathematical Software*, 38(1):1–25, 2011.

[3] Y. Faenza and V. Kaibel. Extended formulations for packing and partitioning orbitopes. *Mathematics of Operations Research*, 34(3):686–697, 2009.

[4] G. Gamrath, T. Fischer, T. Gally, A. M. Gleixner, G. Hendel, T. Koch, S. J. Maher, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, S. Schenker, R. Schwarz, F. Serrano, Y. Shinano, S. Vigerske, D. Weninger, M. Winkler, J. T. Witt, and J. Witzig. The SCIP Optimization Suite 3.2. Technical Report 15-60, ZIB, Takustr. 7, 14195 Berlin, 2016.

[5] E. Gawrilow and M. Joswig. Polymake: A framework for analyzing convex polytopes. *Polytopes – Combinatorics and Computation*, pages 43–74, 2000.

[6] C. Hojny and M. E. Pfetsch. Polytopes associated with symmetry handling. www.optimization-online.org/DB_HTML/2017/01/5835.html, 2017.

[7] V. Kaibel and A. Loos. Finding descriptions of polytopes via extended formulations and liftings. In A. R. Mahjoub, editor, *Progress in Combinatorial Optimization*. Wiley, 2011.

[8] V. Kaibel and M. E. Pfetsch. Packing and partitioning orbitopes. *Mathematical Programming*, 114(1):1–36, 2008.

[9] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

[10] A. Loos. *Describing orbitopes by linear inequalities and projection based tools*. PhD thesis, Universität Magdeburg, 2011.

Research Group Optimization, Department of Mathematics, TU Darmstadt, Germany

E-mail address: {hojny,pfetsch,aschmitt}@opt.tu-darmstadt.de