Computational Aspects of Combinatorial Optimization

Dr. Marc E. Pfetsch

Kumulative Habilitationsschrift an der
Fakultät II – Mathematik und Naturwissenschaften
der Technischen Universität Berlin

Lehrgebiet:
Mathematik

Eröffnung des Verfahrens:	10.10.2007
Verleihung der Lehrbefähigung:	30.04.2008

Gutachter:
Prof. Dr. Martin Grötschel
Prof. Dr. Thomas M. Liebling
Prof. Dr. Rolf H. Möhring
Prof. Dr. Giovanni Rinaldi

Berlin 2008

D 83

List of Publications

The following six papers are contained in this cumulative habilitation thesis.

(1) Volker Kaibel and Marc E. Pfetsch
Packing and Partitioning Orbitopes
Math. Program. **114** (2008), no. 1, pp. 1–36

(2) Volker Kaibel, Matthias Peinhardt, and Marc E. Pfetsch
Orbitopal Fixing
Proc. 12th Integer Programming and Combinatorial Optimization conference (IPCO), M. Fischetti and D. Williamson, eds., LNCS 4513, Springer-Verlag, 2007, pp. 74–88

(3) Ralf Borndörfer, Martin Grötschel, and Marc E. Pfetsch
A Column-Generation Approach to Line Planning in Public Transport
Transportation Sci. **41** (2007), no. 1, pp. 123–132

(4) Michael Joswig and Marc E. Pfetsch
Computing Optimal Morse Matchings
SIAM J. Discrete Math. **20** (2006), no. 1, pp. 11–25

(5) Edoardo Amaldi, Leslie E. Trotter, Jr., and Marc E. Pfetsch
On the Maximum Feasible Subsystem Problem, IISs, and IIS-hypergraphs
Math. Program. **95** (2003), no. 3, pp. 533–554

(6) Marc E. Pfetsch
Branch-And-Cut for the Maximum Feasible Subsystem Problem
SIAM J. Optimization **19** (2008), no. 1, pp. 21–38

Contents

Introduction 1

1. Symmetries in Integer Programs 1
2. Line Planning 2
3. Morse Matchings 3
4. Maximum Feasible Subsystem Problem 4

Paper 1. Packing and Partitioning Orbitopes 5

1. Introduction 5
2. Orbitopes: General Definitions and Basic Facts 9
3. Packing and Partitioning Orbitopes for Cyclic Groups 13
4. Packing and Partitioning Orbitopes for Symmetric Groups 15
5. Concluding Remarks 34

References 36

Paper 2. Orbitopal Fixing 39

1. Introduction 39
2. Orbitopes 42
3. The Geometry of Fixing Variables 44
4. Fixing Variables for Orbitopes 45
5. Computational Experiments 50
6. Concluding Remarks 52

References 53

Paper 3. A Column-Generation Approach to Line Planning in Public Transport 55

1. Introduction 55
2. Related Work 56
3. Line-Planning Model 57
4. Column Generation 61
5. Computational Results 67
6. Conclusions 71

References 71

Paper 4. Computing Optimal Morse Matchings 73

1. Introduction 73
2. Discrete Morse Functions and Morse Matchings 74
3. Properties of Morse Matchings 76
4. Hardness of Optimal Morse Matchings 77
5. An IP-Formulation 80
6. Computational Results 86

References 89

Paper 5. On the Maximum Feasible Subsystem Problem, IISs and IIS-hypergraphs 91
1. Introduction 92
2. Irreducible Infeasible Subsystems 93
3. IIS-hypergraphs 98
4. Feasible Subsystem (FS) Polytope 102
5. Concluding Remarks 109
Appendix 110
References 111

Paper 6. Branch-And-Cut for the Maximum Feasible Subsystem Problem 115
1. Introduction 115
2. Alternative Solution Approaches 117
3. Ingredients for Branch-and-Cut 118
4. Computational Results 124
5. Conclusions 131
References 133

Introduction

This collection contains the following six papers that I submit for obtaining the habilitation at the Technische Universität Berlin, Fakultät II – Mathematik und Naturwissenschaften.

(1) *Packing and Partitioning Orbitopes*
(2) *Orbitopal Fixing*
(3) *A Column-Generation Approach to Line Planning in Public Transport*
(4) *Computing Optimal Morse Matchings*
(5) *On the Maximum Feasible Subsystem Problem, IISs and IIS-hypergraphs*
(6) *Branch-And-Cut for the Maximum Feasible Subsystem Problem*

The papers form a cross-section through my research in combinatorial optimization. They can be grouped into four topics:

- Symmetries in Integer Programs (Papers 1 and 2)
- Line Planning (Paper 3)
- Morse Matchings (Paper 4)
- Maximum Feasible Subsystem Problem (Papers 5 and 6)

In the following, I will outline the main ideas of these topics and papers.

Note. The only changes I made in the papers with respect to the original versions concern the unified layout, e.g., renumbering of theorems and minor reformulations necessary for the modified presentation. Furthermore, I updated some references.

1. Symmetries in Integer Programs

It seems to be folklore knowledge in integer programming that symmetries pose severe problems for linear programming based branch-and-bound methods. The reasons are twofold: The linear programming bounds are weak and many equivalent solutions (with respect to the symmetry) appear in the search tree, although they do not provide new information. These difficulties usually have been resolved by finding alternative nonsymmetric formulations or by adding problem dependent symmetry breaking inequalities. In the recent years, interest in general methods to directly deal with symmetric formulations has increased.

A particular feature, which arises in many symmetric integer programming models, is that of a assignment structure, i.e., the models contain 0/1 variables x_{ij} for $i = 1, \dots, p$, $j = 1, \dots, q$ and constraints

$$\sum_{j=1}^{q} x_{ij} = 1 \qquad \text{for } i = 1, \dots, p. \tag{1}$$

If the problem dependent additional constraints and objective function have the property that permuting columns of the matrix (x_{ij}) preserves feasibility and the objective function value, the corresponding formulation is symmetric, i.e., the full symmetric group acts on the columns. Examples of such formulations arise from the graph coloring problem (see Paper 1, Model (1)) and the graph partitioning problem (see Paper 2, Model (1)).

Paper 1 (*Packing and Partitioning Orbitopes*, written jointly with Volker Kaibel), and Paper 2 (*Orbitopal Fixing*, written jointly with Volker Kaibel and Matthias Peinhardt), deal with a polyhedral approach to handle such assignment-based symmetries in integer programming. The basic idea is to use a lexicographic sorting of the columns of 0/1-matrices (x_{ij}) that fulfill (1); this breaks the symmetry by leaving a single representative in each orbit of the symmetry group. The main object of study are *partitioning orbitopes*, which are the convex hulls of all such lexicographically sorted 0/1-matrices of sizes $p \times q$.

The main results of Paper 1 are as follows. We prove a complete linear description of partitioning orbitopes, which uses exponentially many so-called *shifted column inequalities*. The corresponding separation problem is solvable in linear time. Moreover, except for few exceptions, these inequalities define facets. Similar results hold for the case of *packing orbitopes*, in which the number of ones in each row is at most 1, i.e., (1) is replaced by

$$\sum_{j=1}^{q} x_{ij} \leq 1 \qquad \text{for all } i = 1, \dots, p.$$

Furthermore, complete linear descriptions for the case of cyclic groups acting on the columns are obtained. The corresponding orbitopes, which are the convex hulls of the single representatives of each orbit under the cyclic group, can be described by a polynomial number of inequalities in p and q, and we provide totally unimodular formulations.

In Paper 2, we provide a linear time algorithm to deduce variable fixings depending on the fixings of other variables, using the structure of orbitopes. This can be seen as a node preprocessing or constraint programming approach. For the particular case of the graph partitioning problem, we computationally show that using this approach significantly improves the solution time – also compared to a direct integer programming approach via the symmetry breaking methods employed in CPLEX. It also turns out that this variable fixing method is slightly faster than the approach via the separation of shifted inequalities.

Summarizing, Papers 1 and 2 provide a way to deal with symmetries that arise from assignment-like structures. They can be used as one starting point towards a more detailed and general investigation of symmetries in integer programs.

2. Line Planning

The motivation for Paper 3 (*A Column-Generation Approach to Line Planning in Public Transport*, written jointly with Ralf Borndörfer and Martin

Grötschel) arises from the practical problem of planing lines in a public transport network. Here, given information about the transportation demands of passengers, the problem is to find line routes and frequencies such that the demand can be transported. Two opposing objectives have to be handled: One the one hand the passengers are interested in small traveling times and few transfers. On the other hand, the costs of the computed system have to be taken into account.

The line planning problem is a strategic problem, which decides upon the service level of a public transport system and hence is of social and political interest. In the strategic planning area, much fewer practically relevant integer programming approaches have appeared in the literature than for operational planning problems like vehicle and duty scheduling – let alone cases of uses in practice. One reason is that the inherent multi-objective structure makes optimization approaches more difficult to apply. The long-term goal in this area is to develop decision support tools for practical use.

Paper 3 provides an integer programming model for the line planning problem that allows for the generation of passenger and line paths. We discuss the corresponding pricing problems in a column generation approach. While the pricing of passenger paths can be solved by shortest path methods, the line pricing problem turns out to be NP-hard. We provide a polynomial time algorithm for this pricing problem, if the lengths of the lines are bounded to be $O(\log n)$, where n is the number of nodes in the network; in many practical cases, this is a realistic assumption. Computational experiments for data from the city of Potsdam show that one can compute the LP-relaxation of this model in a few minutes and obtain integer solutions with reasonable quality by a greedy type algorithm.

3. Morse Matchings

Paper 4 (*Computing Optimal Morse Matchings*, written jointly with Michael Joswig) studies a problem that arises in combinatorial topology. It is one of the few examples in which combinatorial optimization tools have been applied in this area.

The basic objects are simplicial complexes, i.e., a collection of (finite) sets closed under taking subsets. Simplicial complexes provide one way of representing many "well-behaved" topological spaces. A *Morse matching* is a matching in the Hasse diagram of a simplicial complex, such that a certain acyclicity condition is fulfilled. Morse matchings are important in combinatorial topology, because they provide a way to obtain a smaller representation of the underlying topological space, starting from a simplicial complex and performing contraction operations. The hope is that the resulting representation allows to deduce topological properties of the space or even classify its topological type.

In the paper, we first show that the problem of finding a maximum size Morse matching is NP-hard and then give an integer programming formulation. We discuss the arising separation problem of the acyclicity condition. It turns out that one needs to find shortest paths in a bipartite graph with conservative weights, i.e., no negative cycles exist. We provide a reduction of this problem to the computation of shortest paths with nonnegative weights. Computational results of a branch-and-cut algorithm show that one can compute optimal Morse matchings for medium-sized instances, especially if the upper bound derived by homology considerations is close to the optimal solution.

4. Maximum Feasible Subsystem Problem

The *maximum feasible subsystem problem* (MAX FS) is to find a largest feasible subsystem of a given infeasible linear inequality system. This has interesting connections to many different combinatorial optimization problems. One example is the problem of finding a solution of a linear equation system with the fewest number of nonzeros. Even more closely related are *irreducible infeasible subsystems* (IISs), i.e., infeasible subsystems such that every proper subsystem is feasible. A feasible subsystem can be obtained by removing at least one inequality of each IIS; this complementary problem to MAX FS can be formulated as a set covering problem (Paper 6 takes this viewpoint).

Paper 5 (*On the Maximum Feasible Subsystem Problem, IISs and IIS-hypergraphs*, written jointly with Leslie Trotter and Edoardo Amaldi) is also contained in my dissertation (*The Maximum Feasible Subsystem Problem and Vertex-Facet Incidence of Polyhedra*, TU Berlin, 2002). The paper gives a theoretical study of the MAX FS problem and structural and algorithmic properties of IISs. We first provide a geometric characterization of IISs as systems that arise by reversing the inequalities describing a simplex plus a linear space. Then we show that the problem to find a smallest IIS is NP-hard and very hard to approximate. The recognition of a given set of indices to be the set of IISs of some infeasible inequality system turns out to be hard as well. We proceed with a study of the feasible subsystem polytope, i.e., the convex hull of incidence vectors of feasible subsystems. We show that inequalities that arise from IISs define facets of this polytope and the corresponding separation problem is NP-hard. Finally, we characterize under which conditions so-called generalized antiweb inequalities define facets.

The empirical counterpart to Paper 5 is given by Paper 6 (*Branch-And-Cut for the Maximum Feasible Subsystem Problem*). It gives a detailed computational study of a branch-and-cut implementation for the MAX FS problem. Several heuristics to separate the inequalities arising from IISs are presented. Further issues of the implementation are discussed: general cutting planes, heuristics, and branching rules. The computational results can be summarized as follows. It turns out that computing optimal MAX FS solutions is quite hard for a number of instances arising from different applications. Although general purpose inequalities like Gomory-cuts or $\{0, \frac{1}{2}\}$-cuts reduce the total number of nodes, they do not significantly reduce the computation time. Nevertheless, the presented algorithm is currently the only way to compute nontrivial upper bounds for MAX FS.

Paper 1

Packing and Partitioning Orbitopes

Volker Kaibel and Marc E. Pfetsch

Packing and Partitioning Orbitopes[1]
Math. Program. **114** (2008), no. 1, pp. 1–36

Abstract. We introduce *orbitopes* as the convex hulls of 0/1-matrices that are lexicographically maximal subject to a group acting on the columns. Special cases are packing and partitioning orbitopes, which arise from restrictions to matrices with at most or exactly one 1-entry in each row, respectively. The goal of investigating these polytopes is to gain insight into ways of breaking certain symmetries in integer programs by adding constraints, e.g., for a well-known formulation of the graph coloring problem.

We provide a thorough polyhedral investigation of packing and partitioning orbitopes for the cases in which the group acting on the columns is the cyclic group or the symmetric group. Our main results are complete linear inequality descriptions of these polytopes by facet-defining inequalities. For the cyclic group case, the descriptions turn out to be totally unimodular, while for the symmetric group case, both the description and the proof are more involved. The associated separation problems can be solved in linear time.

1. Introduction

Symmetries are ubiquitous in discrete mathematics and geometry. They are often responsible for the tractability of algorithmic problems and for the beauty of both the investigated structures and the developed methods. It is common knowledge, however, that the presence of symmetries in integer programs may severely harm the ability to solve them. The reasons for this are twofold. First, the use of branch-and-bound methods usually leads to an

[1] Supported by the DFG Research Center Matheon in Berlin

unnecessarily large search tree, because equivalent solutions are found again and again. Second, the quality of LP relaxations of such programs typically is extremely poor.

A classical approach to "break" such symmetries is to add constraints that cut off equivalent copies of solutions, in hope to resolve these problems. There are numerous examples of this in the literature; we will give a few references for the special case of graph coloring below. Another approach was developed by Margot [11, 12]. He studies a branch-and-cut method that ensures to investigate only one representative of each class of equivalent solutions by employing methods from computational group theory. Furthermore, the symmetries are also used to devise cutting planes. Methods for symmetry breaking in the context of constraint programming have been developed, for instance, by Fahle, Schamberger, and Sellmann [7] and Puget [16].

The main goal of this paper is to start an investigation of the polytopes that are associated with certain symmetry breaking inequalities. In order to clarify the background, we first discuss the example of a well-known integer programming (IP) formulation for the graph coloring problem.

Let $G = (V, E)$ be a loopless undirected graph without isolated nodes. A *(vertex) coloring* of G using at most C colors is an assignment of colors $\{1, \dots, C\}$ to the nodes such that no two adjacent nodes receive the same color. The *graph coloring* problem is to find a vertex coloring with as few colors as possible. This is one of the classical NP-hard problems [9]. It is widely believed to be among the hardest problems in combinatorial optimization. In the following classical IP formulation, $V = \{1, \dots, n\}$ are the nodes of G and C is some upper bound on the number of colors needed.

$$\begin{array}{lrll}
\min & \sum_{j=1}^{C} y_j & & \\
& x_{ij} + x_{kj} \leq y_j & \{i,k\} \in E,\ j \in \{1,\dots,C\} & \text{(i)} \\
& \sum_{j=1}^{C} x_{ij} = 1 & i \in V & \text{(ii)} \\
& x_{ij} \in \{0,1\} & i \in V,\ j \in \{1,\dots,C\} & \text{(iii)} \\
& y_j \in \{0,1\} & j \in \{1,\dots,C\} & \text{(iv)}
\end{array} \qquad (1)$$

In this model, variable x_{ij} is 1 if and only if color j is assigned to node i and variable y_j is 1 if color j is used. Constraints (i) ensure that color j is assigned to at most one of the two adjacent nodes i and k; it also enforces that y_j is 1 if color j is used, because there are no isolated nodes. Constraints (ii) guarantee that each node receives exactly one color.

It is well known that this formulation exhibits symmetry: Given a solution (x, y), any permutation of the colors, i.e., the columns of x (viewed as an $n \times C$-matrix) and the components of y, results in a valid solution with the same objective function value. Viewed abstractly, the symmetric group of order C acts on the solutions (x, y) (by permuting the columns of x and the components of y) in such a way that the objective function is constant along every orbit of the group action. Each orbit corresponds to a symmetry class of feasible colorings of the graph. Note that "symmetry" here always refers to the symmetry of permuting colors, not to symmetries of the graph.

The weakness of the LP-bound mentioned above is due to the fact that the point $(x^\star, y^\star)$ with $x^\star_{ij} = 1/C$ and $y^\star_j = 2/C$ is feasible for the LP relaxation with objective function value 2. The symmetry is responsible for the feasibility of $(x^\star, y^\star)$, since $x^\star$ is the barycenter of the orbit of an arbitrary $x \in \{0,1\}^{n \times C}$ satisfying (ii) in (1).

It turned out that the symmetries make the above IP-formulation for the graph coloring problem difficult to solve. One solution is to develop different formulations for the graph coloring problem. This line has been pursued, e.g., by Mehrotra and Trick [13], who devised a column generation approach. See Figueiredo, Barbosa, Maculan, and de Souza [8] and Cornaz [5] for alternative models.

Another solution is to enhance the IP-model by additional inequalities that cut off as large parts of the orbits as possible, keeping at least one element of each orbit in the feasible region. Méndez-Díaz and Zabala [15] showed that a branch-and-cut algorithm using this kind of symmetry breaking inequalities performs well in practice. The polytope corresponding to (1) was investigated by Campêlo, Corrêa, and Frota [3] and Coll, Marenco, Méndez-Díaz, and Zabala [4]. Ramani, Aloul, Markov, and Sakallah [17] studied symmetry breaking in connection with SAT-solving techniques to solve the graph coloring problem.

The strongest symmetry breaking constraints that Méndez-Díaz and Zabala [14, 15] introduced are the inequalities

$$x_{ij} - \sum_{k=1}^{i-1} x_{k,j-1} \leq 0, \qquad \text{for all } i \text{ and } j \geq 2. \tag{2}$$

From each orbit, they cut off all points except for one representative that is the maximal point in the orbit with respect to a lexicographic ordering. A solution (x, y) of the above IP-model is such a representative if and only if the columns of x are in decreasing lexicographic order. We introduce a generalization and strengthening of Inequalities (2) in Section 4.1.

Breaking symmetries by adding inequalities like (2) does not depend on the special structure of the graph coloring problem. These inequalities single out the lexicographic maximal representative from each orbit (with respect to the symmetric group acting on the columns) of the whole set of all 0/1-matrices with exactly one 1-entry per row. The goal of this paper is to investigate the structure of general "symmetry breaking polytopes" like the convex hull of these representatives. We call these polytopes *orbitopes*. The idea is that general knowledge on orbitopes (i.e., valid inequalities) can be utilized for different symmetric IPs in order to address both the difficulties arising from the many equivalent solutions and from the poor LP-bounds. In particular with respect to the second goal, for concrete applications it will be desirable to combine the general knowledge on orbitopes with concrete polyhedral knowledge on the problem under investigation in oder to derive strengthened inequalities. For the example of graph coloring, we indicate that (and how) this can be done in Section 5. Figure 1 illustrates the geometric situation.

The case of a symmetric group acting on the columns is quite important. It does not only appear in IP-formulations for the graph coloring problem,

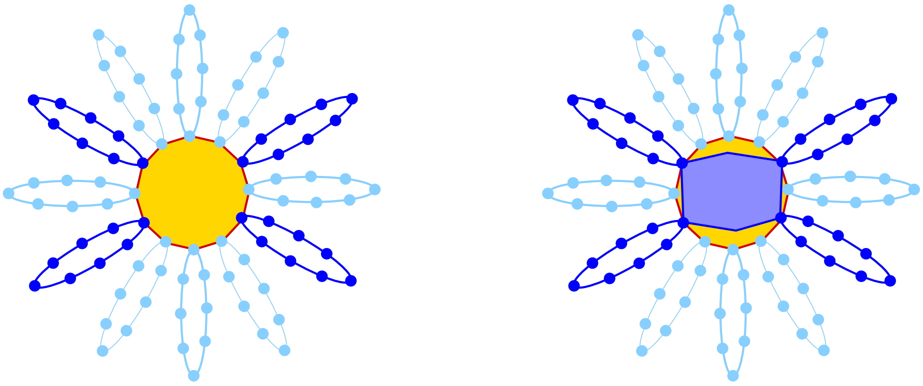


Figure 1: Breaking symmetries by orbitopes. The left figure illustrates an orbitope, i.e., the convex hull of the representatives of a large system of orbits. For a concrete problem, like graph coloring, only a subset of the orbits are feasible (the dark orbits). Combining a (symmetric) IP-formulation for the concrete problem with the orbitope removes the symmetry from the formulation (right figure).

but also in many other contexts like, e.g., block partitioning of matrices [1], k-partitioning in the context of frequency assignment [6], or line-planning in public transport [2]. However, other groups are interesting as well. For instance, in the context of timetabling in public transport systems [19], cyclic groups play an important role.

We thus propose to study different types of orbitopes, depending on the group acting on the columns of the variable-matrix and on further restrictions like the number of 1-entries per row being exactly one (*partitioning*), at most one (*packing*), at least one (*covering*), or arbitrary (*full*).

The main results of this paper are complete and irredundant linear descriptions of packing and partitioning orbitopes for both the symmetric group and for the cyclic group acting on the columns of the variable-matrix. We also provide (linear time) separation algorithms for the corresponding sets of inequalities. While this work lays the theoretical foundations on orbitopes, a thorough computational investigation of the practical usefulness of the results will be the subject of further studies (see also the remarks in Section 5).

The outline of the paper is as follows. In Section 2, we introduce some basic notations and define orbitopes. In Section 2.1 we show that optimization over packing and partitioning orbitopes for symmetric and cyclic groups can be done in polynomial time. In Section 3 we give complete (totally unimodular) linear descriptions of packing and partitioning orbitopes for cyclic groups. Section 4 deals with packing and partitioning orbitopes for symmetric groups, which turn out to be more complicated than their counterparts for cyclic groups. Here, besides (strengthenings of) Inequalities (2), one needs exponentially many additional inequalities, the "shifted column inequalities", which are introduced in Section 4.2. We show that the corresponding separation problem can be solved in linear time, see Section 4.3. Section 4.4 gives a complete linear description, and Section 4.5 investigates the facets of the polytopes. We summarize the results for symmetric groups in Section 4.6 for easier reference. Finally, we close with some remarks in Section 5.

2. Orbitopes: General Definitions and Basic Facts

We first introduce some basic notation. For a positive integer n, we define $[n] := \{1, 2, \dots, n\}$. We denote by $\mathbf{0}$ the 0-matrix or 0-vector of appropriate sizes. Throughout the paper let p and q be positive integers. For $x \in \mathbb{R}^{[p]\times[q]}$ and $S \subseteq [p] \times [q]$, we write

$$x(S) := \sum_{(i,j)\in S} x_{ij}.$$

For convenience, we use $S-(i,j)$ for $S\setminus\{(i,j)\}$ and $S+(i,j)$ for $S\cup\{(i,j)\}$, where $S \subseteq [p]\times[q]$ and $(i,j) \in [p]\times[q]$. If p and q are clear from the context, then $\mathrm{row}_i := \{(i,1),(i,2),\dots,(i,q)\}$ are the entries of the ith row.

Let $\mathcal{M}_{p,q} := \{0,1\}^{[p]\times[q]}$ be the set of 0/1-matrices of size $p \times q$. We define

- ○ $\mathcal{M}^{\leq}_{p,q} := \{x \in \mathcal{M}_{p,q} \,:\, x(\mathrm{row}_i) \leq 1 \text{ for all } i\}$
- ○ $\mathcal{M}^{=}_{p,q} := \{x \in \mathcal{M}_{p,q} \,:\, x(\mathrm{row}_i) = 1 \text{ for all } i\}$
- ○ $\mathcal{M}^{\geq}_{p,q} := \{x \in \mathcal{M}_{p,q} \,:\, x(\mathrm{row}_i) \geq 1 \text{ for all } i\}$.

Let $\prec$ be the lexicographic ordering of $\mathcal{M}_{p,q}$ with respect to the ordering

$$(1,1) < (1,2) < \cdots < (1,q) < (2,1) < (2,2) < \cdots < (2,q) < \cdots < (p,q)$$

of matrix positions, i.e., $A \prec B$ with $A = (a_{ij}), B = (b_{ij}) \in \mathcal{M}_{p,q}$ if and only if $a_{k\ell} < b_{k\ell}$, where (k,ℓ) is the first position (with respect to the ordering above) where A and B differ.

Let $\mathfrak{S}_n$ be the group of all permutations of $[n]$ (*symmetric group*) and let G be a subgroup of $\mathfrak{S}_q$, acting on $\mathcal{M}_{p,q}$ by permuting columns. Let $\mathcal{M}^{\max}_{p,q}(G)$ be the set of matrices of $\mathcal{M}_{p,q}$ that are $\prec$-maximal within their orbits under the group action G.

We can now define the basic objects of this paper.

Definition 2.1 (Orbitopes).

(1) The *full orbitope* associated with the group G is

$$\mathrm{O}_{p,q}(G) := \mathrm{conv}\ \mathcal{M}^{\max}_{p,q}(G).$$

(2) We associate with the group G the following restricted orbitopes:

$$\mathrm{O}^{\leq}_{p,q}(G) := \mathrm{conv}(\mathcal{M}^{\max}_{p,q}(G) \cap \mathcal{M}^{\leq}_{p,q}) \quad \textit{(packing orbitope)}$$
$$\mathrm{O}^{=}_{p,q}(G) := \mathrm{conv}(\mathcal{M}^{\max}_{p,q}(G) \cap \mathcal{M}^{=}_{p,q}) \quad \textit{(partitioning orbitope)}$$
$$\mathrm{O}^{\geq}_{p,q}(G) := \mathrm{conv}(\mathcal{M}^{\max}_{p,q}(G) \cap \mathcal{M}^{\geq}_{p,q}) \quad \textit{(covering orbitope)}$$

Remark. By definition, $\mathrm{O}^{=}_{p,q}(G)$ is a face of both $\mathrm{O}^{\leq}_{p,q}(G)$ and $\mathrm{O}^{\geq}_{p,q}(G)$.

In this paper, we will be only concerned with the cases of G being the *cyclic group* $\mathfrak{C}_q$ containing all q cyclic permutations of $[q]$ (Section 3) or the symmetric group $\mathfrak{S}_q$ (Section 4). Furthermore, we will restrict attention to packing and partitioning orbitopes. For these, we have the following convenient characterizations of vertices:

Observation.

(1) A matrix of $\mathcal{M}_{p,q}$ is contained in $\mathcal{M}^{\max}_{p,q}(\mathfrak{S}_q)$ if and only if its columns are in non-increasing lexicographic order (with respect to the order $\prec$ defined above).

(2) A matrix of $\mathcal{M}^{\leq}_{p,q}$ is contained in $\mathcal{M}^{\max}_{p,q}(\mathfrak{C}_q)$ if and only if its first column is lexicographically not smaller than the remaining ones (with respect to the order $\prec$).

(3) In particular, a matrix of $\mathcal{M}^{=}_{p,q}$ is contained in $\mathcal{M}^{\max}_{p,q}(\mathfrak{C}_q)$ if and only if it has a 1-entry at position $(1,1)$.

2.1. Optimizing over Orbitopes

The main aim of this paper is to provide complete descriptions of $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$, $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$, $\mathrm{O}^{=}_{p,q}(\mathfrak{C}_q)$, and $\mathrm{O}^{\leq}_{p,q}(\mathfrak{C}_q)$ by systems of linear equations and linear inequalities. If these orbitopes admit "useful" linear descriptions then the corresponding linear optimization problems should be solvable efficiently, due to the equivalence of optimization and separation, see Grötschel, Lovász, and Schrijver [10].

We start with the cyclic group operation, since the optimization problem is particularly easy in this case.

Theorem 2.2. Both the linear optimization problem over $\mathcal{M}^{\max}_{p,q}(\mathfrak{C}_q) \cap \mathcal{M}^{\leq}_{p,q}$ and over $\mathcal{M}^{\max}_{p,q}(\mathfrak{C}_q) \cap \mathcal{M}^{=}_{p,q}$ can be solved in time $O(pq)$.

Proof. We first give the proof for the packing case.

For a vector $c \in \mathbb{Q}^{[p]\times[q]}$, we consider the linear objective function

$$\langle c, x\rangle := \sum_{i=1}^{p}\sum_{j=1}^{q} c_{ij}\, x_{ij}.$$

The goal is to find a matrix $A^\star \in \mathcal{M}^{\max}_{p,q}(\mathfrak{C}_q) \cap \mathcal{M}^{\leq}_{p,q}$ such that $\langle c, A^\star\rangle$ is maximal. Let $A^\star$ be such a c-maximal matrix, and let $a^\star \in \{0,1\}^p$ be its first column. If $a^\star = \mathbf{0}$, then $A^\star = \mathbf{0}$ by Part (2) of Observation 2. By the same observation it follows that if $a^\star \neq \mathbf{0}$ and $i^\star \in [p]$ is the minimum row-index i with $a^\star_i = 1$, then $A^\star$ has only zero entries in its first $i^\star$ rows, except for the 1-entry at position $(i^\star, 1)$ (there is at most one 1-entry in each row). Furthermore, each row $i > i^\star$ of $A^\star$ either has no 1-entry or it has its (unique) 1-entry at some position where c is maximal in row i.

Thus, we can compute an optimal solution as follows: (1) For each $i \in [p]$ determine a vector $b^i \in \{0,1\}^q$ that is the zero vector if c does not have any positive entries in row i and otherwise is the j-th standard unit vector, where $j \in [q]$ is chosen such that $c_{ij} = \max\{c_{i\ell} : \ell \in [q]\}$; set $\sigma_i := 0$ in the first case and $\sigma_i := c_{ij}$ in the second. (2) Compute the values $s_p := \sigma_p$ and $s_i := \sigma_i + s_{i+1}$ for all $i = p-1, p-2, \dots, 1$. (3) Determine $i^\star$ such that $c_{i^\star,1} + s_{i^\star+1}$ is maximal among $\{c_{i,1} + s_{i+1} : i \in [p]\}$. (4) If $c_{i^\star,1} + s_{i^\star+1} \leq 0$, then $\mathbf{0}$ is an optimal solution. Otherwise, the matrix whose i-th row equals b^i for $i \in \{i^\star+1, \dots, p\}$ and which is all-zero in the first $i^\star$ rows, except for a 1-entry at position $(i^\star, 1)$, is optimal.

From the description of the algorithm it is easy to see that its running time is bounded by $O(pq)$ (in the unit-cost model).

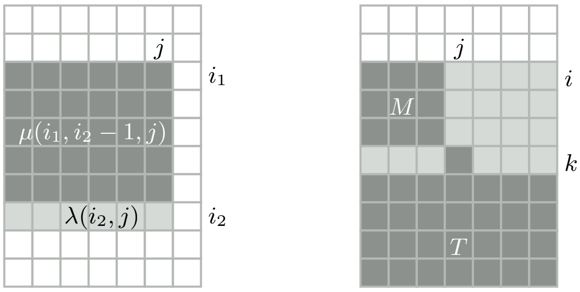


Figure 2: Illustration of the proof of Theorem 2.3. *Left:* Computation of $\mu(i_1, i_2, j)$. *Right:* Computation of $\tau(i, j)$ via the dynamic programming relation (3). Indicated are the matrix $M(i, k-1, j-1)$ and corresponding term $\mu(i, k-1, j-1)$ and matrix $T(k+1, j+1)$ with corresponding term $\tau(k+1, j+1)$.

The partitioning case is then straightforward and even becomes easier due to Part (3) of Observation 2. □

Theorem 2.3. Both the linear optimization problem over $\mathcal{M}^{\max}_{p,q}(\mathfrak{S}_q) \cap \mathcal{M}^{\leq}_{p,q}$ and over $\mathcal{M}^{\max}_{p,q}(\mathfrak{S}_q) \cap \mathcal{M}^{=}_{p,q}$ can be solved in time $O(p^2 q)$.

Proof. We give the proof for the partitioning case, indicating the necessary modifications for the packing case at the relevant points.

As in the proof of Theorem 2.2, we maximize the linear objective function given by $\langle c, x \rangle$ for $c \in \mathbb{Q}^{[p]\times[q]}$. We describe a two-step approach.

In the first step, for $i_1, i_2 \in [p]$ with $i_1 \leq i_2$ and $j \in [q]$, we let $M(i_1, i_2, j)$ be c-maximal among the matrices in $\{0,1\}^{\{i_1, i_1+1, \dots, i_2\}\times[j]}$ with exactly (in the packing case: at most) one 1-entry in every row. Denote by $\mu(i_1, i_2, j)$ the c-value of $M(i_1, i_2, j)$, i.e.,

$$\mu(i_1, i_2, j) = \sum_{k=i_1}^{i_2} \sum_{\ell=1}^{j} c_{k\ell}\, M(i_1, i_2, j)_{k\ell}\,.$$

The values $\mu(i_1, i_2, j)$ can be computed in time $O(p^2 q)$ as follows. First, we compute all numbers $\lambda(i, j) = \max\{c_{i\ell} : \ell \in [j]\}$ (in the packing case: $\lambda(i, j) = \max(0, \{c_{i\ell} : \ell \in [j]\})$) for all $i \in [p]$ and $j \in [q]$. This can clearly be done in $O(pq)$ steps by using the recursions $\lambda(i, j) = \max\{\lambda(i, j-1), c_{ij}\}$ for $j \geq 2$. Then, after initializing $\mu(i, i, j) = \lambda(i, j)$ for all $i \in [p]$ and $j \in [q]$, one computes $\mu(i_1, i_2, j) = \mu(i_1, i_2 - 1, j) + \lambda(i_2, j)$ for all $j \in [q]$, $i_1 = 1, 2, \dots, p$, and $i_2 = i_1 + 1, i_1 + 2, \dots, q$; see Figure 2.

In the second step, for $i \in [p]$ and $j \in [q]$, let $T(i, j)$ be c-maximal among the matrices in $\{0,1\}^{\{i, i+1, \dots, p\}\times[q]}$ with exactly (in the packing case: at most) one 1-entry in every row and with columns $j, j+1, \dots, q$ being in non-increasing lexicographic order. Thus, by Part (1) of Observation 2, $T(1, 1)$ is an optimal solution to our linear optimization problem. Denote by $\tau(i, j)$ the c-value of $T(i, j)$, i.e.,

$$\tau(i, j) = \sum_{k=i}^{p} \sum_{\ell=1}^{q} c_{k\ell}\, T(i, j)_{k\ell}.$$

Let $k \in \{i, i+1, \dots, p+1\}$ be the index of the first row, where $T(i, j)$ has a 1-entry in column j (with $k = p+1$ if there is no such 1-entry); see Figure 2.

Then $T(i,j)$ has a c-maximal matrix T in rows $k+1,\dots,p$ with exactly (in the packing case: at most) one 1-entry per row and lexicographically sorted columns $j+1,\dots,q$ (contributing $\tau(k+1,j+1)$). In row k, there is a single 1-entry at position (k,j) (contributing c_{kj}). And in rows $i,\dots,k-1$, we have a c-maximal matrix M with exactly (in the packing case: at most) one 1-entry per row in the first $j-1$ columns (contributing $\mu(i,k-1,j-1)$) and zeroes in the remaining columns. Therefore, we obtain

$$\tau(i,j) = \mu(i,k-1,j-1) + c_{kj} + \tau(k+1,j+1).$$

Hence, considering all possibilities for k, we have

$$\tau(i,j) = \max\{\, \mu(i,k-1,j-1) + c_{kj} + \tau(k+1,j+1) \;:\; k \in \{i,i+1,\dots,p+1\}\}, \tag{3}$$

for all $i \in [p]$ and $j \in [q]$. For convenience we define $\mu(k_1,k_2,0) = 0$ for $k_1,k_2 \in [p]$ with $k_1 \le k_2$ and $\mu(k,k-1,\ell) = 0$ for all $k \in [p]$ and $\ell \in \{0,1,\dots,q\}$. Furthermore, we set $c_{p+1,\ell} = 0$ for all $\ell \in [q]$. Finally, we define $\tau(p+2,\ell) = \tau(p+1,\ell) = \tau(k,q+1) = 0$ for all $k \in [p]$ and $\ell \in [q+1]$.

Thus, by dynamic programming, we can compute the table $\tau(i,j)$ via Equation (3) in the order $i = p, p-1, \dots, 1$, $j = q, q-1, \dots, 1$. For each pair (i,j) the evaluation of (3) requires no more than $O(p)$ steps, yielding a total running time bound of $O(p^2q)$.

Furthermore, if during these computations for each (i,j) we store a maximizer $k(i,j)$ for k in (3), then we can easily reconstruct the optimal solution $T(1,1)$ from the k-table without increasing the running time asymptotically: For $i \in [p]$, $j \in [q]$ the matrix $T(i,j)$ is composed of $M(i,k(i,j)-1,j-1)$ (if $k(i,j) \ge i+1$ and $j \ge 2$), $T(k(i,j)+1,j+1)$ (if $k(i,j) \le p-1$ and $j \le q-1$), and having 0-entries everywhere else, except for a 1-entry at position $(k(i,j),j)$ (if $k(i,j) \le p$). Each single matrix $M(i_1,i_2,j)$ can be computed in $O((i_2-i_1)j)$ steps. Furthermore, for the matrices $M(i_1,i_2,j)$ needed during the recursive reconstruction of $T(1,1)$, the sets $\{i_1,\dots,i_2\}\times[j]$ are pairwise disjoint (see Figure 2). Thus, these matrices all together can be computed in time $O(pq)$. At the end there might be a single $T(k,q+1)$ to be constructed, which trivially can be done in $O(pq)$ steps. □

Thus, with respect to complexity theory there are no “obstructions” to finding complete linear descriptions of packing and partitioning orbitopes for both the cyclic and the symmetric group action. In fact, for cyclic group actions we will provide such a description in Theorem 3.1 and Theorem 3.2 for the partitioning and packing case, respectively. For symmetric group actions we will provide such a description for partitioning orbitopes in Theorems 4.15 and for packing orbitopes in Theorem 4.16. The algorithm used in the proof of Theorem 2.2 (for cyclic groups) is trivial, while the one described in the proof of Theorem 2.3 (for symmetric groups) is a bit more complicated. This is due to the simpler characterization of the cyclic case in Observation 2 and is reflected by the fact that the proofs of Theorems 4.15 and 4.16 (for symmetric groups) need much more work than the ones of Theorems 3.1 and 3.2 (for cyclic groups).

The algorithms described in the above two proofs heavily rely on the fact that we are considering only matrices with at most one 1-entry per row.

For cyclic group operations, the case of matrices with more ones per row becomes more involved, because we do not have a simple characterization (like the one given in parts 2 and 3 of Observation 2) of the matrices in $\mathcal{M}^{\max}_{p,q}(\mathfrak{C}_q)$ anymore. For the action of the symmetric group, though we still have the characterization provided by Part (1) of Observation 2, the dynamic programming approach used in the proof of Theorem 2.3 cannot be adapted straight-forwardly without resulting in an exponentially large dynamic programming table (unless q is fixed). These difficulties apparently are reflected in the structures of the corresponding orbitopes (see the remarks in Section 5).

3. Packing and Partitioning Orbitopes for Cyclic Groups

From the characterization of the vertices in parts (2) and (3) of Observation 2 one can easily derive IP-formulations of both the partitioning orbitope $\mathrm{O}^{=}_{p,q}(\mathfrak{C}_q)$ and the packing orbitope $\mathrm{O}^{\leq}_{p,q}(\mathfrak{C}_q)$ for the cyclic group $\mathfrak{C}_q$. In fact, it turns out that these formulations do already provide linear descriptions of the two polytopes, i.e., they are totally unimodular. We refer the reader to Schrijver [18, Chap. 19] for more information on total unimodularity.

It is easy to see that for the descriptions given in Theorems 3.1 and 3.2 below, the separation problem can be solved in time $O(pq)$.

Theorem 3.1. The partitioning orbitope $\mathrm{O}^{=}_{p,q}(\mathfrak{C}_q)$ for the cyclic group $\mathfrak{C}_q$ equals the set of all $x \in \mathbb{R}^{[p]\times[q]}$ that satisfy the following linear constraints:

- the equations $x_{11} = 1$ and $x_{1j} = 0$ for all $2 \leq j \leq q$,
- the nonnegativity constraints $x_{ij} \geq 0$ for all $2 \leq i \leq p$ and $j \in [q]$,
- the row-sum equations $x(\mathrm{row}_i) = 1$ for all $2 \leq i \leq p$.

This system of constraints is non-redundant.

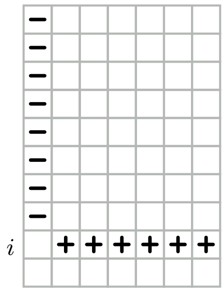
Proof. The constraints $x(\mathrm{row}_i) = 1$ for $i \in [p]$ and $x_{ij} \geq 0$ for $i \in [p], j \in [q]$ define an integral polyhedron, since they describe a transshipment problem (and thus, the coefficient matrix is totally unimodular). Hence, the constraint system given in the statement of the theorem describes an integer polyhedron, because it defines a face of the corresponding transshipment polytope.

By Part (3) of Observation 2, the set of integer points satisfying this constraint system is $\mathcal{M}^{=}_{p,q} \cap \mathcal{M}^{\max}_{p,q}(\mathfrak{C}_q)$. Hence the given constraints completely describe $\mathrm{O}^{=}_{p,q}(\mathfrak{C}_q)$. The non-redundancy follows from the fact that dropping any of the constraints enlarges the set of feasible integer solutions. □

Theorem 3.2. The packing orbitope $\mathrm{O}^{\leq}_{p,q}(\mathfrak{C}_q)$ for the cyclic group $\mathfrak{C}_q$ equals the set of all $x \in \mathbb{R}^{[p]\times[q]}$ that satisfy the following linear constraints:

- the constraints $0 \leq x_{11} \leq 1$ and $x_{1j} = 0$ for all $2 \leq j \leq q$,
- the nonnegativity constraints $x_{ij} \geq 0$ for all $2 \leq i \leq p$ and $j \in [q]$,
- the row-sum inequalities $x(\mathrm{row}_i) \leq 1$ for all $2 \leq i \leq p$,
- the inequalities

$$\sum_{j=2}^{q} x_{ij} - \sum_{k=1}^{i-1} x_{k1} \leq 0 \tag{4}$$



	−						
	−						
	−						
	−						
	−						
	−						
	−						
	−						
i		+	+	+	+	+	+

Figure 3: Example of the coefficient vector for an inequality of type (4); "−" stands for a −1, "+" for a +1.

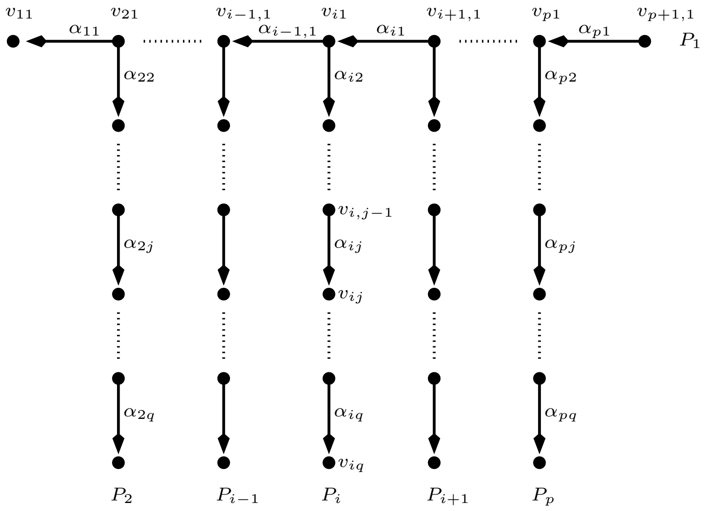


Figure 4: The network matrix constructed in the proof of Theorem 3.2.

for all $2 \leq i \leq p$ (see Figure 3 for an example).

This system of constraints is non-redundant.

Proof. From Part (2) of Observation 2 it follows that an integer point is contained in $\mathrm{O}^{\leq}_{p,q}(\mathfrak{C}_q)$ if and only if it satisfies the constraints described in the statement, where Inequalities (4) ensure that the first column of x is lexicographically not smaller than the other ones (note that we have at most one 1-entry in each row of x). Dropping any of the constraints enlarges the set of integer solutions, which proves the statement on non-redundancy. Thus, as in the proof of the previous theorem, it remains to show that the polyhedron defined by the constraints is integral. We prove this by showing that the coefficient matrix A of the row-sum inequalities $x(\mathrm{row}_i) \leq 1$ (for $2 \leq i \leq p$) and Inequalities (4) (for all $2 \leq i \leq p$) is a network matrix (and thus, totally unimodular). Adding the nonnegativity constraints amounts to adding an identity matrix and preserves total unimodularity, which also holds for the inclusion of $x_{11} \leq 1$ into the system.

In order to establish the claim on the network structure of A, we will identify a directed tree T, whose arcs are in bijection with $[p] \times [q]$ (the set of indices of the columns of A), such that there are pairs of nodes (v_r, w_r)

of T in bijection with the row indices $r \in [2(p-1)]$ of A with the following property. The matrix A has a $(+1)$-entry in row r and column (i,j), if the unique path π_r from node v_r to node w_r in the tree T uses arc (i,j) in its direction from i to j, a (-1)-entry, if π_r uses (i,j) in its reverse direction, and a 0-entry, if π_r does not use (i,j).

For the construction of the tree T, we take a directed path P_1 of length p on nodes $\{v_{11}, v_{21}, \dots, v_{p+1,1}\}$ with arcs $\alpha_{i1} := (v_{i+1,1}, v_{i1})$ for $i \in [p]$; see Figure 4. For each $2 \le i \le p$, we append a directed path P_i of length $q-1$ to node v_{i1}, where P_i has node set $\{v_{i1}, v_{i2}, \dots, v_{iq}\}$ and arcs $\alpha_{ij} := (v_{i,j-1}, v_{ij})$ for $2 \le j \le q$. Choosing the pair $(v_{i+1,1}, v_{iq})$ for the i-th row sum-inequality and the pair (v_{11}, v_{iq}) for the i-th Inequality (4), finishes the proof (using the bijection between the arcs of T and the columns of A indicated by the notation α_{ij}). □

4. Packing and Partitioning Orbitopes for Symmetric Groups

For packing orbitopes $\mathrm{O}^{\le}_{p,q}(\mathfrak{S}_q)$ and partitioning orbitopes $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ with respect to the symmetric group it follows readily from the characterization in Part (1) of Observation 2 that the equations

$$x_{ij} = 0 \qquad \text{for all } i < j \tag{5}$$

are valid. Thus, we may drop all variables corresponding to components in the upper right triangle from the formulation and consider

$$\mathrm{O}^{\le}_{p,q}(\mathfrak{S}_q),\ \mathrm{O}^{=}_{p,q}(\mathfrak{S}_q) \subset \mathbb{R}^{\mathcal{I}_{p,q}} \qquad \text{with} \quad \mathcal{I}_{p,q} := \{(i,j) \in [p] \times [q] \,:\, i \ge j\}.$$

We also adjust the definition of

$$\mathrm{row}_i := \{(i,1), (i,2), \dots, (i, \min\{i,q\})\} \qquad \text{for } i \in [p]$$

and define the jth column for $j \in [q]$ as

$$\mathrm{col}_j := \{(j,j), (j+1,j), \dots, (p,j)\}.$$

Furthermore, we restrict ourselves to the case

$$p \ge q \ge 2$$

in this context. Because of (5), the case of $q > p$ can be reduced to the case $p = q$ and the case of $q = 1$ is of no interest.

The next result shows a very close relationship between packing and partitioning orbitopes for the case of symmetric group actions.

Proposition 4.1. The polytopes $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ and $\mathrm{O}^{\le}_{p-1,q-1}(\mathfrak{S}_{q-1})$ are affinely isomorphic via orthogonal projection of $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ onto the space

$$\mathcal{L} := \{x \in \mathbb{R}^{\mathcal{I}_{p,q}} \,:\, x_{i1} = 0 \text{ for all } i \in [p]\}$$

(and the canonical identification of this space with $\mathbb{R}^{\mathcal{I}_{p-1,q-1}}$).

Proof. The affine subspace

$$\mathcal{A} := \{x \in \mathbb{R}^{\mathcal{I}_{p,q}} \,:\, x(\mathrm{row}_i) = 1 \text{ for all } i\}$$

of $\mathbb{R}^{\mathcal{I}_{p,q}}$ clearly contains $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$. Let $\pi : \mathcal{A} \to \mathbb{R}^{\mathcal{I}_{p-1,q-1}}$ be the orthogonal projection mentioned in the statement (identifying $\mathcal{L}$ in the canonical way with $\mathbb{R}^{\mathcal{I}_{p-1,q-1}}$); note that the first row is removed since it only contains the element $(1,1)$. Consider the linear map $\phi : \mathbb{R}^{\mathcal{I}_{p-1,q-1}} \to \mathbb{R}^{\mathcal{I}_{p,q}}$ defined by

$$\phi(y)_{ij} = \begin{cases} 1 - y(\mathrm{row}_{i-1}) & \text{if } j = 1 \\ y_{i-1,j-1} & \text{otherwise} \end{cases} \qquad \text{for } (i,j) \in \mathcal{I}_{p,q}$$

(where $\mathrm{row}_0 = \varnothing$ and $y(\varnothing) = 0$). This is the inverse of π, showing that π is an affine isomorphism. As we have $\pi(\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)) = \mathrm{O}^{\leq}_{p-1,q-1}(\mathfrak{S}_{q-1})$, this finishes the proof. □

It will be convenient to address the elements in $\mathcal{I}_{p,q}$ via a different "system of coordinates":

$$\langle \eta, j\rangle := (j + \eta - 1, j) \qquad \text{for } j \in [q],\ 1 \leq \eta \leq p - j + 1.$$

Thus (as before) i and j denote the row and the columns, respectively, while η is the index of the diagonal (counted from above) containing the respective element; see Figure 5 (a) for an example. For $(k, j) = \langle \eta, j\rangle$ and $x \in \mathbb{R}^{\mathcal{I}_{p,q}}$, we write $x_{\langle \eta,j\rangle} := x_{(k,j)} := x_{kj}$.

For $x \in \{0,1\}^{\mathcal{I}_{p,q}}$ we denote by $I^x := \{(i,j) \in \mathcal{I}_{p,q} : x_{ij} = 1\}$ the set of all coordinates (positions in the matrix), where x has a 1-entry. Conversely, for $I \subseteq \mathcal{I}_{p,q}$, we use $\chi^I \in \{0,1\}^{\mathcal{I}_{p,q}}$ for the 0/1-point with $\chi^I_{ij} = 1$ if and only if $(i,j) \in I$.

For $(i,j) \in \mathcal{I}_{p,q}$, we define the *column*

$$\mathrm{col}(i,j) = \{(j,j), (j+1,j), \ldots, (i-1,j), (i,j)\} \subseteq \mathcal{I}_{p,q},$$

and for $(i,j) = \langle \eta, j\rangle$ we write $\mathrm{col}\langle \eta, j\rangle := \mathrm{col}(i,j)$. Of course, we have $\mathrm{col}\langle \eta, j\rangle = \{\langle 1,j\rangle, \langle 2,j\rangle, \ldots, \langle \eta,j\rangle\}$.

The rest of this section is organized as follows. First, in Section 4.1, we deal with basic facts about integer points in packing and partitioning orbitopes for the symmetric group. To derive a linear description of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ and $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ that only contains integer vertices, we need additional inequalities, the *shifted column inequalities*, which are introduced in Section 4.2. We then show that the corresponding separation problem can be solved in linear time (Section 4.3). Section 4.4 proves the completeness of the linear description and Section 4.5 investigates the facets of the polytopes.

4.1. Characterization of Integer Points

We first derive a crucial property of the vertices of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$.

Lemma 4.2. Let x be a vertex of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ with $\langle \eta, j\rangle \in I^x$ $(j \geq 2)$. Then we have $I^x \cap \mathrm{col}\langle \eta, j-1\rangle \neq \varnothing$.

Proof. With $\langle \eta, j\rangle = (i,j)$ we have $x_{ij} = 1$, which implies $x_{i,j-1} = 0$ (since x has at most one 1-entry in row i). Thus, $I^x \cap \mathrm{col}\langle \eta, j-1\rangle = \varnothing$ would yield $x_{k,j-1} = 0$ for all $k \leq i$, contradicting the lexicographic order of the columns of x (see Part (1) of Observation 2). □

Definition 4.3 (Column inequality)**.** For $(i,j) \in \mathcal{I}_{p,q}$ and the set $B = \{(i,j), (i,j+1), \ldots, (i, \min\{i,q\})\}$, we call

$$x(B) - x(\mathrm{col}(i-1,j-1)) \leq 0$$

a *column inequality*; see Figure 5 (b) for an example with $(i,j) = (9,5)$.

The column inequalities are strengthenings of the symmetry breaking inequalities

$$x_{ij} - x(\mathrm{col}(i-1, j-1)) \leq 0, \tag{6}$$

introduced by Méndez-Díaz and Zabala [14] in the context of vertex-coloring (see (2) in the introduction).

Proposition 4.4. A point $x \in \{0,1\}^{\mathcal{I}_{p,q}}$ lies in $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ ($\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$) if and only if x satisfies the row-sum constraints $x(\mathrm{row}(i)) \leq 1$ ($x(\mathrm{row}(i)) = 1$) for all $i \in [p]$ and all column inequalities.

Proof. By Lemma 4.2, Inequalities (6) are valid for $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ (and thus, for its face $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ as well). Because of the row-sum constraints, all column inequalities are valid as well. Therefore, it suffices to show that a point $x \in \{0,1\}^{\mathcal{I}_{p,q}}$ that satisfies the row-sum constraints $x(\mathrm{row}(i)) \leq 1$ and all column inequalities is contained in $\mathcal{M}^{\max}_{p,q}(\mathfrak{S}_q)$.

Suppose, this was not the case. Then, by Part (1) of Observation 2, there must be some $j \in [q]$ such that the $(j-1)$-st column of x is lexicographically smaller than the jth column. Let i be minimal with $x_{ij} = 1$ (note that column j cannot be all-zero). Thus, $x_{k,j-1} = 0$ for all $k < i$. This implies $x(\mathrm{col}(i-1, j-1)) = 0 < 1 = x_{ij}$, showing that the column inequality $x(B) - x(\mathrm{col}(i-1, j-1)) \leq 0$ is violated by the point x for the bar $B = \{(i,j), (i,j+1), \ldots, (i, \min\{i,q\})\}$. □

4.2. Shifted Column Inequalities

Proposition 4.4 provides a characterization of the vertices of the packing- and partitioning orbitopes for symmetric groups among the integer points. Different from the situation for cyclic groups (see Theorems 3.1 and 3.2), however, the inequalities in this characterization do not yield complete descriptions of these orbitopes. In fact, we need to generalize the concept of a column inequality in order to arrive at complete descriptions. This will yield exponentially many additional facets (see Proposition 4.13).

Definition 4.5 (Shifted columns)**.** A set $S = \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \ldots, \langle \eta, c_\eta\rangle\} \subset \mathcal{I}_{p,q}$ with $\eta \geq 1$ and $c_1 \leq c_2 \leq \cdots \leq c_\eta$ is called a *shifted column*. It is a *shifting* of each of the columns

$$\mathrm{col}\langle \eta, c_\eta\rangle, \mathrm{col}\langle \eta, c_\eta + 1\rangle, \ldots, \mathrm{col}\langle \eta, q\rangle.$$

Remark.

- As a special case we have column $\mathrm{col}(i,j)$, which is the shifted column $\{\langle 1, j\rangle, \langle 2, j\rangle, \ldots, \langle \eta, j\rangle\}$ for $\langle \eta, j\rangle = (i,j)$.
- By definition, if $S = \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \ldots, \langle \eta, c_\eta\rangle\} \subset \mathcal{I}_{p,q}$ is a shifted column, then so is $\{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \ldots, \langle \eta', c_{\eta'}\rangle\}$ for every $1 \leq \eta' \leq \eta$.

Lemma 4.6. Let x be a vertex of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ with $\langle \eta, j\rangle \in I^x$ ($j \geq 2$). Then we have $I^x \cap S \neq \varnothing$ for all shiftings S of $\mathrm{col}\langle \eta, j-1\rangle$.

Proof. The proof proceeds by induction on j. The case $j = 2$ follows from Lemma 4.2, because the only shifting of $\mathrm{col}\langle \eta, 1\rangle$ is $\mathrm{col}\langle \eta, 1\rangle$ itself. Therefore, let $j \geq 3$, and let $S = \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \ldots, \langle \eta, c_\eta\rangle\}$ be a shifting of $\mathrm{col}\langle \eta, j-1\rangle$ (hence, $c_1 \leq c_2 \leq \cdots \leq c_\eta \leq j-1$). Since by assumption

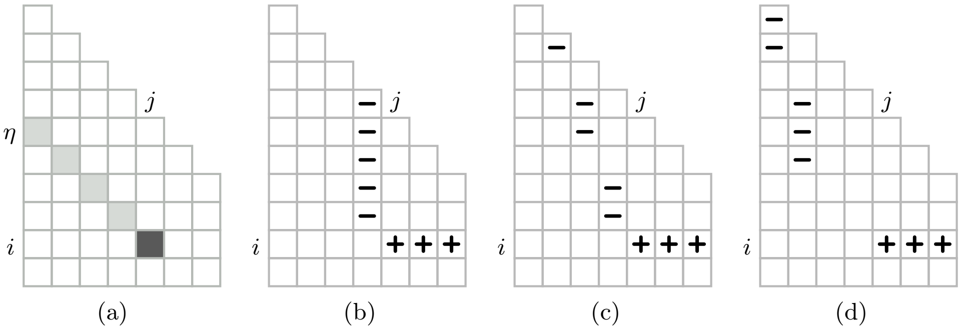


Figure 5: (a) Example for coordinates $(9,5) = \langle 5,5\rangle$. (b)–(d) Shifted column inequalities with leader $\langle 5,5\rangle$, see Definition 4.7. All SCI inequalities are $\leq$-inequalities with right-hand sides zero and "$-$" stands for a (-1)-coefficient, "$+$" for a $(+1)$ coefficient. The shifted column of (c) is $\{\langle 1,2\rangle, \langle 2,3\rangle, \langle 3,3\rangle, \langle 4,4\rangle, \langle 5,4\rangle\}$.

$\langle \eta, j\rangle \in I^x$, Lemma 4.2 yields that there is some $\eta' \leq \eta$ with $\langle \eta', j-1\rangle \in I^x$. If $\langle \eta', j-1\rangle \in S$, then we are done. Otherwise, $c_{\eta'} < j-1$ holds. Therefore, $\{\langle 1,c_1\rangle, \langle 2,c_2\rangle, \dots, \langle \eta', c_{\eta'}\rangle\}$ is a shifting of $(\mathrm{col}\langle \eta', c_{\eta'}\rangle$ and hence of) $\mathrm{col}\langle \eta', j-2\rangle$, which, by the inductive hypothesis, must intersect I^x. □

Definition 4.7 (Shifted column inequalities)**.** For $(i,j) = \langle \eta, j\rangle \in \mathcal{I}_{p,q}$, $B = \{(i,j), (i,j+1), \dots, (i, \min\{i,q\})\}$, and a shifting S of $\mathrm{col}\langle \eta, j-1\rangle$, we call

$$x(B) - x(S) \leq 0$$

a *shifted column inequality (SCI)*. The set B is the *bar* of the SCI, and (i,j) is the *leader* of (the bar of) the SCI. The set S is the *shifted column (SC)* of the SCI. See Figure 5 for examples.

In particular, all column inequalities are shifted column inequalities. The class of shifted column inequalities, however, is substantially richer: It contains exponentially many inequalities (in q).

Proposition 4.8. Shifted column inequalities are valid both for the packing orbitopes $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ and for the partitioning orbitopes $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$.

Proof. As $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ is a face of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$, it is enough to prove the proposition for packing orbitopes $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$. Therefore, let $(i,j) = \langle \eta, j\rangle \in \mathcal{I}_{p,q}$, with $j \geq 2$, and let $S = \{\langle 1,c_1\rangle, \langle 2,c_2\rangle, \dots, \langle \eta, c_\eta\rangle\}$ be a shifting of $\mathrm{col}\langle \eta, j-1\rangle$. Denote by B the bar of the corresponding SCI.

Let $x \in \{0,1\}^{\mathcal{I}_{p,q}}$ be a vertex of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$. If $B \cap I^x = \varnothing$, then clearly $x(B) - x(S) = 0 - x(S) \leq 0$ holds. Otherwise, there is a unique element $(i,j') = \langle \eta', j'\rangle \in B \cap I^x$. As $j' \geq j$, we have $\eta' \leq \eta$. Therefore $S' = \{\langle 1,c_1\rangle, \langle 2,c_2\rangle, \dots, \langle \eta', c_{\eta'}\rangle\} \subseteq S$ is a shifting of $\mathrm{col}\langle \eta', j'-1\rangle$. Thus, by Lemma 4.6, we have $S' \cap I^x \neq \varnothing$. This shows $x(S) \geq x(S') \geq 1$, implying $x(B) - x(S) \leq 1 - 1 = 0$. □

4.3. A Linear Time Separation Algorithm for SCIs

In order to devise an efficient separation algorithm for SCIs, we need a method to compute minimal shifted columns with respect to a given weight

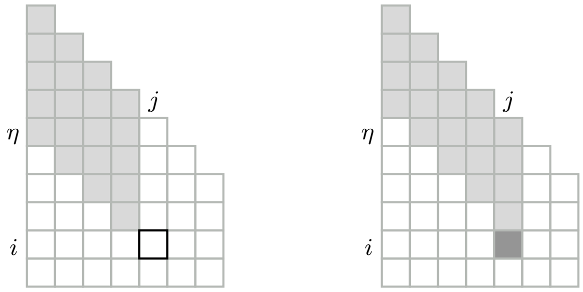


Figure 6: The two cases arising in the dynamic programming algorithm of Section 4.3.

vector $w \in \mathbb{Q}^{\mathcal{I}_{p,q}}$. The crucial observation is the following. Let $S = \{\langle 1,c_1\rangle, \langle 2,c_2\rangle, \dots, \langle \eta, c_\eta\rangle\}$ with $1 \leq c_1 \leq c_2 \leq \cdots \leq c_\eta \leq j$ be a shifting of $\mathrm{col}\langle \eta, j\rangle$ for $\langle \eta, j\rangle \in \mathcal{I}_{p,q}$ with $\eta > 1$. If $c_\eta < j$, then S is a shifting of $\mathrm{col}\langle \eta, j-1\rangle$ (*Case 1*). If $c_\eta = j$, then

$$S - \langle \eta, j\rangle = \{\langle 1,c_1\rangle, \langle 2,c_2\rangle, \dots, \langle \eta-1, c_{\eta-1}\rangle\}$$

is a shifting of $\mathrm{col}\langle \eta-1, j\rangle$ (*Case 2*); see Figure 6.

For all $\langle \eta, j\rangle \in \mathcal{I}_{p,q}$, let $\omega\langle \eta, j\rangle$ be the weight of a w-minimal shifting of $\mathrm{col}\langle \eta, j\rangle$. The table $(\omega\langle \eta, j\rangle)$ can be computed by dynamic programming as follows; we also compute a table of values $\tau\langle \eta, j\rangle \in \{1,2\}$, for each $\langle \eta, j\rangle$, which are needed later to reconstruct the corresponding shifted columns:

(1) For $j = 1, 2, \dots, q$, initialize $\omega\langle 1, j\rangle := \min\{w_{\langle 1,\ell\rangle} : \ell \in [j]\}$.
(2) For $\eta = 2, 3, \dots, p$, initialize $\omega\langle \eta, 1\rangle := \omega\langle \eta-1, 1\rangle + w_{\langle \eta,1\rangle}$.
(3) For $\eta = 2, 3, \dots, p$, $j = 2, 3, \dots, q$ (with $\langle \eta, j\rangle \in \mathcal{I}_{p,q}$): Compute

$$\omega_1 := \omega\langle \eta, j-1\rangle \quad \text{and} \quad \omega_2 := \omega\langle \eta-1, j\rangle + w_{\langle \eta, j\rangle}$$

corresponding to Cases 1 and 2, respectively. Then set

$$\omega\langle \eta, j\rangle = \min\{\omega_1,\ \omega_2\} \quad \text{and} \quad \tau\langle \eta, j\rangle = \begin{cases} 1 & \text{if } \omega_1 \leq \omega_2 \\ 2 & \text{otherwise.} \end{cases}$$

Thus, the tables $(\omega\langle \eta, j\rangle)$ and $(\tau\langle \eta, j\rangle)$ can be computed in time $O(pq)$. Furthermore, for a given $\langle \eta, j\rangle \in \mathcal{I}_{p,q}$, we can compute a w-minimal shifting $S\langle \eta, j\rangle$ of $\mathrm{col}\langle \eta, j\rangle$ in time $O(\eta)$ from the table $(\tau\langle \eta, j\rangle)$: We have $S\langle 1, j\rangle = \{\langle 1, j\rangle\}$ for all $j \in [q]$, $S\langle \eta, 1\rangle = \mathrm{col}\langle \eta, 1\rangle$ for all $\eta \in [p]$, and

$$S\langle \eta, j\rangle = \begin{cases} S\langle \eta, j-1\rangle & \text{if } \tau\langle \eta, j\rangle = 1 \\ S\langle \eta-1, j\rangle \cup \{\langle \eta, j\rangle\} & \text{if } \tau\langle \eta, j\rangle = 2 \end{cases}$$

for all other $\langle \eta, j\rangle$. This proves the following result.

Theorem 4.9. Let $w \in \mathbb{Q}^{\mathcal{I}_{p,q}}$ be a given weight vector. There is an $O(pq)$ time algorithm that simultaneously computes the weights of w-minimal shiftings of $\mathrm{col}\langle \eta, j\rangle$ for all $\langle \eta, j\rangle \in \mathcal{I}_{p,q}$ and a data structure that afterwards, for a given $\langle \eta, j\rangle$, allows to determine a corresponding shifted column in $O(\eta)$ steps.

In particular, we obtain the following:

Corollary 4.10. The separation problem for shifted column inequalities can be solved in linear time $O(pq)$.

Proof. Let a point $x^\star \in \mathbb{Q}^{\mathcal{I}_{p,q}}$ be given. We can compute the $x^\star$-values $\beta(i,j) := x^\star(B(i,j))$ of all bars $B(i,j) = \{(i,j),(i,j+1),\dots,(i,\min\{i,q\})\}$ in linear time in the following way: First, we initialize $\beta(i,\ell) = x^\star_{i\ell}$ for all $i \in [p]$ and $\ell = \min\{i,q\}$. Then, for each $i \in [p]$, we calculate the value $\beta(i,j) = x^\star_{ij} + \beta(i,j+1)$ for $j = \min\{i,q\}-1, \min\{i,q\}-2,\dots,1$.

Using Theorem 4.9 (and the notations introduced in the paragraphs preceding it), we compute the table $(\omega\langle\eta,j\rangle)$ and the mentioned data structure in time $O(pq)$. Then in time $O(pq)$ we check whether there exists an $(i,j) = \langle\eta,j\rangle \in \mathcal{I}_{p,q}$ with $j \geq 2$ and $\omega\langle\eta,j-1\rangle < \beta(i,j)$. If there exists such an $\langle\eta,j\rangle$, we compute the corresponding shifted column $S\langle\eta,j-1\rangle$ (in additional time $O(\eta) \subseteq O(p)$), yielding an SCI that is violated by $x^\star$. Otherwise $x^\star$ satisfies all SCIs. □

Of course, the procedure described in the proof of the corollary can be modified to find a maximally violated SCI if $x^\star$ does not satisfy all SCIs.

4.4. Complete Inequality Descriptions

In this section we prove that nonnegativity constraints, row-sum equations, and SCIs suffice to describe partitioning and packing orbitopes for symmetric groups. The proof will be somewhat more involved than in the case of cyclic groups. In particular, the coefficient matrices are not totally unimodular anymore. In order to see this, consider the three column inequalities

$$x_{3,3} - x_{2,2} \leq 0, \quad x_{4,3} + x_{4,4} - x_{2,2} - x_{3,2} \leq 0, \quad \text{and}$$
$$x_{5,4} + x_{5,5} - x_{3,3} - x_{4,3} \leq 0.$$

The submatrix of the coefficient matrix belonging to these three rows and the columns corresponding to $(2,2)$, $(3,3)$, and $(4,3)$ is the matrix

$$\begin{pmatrix} -1 & +1 & 0 \\ -1 & 0 & +1 \\ 0 & -1 & -1 \end{pmatrix},$$

whose determinant equals -2. Note that the above three inequalities define facets both of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ and $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ for $p \geq q \geq 5$ (see Propositions 4.13 and 4.14, respectively).

Proposition 4.11. The partitioning orbitope $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ is completely described by the nonnegativity constraints, the row-sum equations, and the shifted column inequalities:

$$\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q) = \{\, x \in \mathbb{R}^{\mathcal{I}_{p,q}} \;:\; x \geq \mathbf{0},\; x(\mathrm{row}_i) = 1 \text{ for } i = 1,\dots,p,$$
$$x(B) - x(S) \leq 0 \text{ for all SCIs with SC } S \text{ and bar } B \,\}.$$

Proof. Let P be the polyhedron on the right-hand side of the statement above. From Propositions 4.4 and 4.8 we know already that

$$P \cap \mathbb{Z}^{\mathcal{I}_{p,q}} = \mathrm{O}^{=}_{p,q}(\mathfrak{S}_q) \cap \mathbb{Z}^{\mathcal{I}_{p,q}}$$

holds. Thus, it suffices to show that P is an integral polytope (as $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ is by definition). In the following, we first describe the strategy of the proof.

For the rest of the proof, fix an arbitrary vertex $x^\star$ of P. A *basis* $\mathcal{B}$ of $x^\star$ is a cardinality $|\mathcal{I}_{p,q}|$ subset of the constraints describing P that are satisfied with equality by $x^\star$ with the property that the $|\mathcal{I}_{p,q}|\times|\mathcal{I}_{p,q}|$-coefficient matrix of the left-hand sides of the constraints in $\mathcal{B}$ is non-singular. Thus, the equation system obtained from the constraints in $\mathcal{B}$ has $x^\star$ as its unique solution.

We will show that there exists a basis $\mathcal{B}^\star$ of $x^\star$ that does not contain any SCI. Thus, $\mathcal{B}^\star$ contains a subset of the p row-sum equations and at least $|\mathcal{I}_{p,q}| - p$ nonnegativity constraints. This shows that $x^\star$ has at most p nonzero entries and, since $x^\star$ satisfies the row-sum equations, it has a nonzero entry in every row. Therefore, $\mathcal{B}^\star$ contains all p row-sum equations, and all p nonzero entries must in fact be 1. Hence, $x^\star$ is a 0/1-point. So the existence of such a basis proves the proposition.

The *weight* of a shifted column $S = \{\langle 1, c_1\rangle, \langle 2, c_2, ,\rangle \dots, \langle \eta, c_\eta\rangle\}$ with $1 \le c_1 \le c_2 \le \cdots \le c_\eta < q$ (we will not need shifted columns with $c_\eta = q$ here, as they do not appear in SCIs) is

$$\mathrm{weight}(S) := \sum_{i=1}^{\eta} c_i\, q^i.$$

In particular, if S_1 and S_2 are two shifted columns with $|S_1| < |S_2|$, then we have $\mathrm{weight}(S_1) < \mathrm{weight}(S_2)$. The *weight* of an SCI is the weight of its shifted column, and the *weight* of a basis $\mathcal{B}$ is the sum of the weights of the SCIs contained in $\mathcal{B}$ (note that a shifted column can appear in several SCIs).

A basis of $x^\star$ that contains all row-sum equations and all nonnegativity constraints corresponding to 0-entries of $x^\star$ is called *reduced*. As the coefficient vectors (of the left-hand sides) of these constraints are linearly independent, some reduced basis of $x^\star$ exists. Hence, there is also a reduced basis $\mathcal{B}^\star$ of $x^\star$ of minimal weight.

To prove the proposition, it thus suffices to establish the following claim.

Claim. A reduced basis of $x^\star$ of minimal weight does not contain any SCI.

The proof of Claim 4.4 consists of three parts:

(1) We show that a reduced basis of $x^\star$ does not contain any "trivial SCIs" (Claim 4.4).

(2) We prove that a reduced basis of $x^\star$ of minimal weight satisfies three structural conditions on its (potential) SCIs (Claim 4.4).

(3) Finally, assuming that a reduced basis of $x^\star$ with minimal weight contains at least one SCI, we will derive a contradiction by constructing a different solution $\tilde{x} \neq x^\star$ of the corresponding equation system.

We are now ready to start with Part 1. We call an SCI with shifted column S *trivial* if $x^\star(S) = 0$ holds or if we have $x^\star(S) = 1$ and $x^\star_{k\ell} = 0$ for all $(k,\ell) \in S - (i,j)$ for some $(i,j) \in S$ (thus satisfying $x^\star_{ij} = 1$) (see Figure 7 (a)).

Claim. A reduced basis $\mathcal{B}$ of $x^\star$ does not contain any trivial SCIs.

Proof. Let S be the shifted column S and B be the bar of some SCI that is satisfied with equality by $x^\star$.

If $x^\star(S) = 0$, then the coefficient vector of the SCI is a linear combination of the coefficient vectors of the inequalities $x_{ij} \ge 0$ for $(i,j) \in S \cup B$, which all are contained in $\mathcal{B}$ (due to $x^\star(B) = x^\star(S) = 0$). Since the coefficient

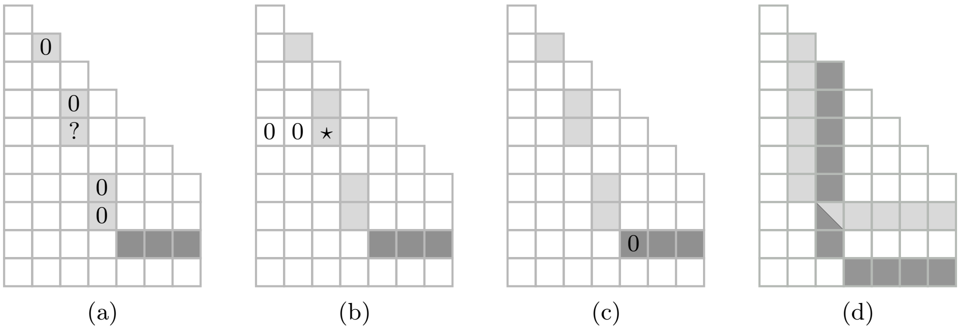


Figure 7: Illustration of trivial SCIs and of the three types of configurations not present in reduced bases of minimal weight, see Claim 4.4. Bars are shown in dark gray, shifted columns in light gray. Figure (a) shows trivial SCIs ("?" refers to a 0 or 1). Figures (b), (c), and (d) refer to parts (1), (2), and (3) of Claim 4.4, respectively ("$\star$" indicates any nonzero number).

vectors of the inequalities in $\mathcal{B}$ form a non-singular matrix, the SCI can not be in $\mathcal{B}$. (By "coefficient vector" we always mean the vector formed by the coefficients of the left-hand side of a constraint.)

If S contains exactly one entry $(k,\ell) \in S$ with $x^\star_{k\ell} = 1$, then we have $x^\star(S) = x^\star(B) = 1$. Let i be the index of the row that contains the bar B. The nonnegativity constraints $x_{rs} \geq 0$ for $(r,s) \in S - (k,\ell)$, $x_{ks} \geq 0$ for $(k,s) \in \text{row}_k - (k,\ell)$, and $x_{is} \geq 0$ for $(i,s) \in \text{row}_i \setminus B$ are contained in $\mathcal{B}$.

Since the coefficient vector of the considered SCI can linearly be combined from the coefficient vectors of these nonnegativity constraints and of the row-sum equations $x(\text{row}_k) = 1$ and $x(\text{row}_i) = 1$, this SCI cannot be contained in $\mathcal{B}$. □

Claim. A minimal weight reduced basis $\mathcal{B}$ of $x^\star$ satisfies the following three conditions:

(1) If (k,ℓ) is contained in the shifted column of some SCI in $\mathcal{B}$, then there exists some $s < \ell$ with $x^\star_{ks} > 0$.
(2) If (i,j) is the leader of an SCI in $\mathcal{B}$, then $x^\star_{ij} > 0$ holds.
(3) If (i,j) is the leader of an SCI in $\mathcal{B}$, then there is no SCI in $\mathcal{B}$ whose shifted column contains (i,j).

See Figure 7, (b)–(d) for an illustration of the three conditions.

Proof. Part (1): Assume there exists an SCI in $\mathcal{B}$ with shifted column S and bar B that contains the first nonzero entry of a row k, i.e., there is $(k,\ell) \in S$ with $x^\star_{k\ell} > 0$ and $x^\star_{ks} = 0$ for all $s < \ell$. Let $S' := S \cap \mathcal{I}_{k-1,q}$ be the entries of S above row k. Let $C = \{(k,1),(k,2),\dots,(k,\ell-1)\}$ and $B' = \text{row}_k \setminus (C + (k,\ell))$. See Figure 8 (1) for an illustration.

Because S' is a shifting of $\text{col}(k-1,\ell)$, $x(B') - x(S') \leq 0$ is an SCI and hence satisfied by $x^\star$. Since we have $|S'| < |S|$ (thus, $\text{weight}(S') < \text{weight}(S)$), it suffices to show that replacing the original SCI $x(B)-x(S) \leq 0$ by $x(B') - x(S') \leq 0$ gives another basis $\mathcal{B}'$ of $x^\star$ (which also is reduced), contradicting the minimality of the weight of $\mathcal{B}$.

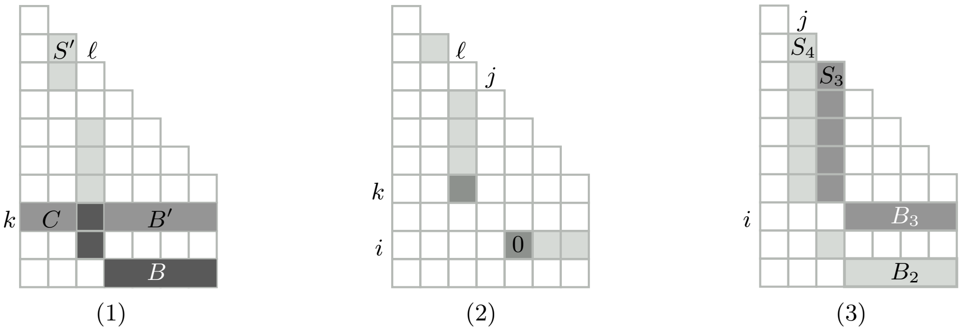


Figure 8: Illustration of the proof of Claim 4.4, parts (1) to (3).

Due to $x^\star(\text{row}_k) = 1$, $x^\star(C) = 0$, $x^\star(B') - x^\star(S') \leq 0$, and $S' + (k, \ell) \subseteq S$ we have

$$1 = x^\star_{k\ell} + x^\star(B') \leq x^\star_{k\ell} + x^\star(S') \leq x^\star(S) = x^\star(B) \leq 1. \tag{7}$$

Therefore, equality must hold throughout this chain. In particular, this shows $x^\star(B') - x^\star(S') = 0$. Thus, its suffices to show that the coefficient matrix of the equation system obtained from $\mathcal{B}'$ is non-singular, which can be seen as follows.

Since $x^\star(S' + (k, \ell)) = 1 = x^\star(S)$ (see (7)), we know that all nonnegativity constraints $x_{rs} \geq 0$ with $(r, s) \in S \setminus (S' + (k, \ell))$ are contained in $\mathcal{B}$ and $\mathcal{B}'$. The same holds for $x_{ks} \geq 0$ with $(k, s) \in C$ and for $x_{is} \geq 0$ with $(i, s) \in \text{row}_i \setminus B$, where row i contains bar B (since $x^\star(B) = 1$ by (7)). Thus, we can linearly combine the coefficient vector of $x(B) - x(S) \leq 0$ from the coefficient vectors of the constraints $x(B') - x(S') \leq 0$, $x(\text{row}_k) = 1$, $x(\text{row}_i) = 1$, and the nonnegativity constraints mentioned above. Since all these constraints are contained in $\mathcal{B}'$, this shows that the coefficient matrix of $\mathcal{B}'$ has the same row-span as that of $\mathcal{B}$, thus proving that it is non-singular as well.

Part (2): Assume that there exists an SCI in $\mathcal{B}$ with leader (i, j), bar B, and shifted column S such that $x^\star_{ij} = 0$. If $S = \{\langle 1, c_1 \rangle, \langle 2, c_2 \rangle, \dots, \langle \eta, c_\eta \rangle\}$, then we have $(i, j) = \langle \eta, j \rangle$. Define $B' := B - (i, j)$, $S' := S - \langle \eta, c_\eta \rangle$, and observe that $B' \neq \varnothing$, $S' \neq \varnothing$, i.e., $|B| > 1$ and $|S| > 1$, because a reduced basis does not contain trivial SCIs by Claim 4.4; see Figure 8 (2). Hence, $x(B') - x(S') \leq 0$ is an SCI. We therefore have:

$$0 = x^\star(B) - x^\star(S) = x^\star(B') - x^\star(S) \leq x^\star(B') - x^\star(S') \leq 0, \tag{8}$$

where the first equation holds because $x(B) - x(S) \leq 0$ is satisfied with equality by $x^\star$ and the second equation follows from $x^\star_{ij} = 0$. Hence, we know that $x^\star(B') - x^\star(S') = 0$. Since we have $|S'| < |S|$ (and consequently $\text{weight}(S') < \text{weight}(S)$), again it remains to show that the coefficient vector of $x(B) - x(S) \leq 0$ can be linearly combined from the coefficient vector of $x(B') - x(S') \leq 0$ and some coefficient vectors of nonnegativity constraints in $\mathcal{B}$ and $\mathcal{B}'$. But this is clear, as we have $x^\star_{ij} = 0$ and $x^\star_{\langle \eta, c_\eta \rangle} = 0$, where the latter follows from (8).

Part (3): Assume that in $\mathcal{B}$ there exists an SCI

$$x(B_1) - x(S_1) \leq 0 \tag{9}$$

with leader $(i, j) = \langle \eta, j \rangle$, bar B_1, and shifted column

$$S_1 = \{\langle 1, c_1 \rangle, \langle 2, c_2 \rangle, \dots, \langle \eta, c_\eta \rangle\}$$

(in particular: $c_\eta < j$) and another SCI

$$x(B_2) - x(S_2) \leq 0 \tag{10}$$

with bar B_2 and shifted column

$$S_2 = \{\langle 1, d_1 \rangle, \langle 2, d_2 \rangle, \dots, \langle \eta, j \rangle, \langle \eta + 1, d_{\eta+1} \rangle, \dots, \langle \tau, d_\tau \rangle\}.$$

Hence, we have $(i, j) = \langle \eta, j \rangle \in S_2$. Define

$$S_3 := \{\langle 1, d_1 \rangle, \langle 2, d_2 \rangle, \dots, \langle \eta - 1, d_{\eta-1} \rangle\}$$

(i.e, the part of S_2 lying strictly above row i) and

$$S_4 := \{\langle 1, c_1 \rangle, \dots, \langle \eta, c_\eta \rangle, \langle \eta + 1, d_{\eta+1} \rangle, \dots, \langle \tau, d_\tau \rangle\}$$

(i.e, S_1 together with the part of S_2 strictly below row i). Clearly, S_3 is a shifting of $\operatorname{col}\langle \eta - 1, j \rangle = \operatorname{col}(i - 1, j)$, and S_4 is a shifted column as well (due to $c_\eta < j \leq d_{\eta+1}$). Thus, with $B_3 = B_1 - (i, j)$, we obtain the SCIs

$$x(B_3) - x(S_3) \leq 0 \tag{11}$$

$$x(B_2) - x(S_4) \leq 0 \tag{12}$$

(see Figure 8 (3)).

Since (9) and (10) are contained in $\mathcal{B}$, we have $x^\star(B_1) - x^\star(S_1) = 0$ and $x^\star(B_2) - x^\star(S_2) = 0$. Adding these two equations yields

$$\big(x^\star(B_3) - x^\star(S_3)\big) + \big(x^\star(B_2) - x^\star(S_4)\big) = 0, \tag{13}$$

because $x^\star_{ij}$ cancels due to $(i, j) \in B_1 \cap S_2$. Since $x^\star$ satisfies the SCIs (11) and (12), Equation (13) shows that in fact we have $x^\star(B_3) - x^\star(S_3) = 0$ and $x^\star(B_2) - x^\star(S_4) = 0$.

It is not clear, however, that we can simply replace (9) and (10) by (11) and (12) in order to obtain a new basis of $x^\star$. Nevertheless, if v_1, v_2, v_3, and v_4 are the coefficient vectors of (9), (10), (11), and (12), respectively, we have $v_1 + v_2 = v_3 + v_4$, which implies

$$v_2 = v_3 + v_4 - v_1. \tag{14}$$

Let $V \subset \mathbb{R}^{\mathcal{I}_{p,q}}$ be the subspace of $\mathbb{R}^{\mathcal{I}_{p,q}}$ that is spanned by the coefficient vectors of the constraints different from (10) in $\mathcal{B}$. Thus, the linear span of $V \cup \{v_2\}$ is the whole space $\mathbb{R}^{\mathcal{I}_{p,q}}$. Due to (14), the same holds for $V \cup \{v_3, v_4\}$ (since $v_1 \in V$). Therefore, there is $\alpha \in \{3, 4\}$ such that $V \cup \{v_\alpha\}$ spans $\mathbb{R}^{\mathcal{I}_{p,q}}$. Let (a) be the corresponding SCI from $\{(11), (12)\}$. Hence, $\mathcal{B}' := \mathcal{B} \setminus \{(10)\} \cup \{(a)\}$ is a (reduced) basis of $x^\star$ as well.

Since we have $|S_3| < |S_2|$ and $\operatorname{weight}(S_4) < \operatorname{weight}(S_2)$ (due to $c_\eta < j$), the weight of $\mathcal{B}'$ is smaller than that of $\mathcal{B}$, contradicting the minimality of the weight of $\mathcal{B}$. □

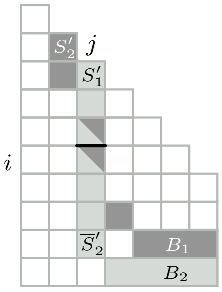


Figure 9: Illustration of sets used in the proof of Claim 4.4.

Before we finish the proof of the proposition by establishing Claim 4.4, we need one more structural result on the SCIs in a reduced basis of $x^\star$. Let $S = \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \dots, \langle \eta, c_\eta\rangle\}$ be any shifted column with $x^\star_{\langle \gamma, c_\gamma\rangle} > 0$ for some $\gamma \in [\eta]$. We call $\langle \gamma, c_\gamma\rangle$ the *first nonzero element* of S if

$$x^\star_{\langle 1,c_1\rangle} = \dots = x^\star_{\langle \gamma-1,c_{\gamma-1}\rangle} = 0$$

holds. Similarly, $\langle \gamma, c_\gamma\rangle$ is called the *last nonzero element* of S if we have

$$x^\star_{\langle \gamma+1,c_{\gamma+1}\rangle} = \dots = x^\star_{\langle \eta,c_\eta\rangle} = 0.$$

Claim. Let $\mathcal{B}$ be a reduced basis of $x^\star$, and let S_1, S_2 be the shifted columns of some SCIs in $\mathcal{B}$ ($S_1 = S_2$ is allowed).

(1) If (i, j) is the first nonzero element of S_1 and $(i, j) \in S_2$, then (i, j) is also the first nonzero element of S_2.

(2) If (i, j) is the last nonzero element of S_1 with $x^\star(S_1) = 1$ and $(i, j) \in S_2$, then (i, j) is also the last nonzero element of S_2 and $x^\star(S_2) = 1$.

(3) If (i, j) is the last nonzero element of S_1 with $x^\star(S_1) = 1$, then (i, j) is not the first nonzero element of S_2.

Proof. Let

$$S_1 = \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \dots, \langle \eta, c_\eta\rangle\} \quad \text{and} \quad S_2 = \{\langle 1, d_1\rangle, \langle 2, d_2\rangle, \dots, \langle \tau, d_\tau\rangle\}$$

be two shifted columns of SCIs with bars B_1 and B_2, respectively, in the reduced basis $\mathcal{B}$ of $x^\star$. Suppose that $(i, j) = \langle \gamma, j\rangle \in S_1 \cap S_2$, i.e., $c_\gamma = j = d_\gamma$ holds. Define

$$S_1' := \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \dots, \langle \gamma-1, c_{\gamma-1}\rangle\},$$
$$S_2' := \{\langle 1, d_1\rangle, \langle 2, d_2\rangle, \dots, \langle \gamma-1, d_{\gamma-1}\rangle\},$$

and $\overline{S}_2' := S_2 \setminus S_2'$, see Figure 9. Since $\langle \gamma, j\rangle \in S_1 \cap S_2$ holds, $S_1' \cup \overline{S_2'}$ is a shifted column and $x(B_2) - x(S_1' \cup \overline{S_2'}) \le 0$ is an SCI. Thus, we obtain

$$x^\star(B_2) - x^\star(S_1') - x^\star(\overline{S}_2') \le 0. \tag{15}$$

Furthermore, since $x(B_2) - x(S_2) \le 0$ is contained in the basis $\mathcal{B}$ of $x^\star$, we have

$$x^\star(B_2) - x^\star(S_2') - x^\star(\overline{S}_2') = 0. \tag{16}$$

Subtracting (16) from (15) yields $x^\star(S_2') - x^\star(S_1') \le 0$. We thus conclude

$$x^\star(S_2') \le x^\star(S_1') \quad \text{and} \quad x^\star(S_1') \le x^\star(S_2') \tag{17}$$

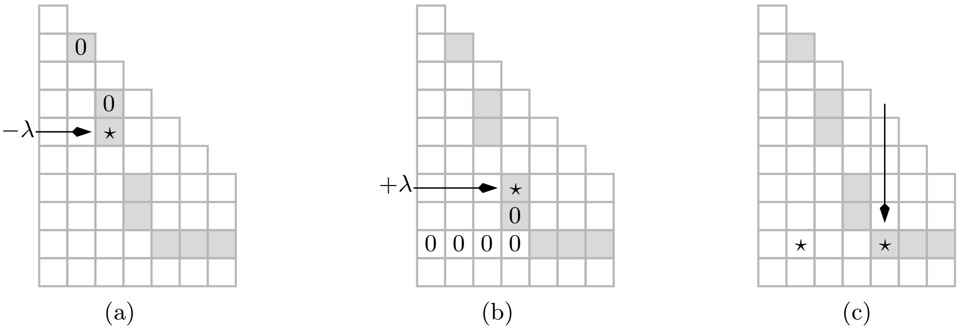


Figure 10: Illustration of the construction of $\tilde{x}$, Steps (1) to (3).

(where the second inequality follows by exchanging the roles of S_1 and S_2 in the argument).

Part (1): If (i,j) is the first nonzero element of S_1, then we have $x^\star(S_1') = 0$. Thus, the first inequality of (17) implies $x^\star(S_2') = 0$, showing that (i,j) is the first nonzero element of S_2.

Part (2): If (i,j) is the last nonzero element of S_1 and $x^\star(S_1) = 1$ holds, then we have $x^\star(S_1' + (i,j)) = 1$. With the second inequality of (17) we obtain:

$$1 = x^\star(S_1' + (i,j)) \le x^\star(S_2' + (i,j)) \le x^\star(S_2) = x^\star(B_2) \le 1,$$

where the last equation holds because $x(B_2) - x(S_2) \le 0$ is contained in $\mathcal{B}$. It follows that $x^\star(S_2) = 1$ and (i,j) is the last nonzero element of S_2.

Part (3): This follows from the first two parts of the claim, since $\mathcal{B}$ does not contain any trivial SCIs by Claim 4.4. □

We will now proceed with the proof of Claim 4.4. Thus, assume that $\mathcal{B}^\star$ is a reduced basis of $x^\star$ of minimal weight and suppose that $\mathcal{B}^\star$ contains at least one SCI. We are going to construct a point $\tilde{x} \neq x^\star$ that satisfies the equation system obtained from $\mathcal{B}^\star$, contradicting the fact the $x^\star$ is the unique solution to this system of equations.

At the beginning, we set $\tilde{x} = x^\star$, and let $\lambda > 0$ be an arbitrary positive number. Then we perform the following four steps (see Figure 10 for illustrations of the first three).

(1) For every (i,j) that is the first nonzero element of the shifted column of at least one SCI in $\mathcal{B}^\star$, we reduce $\tilde{x}_{ij}$ by λ.

(2) For every (i,j) that is the last nonzero element of the shifted column S of at least one SCI in $\mathcal{B}^\star$ with $x^\star(S) = 1$, we increase $\tilde{x}_{ij}$ by λ.

(3) For each $i \in [p]$ and for all $j = \min\{i,q\}, \min\{i,q\} - 1, \dots, 1$ (in this order): If (i,j) is the leader of some SCI in $\mathcal{B}^\star$, we adjust $\tilde{x}_{ij}$ such that, with $B = \{(i,j),(i,j+1),\dots,(i,\min\{i,q\})\}$,

$$\tilde{x}(B) = \begin{cases} 1 & \text{if } x^\star(B) = 1 \\ x^\star(B) - \lambda & \text{otherwise} \end{cases}$$

holds.

(4) For each $i \in [p]$, adjust $\tilde{x}_{ij}$ in order to achieve $\tilde{x}(\mathrm{row}_i) = 1$, where $j = \min\{\ell : x^\star_{i\ell} > 0\}$.

The reason for treating the case $x^\star(S) = 1$ separately in Step 2 will become evident in the proof of Claim 4.4 below.

The following four claims will yield that $\tilde{x}$ is a solution of the equation system corresponding to $\mathcal{B}^\star$.

Claim. After Step 2, for each shifted column S of some SCI in $\mathcal{B}^\star$ we have

$$\tilde{x}(S) = \begin{cases} 1 & \text{if } x^\star(S) = 1 \\ x^\star(S) - \lambda & \text{otherwise.} \end{cases}$$

Proof. Let S be the shifted column of some SCI in $\mathcal{B}^\star$. It follows from Part (1) of Claim 4.4 that the first nonzero element (i, j) of S is the only element in S whose $\tilde{x}$-component is changed (reduced by λ) in Step 1. Thus, after Step 1 we have $\tilde{x}(S) = x^\star(S) - \lambda$.

If $x^\star(S) < 1$, then, by Part (2) of Claim 4.4, $\tilde{x}(S)$ is not changed in Step 2. Otherwise, $x^\star(S) = 1$, and $\tilde{x}_{k\ell}$ is increased by λ in Step 2, where (k, ℓ) is the last nonzero element of S. According to Part (2) of Claim 4.4, no other component of $\tilde{x}$ belonging to some element in S is changed in Step 2. Thus, in both cases the claim holds. □

Claim. No component of $\tilde{x}$ belonging to the shifted column of some SCI in $\mathcal{B}^\star$ is changed in Step 3.

Proof. Let S be the shifted column of some SCI in $\mathcal{B}^\star$. According to Part (3) of Claim 4.4, S does not contain the leader of any SCI in $\mathcal{B}^\star$, since $\mathcal{B}^\star$ is a reduced basis of minimal weight. □

Claim. After Step 3, for each SCI in $\mathcal{B}^\star$ with shifted column S and bar B we have $\tilde{x}(S) = \tilde{x}(B)$.

Proof. For an SCI in $\mathcal{B}^\star$ with shifted column S and bar B, we have $x^\star(S) = x^\star(B)$. Thus, from Claims 4.4 and 4.4 it follows that $\tilde{x}(S) = \tilde{x}(B)$ holds after Step 3. □

Claim. Step 4 does not change any component of $\tilde{x}$ that belongs to the shifted column or the bar of some SCI in $\mathcal{B}^\star$.

Proof. Let (i, j) be such that $x^\star_{i\ell} = 0$ for all $\ell < j$ and $x^\star_{ij} > 0$. By Part (1) of Claim 4.4, (i, j) is not contained in any shifted column of an SCI in $\mathcal{B}^\star$. If (i, j) is contained in the bar B of some SCI in $\mathcal{B}^\star$, then clearly $x^\star(B) = 1$ holds. Thus, after Step 3, we have $\tilde{x}(\mathrm{row}_i) = \tilde{x}(B) = 1$, which shows that $\tilde{x}_{ij}$ is not changed in Step 4. □

We can now finish the proof of the proposition. Claims 4.4 and 4.4 show that $\tilde{x}$ satisfies all SCIs contained in $\mathcal{B}^\star$ with equality. Furthermore, in all steps of the procedure only components $\tilde{x}_{ij}$ with $x^\star_{ij} > 0$ are changed (this is clear for Steps 1, 2, and 4; for Step 3 it follows from Part (2) of Claim 4.4). Since after Step 4, $\tilde{x}$ satisfies all row-sum equations, this proves that $\tilde{x}$ is a solution to the equation system obtained from $\mathcal{B}^\star$.

We assumed that $\mathcal{B}^\star$ contains at least one SCI. Let S be the shifted column of one of these. We know $x^\star(S) > 0$ by Claim 4.4. Thus, let (i, j) be the first nonzero element of S. Hence, after Step 1, we have $\tilde{x}_{ij} = x^\star_{ij} - \lambda$. By Part (3) of Claim 4.4, this still holds after Step 2. As $\tilde{x}_{ij}$ is also not changed in Steps 3 and 4 (see Claims 4.4 and 4.4), we deduce $\tilde{x} \neq x^\star$, contradicting the fact that $x^\star$ is the unique solution to the equation system belonging to $\mathcal{B}^\star$.

This concludes the proof of Proposition 4.11. □

We hope that reading this proof was somewhat enjoyable. Anyway, at least it also gives us a linear description of the packing orbitopes for symmetric groups almost for free.

Proposition 4.12. The packing orbitope $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ is completely described by the nonnegativity constraints, the row-sum inequalities, and the shifted column inequalities:

$$\begin{aligned}\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q) = \{\, x \in \mathbb{R}^{\mathcal{I}_{p,q}} \;:\; & x \geq \mathbf{0},\; x(\mathrm{row}_i) \leq 1 \text{ for } i = 1, \dots, p, \\ & x(B) - x(S) \leq 0 \text{ for all SCIs with SC } S \text{ and bar } B \,\}.\end{aligned}$$

Proof. Let $Q \subset \mathbb{R}^{\mathcal{I}_{p,q}}$ be the polyhedron on the right-hand side of the statement. We define $\mathcal{A} := \{x \in \mathbb{R}^{\mathcal{I}_{p+1,q+1}} \;:\; x(\mathrm{row}_i) = 1 \text{ for all } i \in [p+1]\}$.

The proof of Proposition 4.11 in fact shows that its statement remains true if we drop all SCIs with shifted column S and $S \cap \mathrm{col}_1 \neq \varnothing$ from the linear description. This follows from the fact that, due to $x^\star_{11} = 1$ and Claim 4.4, no such SCI can be contained in any reduced basis of $x^\star$ (using the notations from the proof of Proposition 4.11). Thus we obtain

$$\mathrm{O}^{=}_{p+1,q+1}(\mathfrak{S}_{q+1}) = \mathcal{A} \cap \tilde{Q}, \tag{18}$$

with

$$\begin{aligned}\tilde{Q} = \{x \in \mathbb{R}^{\mathcal{I}_{p+1,q+1}} \;:\; & x(B) - x(S) \leq 0 \text{ for all SCIs with bar } B \\ & \text{and shifted column } S \text{ with } S \cap \mathrm{col}_1 = \varnothing, \\ & x_{ij} \geq 0 \text{ for all } (i,j) \in \mathcal{I}_{p+1,q+1} \setminus \mathrm{col}_1, \\ & x(\mathrm{row}_i - (i,1)) \leq 1 \text{ for all } i = 2, \dots, p+1\},\end{aligned}$$

where the last inequalities are equivalent (with respect to $\mathrm{O}^{=}_{p+1,q+1}(\mathfrak{S}_{q+1})$) to the nonnegativity constraints associated with the elements of col_1 by addition of row-sum equations.

Define $\mathcal{L} := \{x \in \mathbb{R}^{\mathcal{I}_{p+1,q+1}} \;:\; x_{i1} = 0 \text{ for all } i \in [p+1]\}$, and denote by $\tilde{\pi} : \mathbb{R}^{\mathcal{I}_{p+1,q+1}} \to \mathcal{L}$ the orthogonal projection. Since none of the inequalities defining $\tilde{Q}$ has a nonzero coefficient in col_1, we have $\tilde{\pi}^{-1}(\tilde{Q} \cap \mathcal{L}) = \tilde{Q}$, hence $\tilde{Q} \cap \mathcal{L} = \tilde{\pi}(\tilde{Q})$. This yields $\tilde{\pi}(\mathcal{A} \cap \tilde{Q}) = \tilde{\pi}(\mathcal{A}) \cap \tilde{\pi}(\tilde{Q})$, which, due to $\tilde{\pi}(\mathcal{A}) = \mathcal{L}$, implies $\tilde{\pi}(\mathcal{A} \cap \tilde{Q}) = \tilde{Q} \cap \mathcal{L}$. Thus, we obtain

$$\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q) = \tilde{\pi}(\mathrm{O}^{=}_{p+1,q+1}(\mathfrak{S}_{q+1})) = \tilde{\pi}(\mathcal{A} \cap \tilde{Q}) = \tilde{Q} \cap \mathcal{L} = Q,$$

where the first equation is due to Proposition 4.1, the second equation follows from (18), and the final arises from identifying $\mathcal{L}$ with $\mathbb{R}^{\mathcal{I}_{p,q}}$. □

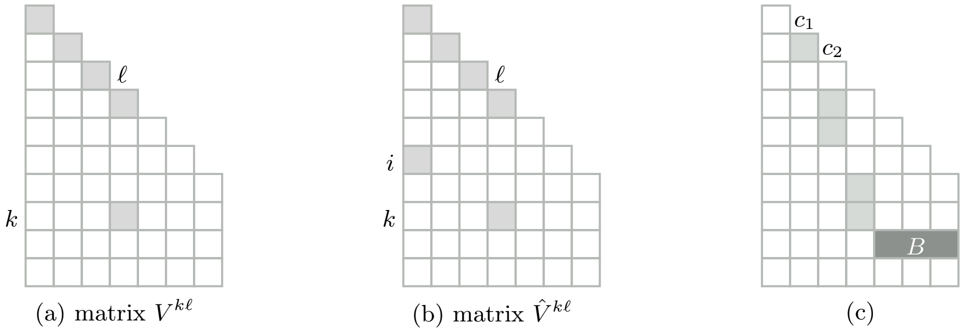


Figure 11: (a)–(b): Illustration of the matrices used in the proof of parts (1) and (3) of Proposition 4.13. (c): Example of an SCI that does not define a facet; see the proof of Part (4) of Proposition 4.13.

4.5. Facets

In this section, we investigate which of the constraints from the linear descriptions of $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ and $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ given in Propositions 4.11 and 4.12, respectively, define facets. This will also yield non-redundant descriptions.

It seems to be more convenient to settle the packing case first and then to carry over the results to the partitioning case. Recall that we assume $2 \leq p \leq q$.

Proposition 4.13.

(1) The packing orbitope $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q) \subset \mathbb{R}^{\mathcal{I}_{p,q}}$ is full dimensional:

$$\dim(\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)) = |\mathcal{I}_{p,q}| = pq - \tfrac{q(q-1)}{2} = \big(p - \tfrac{q-1}{2}\big)q.$$

(2) A nonnegativity constraint $x_{ij} \geq 0$, $(i,j) \in \mathcal{I}_{p,q}$, defines a facet of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$, unless $i = j < q$ holds. The faces defined by $x_{jj} \geq 0$ with $j < q$ are contained in the facet defined by $x_{qq} \geq 0$.

(3) Every row-sum constraint $x(\mathrm{row}_i) \leq 1$ for $i \in [p]$ defines a facet of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$.

(4) A shifted column inequality $x(B) - x(S) \leq 0$ with bar B and shifted column $S = \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \dots, \langle \eta, c_\eta\rangle\}$ defines a facet of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$, unless $\eta \geq 2$ and $c_1 < c_2$ (exception I) or $\eta = 1$ and $B \neq \{\langle 1, c_1 + 1\rangle\}$ (exception II) hold. In case of exception I, the corresponding face is contained in the facet defined by the SCI with bar B and shifted column $\{\langle 1, c_2\rangle, \langle 2, c_2\rangle, \dots, \langle \eta, c_\eta\rangle\}$. In case of exception II, the face is contained in the facet defined by the SCI $x_{\langle 1, c_1+1\rangle} - x_{\langle 1, c_1\rangle} \leq 0$.

Proof. Part (1): For all $(k,\ell) \in \mathcal{I}_{p,q}$, we define $V^{k\ell} = (v^{k\ell}_{ij}) \in \mathbb{R}^{\mathcal{I}_{p,q}}$ by

$$v^{k\ell}_{ij} = \begin{cases} 1 & \text{if } \big(i = j \leq \ell \text{ and } j < q\big) \text{ or } (i,j) = (k,\ell) \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } (i,j) \in \mathcal{I}_{p,q},$$

that is, $V^{k\ell}$ has 1-entries at position (k,ℓ) and on the main diagonal up to column ℓ, except that $v^{k\ell}_{qq} = 0$ unless $(k,\ell) = (q,q)$; see Figure 11 (a). The columns of each $V^{k\ell}$ are in non-increasing lexicographic order. Hence, by Part (1) of Observation 2, each $V^{k\ell}$ is a vertex of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$.

In order to show that these vectors are linearly independent, we fix an arbitrary ordering of the $V^{k\ell}$ that starts with $V^{11}, V^{22}, \dots, V^{q-1,q-1}$. For each $(k,\ell) \in \mathcal{I}_{p,q}$, all points V^{rs} preceding $V^{k\ell}$ have a 0-entry at position (k,ℓ), while $v^{k\ell}_{k\ell} = 1$. This shows that these $|\mathcal{I}_{p,q}|$ vertices of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ are linearly independent. Together with $\mathbf{0}$ this gives $|\mathcal{I}_{p,q}|{+}1$ affinely independent points contained in $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$, proving that $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ is full dimensional. The calculations in the statement are straightforward.

Part (2): For $(i,j) \in \mathcal{I}_{p,q} \setminus \{(j,j) \,:\, j < q\}$ all points $V^{k\ell}$ with $(k,\ell) \neq (i,j)$ are contained in the face defined by $x_{ij} \geq 0$. Since this is also true for $\mathbf{0}$, the face defined by $x_{ij} \geq 0$ contains $|\mathcal{I}_{p,q}|$ affinely independent points (see the proof of Part (1)), i.e., it is a facet of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$.

For every vertex $x^\star \in \mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ contained in the face defined by $x_{jj} \geq 0$ for some $j < q$, we have $x^\star_{\ell\ell} = 0$ for all $\ell \geq j$ (because otherwise the columns of $x^\star$ would not be in non-increasing lexicographic order). This shows that $x^\star$ is contained in the facet defined by $x_{qq} \geq 0$.

Part (3): In order to show that $x(\mathrm{row}_i) \leq 1$ defines a facet of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ for $i \in [p]$, we construct points $\hat{V}^{k\ell}$ (depending on i) from the points $V^{k\ell}$ defined in Part (1) by adding a 1 at position $(i,1)$ if $V^{k\ell}(\mathrm{row}_i) = 0$ (see Figure 11 (b)). The $(|\mathcal{I}_{p,q}| - 1)$ points $\hat{V}^{k\ell}$ for all $(k,\ell) \in \mathcal{I}_{p,q} - (i,1)$, and the unit vector E^{i1} (with a single 1 in position $(i,1)$) satisfy $x(\mathrm{row}_i) = 1$. Furthermore, they are affinely independent, since subtracting E^{i1} from all vectors $\hat{V}^{k\ell}$ yields vectors $\tilde{V}^{k\ell}$, which can be shown to be linearly independent similarly to Part (1); here, we need $(k,\ell) \neq (i,1)$.

Part (4): Let $x(B) - x(S) \leq 0$ be an SCI with bar B, leader $(i,j) = \langle \eta, j \rangle$, and shifted column $S = \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \dots, \langle \eta, c_\eta\rangle\}$.

If $\eta \geq 2$ and $c_1 < c_2$ hold (exception I), then the SCI is the sum of the SCI

$$x_{\langle 1, c_1+1\rangle} - x_{\langle 1, c_1\rangle} \leq 0$$

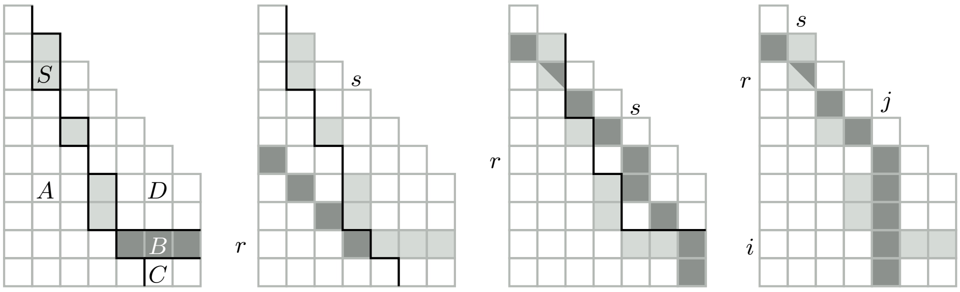
and the SCI with bar B and shifted column $\{\langle 1, c_1+1\rangle, \langle 2, c_2\rangle, \dots, \langle \eta, c_\eta\rangle\}$; see Figure 11 (c). Repeating this argument $(c_2 - c_1 - 1)$ times proves the second statement of Part (4) for exception I.

If $\eta = 1$ and $B = \{\langle 1, j\rangle\}$ with $j > c_1 + 1$ hold (exception II), then the SCI is the sum of the SCIs $x_{\langle 1, c_1+1\rangle} - x_{\langle 1, c_1\rangle} \leq 0, \dots, x_{\langle 1, j\rangle} - x_{\langle 1, j-1\rangle} \leq 0$. This proves the second statement of Part (4) for exception II.

Otherwise, let $\mathcal{V}$ be the set of vertices of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ that satisfy the SCI with equality, and let $\mathcal{L} = \mathrm{lin}(\mathcal{V} \cup \{E^{ij}\})$ be the linear span of $\mathcal{V}$ and the unit vector E^{ij}. We will show that $\mathcal{L} = \mathbb{R}^{\mathcal{I}_{p,q}}$, which proves $\dim(\mathrm{aff}(\mathcal{V})) = |\mathcal{I}_{p,q}| - 1$ (since $\mathbf{0} \in \mathcal{V}$). Hence, the SCI defines a facet of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$.

To show that $\mathcal{L} = \mathbb{R}^{\mathcal{I}_{p,q}}$, we prove that $E^{rs} \in \mathcal{L}$ for all $(r,s) \in \mathcal{I}_{p,q}$. We partition the set $\mathcal{I}_{p,q} \setminus (B \cup S)$ into three parts (see Figure 12 (a)):

$$\begin{aligned} A :=& \{\langle \rho, s\rangle \in \mathcal{I}_{p,q} \,:\, (\rho \leq \eta \text{ and } s < c_\rho) \text{ or } \rho > \eta\}, \\ C :=& \{\langle \rho, s\rangle = (r,s) \in \mathcal{I}_{p,q} \,:\, \rho \leq \eta \text{ and } r > i\}, \text{ and} \\ D :=& \{\langle \rho, s\rangle = (r,s) \in \mathcal{I}_{p,q} \,:\, \rho < \eta,\ s > c_\rho, \text{ and } r < i\}. \end{aligned}$$



(a) All cases (b) Case A, W^{rs} (c) Case D, U^{rs} (d) Case S

Figure 12: Illustration of the constructions in the proof of Part (4) of Proposition 4.13.

For $(r,s) = \langle \rho, s\rangle$, denote by $\operatorname{diag}^{\leq}(r,s) = \{\langle \rho,1\rangle, \langle \rho,2\rangle, \dots, \langle \rho, s\rangle\}$ the diagonal starting at $\langle \rho, 1\rangle = (r-s+1, 1)$ and ending at $\langle \rho, s\rangle = (r,s)$. Similarly, denote by $\operatorname{diag}^{\geq}(r,s) = \{\langle \rho, s\rangle, \langle \rho, s+1\rangle, \dots\} \cap \mathcal{I}_{p,q}$ the diagonal starting at (r,s) and ending in col_q or in row_p.

Claim. For all $(r,s) = \langle \rho, s\rangle \in A \cup C$ we have $E^{rs} \in \mathcal{L}$.

Proof. Denote the incidence vector of $\operatorname{diag}^{\leq}(r,s)$ by $W^{rs} = \chi^{\operatorname{diag}^{\leq}(r,s)}$ (see Figure 12 (b)). Both W^{rs} and $W^{rs} - E^{rs}$ are vertices of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$. We have $\operatorname{diag}^{\leq}(r,s) \cap (B \cup S) = \varnothing$ for $(r,s) \in A$. Furthermore

$$|\operatorname{diag}^{\leq}(r,s) \cap B| = 1 = |\operatorname{diag}^{\leq}(r,s) \cap S|$$

for $(r,s) \in C$. Hence, these two vertices satisfy the SCI with equality and we obtain $E^{rs} = W^{rs} - (W^{rs} - E^{rs}) \in \mathcal{L}$. □

Claim. For all $(r,s) = \langle \rho, s\rangle \in D$ we have $E^{rs} \in \mathcal{L}$.

Proof. Define the set

$$U(r,s) := \operatorname{diag}^{\leq}(r,s) \cup \operatorname{diag}^{\geq}(r+1,s) \cup \big(\{\langle \rho+1, q\rangle, \langle \rho+2, q\rangle, \dots\} \cap \mathcal{I}_{p,q}\big),$$

see Figure 12 (c). Let $U^{rs} := \chi^{U(r,s)}$. By construction, the three points U^{rs}, $U^{rs} - E^{rs}$, and $U^{rs} - E^{r+1,s}$ are vertices of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$.

If $\rho = 1$, we have $|U(r,s) \cap B| = 1$ and $|U(r,s) \cap S| = 1$, where we need $c_1 = c_2$ in case of $s = c_1 + 1$ (notice that in case of $\eta = 1$ we have $D = \varnothing$). Due to $(r,s) \notin B \cup S$, both U^{rs} and $U^{rs} - E^{rs}$ satisfy the SCI with equality. This yields $E^{rs} = U^{rs} - (U^{rs} - E^{rs}) \in \mathcal{L}$.

If $\rho > 1$, then $|U(r,s) \cap S| = 1$ does not hold in all cases (e.g., if $s = c_{\rho+1}$, we have $(r+1,s) \in S$). However, since $\rho > 1$, $U(r-1,s)$ is well-defined and

$$|U(r-1,s) \cap B| = 1 \qquad \text{and} \qquad |U(r-1,s) \cap S| = 1$$

hold. Hence the vertices $U^{r-1,s}$ and $U^{r-1,s} - E^{rs}$ satisfy the SCI with equality, giving $E^{rs} = U^{r-1,s} - (U^{r-1,s} - E^{rs}) \in \mathcal{L}$. □

Claim. For all $(r,s) = \langle \rho, s\rangle \in S$ we have $E^{rs} \in \mathcal{L}$.

Proof. Define the set

$$T(r,s) := \operatorname{diag}^{\leq}(r+j-s, j) \cup \big(\{\langle \rho+1, j\rangle, \langle \rho+2, j\rangle, \dots\} \cap \mathcal{I}_{p,q}\big),$$

see Figure 12 (d). The incidence vector $T^{rs} := \chi^{T(r,s)}$ is a vertex of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$, which, due to $T(r,s) \cap S = \{(r,s)\}$ and $T(r,s) \cap B = \{(i,j)\}$ satisfies the SCI with equality. Thus, from

$$E^{rs} = T^{rs} - E^{ij} - \sum_{(k,\ell)\in T(r,s)\cap A} E^{k\ell} - \sum_{(k,\ell)\in T(r,s)\cap C} E^{k\ell} - \sum_{(k,\ell)\in T(r,s)\cap D} E^{k\ell}$$

we conclude $E^{rs} \in \mathcal{L}$, since $E^{ij} \in \mathcal{L}$ by definition of $\mathcal{L}$, and $E^{k\ell} \in \mathcal{L}$ for all $(k,\ell) \in A \cup C \cup D$ by Claims 4.5 and 4.5. □

Claim. For all $(i,s) = \langle \rho, s\rangle \in B$ we have $E^{rs} \in \mathcal{L}$.

Proof. The vector $W^{is} := \chi^{\mathrm{diag}^{\leq}(i,s)}$ is a vertex of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ that satisfies the SCI with equality. Furthermore, we have

$$E^{is} = W^{is} - E^{rc_\rho} - \sum_{(k,\ell)\in \mathrm{diag}^{\leq}(i,s)\cap A} E^{k\ell} - \sum_{(k,\ell)\in \mathrm{diag}^{\leq}(i,s)\cap D} E^{k\ell},$$

where $(r,c_\rho) := \langle \rho, c_\rho\rangle \in S$. Thus, we conclude $E^{is} \in \mathcal{L}$, since $E^{k\ell} \in \mathcal{L}$ for all $(k,\ell) \in A \cup D \cup S$ by Claims 4.5, 4.5, and 4.5. □

Claims 4.5 to 4.5 show $E^{rs} \in \mathcal{L}$ for all $(r,s) \in \mathcal{I}_{p,q}$. This proves that the SCI defines a facet of $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ (unless exception I or II hold). □

Finally, we carry the results of Proposition 4.13 over to partitioning orbitopes.

Proposition 4.14.

(1) The partitioning orbitope $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q) \subset \mathbb{R}^{\mathcal{I}_{p,q}}$ has dimension

$$\dim(\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)) = |\mathcal{I}_{p-1,q-1}| = |\mathcal{I}_{p,q}| - p = \big(p - \tfrac{q}{2}\big)(q-1).$$

The constraints $x(\mathrm{row}_i) = 1$ form a complete and non-redundant linear description of $\mathrm{aff}(\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q))$.

(2) A nonnegativity constraint $x_{ij} \geq 0$, $(i,j) \in \mathcal{I}_{p,q}$, defines a facet of $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$, unless $i = j < q$ holds. The faces defined by $x_{jj} \geq 0$ with $j < q$ are contained in the facet defined by $x_{qq} \geq 0$.

(3) A shifted column inequality $x(B) - x(S) \leq 0$ with bar B and shifted column $S = \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \dots, \langle \eta, c_\eta\rangle\}$ defines a facet of $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$, unless $c_1 = 1$ (Exception I) or $\eta \geq 2$ and $c_1 < c_2$ (Exception II) or $\eta = 1$ and $B \neq \{\langle 1, c_1 + 1\rangle\}$ (Exception III). In case of Exception I, the corresponding face is contained in the facet defined by $x_{i1} \geq 0$, where i is the index of the row containing B. In case of Exception II, the face is contained in the facet defined by the SCI with bar B and shifted column $\{\langle 1, c_2\rangle, \langle 2, c_2\rangle, \dots, \langle \eta, c_\eta\rangle\}$. In case of Exception III, the face is contained in the facet defined by the SCI $x_{\langle 1,c_1+1\rangle} - x_{\langle 1,c_1\rangle} \leq 0$.

Proof. According to Proposition 4.1, $\mathrm{O}^{\leq}_{p-1,q-1}(\mathfrak{S}_{q-1})$ is affinely isomorphic to $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ via the orthogonal projection of the latter polytope to the space

$$\mathcal{L} := \{x \in \mathbb{R}^{\mathcal{I}_{p,q}} : x_{i1} = 0 \text{ for all } i \in [p]\}$$

(and via the canonical identification of $\mathcal{L}$ and $\mathbb{R}^{\mathcal{I}_{p-1,q-1}}$). This shows the statement on the dimension of $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$; the calculations and the claim on the non-redundancy of the equation system are straightforward.

Furthermore, this projection (which is one-to-one on $\operatorname{aff}(\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)))$ maps every face of $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ that is defined by some inequality

$$\langle a, x\rangle := \sum_{(i,j)\in\mathcal{I}_{p,q}} a_{ij}\, x_{ij} \leq a_0,$$

with $a \in \mathbb{R}^{\mathcal{I}_{p,q}}$, $a_0 \in \mathbb{R}$, and $a_{i1} = 0$ for all $i \in [p]$ to a face of $\mathrm{O}^{\leq}_{p-1,q-1}(\mathfrak{S}_{q-1})$ of the same dimension defined by

$$\sum_{(i,j)\in\mathcal{I}_{p-1,q-1}} a_{i+1,j+1}\, x_{ij} \leq a_0.$$

Conversely, if $\langle \tilde{a}, x\rangle \leq \tilde{a}_0$ defines a face of $\mathrm{O}^{\leq}_{p-1,q-1}(\mathfrak{S}_{q-1})$ for $\tilde{a} \in \mathbb{R}^{\mathcal{I}_{p-1,q-1}}$ and $\tilde{a}_0 \in \mathbb{R}$, then the inequality

$$\sum_{(i,j)\in\mathcal{I}_{p,q}} \tilde{a}_{ij}\, x_{i+1,j+1} \leq \tilde{a}_0$$

defines a face of $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ of the same dimension.

Due to parts (2) and (3) of Proposition 4.13, this proves Part (2) of the proposition, where we use the fact that the inequalities $x_{i1} \geq 0$ are equivalent to $x\big(\mathrm{row}_i - (i,1)\big) \leq 1$ with respect to $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$.

Furthermore, due to Part (4) of Proposition 4.13, the above arguments also imply the statements of Part (3) for $c_1 \geq 2$ (including Exception II and III). Finally, we consider the case $c_1 = 1$ (Exception I). Since we have $x_{1,1} = 1$ for all $x \in \mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$, the equation $x(B) - x(S) = 0$ implies

$$1 \geq x(B) = x(S) \geq x_{1,1} = 1,$$

and hence $x_{i,1} = 0$ (using the row-sum equation for row i containing B). This concludes the proof. □

4.6. Summary of Results on the Symmetric Group

We collect the results on the packing- and partitioning orbitopes for symmetric groups.

Theorem 4.15. The partitioning orbitope $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ (for $p \geq q \geq 2$) with respect to the symmetric group $\mathfrak{S}_q$ equals the set of all $x \in \mathbb{R}^{\mathcal{I}_{p,q}}$ that satisfy the following linear constraints:

- the row-sum equations $x(\mathrm{row}_i) = 1$ for all $i \in [p]$,
- the nonnegativity constraints $x_{ij} \geq 0$ for all $(i,j) \in \mathcal{I}_{p,q} \setminus \{(j,j) : j < q\}$,
- the shifted column inequalities $x(B) - x(S) \leq 0$ for all bars

  $$B = \{(i,j), (i,j+1), \dots, (i, \min\{i,q\})\}$$

  with $(i,j) = \langle \eta, j\rangle \in \mathcal{I}_{p,q}$, $j \geq 2$, and shifted columns

  $$S = \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \dots, \langle \eta, c_\eta\rangle\} \text{ with } 2 \leq c_1 = c_2 \leq \dots \leq c_\eta \leq j-1,$$

  where in case of $\eta = 1$ the last condition reduces to $2 \leq c_1$ and we additionally require $j = c_1 + 1$.

This system of constraints is non-redundant. The corresponding separation problem can be solved in time $O(pq)$.

For the result on the completeness of the description see Proposition 4.11, for the question of redundancy see Proposition 4.14, and for the separation algorithm see Corollary 4.10. Note that the SCI with shifted column $\{(1,1)\}$ and bar $\{(2,2)\}$ defines the same facet of $\mathrm{O}^{=}_{p,q}(\mathfrak{S}_q)$ as the nonnegativity constraint $x_{2,1} \geq 0$.

Theorem 4.16. The packing orbitope $\mathrm{O}^{\leq}_{p,q}(\mathfrak{S}_q)$ (for $p \geq q \geq 2$) with respect to the symmetric group $\mathfrak{S}_q$ equals the set of all $x \in \mathbb{R}^{\mathcal{I}_{p,q}}$ that satisfy the following linear constraints:

- the row-sum inequalities $x(\mathrm{row}_i) \leq 1$ for all $i \in [p]$,
- the nonnegativity constraints $x_{ij} \geq 0$ for all $(i,j) \in \mathcal{I}_{p,q} \setminus \{(j,j) : j < q\}$,
- the shifted column inequalities $x(B) - x(S) \leq 0$ for all bars

$$B = \{(i,j), (i,j+1), \dots, (i, \min\{i,q\})\}$$

with $(i,j) = \langle \eta, j \rangle \in \mathcal{I}_{p,q}$, $j \geq 2$, and shifted columns

$$S = \{\langle 1, c_1 \rangle, \langle 2, c_2 \rangle, \dots, \langle \eta, c_\eta \rangle\} \text{ with } c_1 = c_2 \leq \cdots \leq c_\eta \leq j-1,$$

where in case of $\eta = 1$ we additionally require $j = c_1 + 1$.

This system of constraints is non-redundant. The corresponding separation problem can be solved in time $O(pq)$.

For the result on the completeness of the description see Proposition 4.12, for the question of redundancy see Proposition 4.13, and for the separation algorithm see Corollary 4.10.

5. Concluding Remarks

We close with some remarks on the technique used in the proof of Proposition 4.11, on the combination of SCIs and clique-inequalities for the graph-coloring problem, and on full and covering orbitopes.

The Proof Technique.

Our technique to prove Proposition 4.11 can be summarized as follows. Assume a polytope $Q \subset \mathbb{R}^n$ is described by some (finite) system $\mathcal{Q}$ of linear equations and inequalities. Suppose that $\mathcal{Q}'$ is a subsystem of $\mathcal{Q}$ for which it is known that $\mathcal{Q}'$ defines an integral polytope $Q' \supseteq Q$. One can prove that Q is integral by showing that every vertex $x^\star$ of Q is a vertex of Q' in the following way. Here we call a basis (with respect to $\mathcal{Q}$) of $x^\star$ *reduced* if it contains as many constraints from $\mathcal{Q}'$ as possible:

(1) Starting from an arbitrary reduced basis $\mathcal{B}$ of $x^\star$, construct iteratively a reduced basis $\mathcal{B}^\star$ of $x^\star$ that satisfies some properties that are useful for the second step.

(2) Under the assumption that $\mathcal{B}^\star \not\subseteq \mathcal{Q}'$, modify $x^\star$ to some $\tilde{x} \neq x^\star$ that also satisfies the equation system corresponding to $\mathcal{B}^\star$ (contradicting the fact that $\mathcal{B}^\star$ is a basis).

(In our proof of Proposition 4.11, Step (1) was done by showing that a reduced basis of "minimal weight" has the desired properties.)

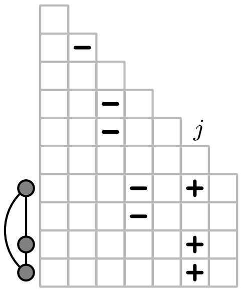


Figure 13: Combination of a clique inequality and an SCI.

Such a proof is conceivable for every 0/1-polytope Q by choosing $Q' = [0,1]^n$ as the whole 0/1-cube and $\mathcal{Q}'$ as the set of the $2n$ trivial inequalities $0 \leq x_i \leq 1$, for $i = 1, \dots, n$ (if necessary, modifying $\mathcal{Q}$ in order to contain them all).

We do not know whether this kind of integrality proof has been used in the literature. It may well be that one can interpret some of the classical integrality proofs in this setting. Anyway, it seems to us that the technique might be useful for other polytopes as well.

The Graph-Coloring Problem.

As mentioned in the introduction, for concrete applications like the graph coloring problem one can (and probably has to) combine the polyhedral knowledge on orbitopes with the knowledge on problem specific polyhedra. We illustrate this by the example of clique inequalities for the graph coloring model (1) described in the introduction.

Fix a color index $j \in [C]$. If $W \subseteq V$ is a clique in the graph $G = (V, E)$, then clearly the inequality $\sum_{i \in W} x_{ij} \leq 1$ is valid. In fact, the strengthened inequalities $\sum_{i \in W} x_{ij} \leq y_j$ are known to be facet-defining for the convex hull of the solutions to (1), see [4]. Suppose that $S \subset \mathcal{I}_{|V|,C}$ is a shifted column and that we have $\eta \leq |S|$ for all $\langle \eta, j \rangle = (i, j)$ with $i \in W$. Then the inequality

$$\sum_{i \in W} x_{ij} - x(S) \leq 0$$

is valid for all solutions to the model obtained from (1) by adding inequalities (2) (which are all "column inequalities" in terms of orbitopes), see Figure 13. The details and a computational study will be the subject of a follow-up paper.

Full and Covering Orbitopes.

As soon as one starts to consider 0/1-matrices that may have more than one 1-entry per row, things seem to become more complicated.

With respect to cyclic group actions, we loose the simplicity of the characterizations in Observation 2. The reason is that the matrices under investigation may have several equal nonzero columns. In particular, the lexicographically maximal column may not be unique.

With respect to the action of the symmetric group, we still have the characterization of the representatives as the matrices whose columns are in non-increasing lexicographic order (see Part 1 of Observation 2). The structures of the respective full and covering orbitopes, however, become much more complicated. In particular, we know from computer experiments that several powers of two arise as coefficients in the facet-defining inequalities. This increase in complexity is reflected by the fact that optimization of linear functionals over these orbitopes seems to be more difficult than over packing and partitioning orbitopes (see the remarks at the end of Section 2.1).

Let us close with a comment on our choice of the set of representatives as the maximal elements with respect to a lexicographic ordering (referring to the row-wise ordering of the components of the matrices). It might be that the difficulties for full and covering orbitopes mentioned in the previous paragraph can be overcome by the choice of a different system of representatives. The choice of representatives considered in this paper, however, seems to be appropriate for the packing and partitioning cases.

Whether the results presented in this paper are useful in practice will turn out in the future. In any case, we hope that the reader shares our view that orbitopes are neat mathematical objects. It seems that symmetry strikes back by its own beauty, even when mathematicians start to fight it.

Acknowledgments

We thank the referees for their work. In particular, we are indepted to one of them for several insightful and constructive remarks, including the proof of Theorem 3.2 that we have in this final version. We furthermore thank Yuri Faenza, Andreas Loos, and Matthias Peinhardt for helpful comments.

References

[1] R. Borndörfer, C. E. Ferreira, and A. Martin, *Decomposing matrices into blocks*, SIAM J. Optim. **9**, no. 1 (1998), pp. 236–269.

[2] R. Borndörfer, M. Grötschel, and M. E. Pfetsch, *A column-generation approach for line planning in public transport*, Transportation Sci. **41**, no. 1 (2006), pp. 123–132.

[3] M. Campêlo, R. Corrêa, and Y. Frota, *Cliques, holes and the vertex coloring polytope*, Inform. Process. Lett. **89**, no. 4 (2004), pp. 159–164.

[4] P. Coll, J. Marenco, I. Méndez Díaz, and P. Zabala, *Facets of the graph coloring polytope*, Ann. Oper. Res. **116** (2002), pp. 79–90.

[5] D. Cornaz, *On forests, stable sets and polyhedras associated with clique partitions*. Preprint, 2006. Available at www.optimization-online.org.

[6] A. Eisenblätter, *Frequency Assignment in GSM Networks: Models, Heuristics, and Lower Bounds*, PhD thesis, TU Berlin, 2001.

[7] T. Fahle, S. Schamberger, and M. Sellmann, *Symmetry breaking*, in Principles and Practice of Constraint Programming CP 2007, 7th International Conference, T. Walsh, ed., LNCS 2239, Springer-Verlag, Berlin/Heidelberg, 2001, pp. 93–107.

[8] R. Figueiredo, V. Barbosa, N. Maculan, and C. de Souza, *New 0-1 integer formulations of the graph coloring problem*, in Proceedings of XI CLAIO, 2002.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.

[10] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Algorithms and Combinatorics 2, Springer-Verlag, Heidelberg, 2nd ed., 1993.

[11] F. Margot, *Pruning by isomorphism in branch-and-cut*, Math. Programming **94**, no. 1 (2002), pp. 71–90.

[12] F. Margot, *Small covering designs by branch-and-cut*, Math. Programming **94**, no. 2–3 (2003), pp. 207–220.

[13] A. Mehrotra and M. A. Trick, *A column generation approach for graph coloring*, INFORMS J. Comput. **8**, no. 4 (1996), pp. 344–354.

[14] I. Méndez-Díaz and P. Zabala, *A polyhedral approach for graph coloring*, Electron. Notes Discrete Math. **7** (2001).

[15] I. Méndez-Díaz and P. Zabala, *A branch-and-cut algorithm for graph coloring*, Discrete Appl. Math. **154**, no. 5 (2006), pp. 826–847.

[16] J.-F. Puget, *Symmetry breaking revisited*, Constraints **10**, no. 1 (2005), pp. 23–46.

[17] A. Ramani, F. A. Aloul, I. L. Markov, and K. A. Sakallah, *Breaking instance-independent symmetries in exact graph coloring*, in Design Automation and Test in Europe Conference, 2004, pp. 324–329.

[18] A. Schrijver, *Theory of linear and integer programming*, John Wiley & Sons, Chichester, 1986. Reprint 1998.

[19] P. Serafini and W. Ukovich, *A mathematical model for periodic scheduling problems.*, SIAM J. Discrete Math. **2**, no. 4 (1989), pp. 550–581.

Paper 2

Orbitopal Fixing

Volker Kaibel, Matthias Peinhardt, and Marc E. Pfetsch
Orbitopal Fixing[1]
Proc. 12th Integer Programming and Combinatorial Optimization conference (IPCO), M. Fischetti and D. Williamson, eds., LNCS 4513, Springer-Verlag, 2007, pp. 74–88

Abstract. The topic of this paper are integer programming models in which a subset of 0/1-variables encode a partitioning of a set of objects into disjoint subsets. Such models can be surprisingly hard to solve by branch-and-cut algorithms if the order of the subsets of the partition is irrelevant. This kind of symmetry unnecessarily blows up the branch-and-cut tree.

We present a general tool, called orbitopal fixing, for enhancing the capabilities of branch-and-cut algorithms in solving such symmetric integer programming models. We devise a linear time algorithm that, applied at each node of the branch-and-cut tree, removes redundant parts of the tree produced by the above mentioned symmetry. The method relies on certain polyhedra, called orbitopes, which have been investigated in [11]. It does, however, not add inequalities to the model, and thus, it does not increase the difficulty of solving the linear programming relaxations. We demonstrate the computational power of orbitopal fixing at the example of a graph partitioning problem motivated from frequency planning in mobile telecommunication networks.

1. Introduction

Being welcome in most other contexts, symmetry causes severe trouble in the solution of many integer programming (IP) models. This paper describes

[1] Supported by the DFG Research Center Matheon *Mathematics for key technologies* in Berlin. During the research of this work the first author was a visiting professor at the Technical University of Berlin.

a method to enhance the capabilities of branch-and-cut algorithms with respect to handling symmetric models of a certain kind that frequently occurs in practice.

We illustrate this kind of symmetry by the example of a graph partitioning problem (another notorious example is the vertex coloring problem). Here, one is given a graph $G = (V, E)$ with nonnegative edge weights $w \in \mathbb{Q}^{E}_{\geq 0}$ and an integer $q \geq 2$. The task is to partition V into q disjoint subsets such that the sum of all weights of edges connecting nodes in the same subset is minimized.

A straight-forward IP model arises by introducing 0/1-variables x_{ij} for all $i \in [p] := \{1, \dots, p\}$ and $j \in [q]$ that indicate whether node i is contained in subset j (where we assume $V = [p]$). In order to model the objective function, we furthermore need 0/1-variables y_{ik} for all edges $\{i, k\} \in E$ indicating whether nodes i and k are contained in the same subset. This yields the following model (see, e.g., [5]):

$$
\begin{aligned}
\min \quad & \sum_{\{i,k\} \in E} w_{ik}\, y_{ik} \\
\text{s.t.} \quad & \sum_{j=1}^{q} x_{ij} = 1 && \text{for all } i \in [p] \\
& x_{ij} + x_{kj} - y_{ik} \leq 1 && \text{for all } \{i,k\} \in E,\ j \in [q] \\
& x_{ij} \in \{0,1\} && \text{for all } i \in [p],\ j \in [q] \\
& y_{ik} \in \{0,1\} && \text{for all } \{i,k\} \in E.
\end{aligned}
\tag{1}
$$

The x-variables describe a 0/1-matrix of size $p \times q$ with exactly one 1-entry per row. They encode the assignment of the nodes to the subsets of the partition. The methods that we discuss in this paper do only rely on this structure and thus can be applied to many other models as well. We use the example of the graph partitioning problem as a prototype application and report on computational experiments in Sect. 5. Graph partitioning problems are discussed in [3, 4, 5], for instance as a relaxation of frequency assignment problems in mobile telecommunication networks. The maximization version is relevant as well [6, 12]. Also capacity bounds on the subsets of the partition (which can easily be incorporated into the model) are of interest, in particular the graph equipartitioning problem [7, 8, 18, 19]. For the closely related clique partitioning problem, see [9, 10].

As it is given above, the model is unnecessarily difficult for state-of-the-art IP solvers. Even solving small instances requires enormous efforts (see Sect. 5). One reason is that every feasible solution (x, y) to this model can be turned into $q!$ different ones by permuting the columns of x (viewed as a 0/1-matrix) in an arbitrary way, thereby not changing the structure of the solution (in particular: its objective function value). Phrased differently, the symmetric group of all permutations of the set $[q]$ operates on the solutions by permuting the columns of the x-variables in such a way that the objective function remains constant along each orbit. Therefore, when solving the model by a branch-and-cut algorithm, basically the same work will be done in the tree at many places. Thus, there should be potential for reducing the running times significantly by exploiting the symmetry. A more subtle second point is that interior points of the convex hulls of the individual orbits are responsible for quite weak linear programming (LP) bounds. We will, however, not address this second point in this paper.

In order to remove symmetry, the above model for the graph partitioning problem is often replaced by models containing only edge variables, see, e.g. [7]. However, for this to work the underlying graph has to be complete, which might introduce many unnecessary variables. Moreover, formulation (1) is sometimes favorable, e.g., if node-weighted capacity constraints should be incorporated.

One way to deal with symmetry is to restrict the feasible region in each of the orbits to a single representative, e.g., to the lexicographically maximal (with respect to the row-by-row ordering of the x-components) element in the orbit. In fact, this can be done by adding inequalities to the model that enforce the columns of x to be sorted in a lexicographically decreasing way. This can be achieved by $O(pq)$ many *column inequalities*. In [11] even a complete (and irredundant) linear description of the convex hull of all 0/1-matrices of size $p \times q$ with exactly one 1-entry per row and lexicographically decreasing columns is derived; these polytopes are called *orbitope*. The description basically consists of an exponentially large superclass of the column inequalities, called *shifted column inequalities*, for which there is a linear time separation algorithm available. We recall some of these results in Sect. 2.

Incorporating the inequalities from the orbitope description into the IP model removes symmetry. At each node of the branch-and-cut tree this ensures that the corresponding IP is infeasible as soon as there is no representative in the subtree rooted at that node. In fact, already the column inequalities are sufficient for this purpose.

In this paper, we investigate a way to utilize these inequalities (or the orbitope that they describe) without adding any of the inequalities to the models explicitly. The reason for doing this is the unpleasant effect that adding (shifted) column inequalities to the models results in more difficult LP relaxations. One way of avoiding the addition of these inequalities to the LPs is to derive logical implications instead: If we are working in a branch-and-cut node at which the x-variables corresponding to index subsets I_0 and I_1 are fixed to zero and one, respectively, then there might be a (shifted) column inequality yielding implications for all representatives in the subtree rooted at the current node. For instance, it might be that for some $(i,j) \notin I_0 \cup I_1$ we have $x_{ij} = 0$ for all feasible solutions in the subtree. In this case, x_{ij} can be fixed to zero for the whole subtree rooted at the current node, enlarging I_0. We call the iterated process of searching for such additional fixings *sequential fixing* with (shifted) column inequalities.

Let us mention at this point that deviating from parts of the literature, we do not distinguish between “fixing“ and “setting“ of variables in this paper.

Sequential fixing with (shifted) column inequalities is a special case of constraint propagation, which is well known from constraint logic programming. Modern IP solvers like SCIP [1] use such strategies also in branch-and-cut algorithms. With orbitopes, however, we can aim at something better:

Consider a branch-and-cut node identified by fixing the variables corresponding to sets I_0 and I_1 to zero and one, respectively. Denote by $W(I_0, I_1)$ the set of all vertices x of the orbitope with $x_{ij} = 0$ for all $(i, j) \in I_0$ and $x_{ij} = 1$ for all $(i, j) \in I_1$. Define the sets $I_0^\star$ and $I_1^\star$ of indices of *all* variables, for which no x in $W(I_0, I_1)$ satisfies $x_{ij} = 1$ for some $(i, j) \in I_0^\star$ or $x_{ij} = 0$ for some $(i, j) \in I_1^\star$. Fixing of the corresponding variables is called *simultaneous fixing* at the branch-and-cut node. Simultaneous fixing is always at least as strong as sequential fixing.

Investigations of sequential and simultaneous fixing for orbitopes are the central topic of the paper. The main contributions and results are the following:

- We present a linear time algorithm for *orbitopal fixing*, i.e., for solving the problem to compute simultaneous fixings for orbitopes (Theorem 4.8).
- We show that, for general 0/1-polytopes, sequential fixing, even with complete and irredundant linear descriptions, is weaker than simultaneous fixing (Theorem 3.2), We clarify the relationships between different versions of sequential fixing with (shifted) column inequalities, where (despite the situation for general 0/1-polytopes) the strongest one is as strong as orbitopal fixing (Theorem 4.7).
- We report on computer experiments (Sect. 5) with the graph partitioning problem described above, showing that orbitopal fixing leads to significant performance improvements for branch-and-cut algorithms.

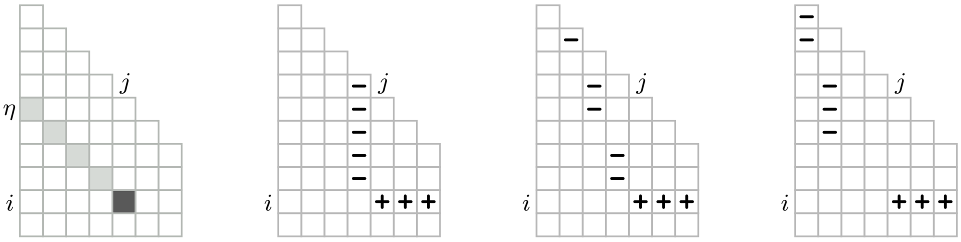
Margot [14, 15, 17] considers a related method for symmetry handling. His approach works for more general types of symmetries than ours. Similarly to our approach, the basic idea is to assure that only (partial) solutions which are lexicographical maximal in their orbit are explored in the branch-and-cut tree. This is guaranteed by an appropriate fixing rule. The fixing and pruning decisions are done by means of a Schreier-Sims table for representing the group action. While Margot's approach is much more generally applicable than orbitopal fixing, the latter seems to be more powerful in the special situation of partitioning type symmetries. One reason is that Margot's method requires to choose the branching variables according to an ordering that is chosen globally for the entire branch-and-cut tree.

Another approach has recently been proposed by Linderoth et al. [13] (in this volume). They exploit the symmetry arising in each node of a branch-and-bound tree when all fixed variables are removed from the model. Thus one may find additional local symmetries. Nevertheless, for partitioning type symmetries one still may miss some part of the (fixed) global symmetry we are dealing with.

We will elaborate on the relations between orbitopal fixing, isomorphism pruning, and orbital branching in more detail in a journal version of the paper.

2. Orbitopes

Throughout the paper, let p and q be integers with $p \geq q \geq 2$. The *orbitope* $\mathrm{O}^{=}_{p,q}$ is the convex hull of all 0/1-matrices $x \in \{0,1\}^{[p]\times[q]}$ with exactly one



(a) (b) (c) (d)

Figure 1: (a) Example for coordinates $(9,5) = \langle 5,5\rangle$. (b), (c), (d) Three shifted column inequalities, the left one of which is a column inequality

1-entry per row, whose columns are in decreasing lexicographical order (i.e., they satisfy $\sum_{i=1}^{p} 2^{p-i}x_{ij} > \sum_{i=1}^{p} 2^{p-i}x_{i,j+1}$ for all $j \in [q-1]$). Let the symmetric group of size q act on $\{0,1\}^{[p]\times[q]}$ via permuting the columns. Then the vertices of $\mathrm{O}^{=}_{p,q}$ are exactly the lexicographically maximal matrices (with respect to the row-by-row ordering of the components) in those orbits whose elements are matrices with exactly one 1-entry per row. As these vertices have $x_{ij} = 0$ for all (i,j) with $i < j$, we drop these components and consider $\mathrm{O}^{=}_{p,q}$ as a subset of the space $\mathbb{R}^{\mathcal{I}_{p,q}}$ with $\mathcal{I}_{p,q} := \{(i,j) \in \{0,1\}^{[p]\times[q]} \ : \ i \geq j\}$. Thus, we consider matrices, in which the i-th row has $q(i) := \min\{i,q\}$ components.

In [11], in the context of more general orbitopes, $\mathrm{O}^{=}_{p,q}$ is referred to as the *partitioning orbitope with respect to the symmetric group.* As we will confine ourselves with this one type of orbitopes in this paper, we will simply call it *orbitope.*

The main result in [11] is a complete linear description of $\mathrm{O}^{=}_{p,q}$. In order to describe the result, it will be convenient to address the elements in $\mathcal{I}_{p,q}$ via a different "system of coordinates": For $j \in [q]$ and $1 \leq \eta \leq p - j + 1$, define $\langle \eta, j\rangle := (j + \eta - 1, j)$. Thus (as before) i and j denote the row and the column, respectively, while η is the index of the diagonal (counted from above) containing the respective element; see Figure 1 (a) for an example.

A set $S = \{\langle 1, c_1\rangle, \langle 2, c_2\rangle, \dots, \langle \eta, c_\eta\rangle\} \subset \mathcal{I}_{p,q}$ with $c_1 \leq c_2 \leq \cdots \leq c_\eta$ and $\eta \geq 1$ is called a *shifted column.* For $(i,j) = \langle \eta, j\rangle \in \mathcal{I}_{p,q}$, a shifted column S as above with $c_\eta < j$, and $B = \{(i,j),(i,j+1),\dots,(i,q(i))\}$, we call $x(B) - x(S) \leq 0$ a *shifted column inequality.* The set B is called its *bar.* In case of $c_1 = \cdots = c_\eta = j-1$ the shifted column inequality is called a *column inequality.* See Figure 1 for examples.

Finally, a bit more notation is needed. For each $i \in [p]$, we define $\mathrm{row}_i := \{(i,j) \ : \ j \in [q(i)]\}$. For $A \subset \mathcal{I}_{p,q}$ and $x \in \mathbb{R}^{\mathcal{I}_{p,q}}$, we denote by $x(A)$ the sum $\sum_{(i,j)\in A} x_{ij}$.

Theorem 2.1 (see [11])**.** The orbitope $\mathrm{O}^{=}_{p,q}$ is completely described by the nonnegativity constraints $x_{ij} \geq 0$, the row-sum equations $x(\mathrm{row}_i) = 1$, and the shifted column inequalities.

In fact, in [11] it is also shown that, up to a few exceptions, the inequalities in this description define facets of $\mathrm{O}^{=}_{p,q}$. Furthermore, a linear time separation algorithm for the exponentially large class of shifted column inequalities is given.

3. The Geometry of Fixing Variables

In this section, we deal with general 0/1-integer programs and, in particular, their associated polytopes. We will define some basic terminology used later in the special treatment of orbitopes, and we are going to shed some light on the geometric situation of fixing variables.

We denote by $[d]$ the set of indices of variables, and by $\mathrm{C}^d = \{x \in \mathbb{R}^d : 0 \le x_i \le 1 \text{ for all } i \in [d]\}$ the corresponding 0/1-cube. For two disjoint subsets $I_0, I_1 \subseteq [d]$ (with $I_0 \cap I_1 = \varnothing$) we call

$$\{x \in \mathrm{C}^d : x_i = 0 \text{ for all } i \in I_0,\ x_i = 1 \text{ for all } i \in I_1\}$$

the *face of* C^d *defined by* (I_0, I_1). All nonempty faces of C^d are of this type.

For a polytope $P \subseteq \mathrm{C}^d$ and for a face F of C^d defined by (I_0, I_1), we denote by $\mathrm{Fix}_F(P)$ the smallest face of C^d that contains $P \cap F \cap \{0,1\}^d$ (i.e., $\mathrm{Fix}_F(P)$ is the intersection of all faces of C^d that contain $P \cap F \cap \{0,1\}^d$). If $\mathrm{Fix}_F(P)$ is the nonempty cube face defined by $(I_0^\star, I_1^\star)$, then $I_0^\star$ and $I_1^\star$ consist of all $i \in [d]$ for which $x_i = 0$ and $x_i = 1$, respectively, holds for all $x \in P \cap F \cap \{0,1\}^d$. In particular, we have $I_0 \subseteq I_0^\star$ and $I_1 \subseteq I_1^\star$, or $\mathrm{Fix}_F(P) = \varnothing$. Thus, if I_0 and I_1 are the indices of the variables fixed to zero and one, respectively, in the current branch-and-cut node (with respect to an IP with feasible points $P \cap \{0,1\}^d$), the node can either be pruned, or the sets $I_0^\star$ and $I_1^\star$ yield the maximal sets of variables that can be fixed to zero and one, respectively, for the whole subtree rooted at this node. Unless $\mathrm{Fix}_F(P) = \varnothing$, we call $(I_0^\star, I_1^\star)$ the *fixing of* P *at* (I_0, I_1). Similarly, we call $\mathrm{Fix}_F(P)$ the *fixing* of P at F.

Remark 3.1. If $P, P' \subseteq \mathrm{C}^d$ are two polytopes with $P \subseteq P'$ and F and F' are two faces of C^d with $F \subseteq F'$, then $\mathrm{Fix}_F(P) \subseteq \mathrm{Fix}_{F'}(P')$ holds.

In general, it is not clear how to compute fixings efficiently. Indeed, computing the fixing of P at $(\varnothing, \varnothing)$ includes deciding whether $P \cap \{0,1\}^d = \varnothing$, which, of course, is NP-hard in general. Instead, one can try to derive as large as possible subsets of $I_0^\star$ and $I_1^\star$ by looking at relaxations of P. In case of an IP that is based on an intersection with an orbitope, one might use the orbitope as such a relaxation. We will deal with the fixing problem for orbitopes in Sect. 4.

If P is given via an inequality description, one possibility is to use the knapsack relaxations obtained from single inequalities out of the description. For each of these relaxations, the fixing can easily be computed. If the inequality system describing P is exponentially large, and the inequalities are only accessible via a separation routine, it might still be possible to decide efficiently whether any of the exponentially many knapsack relaxations allows to fix some variable (see Sect. 4.2).

Suppose, $P = \{x \in \mathrm{C}^d : Ax \le b\}$ and $P_r = \{x \in \mathrm{C}^d : a_r^T x \le b_r\}$ is the knapsack relaxation of P for the rth-row $a_r^T x \le b_r$ of $Ax \le b$, where $r = 1, \dots, m$. Let F be some face of C^d. The face G of C^d obtained by setting $G := F$ and then iteratively replacing G by $\mathrm{Fix}_G(P_r)$ as long as there is some $r \in [m]$ with $\mathrm{Fix}_G(P_r) \subsetneq G$, is denoted by $\mathrm{Fix}_F(Ax \leq b)$. Note that the outcome of this procedure is independent of the choices made for r, due to Remark 3.1. We call the pair $(\tilde{I}_0, \tilde{I}_1)$ defining the cube face $\mathrm{Fix}_F(Ax \leq b)$ (unless this face is empty) the *sequential fixing of $Ax \leq b$ at (I_0, I_1)*. In the context of sequential fixing we often refer to (the computation of) $\mathrm{Fix}_F(P)$ as *simultaneous fixing*.

Due to Remark 3.1 it is clear that $\mathrm{Fix}_F(P) \subseteq \mathrm{Fix}_F(Ax \leq b)$ holds.

Theorem 3.2. In general, even for a system of facet-defining inequalities describing a full-dimensional 0/1-polytope, sequential fixing is weaker than simultaneous fixing.

Proof. The following example shows this. Let $P \subset \mathrm{C}^4$ be the 4-dimensional polytope defined by the trivial inequalities $x_i \geq 0$ for $i \in \{1, 2, 3\}$, $x_i \leq 1$ for $i \in \{1, 2, 4\}$, the inequality $-x_1 + x_2 + x_3 - x_4 \leq 0$ and $x_1 - x_2 + x_3 - x_4 \leq 0$. Let F be the cube face defined by $(\{4\}, \varnothing)$. Then, sequential fixing does not fix any further variable, although simultaneous fixing yields $I_0^\star = \{3, 4\}$ (and $I_1^\star = \varnothing$). Note that P has only 0/1-vertices, and all inequalities are facet defining ($x_4 \geq 0$ and $x_3 \leq 1$ are implied). □

4. Fixing Variables for Orbitopes

For this section, suppose that $I_0, I_1 \subseteq \mathcal{I}_{p,q}$ are subsets of indices of orbitope variables with the following properties:

(P1) $|I_0 \cap \mathrm{row}_i| \leq q(i) - 1$ for all $i \in [p]$
(P2) For all $(i, j) \in I_1$, we have $(i, \ell) \in I_0$ for all $\ell \in [q(i)] \setminus \{j\}$.

In particular, P1 and P2 imply that $I_0 \cap I_1 = \varnothing$. Let F be the face of the 0/1-cube $\mathrm{C}^{\mathcal{I}_{p,q}}$ defined by (I_0, I_1). Note that if P1 is not fulfilled, then we have $\mathrm{O}^{=}_{p,q} \cap F = \varnothing$. The following statement follows immediately from Property P2.

Remark 4.1. If a vertex x of $\mathrm{O}^{=}_{p,q}$ satisfies $x_{ij} = 0$ for all $(i, j) \in I_0$, then $x \in F$.

We assume that the face $\mathrm{Fix}_F(\mathrm{O}^{=}_{p,q})$ is defined by $(I_0^\star, I_1^\star)$, if $\mathrm{Fix}_F(\mathrm{O}^{=}_{p,q})$ is not empty. *Orbitopal fixing* is the problem to compute the simultaneous fixing $(I_0^\star, I_1^\star)$ from (I_0, I_1), or determine that $\mathrm{Fix}_F(\mathrm{O}^{=}_{p,q}) = \varnothing$.

Remark 4.2. If $\mathrm{Fix}_F(\mathrm{O}^{=}_{p,q}) \neq \varnothing$, it is enough to determine $I_0^\star$, as we have $(i, j) \in I_1^\star$ if and only if $(i, \ell) \in I_0^\star$ holds for for all $\ell \in [q(i)] \setminus \{j\}$.

4.1. Intersection of Orbitopes with Cube Faces

We start by deriving some structural results on orbitopes that are crucial in our context. Since $\mathrm{O}^{=}_{p,q} \subset \mathrm{C}^{\mathcal{I}_{p,q}}$ is a 0/1-polytope (i.e., it is integral), we have $\mathrm{conv}(\mathrm{O}^{=}_{p,q} \cap F \cap \{0, 1\}^{\mathcal{I}_{p,q}}) = \mathrm{O}^{=}_{p,q} \cap F$. Thus, $\mathrm{Fix}_F(\mathrm{O}^{=}_{p,q})$ is the smallest cube face that contains the face $\mathrm{O}^{=}_{p,q} \cap F$ of the orbitope $\mathrm{O}^{=}_{p,q}$.

Let us, for $i \in [p]$, define values $\alpha_i := \alpha_i(I_0) \in [q(i)]$ recursively by setting $\alpha_1 := 1$ and, for all $i \in [p]$ with $i \geq 2$,

$$\alpha_i \;:=\; \begin{cases} \alpha_{i-1} & \text{if } \alpha_{i-1} = q(i) \text{ or } (i, \alpha_{i-1}+1) \in I_0 \\ \alpha_{i-1}+1 & \text{otherwise.} \end{cases}$$

The set of all indices of rows, in which the α-value increases, is denoted by

$$\Gamma(I_0) \;:=\; \{i \in [p] \,:\, i \geq 2,\; \alpha_i = \alpha_{i-1}+1\} \cup \{1\}$$

(where, for technical reasons 1 is included).

The following observation follows readily from the definitions.

Remark 4.3. For each $i \in [p]$ with $i \geq 2$ and $\alpha_i(I_0) < q(i)$, the set $S_i(I_0) := \{(k, \alpha_k(I_0)+1) \,:\, k \in [i] \setminus \Gamma(I_0)\}$ is a shifted column with $S_i(I_0) \subseteq I_0$.

Lemma 4.4. For each $i \in [p]$, no vertex of $\mathrm{O}^{=}_{p,q} \cap F$ has its 1-entry in row i in a column $j \in [q(i)]$ with $j > \alpha_i(I_0)$.

Proof. Let $i \in [p]$. We may assume $\alpha_i(I_0) < q(i)$, because otherwise the statement trivially is true. Thus, $B := \{(i,j) \in \mathrm{row}_i \,:\, j > \alpha_i(I_0)\} \neq \varnothing$.

Let us first consider the case $i \in \Gamma(I_0)$. As we have $\alpha_i(I_0) < q(i) \leq i$ and $\alpha_1(I_0) = 1$, there must be some $k < i$ such that $k \notin \Gamma(I_0)$. Let k be maximal with this property. Thus we have $k' \in \Gamma(I_0)$ for all $1 < k < k' \leq i$. According to Remark 4.3, $x(B) - x(S_k(I_0)) \leq 0$ is a shifted column inequality with $x(S_k(I_0)) = 0$, showing $x(B) = 0$ as claimed in the lemma.

Thus, let us suppose $i \in [p] \setminus \Gamma(I_0)$. If $\alpha_i(I_0) \geq q(i) - 1$, the claim holds trivially. Otherwise, $B' := B \setminus \{(i, \alpha_i(I_0)+1)\} \neq \varnothing$. Similarly to the first case, now the shifted column inequality $x(B') - x(S_{i-1}(I_0)) \leq 0$ proves the claim. □

For each $i \in [p]$ we define $\mu_i(I_0) := \min\{j \in [q(i)] \,:\, (i,j) \notin I_0\}$. Because of Property P1, the sets over which we take minima here are non-empty.

Lemma 4.5. If we have $\mu_i(I_0) \leq \alpha_i(I_0)$ for all $i \in [p]$, then the point $x^\star = x^\star(I_0) \in \{0,1\}^{\mathcal{I}_{p,q}}$ with $x^\star_{i,\alpha_i(I_0)} = 1$ for all $i \in \Gamma(I_0)$ and $x^\star_{i,\mu_i(I_0)} = 1$ for all $i \in [p] \setminus \Gamma(I_0)$ and all other components being zero, is contained in $\mathrm{O}^{=}_{p,q} \cap F$.

Proof. Due to $\alpha_i(I_0) \leq \alpha_{i-1}(I_0) + 1$ for all $i \in [p]$ with $i \geq 2$, the point $x^\star$ is contained in $\mathrm{O}^{=}_{p,q}$. It follows from the definitions that $x^\star$ does not have a 1-entry at a position in I_0. Thus, by Remark 4.1, we have $x^\star \in F$. □

We now characterize the case $\mathrm{O}^{=}_{p,q} \cap F = \varnothing$ (leading to pruning the corresponding node in the branch-and-cut tree) and describe the set $I_0^\star$.

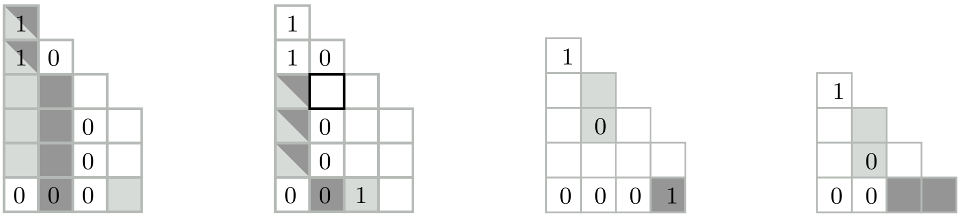
Proposition 4.6.

(1) We have $\mathrm{O}^{=}_{p,q} \cap F = \varnothing$ if and only if there exists $i \in [p]$ with $\mu_i(I_0) > \alpha_i(I_0)$.

(2) If $\mu_i(I_0) \leq \alpha_i(I_0)$ holds for all $i \in [p]$, then the following is true.

  (a) For all $i \in [p] \setminus \Gamma(I_0)$, we have

$$I_0^\star \cap \mathrm{row}_i = \{(i,j) \in \mathrm{row}_i \,:\, (i,j) \in I_0 \text{ or } j > \alpha_i(I_0)\}.$$



(a) (b) (c) (d)

Figure 2: (a): Example for Prop. 4.6 (1). Light-gray entries indicate the entries $(i, \mu_i(I_0))$ and dark-gray entries indicate entries $(i, \alpha_i(I_0))$. (b): Example of fixing an entry to 1 for Prop. 4.6 (2c). As before light-gray entries indicate entries $(i, \mu_i(I_0))$. Dark-gray entries indicate entries $(i, \alpha_i(I_0 \cup \{(s, \alpha_s(I_0))\}))$ with $s = 3$. (c) and (d): Gray entries show the SCIs used in the proofs of Parts 1(a) and 1(b) of Thm. 4.7, respectively.

(b) For all $i \in [p]$ with $\mu_i(I_0) = \alpha_i(I_0)$, we have

$$I_0^\star \cap \mathrm{row}_i = \mathrm{row}_i \setminus \{(i, \alpha_i(I_0))\}.$$

(c) For all $s \in \Gamma(I_0)$ with $\mu_s(I_0) < \alpha_s(I_0)$ the following holds: If there is some $i \geq s$ with $\mu_i(I_0) > \alpha_i(I_0 \cup \{(s, \alpha_s(I_0))\})$, then we have

$$I_0^\star \cap \mathrm{row}_s \;=\; \mathrm{row}_s \setminus \{(s, \alpha_s(I_0))\}.$$

Otherwise, we have

$$I_0^\star \cap \mathrm{row}_s \;=\; \{(s, j) \in \mathrm{row}_s \,:\, (s, j) \in I_0 \text{ or } j > \alpha_s(I_0)\}.$$

Proof. Part 1 follows from Lemmas 4.4 and 4.5.

In order to prove Part 2, let us assume that $\mu_i(I_0) \leq \alpha_i(I_0)$ holds for all $i \in [p]$. For Part 2a, let $i \in [p] \setminus \Gamma(I_0)$ and $(i, j) \in \mathrm{row}_i$. Due to $I_0 \subset I_0^\star$, we only have to consider the case $(i, j) \notin I_0$. If $j > \alpha_i(I_0)$, then, by Lemma 4.4, we find $(i, j) \in I_0^\star$. Otherwise, the point that is obtained from $x^\star(I_0)$ (see Lemma 4.5) by moving the 1-entry in position $(i, \mu_i(I_0))$ to position (i, j) is contained in $\mathrm{O}^{=}_{p,q} \cap F$, proving $(i, j) \notin I_0^\star$.

In the situation of Part 2b, the claim follows from Lemma 4.4 and because $\mathrm{O}^{=}_{p,q} \cap F \neq \varnothing$ (due to Part 1).

For Part 2c, let $s \in \Gamma(I_0)$ with $\mu_s(I_0) < \alpha_s(I_0)$ and define the new set $I_0' := I_0 \cup \{(s, \alpha_s(I_0))\}$. It follows that we have $\mu_i(I_0') = \mu_i(I_0)$ for all $i \in [p]$.

Let us first consider the case that there is some $i \geq s$ with $\mu_i(I_0) > \alpha_i(I_0')$. Part 1 (applied to I_0' instead of I_0) implies that $\mathrm{O}^{=}_{p,q} \cap F$ does not contain a vertex x with $x_{s,\alpha_s(I_0)} = 0$. Therefore, we have $(s, \alpha_s(I_0)) \in I_1^\star$, and thus $I_0^\star \cap \mathrm{row}_s = \mathrm{row}_s \setminus \{(s, \alpha_s(I_0))\}$ holds (where for “$\subseteq$“ we exploit $\mathrm{O}^{=}_{p,q} \cap F \neq \varnothing$ by Part 1, this time applied to I_0).

The other case of Part 2c follows from $s \notin \Gamma(I_0')$ and $\alpha_s(I_0') = \alpha_s(I_0) - 1$. Thus, Part 2a applied to I_0' and s instead of I_0 and i, respectively, yields the claim (because of $(s, \alpha_s(I_0)) \notin I_0^\star$ due to $s \in \Gamma(I_0)$ and $\mathrm{O}^{=}_{p,a} \cap F \neq \varnothing$). □

4.2. Sequential Fixing for Orbitopes

Let us, for some fixed $p \geq q \geq 2$, denote by $\mathcal{S}_{\mathrm{SCI}}$ the system of the nonnegativity inequalities, the row-sum equations (each one written as two inequalities, in order to be formally correct) and all shifted column inequalities.

Thus, according to Theorem 2.1, $\mathrm{O}^{=}_{p,q}$ is the set of all $x \in \mathbb{R}^{\mathcal{I}_{p,q}}$ that satisfy $\mathcal{S}_{\mathrm{SCI}}$. Let $\mathcal{S}_{\mathrm{CI}}$ be the subsystem of $\mathcal{S}_{\mathrm{SCI}}$ containing only the column inequalities (and all nonnegativity inequalities and row-sum equations).

At first sight, it is not clear whether sequential fixing with the exponentially large system $\mathcal{S}_{\mathrm{SCI}}$ can be done efficiently. A closer look at the problem reveals, however, that one can utilize the linear time separation algorithm for shifted column inequalities (mentioned in Sect. 2) in order to devise an algorithm for this sequential fixing, whose running time is bounded by $O(\varrho pq)$, where ϱ is the number of variables that are fixed by the procedure.

In fact, one can achieve more: One can compute sequential fixings with respect to the affine hull of the orbitope. In order to explain this, consider a polytope $P = \{x \in \mathrm{C}^d : Ax \leq b\}$, and let $S \subseteq \mathbb{R}^d$ be some affine subspace containing P. As before, we denote the knapsack relaxations of P obtained from $Ax \leq b$ by $P_1, \dots, P_m$. Let us define $\mathrm{Fix}^S_F(P_r)$ as the smallest cube-face that contains $P_r \cap S \cap \{0,1\}^d \cap F$. Similarly to the definition of $\mathrm{Fix}_F(Ax \leq b)$, denote by $\mathrm{Fix}^S_F(Ax \leq b)$ the face of C^d that is obtained by setting $G := F$ and then iteratively replacing G by $\mathrm{Fix}^S_G(P_r)$ as long as there is some $r \in [m]$ with $\mathrm{Fix}^S_G(P_r) \subsetneq G$. We call $\mathrm{Fix}^S_F(Ax \leq b)$ the *sequential fixing of $Ax \leq b$ at F relative to S*. Obviously, we have $\mathrm{Fix}_F(P) \subseteq \mathrm{Fix}^S_F(Ax \leq b) \subseteq \mathrm{Fix}_F(Ax \leq b)$. In contrast to sequential fixing, sequential fixing relative to affine subspaces *in general* is NP-hard (as it can be used to decide whether a linear equation has a 0/1-solution).

Theorem 4.7.

(1) There are cube-faces F^1, F^2, F^3 with the following properties:

(a) $\mathrm{Fix}_{F^1}(\mathcal{S}_{\mathrm{SCI}}) \subsetneq \mathrm{Fix}_{F^1}(\mathcal{S}_{\mathrm{CI}})$

(b) $\mathrm{Fix}^{\mathrm{aff}(\mathrm{O}^{=}_{p,q})}_{F^2}(\mathcal{S}_{\mathrm{CI}}) \subsetneq \mathrm{Fix}_{F^2}(\mathcal{S}_{\mathrm{SCI}})$

(c) $\mathrm{Fix}^{\mathrm{aff}(\mathrm{O}^{=}_{p,q})}_{F^3}(\mathcal{S}_{\mathrm{SCI}}) \subsetneq \mathrm{Fix}^{\mathrm{aff}(\mathrm{O}^{=}_{p,q})}_{F^3}(\mathcal{S}_{\mathrm{CI}})$

(2) For all cube-faces F, we have $\mathrm{Fix}^{\mathrm{aff}(\mathrm{O}^{=}_{p,q})}_{F}(\mathcal{S}_{\mathrm{SCI}}) = \mathrm{Fix}_F(\mathrm{O}^{=}_{p,q})$.

Proof. For Part 1(a), we chose $p = 5$, $q = 4$, and define the cube-face F_1 via $I^1_0 = \{(3,2),(5,1),(5,2),(5,3)\}$ and $I^1_1 = \{(1,1),(5,4)\}$. The shifted column inequality with shifted column $\{(2,2),(3,2)\}$ and bar $\{(5,4)\}$ allows to fix x_{22} to one (see Fig. 2 (c)), while no column inequality (and no nonnegativity constraint and no row-sum equation) allows to fix any variable.

For Part 1(b), let $p = 4$, $q = 4$, and define F^2 via the fixing sets $I^2_0 = \{(3,2),(4,1),(4,2)\}$ and $I^2_1 = \{(1,1)\}$. Exploiting $x_{43} + x_{44} = 1$ for all $x \in \mathrm{aff}(\mathrm{O}^{=}_{p,q}) \cap F^2$, we can use the column inequality with column $\{(2,2),(3,2)\}$ and bar $\{(4,3),(4,4)\}$ to fix x_{22} to one (see Fig. 2 (d)), while no fixing is possible with $\mathcal{S}_{\mathrm{SCI}}$ only.

For Part 1(c), we use $F^3 = F^1$. The proof of Part 2 is omitted here. □

The different versions of sequential fixing for partitioning orbitopes are dominated by each other in the following sequence: $\mathcal{S}_{\mathrm{CI}} \rightarrow \{\mathcal{S}_{\mathrm{SCI}}, \text{affine } \mathcal{S}_{\mathrm{CI}}\} \rightarrow$ affine $\mathcal{S}_{\mathrm{SCI}}$, which finally is as strong as orbitopal fixing. For each of the arrows there exists an instance for which dominance is strict. The examples in the proof of Theorem 4.7 also show that there is no general relation between $\mathcal{S}_{\mathrm{SCI}}$ and affine $\mathcal{S}_{\mathrm{CI}}$.

Algorithm 1 Orbitopal Fixing

```
 1: Set $I_0^\star \leftarrow I_0$, $I_1^\star \leftarrow I_1$, $\mu_1 \leftarrow 1$, $\alpha_1 \leftarrow 1$, and $\Gamma = \varnothing$.
 2: for $i = 2, \dots, p$ do
 3:   compute $\mu_i \leftarrow \min\{j : (i,j) \notin I_0\}$.
 4:   if $\alpha_{i-1} = q(i)$ or $(i, \alpha_{i-1}+1) \in I_0$ then
 5:     $\alpha_i \leftarrow \alpha_{i-1}$
 6:   else
 7:     $\alpha_i \leftarrow \alpha_{i-1} + 1$, $\Gamma \leftarrow \Gamma \cup \{i\}$
 8:   if $\mu_i > \alpha_i$ then
 9:     return "Orbitopal fixing is empty"
10:   Set $I_0^\star \leftarrow I_0^\star \cup \{(i,j) : j > \alpha_i\}$.
11:   if $|I_0^\star \cap \text{row}_i| = q(i) - 1$ then
12:     set $I_1^\star \leftarrow I_1^\star \cup (\text{row}_i \setminus I_0^\star)$.
13: for all $s \in \Gamma$ with $(s, \alpha_s) \notin I_1^\star$ do
14:   Set $\beta_s \leftarrow \alpha_s - 1$.
15:   for $i = s+1, \dots, p$ do
16:     if $\beta_{i-1} = q(i)$ or $(i, \beta_{i-1}+1) \in I_0$ then
17:       $\beta_i \leftarrow \beta_{i-1}$
18:     else
19:       $\beta_i \leftarrow \beta_{i-1} + 1$
20:     if $\mu_i > \beta_i$ then
21:       $I_1^\star \leftarrow I_1^\star \cup \{(s, \alpha_s)\}$ and $I_0^\star \leftarrow \text{row}_s \setminus \{(s, \alpha_s)\}$.
22:       Proceed with the next $s$ in Step 13.
```

In particular, we could compute orbitopal fixings by the polynomial time algorithm for sequential fixing relative to $\text{aff}(\text{O}_{p,q}^{=})$. It turns out, however, that this is not the preferable choice. In fact, we will describe below a linear time algorithm for solving the orbitopal fixing problem directly.

4.3. An Algorithm for Orbitopal Fixing

Algorithm 1 describes a method to compute the simultaneous fixing $(I_0^\star, I_1^\star)$ from (I_0, I_1) (which are assumed to satisfy Properties P1 and P2). Note that we use β_i for $\alpha_i(I_0 \cup \{(s, \alpha_s(I_0))\})$.

Theorem 4.8. A slight modification of Algorithm 1 solves the orbitopal fixing problem in time $O(pq)$.

Proof. The correctness of the algorithm follows from the structural results given in Proposition 4.6.

In order to prove the statement on the running time, let us assume that the data structures for the sets I_0, I_1, $I_0^\star$, and $I_1^\star$ allow both membership testing and addition of single elements in constant time (e.g., the sets can be stored as bit vectors).

As none of the Steps 3 to 12 needs more time than $O(q)$, we only have to take care of the second part of the algorithm starting in Step 13. (In fact, used verbatim as described above, the algorithm might need time $\Omega(p^2)$.)

For $s, s' \in \Gamma$ with $s < s'$ denote the corresponding β-values by β_i $(i \geq s)$ and by β'_i $(i \geq s')$, respectively. We have $\beta_i \leq \beta'_i$ for all $i \geq s'$, and furthermore, if equality holds for one of these i, we can deduce $\beta_k = \beta'_k$ for all $k \geq i$. Thus, as soon as a pair (i, β_i) is used a second time in Step 20, we can break the for-loop in Step 15 and reuse the information that we have obtained earlier.

This can, for instance, be organized by introducing, for each $(i, j) \in \mathcal{I}_{p,q}$, a flag $f(i, j) \in \{\text{red}, \text{green}, \text{white}\}$ (initialized by white), where $f(i, j) = \text{red} / \text{green}$ means that we have already detected that $\beta_i = j$ eventually leads to a positive/negative test in Step 20. The modifications that have to be applied to the second part of the algorithm are the following: The selection of the elements in Γ in Step 13 must be done in increasing order. Before performing the test in Step 20, we have to check whether $f(i, \beta_i)$ is green. If this is true, then we can proceed with the next s in Step 13, after setting all flags $f(k, \beta_k)$ to green for $s \leq k < i$. Similarly, we set all flags $f(k, \beta_k)$ to red for $s \leq k \leq i$, before switching to the next s in Step 22. And finally, we set all flags $f(k, \beta_k)$ to green for $s \leq k \leq p$ at the end of the body of the s-loop starting in Step 13.

As the running time of this part of the algorithm is proportional to the number of flags changed from white to red or green, the total running time indeed is bounded by $O(pq)$ (since a flag is never reset). □

5. Computational Experiments

We performed computational experiments for the graph partitioning problem mentioned in the introduction. The code is based on the SCIP 0.90 framework by Achterberg [1], and we use CPLEX 10.01 as the basic LP solver. The computations were performed on a 3.2 GHz Pentium 4 machine with 2 GB of main memory and 2 MB cache running Linux. All computation times are CPU seconds and are subject to a *time limit of four hours.* Since in this paper we are not interested in the performance of heuristics, we initialized all computations with the *optimal primal solution.* We compare different variants of the code by counting *winning* instances. An instance is a winner for variant A compared to variant B, if A finished within the time limit and B did not finish or needed a larger CPU time; if A did not finish, then the instance is a winner for A in case that B did also not finish, leaving, however, a larger gap than A. If the difference between the times or gaps are below 1 sec. and 0.1, respectively, the instance is not counted.

In all variants, we fix the variables x_{ij} with $j > i$ to zero. Furthermore, we heuristically separate general clique inequalities $\sum_{i,j \in C} y_{ij} \geq b$, where

$$b = \frac{1}{2}t(t-1)(q-r) + \frac{1}{2}t(t+1)r$$

and $C \subseteq V$ is a clique of size $tq + r > q$ with integers $t \geq 1$, $0 \leq r < q$ (see [3]). The separation heuristic for a fractional point $y^\star$ follows ideas of Eisenblätter [5]. We generate the graph $G' = (V, E')$ with $\{i, k\} \in E'$ if and only if $\{i, k\} \in E$ and $y^\star_{ik} < b(b+1)/2$, where $y^\star$ is the y-part of an LP-solution. We search for maximum cliques in G' with the branch-and-bound method implemented in SCIP (with a branch-and-bound node limit of 10 000) and check whether the corresponding inequality is violated.

Our default branching rule combines *first index* and *reliability branching.* We branch on the first fractional x-variable in the row-wise variable order used for defining orbitopes, but we skip columns in which a 1 has appeared before. If no such fractional variable could be found, we perform reliability branching as described by Achterberg, Koch, and Martin [2].

Table 1: Results of the branch-and-cut algorithm. All entries are rounded averages over three instances. CPU times are given in seconds.

			basic		Iso Pruning		OF		
n	m	q	nsub	cpu	nsub	cpu	nsub	cpu	#OF
30	200	3	1 082	6	821	4	697	5	6
30	200	6	358	1	122	0	57	0	25
30	200	9	1	0	1	0	1	0	0
30	200	12	1	0	1	0	1	0	0
30	300	3	3 470	87	2 729	64	2 796	69	7
30	300	6	89 919	445	63 739	168	8 934	45	353
30	300	9	8 278	19	5 463	5	131	0	73
30	300	12	1	0	1	0	1	0	0
30	400	3	11 317	755	17 433	800	9 864	660	8
30	400	6	458 996	14 400	1 072 649	11 220	159 298	3 142	1 207
30	400	9	2 470 503	14 400	1 048 256	2 549	70 844	450	7 305
30	400	12	3 668 716	12 895	37 642	53	2 098	12	1 269
50	560	3	309 435	10 631	290 603	14 400	288 558	10 471	10
50	560	6	1 787 989	14 400	3 647 369	14 400	1 066 249	9 116	4 127
50	560	9	92	0	2 978	5	10	0	10
50	560	12	1	0	1	0	1	0	0

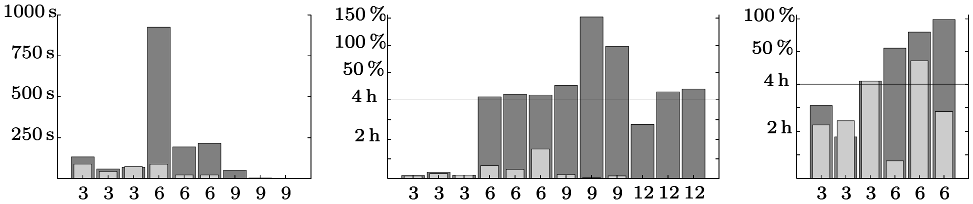


Figure 3: Computation times/gaps for the basic version (dark gray) and the version with orbitopal fixing (light gray). From left to right: instances with $n = 30$, $m = 300$, instances for $n = 30$, $m = 400$, instances for $n = 50$, $m = 560$. The number of partitions q is indicated on the x-axis. Values above 4 hours indicate the gap in percent.

We generated random instances with n vertices and m edges of the following types. For $n = 30$ we used $m = 200$ (*sparse*), 300 (*medium*), and 400 (*dense*). Additionally, for $n = 50$ we choose $m = 560$ in search for the limits of our approach. For each type we generated three instances by picking edges uniformly at random (without recourse) until the specified number of edges is reached. The edge weights are drawn independently uniformly at random from the integers $\{1, \dots, 1000\}$. For each instance we computed results for $q = 3$, 6, 9, and 12.

In a first experiment we tested the speedup that can be obtained by performing orbitopal fixing. For this we compare the variant (*basic*) without symmetry breaking (except for the zero-fixing of the upper right x-variables) and the version in which we use orbitopal fixing (*OF*); see Table 1 for the results. Columns *nsub* give the number of nodes in the branch-and-bound tree. The results show that orbitopal fixing is clearly superior (OF winners: 26, basic winners: 3), see also Figure 3.

Table 1 shows that the sparse instances are extremely easy, the instances with $m = 300$ are quite easy, while the dense instances are hard. One effect is that often for small m and large q the optimal solution is 0 and hence no work has to be done. For $m = 300$ and 400, the hardest instances arise when $q = 6$. It seems that for $q = 3$ the small number of variables helps, while for $q = 12$ the small objective function values help. Of course, symmetry breaking methods become more important when q gets larger.

In a second experiment we investigated the symmetry breaking capabilities built into CPLEX. We suspect that it breaks symmetry within the tree, but no detailed information was available. We ran CPLEX 10.01 on the IP formulation stated in Sect. 1. In one variant, we fixed variables x_{ij} with $j > i$ to zero, but turned symmetry breaking off. In a second variant, we turned symmetry breaking on and did not fix variables to zero (otherwise CPLEX seems not to recognize the symmetry). These two variants performed about equally good (turned-on winners: 13, turned-off winners: 12). The variant with no symmetry breaking and no fixing of variables performed extremely badly. The results obtained by the OF-variant above are clearly superior to the best CPLEX results (CPLEX could not solve 10 instances within the time limit, while OF could not solve 2). Probably this is at least partially due to the separation of clique inequalities and the special branching rule in our code.

In another experiment, we turned off orbitopal fixing and separated shifted column inequalities in every node of the tree. The results show that the OF-version is slightly better than this variant (OF winners: 13, SCI winners: 10), but the results are quite close (OF average time: 1563.3, SCI average time: 1596.7). Although by Part 2 of Theorem 4.7, orbitopal fixing is not stronger than fixing with SCIs (with the same branching decisions), the LPs get harder and the process slows down a bit.

Finally, we compared orbitopal fixing to the isomorphism pruning approach of Margot. We implemented the *ranked branching rule* (see [16]) adapted to the special symmetry we exploit, which simplifies Margot's algorithm significantly. It can be seen from Table 1 that isomorphism pruning is inferior to both orbitopal fixing (OF winners: 25, isomorphism pruning winners: 3) and shifted column inequalities (26:2), but is still a big improvement over the basic variant (23:7).

6. Concluding Remarks

The main contribution of this paper is a linear time algorithm for the orbitopal fixing problem, which provides an efficient way to deal with partitioning type symmetries in integer programming models. The result can easily be extended to "packing orbitopes" (where, instead of $x(\mathrm{row}_i) = 1$, we require $x(\mathrm{row}_i) \leq 1$). Our proof of correctness of the procedure uses the linear description of $\mathrm{O}^{=}_{p,q}$ given in [11]. However, we only need the validity of the shifted column inequalities in our arguments. In fact, one can devise a similar procedure for the case where the partitioning constraints $x(\mathrm{row}_i) = 1$ are replaced by covering constraints $x(\mathrm{row}_i) \geq 1$, though, for the corresponding "covering orbitopes" no complete linear descriptions are known at this time. A more detailed treatment of this will be contained in a journal version of the paper, which will also include comparisons to the isomorphism pruning method [14, 15, 17] and to orbital branching [13].

References

[1] T. Achterberg, *SCIP – A framework to integrate constraint and mixed integer programming*, Report 04-19, Zuse Institute Berlin, 2004. Available online at http://www.zib.de/Publications/abstracts/ZR-04-19/.

[2] T. Achterberg, T. Koch, and A. Martin, *Branching rules revisited*, Oper. Res. Lett. **33**, no. 1 (2005), pp. 42–54.

[3] S. Chopra and M. Rao, *The partition problem*, Math. Program. **59**, no. 1 (1993), pp. 87–115.

[4] S. Chopra and M. Rao, *Facets of the k-partition polytope*, Discrete Appl. Math. **61**, no. 1 (1995), pp. 27–48.

[5] A. Eisenblätter, *Frequency Assignment in GSM Networks: Models, Heuristics, and Lower Bounds*, PhD thesis, TU Berlin, 2001.

[6] J. Falkner, F. Rendl, and H. Wolkowicz, *A computational study of graph partitioning*, Math. Program. **66**, no. 2 (1994), pp. 211–239.

[7] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel, and L. A. Wolsey, *Formulations and valid inequalities of the node capacitated graph partitioning problem*, Math. Program. **74**, no. 3 (1996), pp. 247–266.

[8] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel, and L. A. Wolsey, *The node capacitated graph partitioning problem: A computational study*, Math. Program. **81**, no. 2 (1998), pp. 229–256.

[9] M. Grötschel and Y. Wakabayashi, *A cutting plane algorithm for a clustering problem*, Math. Prog. **45**, no. 1 (1989), pp. 59–96.

[10] M. Grötschel and Y. Wakabayashi, *Facets of the clique partitioning polytope*, Math. Prog. **47**, no. 3 (1990), pp. 367–387.

[11] V. Kaibel and M. E. Pfetsch, *Packing and partitioning orbitopes*, Math. Program. (2007). In press.

[12] G. Kochenberger, F. Glover, B. Alidaee, and H. Wang, *Clustering of microarray data via clique partitioning*, J. Comb. Optim. **10**, no. 1 (2005), pp. 77–92.

[13] J. Linderoth, J. Ostrowski, F. Rossi, and S. Smriglio, *Orbital branching*, in Proceedings of IPCO XII, M. Fischetti and D. Williamson, eds., LNCS 4513, Springer-Verlag, 2007, pp. 106–120.

[14] F. Margot, *Pruning by isomorphism in branch-and-cut*, Math. Program. **94**, no. 1 (2002), pp. 71–90.

[15] F. Margot, *Exploiting orbits in symmetric ILP*, Math. Program. **98**, no. 1–3 (2003), pp. 3–21.

[16] F. Margot, *Small covering designs by branch-and-cut*, Math. Program. **94**, no. 2–3 (2003), pp. 207–220.

[17] F. Margot, *Symmetric ILP: Coloring and small integers*, Discrete Opt. **4**, no. 1 (2007), pp. 40–62.

[18] A. Mehrotra and M. A. Trick, *Cliques and clustering: A combinatorial approach*, Oper. Res. Lett. **22**, no. 1 (1998), pp. 1–12.

[19] M. M. Sørensen, *Polyhedral computations for the simple graph partitioning problem*, working paper L-2005-02, Århus School of Business, 2005.

Paper 3

A Column-Generation Approach to Line Planning in Public Transport

Ralf Borndörfer, Martin Grötschel, and Marc E. Pfetsch
A Column-Generation Approach to Line Planning in Public Transport[1]
Transportation Sci. **41** (2007), no. 1, pp. 123–132

Abstract. The *line-planning problem* is one of the fundamental problems in strategic planning of public and rail transport. It involves finding lines and corresponding frequencies in a transport network such that a given travel demand can be satisfied. There are (at least) two objectives: the transport company wishes to minimize operating costs and the passengers want to minimize traveling times. We propose a new multicommodity flow model for line planning. Its main features, in comparison to existing models, are that the passenger paths can be freely routed and lines are generated dynamically. We discuss properties of this model, investigate its complexity, and present a column-generation algorithm for its solution. Computational results with data for the city of Potsdam, Germany, are reported.

1. Introduction

The *strategic planning* process in public and rail transport is usually divided into consecutive steps of *network design*, *line planning*, and *timetabling*. Each step can be supported by operations research methods, see for instance the survey articles of Odoni, Rousseau, and Wilson [20] and of Bussieck, Winter, and Zimmermann [7].

This article is about the *line-planning problem* (LPP) in public transport. The problem is to design line routes and their frequencies in a street

[1] Supported by the DFG Research Center Matheon in Berlin

or track network such that a transportation volume, given by a so-called *origin-destination matrix* (OD-matrix), can be routed. The frequency of a line is supposed to indicate a basic timetable period and controls the lines' transportation capacity. There are two competing objectives: on the one hand to minimize the operating costs of lines, and on the other hand to minimize user discomfort. User discomfort is usually measured by the total passenger traveling time or the number of transfers during the ride, or both.

The recent literature on the LPP mainly deals with railway networks. One common assumption is the so-called *system split*, which fixes the traveling paths of the passengers *before* the lines are known. A second common assumption is that an optimal line plan can be chosen from a (small) precomputed set of lines. Third, maximization of *direct travelers* (that travel without transfers) is often considered as the objective. In such an approach, transfer waiting times do not play a role.

This article proposes a new, extended multicommodity flow model for the LPP. The model minimizes a combination of total passenger traveling time and operating costs. It generates line routes dynamically, handles frequencies by means of continuous frequency variables, and allows passengers to change their routes according to the computed line system; in particular, we do not assume a system split. These properties aim at line-planning scenarios in public transport, in which we see less justification for a system split and fewer restrictions in line design than one seems to have in railway line planning. The goal of this article is to show that such a model is tractable and can be used to optimize the line plan of a medium-sized town.

The paper is organized as follows. Section 2 surveys the literature on the LPP. Section 3 introduces and discusses our model. Section 4 presents a column-generation solution approach. We show that the pricing problem for the passenger variables is a shortest path problem, while the pricing problem for the lines turns out to be an NP-hard longest path problem. However, if only lines of logarithmic length with respect to the number of nodes are considered, the pricing problem can be solved in polynomial time. In Section 5, computational results on a practical problem for the city of Potsdam, Germany, are reported. We end with conclusions in Section 6.

2. Related Work

This section provides a short overview of the literature for the line-planning problem. Additional information can be found in the article of Ceder and Israeli [8], which covers the literature up to the beginning of the 1990s; see also Odoni, Rousseau, and Wilson [20] and Bussieck, Winter, and Zimmermann [7].

The first approaches to the line-planning problem had the idea to assemble lines from short pieces in an iterative (and often interactive) process. An early example is the so-called skeleton method described by Silman, Barzily, and Passy [25], that chooses the endpoints of a route and several intermediate nodes which are then joined by shortest paths with respect to length or traveling time; for a variation see Dubois, Bel, and Llibre [13]. In a similar way, Sonntag [26] and Pape, Reinecke, and Reinecke [21] constructed lines by adjoining small pieces of streets/tracks to maximize the number of direct travelers.

Successive approaches precompute some set of lines in a first phase and choose a line plan from this set in a second phase; all articles discussed in the remainder of this section use this idea. For example, Ceder and Wilson [9] described an enumeration method to generate lines whose length is within a certain factor from the length of the shortest path, while Mandl [19] proposed a local search strategy to optimize over such a set. Ceder and Israeli [8, 18] introduced a quadratic set covering approach.

An important line of developments is based on the concept of the so-called *system split*. Its starting point is a classification of the links of a transportation system into levels of different speed, as is common in railway systems. Assuming that travelers are likely to change to fast levels as early and leave them as late as possible, the passengers are distributed onto several paths in the system—using Kirchhoff-like rules at the transit points—before any lines are known. This fixes the passenger flow on each individual link in the network. The system split was promoted by Bouma and Oltrogge [3], who used it to develop a branch-and-bound-based software system for the planning and analysis of the line system of the Dutch railway network.

Recently, advanced integer programming techniques have been applied to the line-planning problem. Bussieck, Kreuzer, and Zimmermann [5] (see also Bussieck [4]) and Claessens, van Dijk, and Zwaneveld [10] both propose cut-and-branch approaches to select lines from a previously generated set of potential lines and report computations on real-world railway data. Both articles deal with homogeneous transport systems, which can be assumed after a system-split is performed as a preprocessing step. Bussieck, Lindner, and Lübbecke [6] extend this work by incorporating nonlinear components into the model. Goossens, van Hoesel, and Kroon [16, 17] show that practical railway problems can be solved within reasonable time and quality by a branch-and-cut approach, even for the simultaneous optimization of several transportation systems. Schöbel and Scholl [23, 24] study a Dantzig-Wolfe decomposition approach to route passengers through an expanded line-network to minimize the number of transfers or the transfer time.

3. Line-Planning Model

We typeset vectors in bold face, scalars in normal face. If $\boldsymbol{v} \in \mathbb{R}^J$ is a real valued vector and I a subset of J, we denote by $\boldsymbol{v}(I)$ the sum over all components of $\boldsymbol{v}$ indexed by I, i.e., $\boldsymbol{v}(I) := \sum_{i \in I} v_i$.

For the line-planning problem (LPP), we are given a number M of transportation *modes* (bus, tram, subway, etc.), an undirected multigraph $G = (V, E) = (V, E_1 \dot{\cup} \ldots \dot{\cup} E_M)$ representing a multimodal transportation network, *terminal sets* $\mathcal{T}_1, \ldots, \mathcal{T}_M \subseteq V$ of nodes for each mode where lines can start and end, line *operating costs* $\boldsymbol{c}^1 \in \mathbb{Q}_+^{E_1}, \ldots, \boldsymbol{c}^M \in \mathbb{Q}_+^{E_M}$ on the edges, *fixed costs* $C_1, \ldots, C_M \in \mathbb{Q}_+$ for the set-up of a line for each mode, *vehicle capacities* $\kappa_1, \ldots, \kappa_M \in \mathbb{Q}_+$ for each mode, and *edge capacities* $\boldsymbol{\Lambda} \in \mathbb{Q}_+^E$. Denote by $G_i = (V, E_i)$ the subgraph of G corresponding to

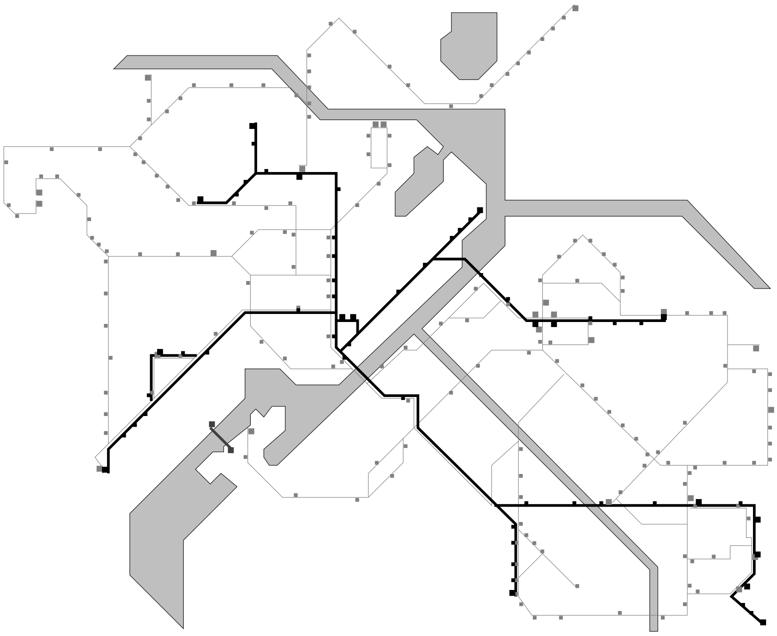


Figure 1: Multimodal transportation network in Potsdam. Black: tram, light gray: bus, dark gray: ferry, large nodes: terminals, small nodes: stations, grey: rivers and lakes.

mode i. See Figure 1 for an example network and Table 1 for a list of notation that we use throughout the paper.

A *line of mode* i is a path in G_i connecting two (different) terminals of $\mathcal{T}_i$. Note that paths are always *simple*, i.e., the repetition of nodes is not allowed; it is possible to consider additional constraints on the formation of lines such as a maximum length, etc. Let $c_\ell := \sum_{e\in\ell} c_e^i$ be the operating cost of line ℓ of mode i, $C_\ell := C_i$ be its fixed cost, and $\kappa_\ell := \kappa_i$ be its vehicle capacity. Let $\mathcal{L}$ be the set of all feasible lines. Furthermore, $\mathcal{L}_e := \bigcup\{\ell \in \mathcal{L} \ : \ e \in \ell\}$ is the set of lines that use edge $e \in E$.

The problem formulation further involves a (not necessarily symmetric) *origin-destination matrix* (OD-matrix) $(d_{st}) \in \mathbb{Q}_+^{V\times V}$ of travel demands, i.e., d_{st} is the number of passengers who want to travel from node s to node t. Let $D := \{(s,t) \in V \times V \ : \ d_{st} > 0\}$ be the set of all *OD-pairs*.

Finally, we derive a directed *passenger route graph* (V, A) from $G = (V, E)$ by replacing each edge $e \in E$ with two antiparallel arcs $a(e)$ and $\overline{a}(e)$; conversely, let $e(a) \in E$ be the undirected edge corresponding to $a \in A$. For simplicity of notation, we denote this digraph also by $G = (V, A)$. We are given *traveling times* $\tau_a \in \mathbb{Q}_+$ for every arc $a \in A$. For an OD-pair $(s,t) \in D$, an (s,t)-*passenger path* is a directed path in (V, A) from s to t. Let $\mathcal{P}_{st}$ be the set of all (s,t)-passenger paths, $\mathcal{P} := \bigcup\{p \in \mathcal{P}_{st} \ : \ (s,t) \in D\}$ the set of all passenger paths, and $\mathcal{P}_a := \bigcup\{p \in \mathcal{P} \ : \ a \in p\}$ the set of all passenger paths that use arc a. The *traveling time* of a passenger path p is defined as $\tau_p := \sum_{a\in p} \tau_a$.

With this notation, the LPP can be modeled using three kinds of variables:

Table 1: Notation and terminology.

G	multimodal transport network	G_i	subnetwork for mode i
$\mathcal{T}_i$	terminals for mode i	$\boldsymbol{c}^i$	line operating costs for mode i
c_ℓ	operating costs for line ℓ	C_i	line fixed costs for mode i
κ_i	vehicle capacity for mode i	κ_ℓ	vehicle capacity for line ℓ
$\mathcal{L}$	set of all lines	$\mathcal{L}_e$	lines using edge e
D	set of OD-pairs	d_{st}	travel demand between s and t
τ_a	traveling time on arc a	τ_p	traveling time on path p
$\mathcal{P}$	set of all passenger paths	$\mathcal{P}_{st}$	paths between s and t
y_p	passenger flow on path p	x_ℓ	whether line ℓ is used
f_ℓ	frequency of line ℓ	Λ_e	frequency bounds for edge e

$y_p \in \mathbb{R}_+$ the flow of passengers traveling from s to t on path $p \in \mathcal{P}_{st}$,
$f_\ell \in \mathbb{R}_+$ the frequency of line $\ell \in \mathcal{L}$,
$x_\ell \in \{0,1\}$ a decision variable for using line $\ell \in \mathcal{L}$.

$$\text{(LPP)} \quad \min \ \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}$$

$$\begin{array}{rll}
\boldsymbol{y}(\mathcal{P}_{st}) = d_{st} & \forall\, (s,t) \in D & \text{(i)} \\
\boldsymbol{y}(\mathcal{P}_a) - \displaystyle\sum_{\ell : e(a) \in \ell} \kappa_\ell f_\ell \leq 0 & \forall\, a \in A & \text{(ii)} \\
\boldsymbol{f}(\mathcal{L}_e) \leq \Lambda_e & \forall\, e \in E & \text{(iii)} \\
\boldsymbol{f} \leq F\boldsymbol{x} & & \text{(iv)} \\
x_\ell \in \{0,1\} & \forall\, \ell \in \mathcal{L} & \text{(v)} \\
f_\ell \geq 0 & \forall\, \ell \in \mathcal{L} & \text{(vi)} \\
y_p \geq 0 & \forall\, p \in \mathcal{P}. & \text{(vii)}
\end{array}$$

The *passenger flow constraints* (i) and the nonnegativity constraints (vii) model a multicommodity flow problem for the passenger flow, where the commodities correspond to the OD-pairs $(s,t) \in D$. This part guarantees that the demand is routed. The *capacity constraints* (ii) link the passenger paths with the line paths to ensure sufficient transportation capacity on each arc. The *frequency constraints* (iii) bound the total frequency of lines using an edge. Inequalities (iv) link the frequencies with the decision variables for the use of lines; they guarantee that the frequency of a line is zero whenever it is not used. Here, F is an upper bound on the frequency of a line; for technical reasons, we assume that $F \geq \Lambda_e$ for all $e \in E$, see Section 4 for more information.

Let us discuss some properties of the model before we investigate its algorithmic tractability.

Objectives: The objective of the model has two competing parts, namely, to minimize total passenger traveling time $\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ and to minimize costs $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}$. Here, $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x}$ is the fixed cost for setting up lines, and $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}$ is the variable cost for operating these lines at frequencies $\boldsymbol{f}$. The model allows to adjust the relative importance of one part over the other by an appropriate scaling of the respective objective coefficients. Including fixed costs allows to consider objectives such as minimizing the number of lines; note that LPP is a linear program (LP) if all fixed costs are zero.

OD-Matrices: Each entry in an OD-matrix gives the number of passengers who want to travel from one point in the network to another point within a fixed time horizon. It is well known that such data have certain deficiencies. For instance, OD-matrices depend on the geometric discretization used, they are highly aggregated, they give only a snapshot type of view, it is often questionable how well the entries represent the real situation, and they should only be used when the transportation demand can be assumed to be fixed. However, OD-matrices are at present the industry standard for estimating transportation demand. It is already quite an art and rather costly to assemble this data, and currently, no alternative is in sight.

Time horizon: The LPP implicitly contains a time horizon via the OD-matrix. Usually, OD-data are aggregated over one day, but it is similarly appropriate to consider, for instance, peak traffic in rush hours. In fact, the asymmetry of demands in rush hours was one of the reasons why we consider directed passenger paths.

Passenger Routes: Because the traveling times $\boldsymbol{\tau}$ are nonnegative, we can assume passenger routes to be (simple) paths.

Our model does not fix passenger paths according to a system split, but allows to freely route passengers according to the computed lines. This is targeted at local public transport systems, where, in our opinion, people determine their traveling paths according to the line system and not only according to the network topology. Except for the work of Schöbel and Scholl [23, 24], which is independent of ours, such routings have not been considered in the context of line planning before.

Our model computes a set of passenger paths that minimize the total traveling times $\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ in the sense of a system optimum. However, in our case, with a linear objective function and linear capacities, it can be shown that the resulting system optimum is also a user equilibrium, namely, the so-called Beckmann user equilibrium, see Correa, Schulz, and Stier Moses [11]. We do not address the question of why passengers should choose this equilibrium out of several possible equilibria that can arise in routing with capacities.

The routing in our model allows for passengers paths of arbitrary travel times, which may force some passengers to long detours. We remark that this problem could be handled by introducing appropriate bounds on the travel times of paths. This would, however, turn the pricing problem for the passenger paths into an NP-hard resource-constrained shortest path problem; see Section 4.1. Note also that such an approach would measure travel times with respect to shortest paths in the underlying network (independent of any line system). Ideally, however, one would like to compare to the shortest paths using only arcs covered by the computed line system.

Line Routes: The literature generally takes line routes as (simple) bidirected paths, and we do the same in this article. In fact, a restriction forcing some sort of simplicity is necessary to prevent repetitions around cycles. As a slight generalization of the concept of simplicity, one could investigate the case in which one assumes that every line route is bounded in length or "almost" simple, i.e., no node is repeated within a given interval.

It is easy to incorporate additional constraints on the formation of individual lines and constraints on sets of lines, e.g., that the length of a line should not deviate too much from a shortest path between its endpoints or bounds on the number of lines using an edge. Such constraints are important in practice. In this article we consider bounds on the number of edges in a line. Let us give two arguments why this case is practically relevant.

The first argument is based on an idea of a transportation network as a planar graph, probably of high connectivity. Suppose this network occupies a square, in which n nodes are evenly distributed. A typical line starts in the outer regions of the network, passes through the center, and ends in another outer region; we would expect such a line to be of length $O(\sqrt{n})$.

Real networks, however, are not only (more or less) planar, but often resemble trees. But in a *balanced* and preprocessed tree, where each node degree is at least three, the length of a path between any two nodes is only $O(\log n)$.

Transfers: Transfers between lines are currently ignored in our model, because constraints (ii) only control the total capacity on edges and not the assignment of passengers to lines. The problem are not transfers between different modes, which can be handled by linking the mode networks G_i with appropriate transfer edges, weighted by estimated transfer times. In principle, a similar trick could be used for transfers between lines of the same mode, using an appropriate expansion of the graph. However, this greatly increases the complexity of the model, and it introduces degeneracy; it is unclear whether such a model remains tractable for practical data.

Frequencies: Frequencies indicate the (approximate) number of times vehicles need to be employed to serve the demand over the time horizon. In a real-world line plan, frequencies often have to produce a regular timetable and, hence, are not allowed to take arbitrary fractional values. Our model, however, treats frequencies as continuous values. This is a simplification. We have introduced fixed costs to reduce the number of lines and decrease the likelihood of low frequencies. In addition, we could have forced our model to accept only a finite number of frequencies by enumerating lines with fixed frequencies in a similar way as Claessens, van Dijk, and Zwaneveld [10] and Goossens, van Hoesel, and Kroon [16, 17]; but the resulting model would be much harder to solve. However, as the frequencies mainly are used to adjust line capacities, we do (at present) not care so much about "nice" frequencies and view the fractional values as approximations or clues to "sensible" values.

4. Column Generation

The LP relaxation of (LPP) can be simplified by eliminating the $\boldsymbol{x}$-variables. In fact, since (LPP) minimizes over nonnegative costs, one can assume w.l.o.g. that inequalities (iv) above are satisfied with equality, i.e., there is an optimal LP solution such that $Fx_\ell = f_\ell \Leftrightarrow x_\ell = f_\ell/F$ for all lines ℓ. Substituting for $\boldsymbol{x}$, we observe that the inequalities $f_\ell \leq F$ remaining after the elimination are dominated by inequalities (iii) and, hence, can be omitted (recall that we assumed $F \geq \Lambda_e$). Setting $\gamma_\ell = C_\ell/F + c_\ell$, we arrive at the following equivalent, but simpler, linear program:

$$\text{(LP)} \qquad \min\ \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{f}$$

$$\begin{aligned} \boldsymbol{y}(\mathcal{P}_{st}) &= d_{st} && \forall\,(s,t) \in D && \text{(i)}\\ \boldsymbol{y}(\mathcal{P}_a) - \sum_{\ell : e(a) \in \ell} \kappa_\ell f_\ell &\le 0 && \forall\, a \in A && \text{(ii)}\\ \boldsymbol{f}(\mathcal{L}_e) &\le \Lambda_e && \forall\, e \in E && \text{(iii)}\\ f_\ell &\ge 0 && \forall\, \ell \in \mathcal{L} && \text{(iv)}\\ y_p &\ge 0 && \forall\, p \in \mathcal{P}. && \text{(v)} \end{aligned}$$

Note that (LP) contains only a polynomial number of inequalities (apart from the nonnegativity constraints (iv) and (v)).

We aim at solving (LP) with a column-generation approach (see Barnhart et al. [2] for an introduction) and therefore investigate the corresponding pricing problems. These pricing problems are studied in terms of the dual of (LP). Denote the variables of the dual as follows: $\boldsymbol{\pi} = (\pi_{st}) \in \mathbb{R}^D$ (flow constraints (i)), $\boldsymbol{\mu} = (\mu_a) \in \mathbb{R}^A$ (capacity constraints (ii)), and $\boldsymbol{\eta} \in \mathbb{R}^E$ (frequency constraints (iii)). The dual of (LP) is:

$$\max\ \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\pi} - \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\eta}$$

$$\begin{aligned} \pi_{st} - \boldsymbol{\mu}(p) &\le \tau_p && \forall\, p \in \mathcal{P}_{st},\ (s,t) \in D\\ \kappa_\ell\, \boldsymbol{\mu}(\ell) - \boldsymbol{\eta}(\ell) &\le \gamma_\ell && \forall\, \ell \in \mathcal{L}\\ \boldsymbol{\mu},\ \boldsymbol{\eta} &\ge 0, \end{aligned}$$

where

$$\boldsymbol{\mu}(\ell) = \sum_{e \in \ell} \big(\mu_{a(e)} + \mu_{\overline{a}(e)}\big).$$

It will turn out that the pricing problem for the line variables f_ℓ is a longest path problem; the pricing problem for the passenger variables y_p, however, is a shortest path problem.

4.1. Pricing of the Passenger Variables

The reduced cost $\overline{\tau}_p$ for variable y_p with $p \in \mathcal{P}_{st}$, $(s,t) \in D$, is

$$\overline{\tau}_p = \tau_p - \pi_{st} + \boldsymbol{\mu}(p) = \tau_p - \pi_{st} + \sum_{a \in p} \mu_a = -\pi_{st} + \sum_{a \in p} (\mu_a + \tau_a).$$

The pricing problem for the $\boldsymbol{y}$-variables is to find a path p such that $\overline{\tau}_p < 0$ or to conclude that no such path exists. This easily can be done in polynomial time as follows. For all $(s,t) \in D$, we search for a shortest (s,t)-path p with respect to the nonnegative weights $(\mu_a + \tau_a)$ on the arcs; we can, for instance, use Dijkstra's algorithm. If the length of this path p is less than π_{st}, then y_p is a candidate variable to be added to the LP, otherwise, we proved that no such path exists (for the pair (s,t)). Note that we can assume that each passenger path is simple: just remove cycles of length 0 – or trust Dijkstra's algorithm, which produces only simple paths.

4.2. Pricing of the Line Variables

The pricing problem for line variables f_ℓ is more complicated. The reduced cost $\overline{\gamma}_\ell$ for a variable f_ℓ is

$$\overline{\gamma}_\ell = \gamma_\ell - \kappa_\ell\,\boldsymbol{\mu}(\ell) + \boldsymbol{\eta}(\ell) = \gamma_\ell - \sum_{e\in\ell}\big(\kappa_\ell\,(\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e\big).$$

The corresponding pricing problem consists of finding a (simple) path ℓ of mode i such that

$$\begin{array}{rcl} 0 > \overline{\gamma}_\ell & = & \gamma_\ell - \sum_{e\in\ell}\big(\kappa_\ell\,(\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e\big) \\ & = & C_\ell/F + c_\ell - \sum_{e\in\ell}\big(\kappa_\ell\,(\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e\big) \\ & = & C_i/F + \sum_{e\in\ell} c^i_e - \sum_{e\in\ell}\big(\kappa_i\,(\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e\big) \\ & = & C_i/F + \sum_{e\in\ell}\big(c^i_e - \kappa_i\,(\mu_{a(e)} + \mu_{\overline{a}(e)}) + \eta_e\big) \\ \Leftrightarrow & \multicolumn{2}{l}{\sum_{e\in\ell}(\kappa_i\,(\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e - c^i_e) > C_i/F.} \end{array}$$

This problem turns out to be a maximum weighted path problem, because the weights $(\kappa_i\,(\mu_{a(e)} + \mu_{\overline{a}(e)}) - \eta_e - c^i_e)$ are not restricted in sign. Hence, the pricing problem for the line variables is NP-hard [15]. This shows that solving the LP relaxation (LP) is NP-hard as well. In fact, we can prove the stronger result that the line-planning problem itself is NP-hard, even with fixed costs zero, independent of the model (Proposition 4.1 implies that (LP) is NP-hard, because (LPP) is equivalent to (LP) for fixed costs 0).

Proposition 4.1. The line-planning problem LPP is NP-hard, even with fixed costs 0.

Proof. We reduce the Hamiltonian path problem, which is strongly NP-complete [15], to the LPP with fixed costs 0. Let (H, s, t) be an instance of the Hamiltonian path problem, i.e., $H = (V, E)$ is a graph and s and t are two distinct nodes of H.

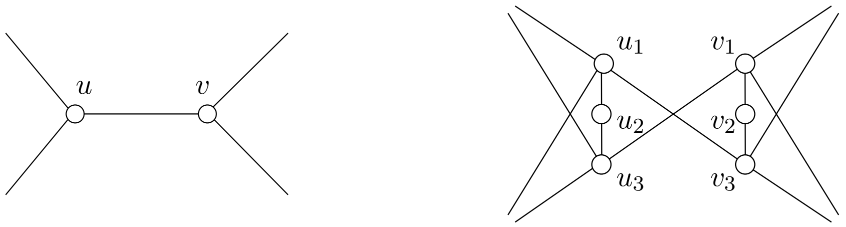


Figure 2: Example for the node splitting gadget in the proof of Proposition 4.1

For the reduction, we are going to derive an appropriate instance of LPP. The underlying network is formed by a graph $H' = (V', E')$, which arises from H by splitting each node v into three copies v_1, v_2, and v_3. For each node $v \in V$, we add edges $\{v_1, v_2\}$ and $\{v_2, v_3\}$ to E' and for each edge $\{u, v\}$ the edges $\{u_1, v_3\}$ and $\{u_3, v_1\}$, see Figure 2. Our instance of LPP contains just a single mode with only two terminals s_1 and t_3 such that every line must start at s_1 and end at t_3. The demands are $d_{v_1 v_2} = 1$ $(v \in V)$ and 0 otherwise, and the capacity of every line is 1 For every $e \in E'$, we set Λ_e to some high value (e.g., to $|V'|$). The cost of all edges is set to 0, except for the edges incident to s_1, for which the costs are set to 1. The traveling times are set to 0 everywhere. It follows that the value of a solution to LPP is the sum of the frequencies of all lines.

Assume that $p = (s, v^1, \dots, v^k, t)$ (for $v^1, \dots, v^k \in V$) is an (s,t)-Hamiltonian path in H. Then $p' = (s_1, s_2, s_3, v^1_1, v^1_2, v^1_3, \dots, v^k_1, v^k_2, v^k_3, t_1, t_2, t_3)$ is an (s_1, t_3)-Hamiltonian path in H', which gives rise to an optimal solution of LPP. Namely, we can take p' as the route of a single line with frequency 1 and route the demands $d_{v_1v_2} = 1$ for every $v \in V$ on this line directly from v_1 to v_2. As the frequency of p' is 1, the objective value of this solution is also 1. On the other hand, every solution to LPP must have value at least one, because every line has to pass an edge incident to s_1 and the sum of the frequencies of lines visiting an arbitrary edge of type $\{v_1, v_2\}$, for $v \in V$, is at least 1. This proves that LPP has a solution of value 1, if (H, s, t) contains a Hamiltonian path.

For the converse, assume that there exists a solution to LPP of value 1, for which we ignore lines with frequency 0. We know that every edge $\{v_1, v_2\}$ ($v \in V$) is covered by at least one line of the solution. If every line contains all edges $\{v_1, v_2\}$ ($v \in V$), each such line gives rise to a Hamiltonian path (because the line paths are simple) and we are done. Otherwise, there must be an edge $e = \{v_1, v_2\}$ ($v \in V$) that is not covered by all of the lines. Because the lines have to provide enough capacity, the sum of the frequencies of the lines covering e is at least 1. However, the edges incident to s_1 are covered by the lines covering edge e plus at least one more line of nonzero frequency. Hence, the total sum of all frequencies is larger than one, which is a contradiction to the assumption that the solution has value 1.

This shows that there exists an (s,t)-Hamiltonian path in H if and only if an optimal solution of LPP with respect to H' has value 1. □

4.3. Pricing of Length Restricted Lines

Let us now consider the pricing problem for line-planning problems with bounds on the lengths of the lines, i.e., the number of edges of a line. Consider for this purpose the graph $G = (V, E)$ (for simplicity of notation with only one mode) with arbitrary edge weights $w_e \in \mathbb{Q}$ for all $e \in E$, and a source node s and a sink node t. We let $n = |V|$ and $m = |E|$. In this setting, the line-pricing problem is to find a maximum weight path from s to t with respect to $\boldsymbol{w}$. We first show that this problem is NP-hard for the case in which the length of a line is bounded by $O(\sqrt{n})$.

Proposition 4.2. It is NP-hard to compute a maximum weight path from s to t of length at most k, if $k \in O\big(n^{1/N}\big)$ for any fixed $N \in \mathbb{N} \setminus \{0\}$.

Proof. Let (H, s, t) be an instance of the Hamiltonian path problem, in which H is a graph with n nodes. We add $(n^N - n)$ isolated nodes to H in order to obtain a graph H' with n^N nodes; note that n^N is polynomial in n for fixed N. Let the weights on the edges be 1. If we could find a maximum weight path from s to t with at most $k = (n^N)^{1/N} = n$ edges in polynomial time, we could solve the Hamiltonian path problem for H in polynomial time. □

We now provide a result that shows that the maximum weighted path problem can be solved in polynomial time in the case when the lengths of the paths are at most $O(\log n)$. Our method is a direct generalization of work by Alon, Yuster, and Zwick [1] on the unweighted case; it works both for directed and undirected graphs.

Alon et al. consider the problem to find simple paths of fixed length $k-1$ in a graph. Their basic idea is to randomly color the nodes of the graph with k colors and only allow paths that use distinct colors for each node; such paths are called *colorful* with respect to the coloring and are necessarily simple. Choosing a coloring $c : V \to \{1, \dots, k\}$ uniformly at random, every path using at most $k-1$ edges has a chance of at least $k!/k^k > e^{-k}$ to be colorful with respect to c. If we repeat this process $\alpha \cdot e^k$ times with $\alpha > 0$, the probability that a given path p with at most $k-1$ edges is never colorful is less than

$$\bigl(1 - e^{-k}\bigr)^{\alpha \cdot e^k} < e^{-\alpha}.$$

Hence, the probability that p is colorful at least once is at least $1 - e^{-\alpha}$. The search for such colorful paths can be performed using dynamic programming, which leads to an algorithm running in $m \cdot 2^{O(k)}$ expected time. This algorithm is then derandomized.

These arguments yield the following result for the weighted undirected case, which is easily seen to be valid for directed graphs as well.

Proposition 4.3. Let $G = (V, E)$ be a graph with m edges, k be a fixed number, and $c : V \to \{1, \dots, k\}$ be a coloring of the nodes of G. Let s be a node in G and (w_e) be edge weights. Then a colorful maximum weight path with respect to $\boldsymbol{w}$ using at most $k-1$ edges from s to every other node can be found in time $O\bigl(m \cdot k \cdot 2^k\bigr)$, if such paths exist.

Proof. We find the maximum weight of such paths by dynamic programming. Let $v \in V$, $i \in \{1, \dots, k\}$, and $C \subseteq \{1, \dots, k\}$ with $|C| \leq i$. Define $w(v, C, i)$ to be the weight of the maximum weight colorful path with respect to $\boldsymbol{w}$ from s to v using at most $i-1$ edges and using the colors in C. Hence, for each iteration i, we store the set of colors of all maximum weight colorful paths from s to v using at most $i-1$ edges. Note that we do not store the set of paths, only their colors. Hence, at each node, we store at most 2^i entries. The entries of the table are initialized with minus infinity, and we set $w(s, \{c(s)\}, 1) = 0$.

At iteration $i \geq 1$, let (u, C, i) be an entry in the dynamic programming table. If for some edge $e = \{u, v\} \in E$ we have $c(v) \notin C$, let $C' = C \cup \{c(v)\}$ and set

$$w(v, C', i+1) = \max\bigl\{w(u, C, i) + w_e,\ w(v, C', i+1),\ w(v, C', i)\bigr\}.$$

The term $w(v, C', i+1)$ accounts for the cases in which we already found a path to v (using at most i edges) with higher weight, whereas $w(v, C', i)$ makes sure that paths using at most $i-1$ edges to v are accounted for. After iteration $i = k$, we take the maximum of all entries corresponding to each node v, which is the wanted result. The number of updating steps is bounded by

$$\sum_{i=0}^{k} i \cdot 2^i \cdot m = m \cdot \bigl(2 + 2^{k+1}(k-1)\bigr) = \mathrm{O}\left(m \cdot k \cdot 2^k\right).$$

The sum on the left side of this equation arises as follows. In iteration i, m edges are considered; each edge $\{u, v\}$ starts at node u, to which at most 2^i labels $w(u, C, i)$ are associated, one for each possible set C; for each such set, checking whether $c(v) \in C$ takes time $O(i)$. The summation formula itself can be proved by induction, see also [22, Exc. 5.7.1, p. 95]. The algorithm can be easily modified to actually find the maximum weight paths. □

We can use Proposition 4.3 to produce an algorithm that finds a maximum weight path in $\alpha\, e^k\, O\big(mk2^k\big) = \alpha\, O\big(m \cdot 2^{O(k)}\big)$ time with high probability. Then a derandomization can be performed by a clever enumeration of colorings such that each path with at most $k-1$ edges is colorful with respect to at least one such coloring. Alon et al. combine several techniques to show that $2^{O(k)} \cdot \log n$ colorings suffice. Applying this result we obtain the following.

Theorem 4.4. Let $G = (V, E)$ be a graph with n nodes and m edges and k be a fixed number. Let s be a node in G and (w_e) be edge weights. Then a maximum weight path with respect to $\boldsymbol{w}$ using at most $k-1$ edges from s to every other node can be found in time $O\big(m \cdot 2^{O(k)} \cdot \log n\big)$, if such paths exist.

If $k \in O(\log n)$, this yields a polynomial time algorithm. Hence, by the discussion above, we get the following result.

Corollary 4.5. The LP relaxation of (LPP) can be solved in polynomial time, if the lengths of the lines are most k, with $k \in O(\log n)$.

4.4. Algorithm

We used the results of the previous sections to implement a column-generation algorithm for the solution of the model (LPP) with length-restricted lines. As an overall objective function, we used the weighted sum

$$\lambda\,(\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}) + (1-\lambda)\,\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y},$$

where $\lambda \in [0, 1]$ is a parameter weighing the two parts.

The algorithm solves the LP relaxation in a first phase and constructs a feasible line plan using a greedy type heuristic in a second phase.

To solve the LP relaxation, our algorithm iteratively prices out passenger and line path variables until no improving variables are found. We solve the master LP with the barrier algorithm and, toward the end of the process, with the primal simplex algorithm of CPLEX 9.1. We check for new passenger path variables for all OD-pairs using Dijkstra's algorithm, see Section 4.1, until no more improving passenger paths are found. If we do not find an improving passenger path, we price out line variables for all line modes and all feasible terminal pairs. We have implemented two different methods for the pricing of (simple) line paths, namely, we either use an enumeration or the randomized coloring algorithm of Section 4.3 (we do not derandomize the algorithm). If an improving passenger or line path has been found, another iteration is started; otherwise, the LP is solved.

In the second phase, our algorithm tries to construct a good integer solution from a line pool consisting of the lines having nonzero frequencies in the optimal LP solution. The heuristic is motivated by the observation that the solution of the LP relaxation of a line-planning problem often contains lines with very low frequencies. We try to remove these lines by a simple greedy method based on a strong branching selection criterion. In the beginning, the $\boldsymbol{x}$-variables of all lines in the pool are set to 1. In each iteration, we tentatively remove a line (set its x-variable to 0), compute the objective $\lambda\, \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f} + (1-\lambda)\, \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ of the LP obtained by fixing the line variables as described, pricing passenger variables as needed, and add the fixed costs $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x}$ of all lines that are fixed to 1. After probing candidate lines with the smallest $\boldsymbol{f}$-values in this way, we permanently delete the line whose removal resulted in the smallest objective. We repeat this elimination as long as the remaining set of lines is still feasible, i.e., all demands can be routed, and the objective function decreases.

5. Computational Results

In this section, we report on computational experience with line-planning problems for the city of Potsdam, Germany. The experiments originate from a joint project with the two local public transport companies, ViP Verkehrsgesellschaft GmbH and Havelbus Verkehrsgesellschaft mbH, the city of Potsdam, and the software company IVU Traffic Technologies AG.

Potsdam is a medium sized town near Berlin; it has about 150,000 inhabitants. Its public transportation system uses city buses and trams (operated by ViP) and regional buses (operated by Havelbus). Additionally, regional trains connect Potsdam to its surroundings (operated by Deutsche Bahn AG) and a city railroad (operated by S-Bahn Berlin) provides connections to Berlin. Because regional trains and the city railroad are not operated by ViP and Havelbus, the associated lines routes are assumed to be fixed.

5.1. Data

Our data consists of a multimodal traffic network of Potsdam and an associated OD-matrix, which had been used by IVU in a consulting project for planning the Potsdam network (Nahverkehrsplan). The data represents the 1998 line system of Potsdam. It has 27 bus lines and 4 tram lines. Including line variants, the total number of lines was 80. The network has 951 nodes, including 111 OD-nodes, and 1,321 edges. The maximum length of a line is 47 edges.

The network was preprocessed as follows. We removed isolated nodes. Then, we iteratively removed "leaves" in the graph—i.e., nodes with only one neighbor—and iteratively contracted nodes with two neighbors. The preprocessed graph has 410 nodes, 106 of which were OD-nodes, and 891 edges. We remark that although such preprocessing steps are conceptually easy, the data handling can be quite intricate in practice; for instance, our data included information on possible turnings of a line at road/rail crossings, which must be updated in the course of the preprocessing.

The OD-matrix was also modified. Nodes with zero traffic were removed. The original time horizon was one day, but we wanted to construct a line plan for the peak hour. We therefore scaled the matrix to 40% in an (admittedly

Table 2: Experimental results of line planning for $\lambda = 0.9978$.

Optimized LP solution – enumeration:		
total traveling time:	108,360,036.33	[scaled: 238,392.08]
total line cost:	233,776.86	[scaled: 233,262.55]
LP objective value:	471,654.63	
active line/pass. var.:	60/4,879	transfers: 8,777/64,607
Optimized LP solution – randomized coloring – 5 trials:		
total traveling time:	108,396,741.75	[scaled: 238,472.83]
total line cost:	239,099.73	[scaled: 238,573.71]
LP objective value:	477,046.54	
active line/pass. var.:	61/4,880	transfers: 9,143/66,546
Optimized LP solution – randomized coloring – 15 trials:		
total traveling time:	108,491,234.25	[scaled: 238,680.72]
total line cost:	237,422.50	[scaled: 236,900.17]
LP objective value:	475,580.88	
active line/pass. var.:	62/4,885	transfers: 9,387/68,049
Optimized integer solution – greedy heuristic:		
total traveling time:	112,581,291.50	[scaled: 247,678.84]
total line cost:	287,060.90	[scaled: 286,429.37]
integer objective value:	818,491.68	
active line/pass. var.:	30/4,767	transfers: 8,638/60,539
Reference LP solution:		
total traveling time:	105,269,846.00	[scaled: 231,593.66]
total line cost:	501,376.24	[scaled: 500,273.21]
LP objective value:	731,866.87	
active line/pass. var.:	61/4,857	transfers: 8,618/63,310
Reference integer solution – greedy heuristic:		
total traveling time:	106,952,869.00	[scaled: 235,296.31]
total line cost:	562,964.54	[scaled: 561,726.02]
integer objective value:	1,213,221.49	
active line/pass. var.:	44/4,814	transfers: 9,509/70,525

rough) attempt to simulate afternoon traffic (3 p.m. to 6 p.m.). Note that the resulting matrix is still quite symmetric (the maximum difference between each of the two directions was 25) whereas a real afternoon OD-matrix would not be symmetric. The scaled OD-matrix had 4685 nonzeros and the total scaled travel demand was 42796.

All traveling times are measured in seconds and we always restricted the maximum length of a line to 55 edges. Because no data was available on line costs, we decided on $C_\ell = 10000$ (fixed costs) for each line ℓ and $c_e^i = 100$ (operating costs) for each edge e and mode i. Hence, we do not distinguish between costs of different modes (an unrealistic assumption in practice).

5.2. Experiments

Table 2 reports the results of several computational experiments with the data and implementation we have described. All experiments were performed using a 3.4 GHz Pentium 4 machine running Linux. In the table, the *total traveling time* is $\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ and the *total line cost* is $\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{f}$, the *scaled* values are $(1-\lambda)\,\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$ and $\lambda\,\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{f}$, respectively; all four values refer to the LP relaxation (LP). The *LP objective value* is $\lambda\,\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{f} + (1-\lambda)\,\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$, the *integer objective value* refers to $\lambda\,(\boldsymbol{C}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{f}) + (1-\lambda)\,\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}$. The last line in each block of results gives the number of active (i.e., nonzero) line and passenger variables, and the number of passenger transfers (first number) that were needed as well as the number of transfering passengers (second number). Note that we can compute transfers from passenger routes as an afterthought, although our optimization model is currently insensitive to them.

Let us point out explicitly that we do not claim that our results are already practically significant; we only want to show that there is potential to apply our methods to practical data. For example, our costs are not realistic. Therefore, the frequencies we compute cannot be compared to ones used in practice. To allow some adaptation to our cost model, we let the frequencies of all lines be variable, in particular, the frequencies of the city railroad and regional train lines.

In our first experiment, we solved the LP relaxation (LP) of the Potsdam problem, pricing lines either by enumeration or by the randomized coloring method of Section 4.3, see top of Table 2. We set $\lambda = 0.9978$, which roughly balances the two parts of the objective function. The resulting LP had 5761 rows. Using enumeration, we obtained an optimal solution after 451 seconds and 283 iterations (i.e., solutions of the master LP), of which 15 were used to price lines. The pricing problems needed a total time of 183 seconds of which most was used for the pricing of line paths. Hence, more than half the time is spent for solving the master LPs.

We repeated this experiment using the randomized coloring algorithm with 5 and 15 trials for line pricing. With 5 trials, we needed 397 master LPs and 394 seconds in total; line pricing used only 99 seconds. One can see, however, that the objective is about 1% higher than for the enumeration variant. Using 15 trials resulted in 269 master LPs and 473 seconds in total. Line pricing now uses 265 seconds, and the difference in the objective function relative to the enumeration variant is reduced to 0.8%. Hence, one can achieve a good approximation of the optimal value using randomized line pricing, although approaching the optimum solution comes at the cost of larger computation times.

We also investigated in more detail the passenger routing of our LP solution for the enumeration variant. To connect the 4,685 OD-pairs only 4,879 paths are needed, i.e., most OD-pairs are connected by a unique path. The total traveling time is 108,360,036.33 seconds, see Table 2. For comparison, when we ignore capacities and route all passengers between every OD-pair on the fastest path in the final line system, the total traveling time is 95,391,460 seconds. This relative difference of 12% seems to be an acceptable deviation.

In our second experiment, we computed two integer solutions for (LPP) associated with the parameter $\lambda = 0.9978$, as above. The first solution is obtained by rounding all nonzero $\boldsymbol{x}$-variables in the solution of the LP relaxation, computed with the enumeration variant, to 1. The (integer) objective of this rounded solution is 1,058,079.69, which leads to a gap of 55%

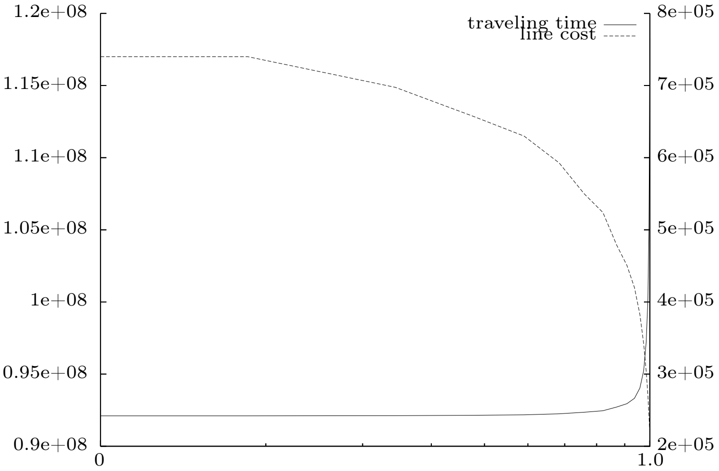


Figure 3: Total traveling time (solid, left axis) and total line cost (dashed, right axis) in dependence on λ (x-axis in logscale).

compared to the LP relaxation value of 471,654.63. The second solution is obtained by the greedy algorithm described in Section 4.4, starting from the same LP solution (only lines for city buses, trams, and regional buses were removed). It has 30 lines (17 bus lines and 2 tram lines), down from 60 in the first solution, see Table 2; it took 1,368 seconds to compute. The final (scaled) operating costs are 286,429.37, while the final fixed costs are $\lambda \cdot 300,000 = 299,340$. The integer objective of 818,491.68 has a gap of 42% with respect to the LP relaxation value of 471,654.63. Note that the results heavily depend on the cost structure: decreasing the fixed costs automatically reduces the gap. In our context, with high fixed costs, emphasis is on reducing the number of lines (recall that the costs were artificial). The result obtained seems to be quite good, given that the original line system contained 27 bus lines and 4 tram lines; it seems unlikely that one can further reduce the number. Furthermore, the lower bound of the LP relaxations typically is very weak for such fixed-cost problems. Still, more research is needed to provide better lower bounds and primal solutions.

We compare the LP and integer solutions to "reference solutions" shown in the lower part of Table 2. The reference LP solution is obtained by fixing the paths of the original lines of Potsdam and then solving the resulting LP relaxation without generating new lines, but allowing the frequencies of the lines to change. The reference integer solution is obtained by applying the greedy heuristic to the reference LP solution. The results show that allowing the generation of new line paths reduces line costs in both cases to roughly 50% and the total objective to roughly 2/3 of the original values, while the total traveling time increases by a small percent. Hence, in these experiments, the greedy algorithm has not changed the relative improvement obtained from optimizing lines.

Our third experiment investigates the influence of the parameter λ on the solution. We computed the solutions to the LP relaxation for 21 different values of λ_i, taking $\lambda_i = 1 - \big(1 - i/20\big)^4$, for $i = 0, \ldots, 20$. This collects increasingly more samples near $\lambda = 1$, a region where the total traveling time and total line cost are about equal.

The results are plotted in Figure 3. This figure shows the total traveling time and the total line cost depending on λ. The extreme cases are as expected: For $\lambda = 0$, the line costs do not contribute to the objective and are therefore high, while the total traveling time is low. For $\lambda = 1$, only the total line cost contributes to the objective and is therefore minimized as much as possible at the cost of increasing the total traveling time. With increasing λ, the total line cost monotonically decreases, while the total traveling time increases. Note that each computed pair of total traveling time and line cost constitutes a Pareto optimal point, i.e., is not dominated by any other attainable combination. Conversely, any Pareto optimal solution of the LP relaxation can be obtained as the solution for some $\lambda \in [0,1]$, see, e.g., Ehrgott [14].

6. Conclusions

We proposed a new model for line planning in public transport that allows to generate lines dynamically and to freely route passengers according to the computed lines. The model allows to deal with manifold requirements from practice. We showed that line-planning problems for a medium-sized town can be solved within reasonable quality with integer programming techniques. Our computational results indicate significant optimization potential. Our results on the polynomial time solvability of the LP relaxation for the case of logarithmic line lengths raises our hope that the model is suited to deal with larger problems as well.

Acknowledgment

The authors thank Volker Kaibel for pointing out Proposition 4.2.

References

[1] N. Alon, R. Yuster, and U. Zwick, *Color-coding*, J. Assoc. Comput. Mach. **42**, no. 4 (1995), pp. 844–856.

[2] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, *Branch-and-price: Column generation for solving huge integer programs*, Oper. Res. **46**, no. 3 (1998), pp. 316–329.

[3] A. Bouma and C. Oltrogge, *Linienplanung und Simulation für öffentliche Verkehrswege in Praxis und Theorie*, Eisenbahntechnische Rundschau **43**, no. 6 (1994), pp. 369–378.

[4] M. R. Bussieck, *Optimal lines in public rail transport*, PhD thesis, TU Braunschweig, 1997.

[5] M. R. Bussieck, P. Kreuzer, and U. T. Zimmermann, *Optimal lines for railway systems*, Eur. J. Oper. Res. **96**, no. 1 (1997), pp. 54–63.

[6] M. R. Bussieck, T. Lindner, and M. E. Lübbecke, *A fast algorithm for near optimal line plans*, Math. Methods Oper. Res. **59**, no. 2 (2004).

[7] M. R. Bussieck, T. Winter, and U. T. Zimmermann, *Discrete optimization in public rail transport*, Math. Program. **79**, no. 1–3 (1997), pp. 415–444.

[8] A. CEDER AND Y. ISRAELI, *Scheduling considerations in designing transit routes at the network level*, in Proc. of the Fifth International Workshop on Computer-Aided Scheduling of Public Transport (CASPT), Montréal, Canada, 1990, M. Desrochers and J.-M. Rousseau, eds., Lecture Notes in Economics and Mathematical Systems 386, Springer-Verlag, Berlin, Heidelberg, 1992, pp. 113–136.

[9] A. CEDER AND N. H. M. WILSON, *Bus network design*, Transportation Res. **20B**, no. 4 (1986), pp. 331–344.

[10] M. T. CLAESSENS, N. M. VAN DIJK, AND P. J. ZWANEVELD, *Cost optimal allocation of rail passanger lines*, Eur. J. Oper. Res. **110**, no. 3 (1998), pp. 474–489.

[11] J. R. CORREA, A. S. SCHULZ, AND N. E. STIER MOSES, *Selfish routing in capacitated networks*, Math. Oper. Res. **29** (2004), pp. 961–976.

[12] J. R. DADUNA, I. BRANCO, AND J. M. P. PAIXÃO, eds., *Proc. of the Sixth International Workshop on Computer-Aided Scheduling of Public Transport (CASPT), Lisbon, Portugal, 1993*, Lecture Notes in Economics and Mathematical Systems 430, Springer-Verlag, Berlin, Heidelberg, 1995.

[13] D. DUBOIS, G. BEL, AND M. LLIBRE, *A set of methods in transportation network synthesis and analysis*, J. Oper. Res. Soc. **30**, no. 9 (1979), pp. 797–808.

[14] M. EHRGOTT, *Multicriteria optimization*, Springer-Verlag, Berlin, 2nd ed., 2005.

[15] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.

[16] J.-W. H. M. GOOSSENS, S. VAN HOESEL, AND L. G. KROON, *On solving multi-type line planning problems*, METEOR Research Memorandum RM/02/009, University of Maastricht, 2002.

[17] J.-W. H. M. GOOSSENS, S. VAN HOESEL, AND L. G. KROON, *A branch-and-cut approach for solving railway line-planning problems*, Transportation Sci. **38**, no. 3 (2004), pp. 379–393.

[18] Y. ISRAELI AND A. CEDER, *Transit route design using scheduling and multi-objective programming techniques*, in Daduna et al. [12], pp. 56–75.

[19] C. E. MANDL, *Evaluation and optimization of urban public transportation networks*, Eur. J. Oper. Res. **5** (1980), pp. 396–404.

[20] A. R. ODONI, J.-M. ROUSSEAU, AND N. H. M. WILSON, *Models in urban and air transportation*, in Handbooks in OR & MS 6, S. M. Pollock et al., ed., North Holland, 1994, ch. 5, pp. 107–150.

[21] U. PAPE, Y.-S. REINECKE, AND E. REINECKE, *Line network planning*, in Daduna et al. [12], pp. 1–7.

[22] M. PETKOVSEK, H. S. WILF, AND D. ZEILBERGER, *$A = B$*, A. K. Peters, Wellesley, MA, 1996.

[23] A. SCHÖBEL AND S. SCHOLL, *Line planning with minimal travelling time*, Tech. Report 1-2005, University of Göttingen, Germany, 2005.

[24] S. SCHOLL, *Customer-Oriented Line Planning*, PhD thesis, University of Göttingen, 2005.

[25] L. A. SILMAN, Z. BARZILY, AND U. PASSY, *Planning the route system for urban buses*, Comput. Oper. Res. **1** (1974), pp. 201–211.

[26] H. SONNTAG, *Ein heuristisches Verfahren zum Entwurf nachfrageorientierter Linienführung im öffentlichen Personennahverkehr*, Z. Oper. Res. **23** (1979), pp. B15–B31.

Paper 4

Computing Optimal Morse Matchings

Michael Joswig and Marc E. Pfetsch
Computing Optimal Morse Matchings[1]
SIAM J. Discrete Math. **20** (2006), no. 1, pp. 11–25

Abstract. Morse matchings capture the essential structural information of discrete Morse functions. We show that computing optimal Morse matchings is NP-hard and give an integer programming formulation for the problem. Then we present polyhedral results for the corresponding polytope and report on computational results.

1. Introduction

Discrete Morse theory was developed by Forman [8, 10] as a combinatorial analog to the classical smooth Morse theory. Applications to questions in combinatorial topology and related fields are numerous: e.g., Babson et al. [3], Forman [9], Shareshian [30], Batzies and Welker [4], and Jonsson [19].

It turns out that the topologically relevant information of a discrete Morse function f on a simplicial complex can be encoded as a (partial) matching in its Hasse diagram (considered as a graph), the *Morse matching* of f. A matching in the Hasse diagram is Morse if it satisfies a certain, entirely combinatorial, acyclicity condition. Unmatched k-dimensional faces are called *critical*; they correspond to the critical points of index k of a smooth Morse function. The total number of noncritical faces equals twice the number of edges in the Morse matching. The purpose of this paper is to study algorithms which compute maximum Morse matchings of a given finite simplicial complex. This is equivalent to finding a Morse matching with as few critical faces as possible.

[1] The authors' research was partially supported by the DFG Research Center Matheon in Berlin.

A Morse matching M can be interpreted as a discrete flow on a simplicial complex Δ. The flow indicates how Δ can be deformed into a more compact description as a CW complex with one cell for each critical face of M. Naturally one is interested in a most compact description, which leads to the combinatorial optimization problem described above. This way optimal (or even sufficiently good) Morse matchings of Δ can help to recognize the topological type of a space given as a finite simplicial complex. The latter problem is known to be undecidable even for highly structured classes of topological spaces, such as smooth 4-manifolds. We have to admit, however, that so far no new topological results have been obtained by our approach.

Optimization of discrete Morse matchings has been studied by Lewiner, Lopes, and Tavares [23, 24]. Hersh [17] investigated heuristic approaches to the maximum Morse matching problem with applications to combinatorics. Morse matchings can also be interpreted as pivoting strategies for homology computations; see [20]. Furthermore, the set of all Morse matchings of a given simplicial complex itself has the structure of a simplicial complex; see [6].

The paper is structured as follows. First we show that computing optimal Morse matchings is NP-hard. This issue has been addressed previously by Lewiner, Lopes, and Tavares [24], but their argument omits details which to us seem quite important to address carefully. Then we give an integer programming (IP) formulation for the problem. The formulation consists of two parts: one for the matching conditions and one for the acyclicity constraints. This turns out to be related to the acyclic subgraph problem studied by Grötschel, Jünger, and Reinelt [14]. We derive polyhedral results for the corresponding polytope. In particular, we give two different polynomial time algorithms for the separation of the acyclicity constraints. The paper closes with computational results.

Like most of discrete Morse theory, also most of our results extend to arbitrary finite regular CW-complexes. We stick to the simplicial setting, however, to simplify the presentation.

2. Discrete Morse Functions and Morse Matchings

We will first introduce discrete Morse functions as developed by Forman. The essential structure of discrete Morse functions is captured by so-called Morse matchings; see Forman [8] and Chari [5]. It turns out that this latter formulation directly leads to a combinatorial optimization problem in which one wants to maximize the size of a Morse matching.

We first need some notation. Let Δ be a *(finite abstract) simplicial complex*, i.e., a set of subsets of a finite set V with the following property: if $F \in \Delta$ and $G \subseteq F$, then $G \in \Delta$; in other words, Δ is an independence system with ground set V. In the following we will ignore $\varnothing$ as a member of Δ. The elements in V are called *vertices* and the sets in Δ are called *faces*. The *dimension* of a face F is $\dim F := |F| - 1$. Let $d = \max\{\dim F : F \in \mathcal{F}\}$ be the dimension of Δ. We often write i-faces for i-dimensional faces. Let $\mathcal{F}$ be the set of faces of Δ and let $f_i = f_i(\Delta)$ be the number of faces of dimension $i \geq 0$. The maximal faces with respect to inclusion are called *facets* and 1-faces are called edges. The complex Δ is *pure*, if all facets have the same dimension. For F, $G \in \Delta$, we write $F \prec G$ if $F \subset G$ and $\dim F = \dim G - 1$, i.e., "$\prec$" denotes the covering relation in the Boolean lattice. The *graph* of Δ is the (abstract) graph on V in which two vertices are connected by an edge if there exists a 1-face containing both vertices. Throughout this paper we assume that Δ is *connected*, i.e., its graph is connected. This is no loss of generality since the connected components can be treated separately.

The *size* of Δ is defined as the coding length of its face lattice, i.e., if Δ has n faces, then $\operatorname{size} \Delta = O(n \cdot d \cdot \log n)$. Statements about the complexity of algorithms in the subsequent sections are always with respect to this notion of size.

A function $f : \Delta \to \mathbb{R}$ is a *discrete Morse function* if for every $G \in \Delta$ the sets

$$\{F : F \prec G,\ f(G) \leq f(F)\} \quad \text{and} \quad \{H : G \prec H,\ f(H) \leq f(G)\} \tag{1}$$

both have cardinality at most 1. The first set includes the faces covered by face G which are not assigned a lower value than G, while the second set includes the faces covering G which are not assigned a higher value. The face G is *critical* if both sets have cardinality 0. A simple example of a discrete Morse function can be obtained by setting $f(F) = \dim F$ for every $F \in \Delta$. With respect to this function every face is critical.

Discrete Morse functions are interesting because they can be used to deform a simplicial complex into a (smaller) CW-complex that has a cell for each critical face; see Section 3.

Consider the *Hasse diagram* $H = (\mathcal{F}, A)$ of Δ, that is, a directed graph on the faces of Δ with an arc $(G, F) \in A$ if $F \prec G$; note that the arcs lead from higher to lower dimensional faces. Let $M \subset A$ be a matching in H, i.e., each face is incident to at most one arc in M. Let $H(M)$ be the directed graph obtained from H by reversing the direction of the arcs in M. Then M is a *Morse matching* of Δ if $H(M)$ does not contain directed cycles, i.e., is acyclic (in the directed sense). Morse matchings are also often called *acyclic matchings*. Given $M \subset A$, one can decide in linear time (in the size of Δ) whether it is a Morse matching: the matching conditions are trivial and acyclicity of $H(M)$ can be checked by depth first search in linear time (see, e.g., Korte and Vygen [22]).

There is the following relation between Morse functions and Morse matchings; see Forman [8] and Chari [5]. Let f be a discrete Morse function and let M be the set of arcs $(G, F) \in A$ such that $f(G) \leq f(F)$, i.e., f is not decreasing on these arcs. A simple proof shows that at most one of the sets in (1) can have cardinality one. This shows that M is a matching. Since the order given by f can be refined to a linear ordering of the faces of Δ, the directed graph $H(M)$ is in fact acyclic and therefore a Morse matching. To construct a discrete Morse function from a Morse matching, compute a linear ordering extending $H(M)$ (which is acyclic) and then number the faces consecutively in the reverse order.

Although we lose the concrete numbers attached to the faces when going from a discrete Morse function f to the corresponding Morse matching M, we do not lose the information about critical faces: Critical faces of f are exactly the unmatched faces of M. Hence, by maximizing $|M|$ we minimize the number of critical faces of f. In fact, the number of critical faces is $|\mathcal{F}| - 2\,|M|$. For $0 \leq j \leq d$, let $c_j = c_j(M)$ be the number of critical faces of dimension j and let $c(M)$ be the total number of critical faces.

It seems helpful to briefly describe the case of Morse matchings for a one-dimensional simplicial complex Δ. Then Δ represents the incidences of a graph G. A Morse matching M of Δ matches edges with nodes of G. Let $\tilde{G}$ be the following oriented subgraph of G: take all edges which are matched in M and orient them towards its matched node. Since M is a matching, this construction is well defined and the in-degree of each node is at most one. The acyclicity property shows that $\tilde{G}$ contains no directed cycles and hence is a branching, i.e., the underlying graph is a forest and each (weakly) connected component has a unique root. Therefore, the Morse matchings on a graph G are in one-to-one correspondence with orientations of subgraphs of G which are branchings.

Building on this idea, Lewiner, Lopes, and Tavares [23] computed maximum Morse matchings, i.e., Morse matchings with maximal cardinality, for combinatorial 2-manifolds. In [24] they developed a heuristic for computing Morse matchings for arbitrary simplicial complexes. In the general case, however, this problem is NP-hard, as shown in Section 4.

3. Properties of Morse Matchings

In this section we briefly review some important properties of Morse matchings which we need in what follows.

Let F be a facet of Δ and let G be a facet of F, which is not contained in any other facet of Δ. The operation of transforming Δ to $\Delta \setminus \{F, G\}$ is called a *simplicial* or *elementary collapse*. We will simply use collapse in the following.

Proposition 3.1 (Forman [8]). Let Δ be a simplicial complex and Σ a subcomplex of Δ. Then there exists a sequence of collapses from Δ to Σ if and only if there exists a discrete Morse function such that $\Delta \setminus \Sigma$ contains no critical face.

Forman [8] also proved the following result, which describes one of the most interesting features of Morse matchings:

Theorem 3.2. Let Δ be a simplicial complex and M be a Morse matching on Δ. Then Δ is homotopy equivalent to a CW-complex containing a cell of dimension i for each critical face of dimension i.

We refer to Munkres [27] for more information on CW-complexes. By Theorem 3.2 we can hope for a compact representation of the topology of Δ (up to homotopy) by computing a Morse matching with few critical faces. This is the main motivation for the combinatorial optimization problem studied in this paper.

Let K be a field and let $\beta_j = \beta_j(K)$ be the Betti number for dimension j over K for Δ; see again Munkres [27] for details. Forman [8] proved the following bounds on the number of critical faces c_j of a Morse matching M:

Theorem 3.3 (*Weak Morse inequalities*)**.** Let K be a field, Δ be a simplicial complex, and M a Morse matching for Δ. We have

$$c_j \geq \beta_j \qquad \text{for all } j = 0, \ldots, d \tag{2}$$

and

$$c_0 - c_1 + c_2 - \cdots + (-1)^d c_d = \beta_0 - \beta_1 + \beta_2 - \cdots + (-1)^d \beta_d. \tag{3}$$

The Betti numbers over $\mathbb{Q}$ and finite fields can easily be obtained in polynomial time (in the size of Δ), by computing the ranks of the boundary matrices for each dimension. Although harder to compute (see Iliopoulos [18]), the homology over $\mathbb{Z}$ can be used to choose among the finite fields or $\mathbb{Q}$, in order to obtain the strongest form of the Morse inequalities (2).

4. Hardness of Optimal Morse Matchings

In this section we prove NP-hardness of the problem to compute a maximum Morse matching, i.e., to find a Morse matching M with maximal cardinality. As we saw previously, this is equivalent to minimizing the number of critical faces.

We want to reduce the following *collapsibility problem*, introduced by Eğecioğlu and Gonzalez [7], to the problem of finding an optimal Morse matching: Given a connected pure 2-dimensional simplicial complex Δ that is embeddable in $\mathbb{R}^3$ and an integer k, decide whether there exists a subset $\mathcal{K}$ of the facets of Δ with $|\mathcal{K}| \leq k$ such that there exists a sequence of collapses which transforms $\Delta \setminus \mathcal{K}$ to a 1-dimensional complex. Eğecioğlu and Gonzalez proved that this collapsibility problem is strongly NP-complete. Using Proposition 3.1, this result reads as follows in terms of discrete Morse theory.

Theorem 4.1. Given a connected pure 2-dimensional simplicial complex Δ that is embeddable in $\mathbb{R}^3$ and a nonnegative integer k, it is NP-complete in the strong sense to decide whether there exists a Morse matching with at most k critical 2-faces.

When k is fixed, we can try all possible sets $\mathcal{K}$ of size at most k and then decide whether the resulting complex is collapsible to a 1-dimensional complex in polynomial time. Therefore we let k be part of the input.

We need the following construction. Consider a Morse matching M for a simplicial complex Δ, with $\dim \Delta \geq 1$. Let $\Gamma(M)$ be the graph obtained from the graph of Δ by removing all edges (1-faces) matched with 2-faces. Note that $\Gamma(M)$ contains all vertices of Δ.

Lemma 4.2. The graph $\Gamma(M)$ is connected.

Proof. Without loss of generality we assume that $\dim \Delta \geq 2$. Otherwise, $\Gamma(M)$ coincides with the graph of Δ, which is connected (recall that Δ is connected).

Suppose that $\Gamma(M)$ is disconnected. Let N be its set of nodes in a connected component of $\Gamma(M)$, and let C be the set of *cut edges*, that is, edges of Δ with one vertex in N and one vertex in its complement. Since Δ is connected, C is not empty. By definition of $\Gamma(M)$, each edge in C is matched to a unique 2-face.

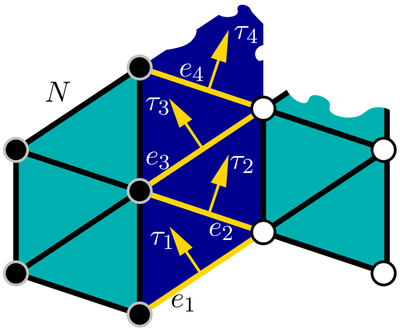


Figure 1: Illustration of the proof of Lemma 4.2.

Consider the directed subgraph D of the Hasse diagram consisting of the edges in C and their matching 2-faces. The standard direction of arcs in the Hasse diagram (from the higher to the lower dimensional faces) is reversed for each matching pair of M, i.e., D is a subgraph of $H(M)$.

We construct a directed path in D as follows; see Figure 1. Start with any node of D corresponding to a cut edge e_1. Go to the node of D determined by the unique 2-face τ_1 to which e_1 is matched to. Then τ_1 contains at least one other cut edge e_2, otherwise e_1 cannot be a cut edge. Now iteratively go to e_2, then to its unique matching 2-face τ_2, choose another cut edge e_3, and so on. We observe that we obtain a directed path $e_1, \tau_1, e_2, \tau_2, \ldots$ in D, i.e., the arcs are directed in the correct direction.

Since we have a finite graph at some point the path must arrive at a node of D which we have visited already. Hence, D (and therefore also $H(M)$) contains a directed cycle, which is a contradiction since M is a Morse matching. □

Now pick an arbitrary node r and any spanning tree of $\Gamma(M)$ (which can be computed in polynomial time; see Korte and Vygen [22]) and direct all edges away from r. This yields a maximum Morse matching on $\Gamma(M)$; see the end of Section 2. It is easy to see that replacing the part of M on $\Gamma(M)$ with this matching yields a Morse matching. This Morse matching has only one critical vertex (the root r). Note that every Morse matching contains at least one critical vertex; this can be seen from the Morse inequalities (2) in Theorem 3.3. Furthermore, the total number of critical faces can only decrease, since we computed an optimal Morse matching on $\Gamma(M)$. The number of critical i-faces for $i \geq 2$ stays the same. We have thus proved the following corollary, which is also implicit in Forman [8].

Corollary 4.3. Let M be a Morse matching on Δ. Then we can compute a Morse matching M' in polynomial time which has exactly one critical vertex and the same number of critical faces of dimension 2 or higher as M, such that $c(M') \leq c(M)$.

We can now prove the hardness result.

Theorem 4.4. Given a simplicial complex Δ and a nonnegative integer c, it is strongly NP-complete to decide whether there exists a Morse matching with at most c critical faces, even if Δ is connected, pure, 2-dimensional, and can be embedded in $\mathbb{R}^3$.

Proof. Clearly this problem is in NP. So let (Δ, k) be an input for the collapsibility problem. We claim that there exists a Morse matching with at most k critical 2-faces if and only if there exists a Morse matching with at most $g(k) := 2(k+1) - \chi(\Delta)$ critical faces altogether. Here, $\chi(\Delta) = \beta_0 - \beta_1 + \cdots + (-1)^d \beta_d$ is the Euler characteristic, which can be computed in polynomial time; see Section 3. Hence g is a polynomial-time computable function. Using Theorem 4.1 then finishes the proof.

So assume that M is a Morse matching on Δ with at most k critical 2-faces. We use Corollary 4.3 to compute a Morse matching M', in polynomial time, such that $c_0(M') = 1$, $c_2(M') = c_2(M)$, and $c(M') \leq c(M)$. By (3) of Theorem 3.3, we have $c_1(M') = c_2(M') + 1 - \chi(\Delta)$. Since $c(M') = c_0(M') + c_1(M') + c_2(M')$ it follows that

$$c_2(M) = c_2(M') = \tfrac{1}{2}(c(M') + \chi(\Delta)) - 1. \tag{4}$$

Solving for $c(M')$, it follows that M' has at most $2(k+1) - \chi(\Delta)$ critical faces altogether.

Conversely, assume there exists a Morse matching M with at most $g(k)$ critical faces. Computing M' as above, we obtain by (4), that

$$c_2(M) = c_2(M') \leq \tfrac{1}{2}(g(k) + \chi(\Delta)) - 1 = k,$$

which completes the proof. □

Lewiner, Lopes, and Tavares [24] showed that it is NP-hard to compute an optimal Morse matching, but their proof omits an argument similar to Lemma 4.2 above. We therefore provided a proof for it.

Since there exists a Morse matching with at most c critical faces if and only if there exists a Morse matching of size at least $\frac{1}{2}(|\mathcal{F}| - c)$, we proved the following corollary.

Corollary 4.5. Let Δ be as in Theorem 4.4 and m be a nonnegative integer. Then it is NP-complete in the strong sense to decide whether there exists a Morse matching of size at least m.

We do not know about the complexity status for this problem with m fixed.

Eğecioğlu and Gonzalez [7] additionally proved that the collapsibility problem is as hard to approximate as the set covering problem. In particular, the collapsibility problem cannot be approximated better than within a logarithmic factor in polynomial time, unless P = NP. Using this, Lewiner, Lopes, and Tavares [24] claimed that the problem to compute a Morse matching minimizing the number of critical faces is hard to approximate. However, the function g used in the proof above is not "approximation preserving" and we do not see how the nonapproximability result carries over.

Similarly, the problem to approximate the size of a Morse matching seems to be open.

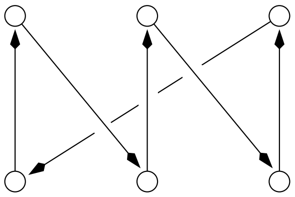


Figure 2: Example for a directed cycle of size 6; at least three arcs with reversed orientation (pointing "up") are necessary to close a 6-cycle in the Hasse diagram of a simplicial complex.

5. An IP-Formulation

In this section we introduce an integer programming formulation for the problem to compute a Morse matching of maximal size. From now on we assume that $\dim \Delta \geq 1$, since the other cases are uninteresting in our context.

We use the following notation. We depict vectors in bold font. Let $\boldsymbol{e}_i$ be the ith unit vector and let $\mathbb{1}$ be the vector of all ones. For any vector $\boldsymbol{x} \in \mathbb{R}^n$ and $I \subseteq \{1, \ldots, n\}$ we define

$$\boldsymbol{x}(I) := \sum_{i \in I} x_i.$$

Furthermore, for $S \subseteq \{1, \ldots, n\}$, $\boldsymbol{I}(S) \in \mathbb{R}^n$ denotes the incidence vector of S.

For a node v in a directed graph, let $\delta(v)$ be the arcs incident to v, i.e., the arcs having v as one of their endnodes. For a subset $A' \subseteq A$, we denote by $N(A')$ the nodes incident to at least one arc in A'. Throughout this article, all directed or undirected cycles are assumed to be *simple*, i.e., without node repetitions.

For ease of notation, we consider the Hasse diagram H as directed or undirected depending on the context; we will explicitly say *directed* when we refer to the directed version.

We split H into d levels $H_0 = (\mathcal{F}^0, A_0)$, $\ldots$, $H_{d-1} = (\mathcal{F}^{d-1}, A_{d-1})$, where H_i denotes the level of the Hasse diagram between faces of dimension i and $i+1$. Then A is the disjoint union of $A_0, \ldots, A_{d-1}$ and $\mathcal{F}^{i-1} \cap \mathcal{F}^i$ consists of the faces of dimension i. Recall that the arcs in the Hasse diagram are directed from the higher to the lower dimensional faces.

Let $M \subset A$ be a Morse matching of Δ. By definition, its incidence vector $\boldsymbol{x} = \boldsymbol{I}(M) \in \{0,1\}^A$ satisfies the *matching inequalities*

$$\boldsymbol{x}(\delta(F)) \leq 1 \quad \forall\, F \in \mathcal{F}. \tag{5}$$

Now assume that for some $M \subseteq A$ there exists a directed cycle D in $H(M)$. Then in D "up" and "down" arcs alternate; for an example, see Figure 2. In particular, the size of D is always even. Hence, $\frac{1}{2}|D|$ arcs are contained in M, i.e., are reversed in $H(M)$. We will use the following well-known observation.

Observation. Let $M \subset A$ be a matching. If D is a directed cycle in $H(M)$, the edges in D can only belong to one level H_i ($i \in \{0, \ldots, d-1\}$), i.e., we have $\{\dim F \,:\, F \in N(D)\} = \{i, i+1\}$.

Putting these arguments together we obtain: If M is acyclic, $\boldsymbol{x} = \boldsymbol{I}(M)$ satisfies the following *cycle inequalities*:

$$\boldsymbol{x}(C) \leq \tfrac{1}{2}|C| - 1 \quad \forall\, C \in \mathcal{C}_i,\ i = 1, \dots, d-1, \tag{6}$$

where $\mathcal{C}_i$ are the cycles in H_i.

Conversely, it is easy to see that every $\boldsymbol{x} \in \{0,1\}^A$ which fulfills inequalities (5) and (6) is the incidence vector of a Morse matching. Hence, we arrive at the following IP formulation for the problem to find a maximum Morse matching:

$$\begin{array}{lrll}
(\textsc{MaxMM}) & \max & \mathbb{1}^{\mathrm{T}}\boldsymbol{x} & \\
& \text{s.t.}\ \boldsymbol{x}(\delta(F)) \leq 1 & & \forall\, F \in \mathfrak{F} \\
& \boldsymbol{x}(C) \leq \tfrac{1}{2}|C| - 1 & & \forall\, C \in \mathcal{C}_i,\ i = 1, \dots, d-1 \\
& \boldsymbol{x} \in \{0,1\}^A. & &
\end{array}$$

This formulation can easily be extended to arbitrary weights on the arcs, i.e., replacing $\mathbb{1}$ in the objective function by an arbitrary nonnegative vector $\boldsymbol{w}$.

A different view on this optimization problem is to find directed spanning trees in the hypergraph defined by H_i and to patch them together (see Warme et al. [31] for spanning trees in hypergraphs).

We define the corresponding polytope as

$$P_M = \operatorname{conv}\big\{\boldsymbol{x} \in \{0,1\}^A \ :\ \boldsymbol{x} \text{ satisfies (5) and (6)}\big\}.$$

Let M be a Morse matching and $\boldsymbol{x} = \boldsymbol{I}(M)$ be its incidence vector. Then $F \in \mathfrak{F}$ is a critical face with respect to M if and only if it is unmatched by M, i.e., $\boldsymbol{x}(\delta(F)) = 0$. Hence, the total number of critical faces is

$$c(M) = \sum_{F \in \mathfrak{F}} \Big(1 - \sum_{a \in \delta(F)} x_a\Big) = |\mathfrak{F}| - 2 \sum_{a \in A} x_a = |\mathfrak{F}| - 2\,\mathbb{1}^{\mathrm{T}}\boldsymbol{x}, \tag{7}$$

since every arc is incident to exactly two nodes. Using this formula one can easily switch between the number of critical faces and the number of arcs in a Morse matching.

The LP relaxation of MaxMM can be strengthened by using the weak Morse inequalities (2) of Theorem 3.3. Applying (7), this yields the following *Betti inequality* for dimension i:

$$\sum_{F:\dim F = i} \Big(1 - \sum_{a \in \delta(F)} x_a\Big) \geq \beta_i \quad \Leftrightarrow \quad \sum_{F:\dim F = i}\ \sum_{a \in \delta(F)} x_a \leq f_i - \beta_i. \tag{8}$$

Observe that we can choose the field in Theorem 3.3 to employ the Morse inequalities in their strongest form.

Example 5.1. This can be illustrated by the real projective plane $\mathbb{RP}_2$. The Betti numbers with respect to $\mathbb{Q}$ and $\mathbb{Z}_2$ are $\boldsymbol{\beta}(\mathbb{Q}) = (1,0,0)$ and $\boldsymbol{\beta}(\mathbb{Z}_2) = (1,1,1)$, respectively. The resulting lower bounds are $(1,1,1)$, i.e., we have at least three critical faces in any Morse matching (this is, in fact, optimal).

Remark 5.2. The cycle inequalities (6) are similar to the cycle inequalities for the acyclic subgraph problem (ASP); see Jünger [21], and Grötschel,

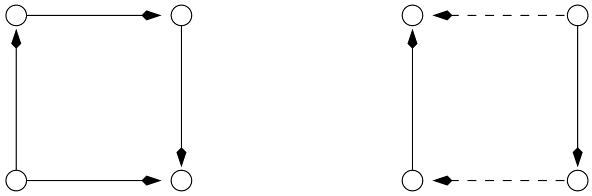


Figure 3: Example of a nonmonotone behavior of acyclic matchings. The directed graph on the right, obtained from the left graph by reversing the dashed arcs, is acyclic. However, if the top arc is set to its original orientation, the graph is not acyclic anymore. This shows that subsets of acyclic matchings are not necessarily acyclic.

Jünger, and Reinelt [14]. The separation problem for (6), however, is more complicated than the corresponding problem for ASP; see Section 5.2.

Furthermore, there is a similarity to the relation between the ASP and the linear ordering problem (see Reinelt [28], and Grötschel, Jünger, and Reinelt [13]): an alternative formulation for our problem can be obtained by modeling discrete Morse functions as linear orders on the faces, subject to matching requirements. Since this formulation is based on the relation between faces, it leads to quadratically many variables in the number of faces; therefore we have opted for the above formulation, at the cost of having to solve the separation problem for the cycle inequalities; see Section 5.2.

5.1. Facial Structure of $\boldsymbol{P_M}$

It is easy to see that P_M is a full dimensional polytope and $x_a \geq 0$ defines a facet for every $a \in A$. Furthermore, P_M is monotone, since every subset of a Morse matching is a Morse matching. It is well known that this implies that every facet defining inequality $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{x} \leq \beta$ not equivalent to the nonnegativity inequalities fulfills $\boldsymbol{\alpha} \geq 0$, $\beta > 0$; see Hammer, Johnson, and Peled [16].

Interestingly, if we consider acyclic matchings as defined above for arbitrary acyclic directed graphs, the collection of such acyclic matchings is not necessarily monotone anymore; see the example in Figure 3. Therefore, the structure of the generalized problem is likely to be more complicated.

We have the following two results.

Proposition 5.3. The matching inequalities $\boldsymbol{x}(\delta(F)) \leq 1$ define facets of P_M for $F \in \mathcal{F}$, except if $|\delta(F)| = 1$, in which case F is a vertex.

Proof. Let F be a face with $|\delta(F)| > 1$ (note that $|\delta(F)| = 0$ does not occur since $\dim \Delta \geq 1$ and Δ is connected). We can assume that $A = \{a_1, \dots, a_k, a_{k+1}, \dots, a_m\}$, where $\delta(F) = \{a_1, \dots, a_k\}$. For $i = k+1, \dots, m$, observe that a_i cannot be adjacent to every arc in $\delta(F)$: since $|\delta(F)| > 1$, a_i would either be incident to at least two nodes of the same dimension or to two nodes whose dimensions are two apart, which is impossible. Therefore, choose $p(i) \in \{1, \dots, k\}$ such that a_i and $a_{p(i)}$ are not adjacent. It follows that $\boldsymbol{e}_i + \boldsymbol{e}_{p(i)} \in P_M$. Then

$$\boldsymbol{e}_1, \dots, \boldsymbol{e}_k, \boldsymbol{e}_{k+1} + \boldsymbol{e}_{p(k+1)}, \dots, \boldsymbol{e}_m + \boldsymbol{e}_{p(m)}$$

are affinely independent and fulfill $\boldsymbol{x}(\delta(F)) = 1$. □

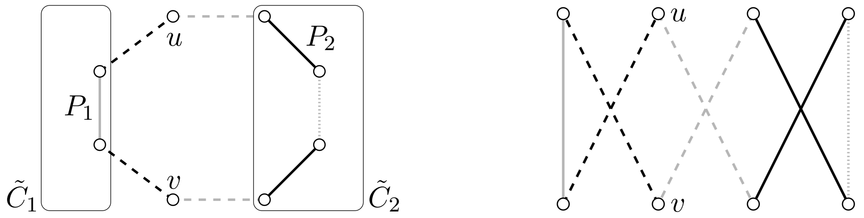


Figure 4: Illustration of the first case in the proof of Theorem 5.4. The sets P_1 and P_2 are shown by continuous lines. The edges in C_1 are drawn gray and hence $P_1 \subset C_1$; edges in C_2 are drawn black. The dashed edges incident to u and v are not considered. The right-hand side shows the graph embedded in the Hasse diagram.

It follows that the inequalities $x_a \leq 1$, $a \in A$, never define facets, since each arc has a nonvertex endpoint.

Theorem 5.4. The cycle inequalities (6) define facets of P_M.

Proof. We extend the corresponding proof by Jünger [21] for the ASP.

Let C be a cycle in H. Without loss of generality assume that $A = \{a_1, \ldots, a_k, a_{k+1}, \ldots, a_m\}$, where $C = (a_1, \ldots, a_k)$ and k is even. We will construct affinely independent feasible vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k, \boldsymbol{v}_{k+1}, \ldots, \boldsymbol{v}_m$ satisfying the cycle inequality corresponding to C with equality.

Let $C_1 = \{a_1, a_3, \ldots, a_{k-1}\}$ and $C_2 = \{a_2, a_4, \ldots, a_k\}$. Hence C_1 and C_2 are the “up” and “down” arcs in C.

Define

$$\boldsymbol{v}_i = \begin{cases} \boldsymbol{I}(C_1 \setminus \{a_i\}) & \text{if } a_i \in C_1 \\ \boldsymbol{I}(C_2 \setminus \{a_i\}) & \text{if } a_i \in C_2 \end{cases} \qquad \text{for } i = 1, \ldots, k.$$

Hence, for $i = 1, \ldots, k$ we have $\boldsymbol{v}_i(C) = \frac{k}{2} - 1$.

For $i = k+1, \ldots, m$, consider $a_i = \{u, v\} \notin C$. We have four cases.

- ▷ $u, v \in N(C)$: Let $\tilde{C} := C \setminus \big(\delta(u) \cup \delta(v)\big)$. We have that $|\tilde{C}| = k - 4$ (since there exist no odd cycles) and $\tilde{C}$ splits into two odd nonempty parts, $\tilde{C}_1$ and $\tilde{C}_2$, which are both paths. Let $k_1 := |\tilde{C}_1|$ and $k_2 := |\tilde{C}_2|$; k_1 and k_2 are odd, since u and v are on opposite sides of the bipartition. We choose a subset $P_1 \subset \tilde{C}_1$ by taking every second arc in order to get $|P_1| = \frac{k_1+1}{2}$; similarly we choose $P_2 \subset \tilde{C}_2$ with $|P_2| = \frac{k_2+1}{2}$. By construction either $P_i \subset C_1$ or $P_i \subset C_2$ and either $P_i \cap C_2 = \varnothing$ or $P_i \cap C_1 = \varnothing$ for $i = 1, 2$. An easy calculation shows that $|P_1 \cup P_2| = \frac{k}{2} - 1$; see Figure 4 for an illustration of this case. Then define $\boldsymbol{v}_i := \boldsymbol{I}(P_1 \cup P_2 \cup \{a_i\})$.
- ▷ $u \notin C$, $v \in C$: Here we define $\boldsymbol{v}_i := \boldsymbol{I}(C_1 \setminus \delta(v) \cup \{a_i\})$.
- ▷ $u \in C$, $v \notin C$: Define $\boldsymbol{v}_i := \boldsymbol{I}(C_1 \setminus \delta(u) \cup \{a_i\})$.
- ▷ $u, v \notin C$: Choose any $a \in C_1$ and define $\boldsymbol{v}_i := \boldsymbol{I}(C_1 \setminus \{a\} \cup \{a_i\})$.

It is easy to check in each case that $\boldsymbol{v}_i \in P_M$ and that $\boldsymbol{v}_i(C) = \frac{k}{2} - 1$.

It can be shown that the m vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ are affinely independent, which concludes the proof. □

The separation problem for the cycle inequalities is discussed in the next section.

5.2. Separating the Cycle Inequalities

Of course, there are exponentially many cycle inequalities (6). Hence we have to deal with the separation problem for these inequalities.

We can assume that we are given $\boldsymbol{x}^* \in [0,1]^A$, which satisfies all matching inequalities (5). We consider the separation problem for each graph H_i in turn, $i = 0, \dots, d-1$. The problem is to find an undirected cycle C in H_i such that

$$\boldsymbol{x}^*(C) > \tfrac{1}{2}|C| - 1$$

or conclude that no such cycle exists. In the next sections we describe two methods to solve this problem in polynomial time.

5.2.1. Undirected Shortest Path with Conservative Weights

A well-known trick to solve the above separation problem is to apply an affine transformation and obtain a shortest cycle problem. The transformation suitable for our needs is $\boldsymbol{x}' = \frac{1}{2}\mathbb{1} - \boldsymbol{x}$, which yields

$$\boldsymbol{x}(C) \leq \tfrac{1}{2}|C| - 1 \qquad \Leftrightarrow \qquad \boldsymbol{x}'(C) \geq 1.$$

The separation problem can now be solved as follows: compute a shortest cycle in H_i with respect to the weights $\frac{1}{2}\mathbb{1} - \boldsymbol{x}^*$. If its weight is at most 1, this cycle yields a violated cycle inequality, otherwise no such cycle exists.

However, the weights can be negative and we have to rule out negative cycles in order to apply polynomial time methods from the literature; that is, we want the weights to be *conservative.*

Lemma 5.5. There exists no cycle of negative weight in H_i with respect to $\frac{1}{2}\mathbb{1} - \boldsymbol{x}^*$, for $0 \leq i \leq d-1$.

Proof. Let $C = (a_1, \dots, a_k)$ be a cycle in H_i and let $F_1, \dots, F_k$ be the faces that are visited by C. Recall that $\boldsymbol{x}^*$ satisfies the matching inequalities. We obtain

$$\sum_{j=1}^{k} \sum_{a \in \delta(F_i) \cap C} x_a^* \;=\; 2 \sum_{a \in C} x_a^* \;=\; 2\,\boldsymbol{x}^*(C), \tag{9}$$

since each edge weight is counted twice in the first term. Applying the matching inequalities (5) on the left-hand side yields that $\boldsymbol{x}^*(C) \leq \frac{1}{2}k = \frac{1}{2}|C|$. Hence, the weight of C with respect to $\frac{1}{2}\mathbb{1} - \boldsymbol{x}^*$ can be bounded as follows:

$$\sum_{a \in C} \left(\tfrac{1}{2} - x_a^*\right) = \tfrac{1}{2}|C| - \boldsymbol{x}^*(C) \geq 0,$$

which proves the lemma. □

We have now reduced the separation problem to finding a shortest cycle in a weighted undirected graph $G = (V, E)$ without negative cycles.

By using T-join techniques, one can compute a shortest path in an undirected graph without negative cycles in $O(n_i(m_i + n_i \log n_i))$ time, where in this formula $n_i = |\mathcal{F}^i|$ and $m_i = |A_i|$; see Schrijver [29, Chapter 29]. It follows that a shortest cycle can be computed in $O(m_i n_i(m_i + n_i \log n_i))$ time.

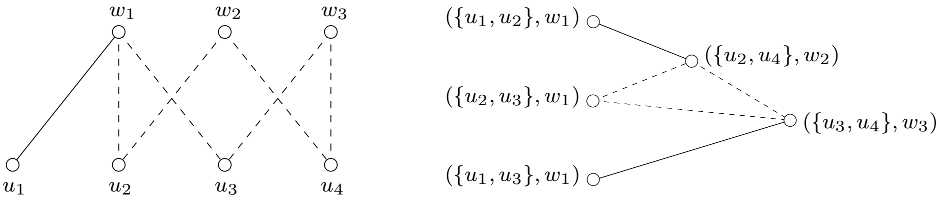


Figure 5: Example of the construction in Section 5.2.2. *Left*: original graph G. *Right:* constructed graph G'. The 6-cycle on the left corresponds to the 3-cycle on the right (both shown with dashed lines).

Since $|A_i| \leq (i+2)n_i$, this leads to an $O\big((d+1)^2n^3 + (d+1)n^3 \log n\big)$ overall algorithm, where $n := |\mathcal{F}|$ is the number of faces and d is the dimension of the complex.

5.2.2. Transforming the Graph

Another method for the separation problem of cycle inequalities, which is easier to implement, works as follows.

Let $G = (U \mathbin{\dot\cup} W, E)$ be a bipartite graph, e.g., $G = H_i$ (with $i \in \{0,\dots,d-1\}$), the ith level of the Hasse diagram. Let $\ell : E \to \mathbb{R}_{\geq 0}$ be a length function for the edges of G. In the following we write $\ell(u,v) = \ell(v,u)$ for the length $\ell(\{u,v\})$.

We construct a graph $G' = (V', E')$ and lengths $\ell' : E' \to \mathbb{R}_{\geq 0}$ as follows; see Figure 5 for an example. The set of nodes of G' is

$$\big\{(\{u,u'\},w) \;:\; u,u' \in U,\; u \neq u',\; w \in W,\; \{u,w\} \in E,\; \{u',w\} \in E\big\}.$$

Hence, G' has a node for each path with two edges in G. There is an edge between two nodes $(\{u_1,u_1'\},w_1)$ and $(\{u_2,u_2'\},w_2)$ if

$$|\{u_1,u_1'\} \cap \{u_2,u_2'\}| = 1 \quad \text{and} \quad w_1 \neq w_2.$$

The length of such an edge e' is defined by

$$\ell'(e') = \tfrac{1}{2}\big(\ell(u_1,w_1) + \ell(u_1',w_1) + \ell(u_2,w_2) + \ell(u_2',w_2)\big).$$

Hence, G' contains an edge for each path with four edges in G and its length is the length of this path divided by 2. We now consider the relation of cycles in G and G'.

Lemma 5.6. $C = (u_0,w_0,u_1,w_1,\dots,w_{k-1},u_1)$ is a cycle in G with $k > 1$ of length $\ell(C)$ if and only if

$$C' = \big((\{u_0,u_1\},w_0),(\{u_1,u_2\},w_1),\dots,(\{u_{k-1},u_1\},w_{k-1}),(\{u_0,u_1\},w_0)\big)$$

is a cycle in G' with $\ell'(C') = \ell(C)$.

We omit the straightforward proof.

The previous lemma does not cover cycles in G of length four. These do not occur for the case of $G = H_i$, since H_i is a level in the Hasse diagram of a *simplicial* complex. Moreover, cycles of length four can readily be detected in the construction of G' and handled accordingly (there is only a polynomial number of them).

To solve our separation problem, let $G = H_i$, $i \in \{0, \dots, d-1\}$, and $\ell(e) = x_e^*$ for $e \in G$. Then we have $\ell'(e') \in [0,1]$ for each $e' \in E'$, because of the matching inequalities. We now set $\tilde{\ell}(e') = 1 - \ell'(e')$ for $e' \in G'$ and hence $\tilde{\ell}(e') \in [0,1]$. Let C be a cycle in G with at least six edges and C' be the corresponding cycle in G'. Note that $|C'| = \frac{1}{2}|C|$. We then have the following:

$$\begin{aligned}
&\tilde{\ell}(C') = \sum_{e' \in C'} \tilde{\ell}(e') = \sum_{e' \in C'} (1 - \ell'(e')) < 1 \\
\Leftrightarrow \quad & \sum_{e' \in C'} \ell'(e') > |C'| - 1 \\
\Leftrightarrow \quad & \ell'(C') > |C'| - 1 \\
\Leftrightarrow \quad & \ell(C) > \tfrac{1}{2}|C| - 1 \qquad\qquad \text{(by Lemma 5.6).}
\end{aligned}$$

Hence, C violates the cycle inequality (6) if and only if $\tilde{\ell}(C') < 1$. Since $\tilde{\ell}(e') \geq 0$, we can use the Floyd-Warshall algorithm to solve the separation problem in time $O\big(|V'|^3\big)$; see Korte and Vygen [22].

If $G = H_i$ and W is the part arising from the higher dimensional faces, we have $|V'| = \binom{i+2}{2}|W| = \binom{i+2}{2} f_{i+1}$. This leads to an $O\big((d+1)^6 n^3\big)$ algorithm for separating cycle inequalities, which is roughly as fast as the method discussed in Section 5.2.1, but much easier to implement.

6. Computational Results

In this section we report on computational experience with a branch-and-cut algorithm along the lines of Section 5. The C++ implementation uses the framework SCIP (Solving Constraint Integer Programs) by Achterberg, see [1]. It furthermore builds on `polymake`; see [11, 12]. As an LP solver we used CPLEX 9.0.

As the basis of our implementation we take the formulation of MaxMM in Section 5. Matching inequalities (5) and Betti inequalities (8) (together with variable bounds) form the initial LP. The computation of the simplicial homology from which the Betti numbers are computed is very fast, because the examples are small; its running time is not included in the following. Cycle inequalities (6) are separated as described in Section 5.2.2. Additionally, Gomory cuts are added. As a branching rule we use *reliability branching* implemented in SCIP, a variable branching rule introduced by Achterberg, Koch, and Martin [2].

We implemented the following primal heuristic. First a simple greedy algorithm is run: We start with the empty matching $M = \varnothing$. We add arcs of the Hasse diagram to M in the order of decreasing value of the current LP solution as long as M stays an acyclic matching (which can easily be tested). Then the outcome is iteratively improved by a method described in Forman [8]: One searches for a unique path between two critical faces in $H(M)$. Such a path is alternating with respect to M. Then M can be augmented along the path (the new matching is the symmetric difference

Table 1: Computational results of the branch-and-cut algorithm with separating cycle inequalities and Gomory cuts.

name	n	m	d	nodes	depth	time	β	c
solid_2_torus	24	42	2	1	0	0.00	2	2
simon2	31	60	2	1	0	0.00	1	1
projective ($\mathbb{RP}_2$)	31	60	2	1	0	0.01	3	3
bjorner	32	63	2	1	0	0.05	2	2
nonextend	39	77	2	6	5	0.16	1	1
simon	41	82	2	1	0	0.18	1	1
dunce	49	99	2	385	10	2.62	1	3
c-ns3	63	128	2	349	10	3.47	1	3
c-ns	75	152	2	28	10	1.95	1	3
c-ns2	79	159	2	14	7	1.11	1	1
ziegler	119	310	3	1	0	0.01	1	1
gruenbaum	167	434	3	1	0	25.24	1	1
lockeberg	216	600	3	1	0	36.25	2	2
rudin	215	578	3	77	30	103.78	1	1
mani-walkup-D	392	1112	3	111	23	512.81	2	2
mani-walkup-C	464	1312	3	135	83	1658.02	2	2
MNSB	103	267	3	12	10	73.39	1	1
MNSS	250	698	3	292	110	750.36	2	2
CP2	255	864	4	230	80	558.14	3	3

of M and the path). As is easily seen, this generates an acyclic matching, because the path is unique. This heuristic turns out to be extremely successful; see below.

We tested the implementation on a set of simplicial complexes collected by Hachimori; see [15] for more details. This test set was also used by Lewiner et al. [24]. Additionally, we considered the following complexes: `CP2` (complex projective plane), `CP2+CP2` (connected sum of `CP2` with itself), `MNSB` (vertex minimal nonshellable ball), and `MNSS` (nonshellable sphere with the fewest number of vertices known). The last two examples are due to Lutz [25, 26].

All computational experiments were run on a 3 GHz Pentium machine running Linux. In the tables of computational results, n denotes the number of faces, m the number of arcs in the Hasse diagram (= number of variables), d the dimension, *nodes* the number of nodes in the branch-and-bound tree, *depth* the maximal depth in the tree, *time* the computation time in seconds, β the lower bound obtained by adding all Betti inequalities (8), and c the number of critical faces in the optimal solution.

Our implementation could not solve the larger problems of Hachimori's collection in reasonable time: `bing`, `knot`, `poincare`, `nonpl_sphere`, and `nc_sphere`. In fact, for `poincare` we ran our code in different settings, each for about a week – without success.

Table 1 shows the results of a computation where we separate cycle inequalities and Gomory cuts and run the heuristic every 10th level. At most seven separation rounds of cycle inequalities were performed at a node. We do not report results on the problems by Moriyama and Takeuchi in

Table 2: Computational results of the branch-and-cut algorithm without separation.

name	n	m	d	nodes	depth	time	β	c
solid_2_torus	24	42	2	1	0	0.00	2	2
simon2	31	60	2	1	0	0.01	1	1
projective ($\mathbb{RP}_2$)	31	60	2	1	0	0.00	3	3
bjorner	32	63	2	1	0	0.01	2	2
nonextend	39	77	2	3	2	0.02	1	1
simon	41	82	2	4	3	0.02	1	1
dunce	49	99	2	168367	42	145.60	1	3
c-ns3	63	128	2	3665581	53	3940.40	1	3
c-ns	75	152	2	16625713	58	19359.69	1	3
c-ns2	79	159	2	4	3	0.03	1	1
ziegler	119	310	3	1	0	0.01	1	1
gruenbaum	167	434	3	21	20	0.68	1	1
lockeberg	216	600	3	1	0	0.05	2	2
rudin	215	578	3	81	80	3.18	1	1
mani-walkup-D	392	1112	3	107	100	2.00	2	2
mani-walkup-C	464	1312	3	1498	456	30.54	2	2
MNSB	103	267	3	1	0	0.01	1	1
MNSS	250	698	3	163	126	4.63	2	2
CP2	255	864	4	198	190	4.77	3	3
CP2+CP2	460	1592	4	5178	534	110.21	4	4

Hachimori's collection – they all could be solved within a second. The version with cut separation could not solve `CP2+CP2` within 90 minutes.

For most problems the bound obtained by adding Betti inequalities (8), as indicated in column "β", is tight. This means that the algorithm is done once an optimal solution is found. This usually happens very fast and shows that the heuristic is efficient. In fact, there are only three problems for which the bound is not tight and could be solved by our algorithm (`dunce`, `c-ns`, and `c-ns3`). These three problems are solved easily by the version with cut separation. In our problem set there exists no hard but still solvable problem with a "Betti bound" which is not sharp. We therefore cannot estimate the limits of our implementation for these cases (`poincare` is the next larger problem of this kind with 1112 variables, but we could not solve it).

The tractability of problems with a tight "Betti bound" is supported by the results obtained by running the implementation without any separation; see Table 2. Only integer solutions are checked whether they are acyclic and the heuristic is run every 10th level. This essentially is a test of the performance of the primal heuristic. Indeed, all problems with tight "Betti bound" were solved within a few seconds (`CP2+CP2` and `mani-walkup-C` being the exception, but could be solved within two minutes). The results for the problems `c-ns`, `c-ns3`, and `dunce` show that the cycle inequalities and Gomory cuts are very effective in reducing the number of nodes in the tree and the computing time for problems where the "Betti bound" is not sharp.

Summarizing, we can say that our implementation can solve large instances with up to about 1500 variables if the bounds from the Betti numbers are tight and small instances with up to about 150 variables if the bounds are not tight. In all the instances computed so far, the topology of the spaces involved was known. In the future, we plan to apply our techniques to other cases.

Acknowledgments

We are indebted to Tobias Achterberg for his support of the implementation. We also thank both referees for their helpful comments.

References

[1] T. Achterberg, *SCIP – a framework to integrate constraint and mixed integer programming.* ZIB-Report 04-19, 2004.

[2] T. Achterberg, T. Koch, and A. Martin, *Branching rules revisited*, Oper. Res. Lett. **33**, no. 1 (2005), pp. 42–54.

[3] E. Babson, A. Björner, S. Linusson, J. Shareshian, and V. Welker, *Complexes of not i-connected graphs*, Topology **38**, no. 2 (1999), pp. 271–299.

[4] E. Batzies and V. Welker, *Discrete Morse theory for cellular resolutions*, J. Reine Angew. Math. **543** (2002), pp. 147–168.

[5] M. K. Chari, *On discrete Morse functions and combinatorial decompositions*, Discrete Math. **217**, no. 1–3 (2000), pp. 101–113.

[6] M. K. Chari and M. Joswig, *Complexes of discrete Morse functions*, Discrete Math. **302** (2005), pp. 39–51.

[7] Ö. Eğecioğlu and T. F. Gonzalez, *A computationally intractable problem on simplicial complexes*, Comut. Geom. **6** (1996), pp. 85–98.

[8] R. Forman, *Morse theory for cell-complexes*, Advances in Math. **134** (1998), pp. 90–145.

[9] R. Forman, *Morse theory and evasiveness*, Combinatorica **20**, no. 4 (2000), pp. 489–504.

[10] R. Forman, *A user's guide to discrete Morse theory*, Sém. Lothar. Combin. **48** (2002), pp. Art. B48c, 35 pp.

[11] E. Gawrilow and M. Joswig, `polymake`: *a framework for analyzing convex polytopes*, in Polytopes – Combinatorics and Computation, G. Kalai and G. M. Ziegler, eds., DMV Seminar 29, Birkhäuser, Basel, 2000, pp. 43–74.

[12] E. Gawrilow and M. Joswig, `polymake`: *Version 2.1.0.* http://www.math.tu-berlin.de/polymake, 2004. With contributions by T. Schröder and N. Witte.

[13] M. Grötschel, M. Jünger, and G. Reinelt, *A cutting plane algorithm for the linear ordering problem*, Oper. Res. **32** (1984), pp. 1195–1220.

[14] M. Grötschel, M. Jünger, and G. Reinelt, *On the acyclic subgraph polytope*, Math. Program. **33** (1985), pp. 28–42.

[15] M. Hachimori, *Simplicial complex library.* http://infoshako.sk.tsukuba.ac.jp/~hachi/math/library/index_eng.html, 2001.

[16] P. L. Hammer, E. L. Johnson, and U. N. Peled, *Facets of regular 0-1 polytopes*, Math. Program. **8** (1975), pp. 179–206.

[17] P. Hersh, *On optimizing discrete Morse functions*, Adv. in Appl. Math. (2005). To appear.

[18] C. S. Iliopoulos, *Worst-case complexity bounds on algorithms for computing the canonical structure of finite Abelian groups and the Hermite and Smith normal forms of an integer matrix*, SIAM J. Comput. **18**, no. 4 (1989), pp. 658–669.

[19] J. JONSSON, *On the topology of simplicial complexes related to 3-connected and Hamiltonian graphs*, J. Combin. Theory Ser. A **104**, no. 1 (2003), pp. 169–199.

[20] M. JOSWIG, *Computing invariants of simplicial manifolds.* Preprint, available at arXiv math.AT/0401176, 2004.

[21] M. JÜNGER, *Polyhedral combinatorics and the acyclic subdigraph problem*, Research and Exposition in Mathematics 7, Heldermann Verlag, Berlin, 1985.

[22] B. KORTE AND J. VYGEN, *Combinatorial optimization. Theory and algorithms*, Algorithms and Combinatorics 21, Springer, Berlin, 2nd ed., 2002.

[23] T. LEWINER, H. LOPES, AND G. TAVARES, *Optimal discrete Morse functions for 2-manifolds*, Comput. Geom. **26**, no. 3 (2003), pp. 221–233.

[24] T. LEWINER, H. LOPES, AND G. TAVARES, *Towards optimality in discrete Morse theory*, Exp. Math. **12**, no. 3 (2003), pp. 271–285.

[25] F. H. LUTZ, *Small examples of non-constructible simplicial balls and spheres*, SIAM J. Discrete Math **18** (2004), pp. 103–109.

[26] F. H. LUTZ, *A vertex-minimal non-shellable simplicial 3-ball with 9 vertices and 18 facets*, Electronic Geometry Models , no. 2003.05.004 (2004). www.eg-models.de.

[27] J. R. MUNKRES, *Elements of Algebraic Topology*, Addison-Wesley, Menlo Park CA, 1984.

[28] G. REINELT, *The linear ordering problem: Algorithms and applications*, Research and Exposition in Mathematics 8, Heldermann Verlag, Berlin, 1985.

[29] A. SCHRIJVER, *Combinatorial Optimization: Polyhedra and Efficiency*, Algorithms and Combinatorics 24, Springer, Berlin Heidelberg, 2003.

[30] J. SHARESHIAN, *Discrete Morse theory for complexes of* 2*-connected graphs*, Topology **40**, no. 4 (2001), pp. 681–701.

[31] D. M. WARME, P. WINTER, AND M. ZACHARIASEN, *Exact solutions to large-scale plane steiner tree problems*, in Proceedings of the 10th annual ACM-SIAM symposium on discrete algorithms, SIAM, Philadelphia, 1999, pp. 979–980.

Paper 5

On the Maximum Feasible Subsystem Problem, IISs and IIS-hypergraphs

Edoardo Amaldi, Leslie E. Trotter, Jr., and Marc E. Pfetsch

On the Maximum Feasible Subsystem Problem, IISs, and IIS-hypergraphs[1]
Math. Program. **95** (2003), no. 3, pp. 533–554

Abstract. We consider the Max FS problem: For a given infeasible linear system $A\boldsymbol{x} \leq \boldsymbol{b}$, determine a feasible subsystem containing as many inequalities as possible. This problem, which is NP-hard and also difficult to approximate, has a number of interesting applications in a wide range of fields. In this paper we examine structural and algorithmic properties of Max FS and of *Irreducible Infeasible Subsystems* (IISs), which are intrinsically related since one must delete at least one constraint from each IIS to attain feasibility. First we provide a new *simplex decomposition* characterization of IISs and prove that finding a smallest cardinality IIS is very difficult to approximate. Then we discuss structural properties of IIS-hypergraphs, i.e., hypergraphs in which each edge corresponds to an IIS, and show that recognizing IIS-hypergraphs subsumes the Steinitz problem for polytopes and hence is NP-hard. Finally we investigate rank facets of the *Feasible Subsystem polytope* whose vertices are incidence vectors of feasible subsystems of a given infeasible system. In particular, using the IIS-hypergraph structural result, we show that only two very specific types of rank inequalities induced by generalized antiwebs (which generalize cliques, odd holes and antiholes to general independence systems) can arise as facets.

[1] Part of this work was done while the first two authors were with the School of OR&IE, Cornell University, USA. A preliminary version appeared in the Proceedings of the 10th IPCO conference [7], held in Graz, Austria, June 1999. This work was supported by NSF grant DMS-9527124.

1. Introduction

We consider the following combinatorial optimization problem related to infeasible linear inequality systems.

Max FS: *Given an infeasible system* $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ *with* $A \in \mathbb{R}^{m\times n}$ *and* $\boldsymbol{b} \in \mathbb{R}^m$*, find a feasible subsystem containing as many inequalities as possible.*

Weighted and unweighted versions of this problem have a number of interesting applications in various fields such as operations research, computational geometry, statistical discriminant analysis and machine learning (see [2, 10, 29, 31, 34, 39, 43] and the references therein).

In linear programming (LP) it arises when the formulation phase yields infeasible models and one wishes to diagnose and resolve infeasibility by deleting as few constraints as possible, which is the complementary version of Max FS [19, 28, 40]. In most situations this cannot be done by inspection and the need for effective algorithmic tools has become more acute with the considerable increase in model size. This type of questions was first addressed in [48]. The reader is referred to [27] for a survey on redundant and implied relations of inequality systems as well as on infeasibility issues. From the computational complexity point of view, Max FS is NP-hard [46] even when the matrix A is totally unimodular and $\boldsymbol{b}$ is integer; it can be approximated within a factor 2 but it does not admit a polynomial-time approximation scheme, unless P = NP [4]. The above-mentioned complementary version, in which the goal is to delete as few inequalities as possible in order to achieve feasibility, is equivalent to solve to optimality but is much harder to approximate than Max FS [5, 8].

Not surprisingly, minimal infeasible subsystems, discussed for instance in Motzkin's thesis [37], play a key role in the study of Max FS. An infeasible subsystem Σ' of Σ is an *Irreducible Infeasible Subsystem* (IIS) if every proper subsystem of Σ' is feasible. In order to help the modeler resolve infeasibility of large linear inequality systems, attention was first devoted to the problem of identifying IISs, with a small and possibly minimum number of inequalities [28]; see [20, 22, 47] for some heuristics and [18] for implementations in commercial solvers such as CPLEX and MINOS. Clearly, in the presence of many overlapping IISs, this does not provide enough information to repair the original system. To achieve feasibility, one must delete at least one inequality from each IIS. If all IISs were known, the complementary version of Max FS could be formulated as the following covering problem [26].

Min IIS Cover: *Given an infeasible system* $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ *with* $A \in \mathbb{R}^{m\times n}$ *and* $\boldsymbol{b} \in \mathbb{R}^m$ *and the set* $\mathcal{C}$ *of all its IISs, minimize* $\sum_{i=1}^{m} y_i$ *subject to* $\sum_{i\in C} y_i \geq 1 \; \forall C \in \mathcal{C}$*,* $y_i \in \{0,1\}$*,* $1 \leq i \leq m$*.*

Note that $|\mathcal{C}|$ can grow exponentially with m and n [17].

An exact algorithm based on a partial cover formulation is proposed in [39, 40] and several heuristics are described in [10, 19, 21, 34]; a collection of infeasible LPs is maintained in the Netlib Repository [38]. In [44, 45] the class of hypergraphs representing the IISs of infeasible systems is studied and it is shown that in some special cases Max FS and Min IIS Cover can be solved in polynomial time in the number of IISs.

Although Max FS with 0-1 variables can be easily shown to admit as a special case the graphical problem of finding a maximum stable set of nodes [4], it has a different structure when the variables are real-valued. Note that, since linear system feasibility can be checked in polynomial time, Max FS structure also differs substantially from that of the maximum satisfiability problem aimed at satisfying a maximum number of disjunctive Boolean clauses. The reader is referred to [25] for the exact definitions of these well-known problems.

Variants of the classical Agmon-Motzkin-Schoenberg relaxation method for solving linear inequality systems have also been investigated and used, among others, in machine learning as well as image and signal processing applications (see e.g. [2, 3, 6, 24]). The implicit enumeration technique described in [29] for optimizing general functions of a set of linear relations can, in principle, also be applied to the special case of Max FS. As to more recent work on problems related to Max FS and IISs let us mention, for instance, Håstad's breakthrough [30] which bridges the approximability gap for Max FS on $GF(p)$, and the problems of determining minimum or minimal witnesses of infeasibility in network flows [1].

In this paper we investigate some structural and algorithmic properties of IISs, of IIS-hypergraphs in which each edge corresponds to an IIS, and of the feasible subsystem polytope defined by the convex hull of incidence vectors of feasible subsystems of a given infeasible system. In Section 2 we provide a new IIS simplex decomposition characterization and prove that finding a smallest cardinality IIS is very difficult to approximate. In Section 3 we first discuss the connection between IIS-hypergraphs and vertex-facet incidences of polyhedra which is needed in the sequel. Based on this connection we also derive that the problem of recognizing IIS-hypergraphs is NP-hard since it subsumes the well-known Steinitz problem for polytopes. In Section 4 we investigate rank facets of the feasible subsystem polytope. In particular, we focus attention on the rank inequalities arising from generalized antiwebs, which generalize cliques, odd holes and antiholes to general independence systems [33]. Finally, the appendix contains the proof of a result stated in Section 3 which completes the discussion but is not required in Section 4.

Below we denote the ith row of the matrix $A \in \mathbb{R}^{m \times n}$ by $\boldsymbol{a}^i \in \mathbb{R}^n$, $1 \leq i \leq m$; for $S \subseteq [m] := \{1, \ldots, m\}$, A_S denotes the $|S| \times n$ matrix consisting of the rows of A indexed by S. By identifying the ith inequality of the system Σ (i.e., $\boldsymbol{a}^i \boldsymbol{x} \leq b_i$) with index i itself, $[m]$ may also refer to Σ.

2. Irreducible Infeasible Subsystems

First we briefly recall the main structural results regarding IISs. For notational simplicity, we use the same A and $\boldsymbol{b}$, with $A \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, to denote either the original system Σ or one of its IISs.

The known characterizations of IISs are based on the following version of the Farkas Lemma:

For any linear inequality system $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$, either $A\boldsymbol{x} \leq \boldsymbol{b}$ is feasible or there exists $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{y} \geq \boldsymbol{0}$, such that $\boldsymbol{y}A = \boldsymbol{0}$ and $\boldsymbol{y}\boldsymbol{b} < 0$, but not both.

Theorem 2.1 (Motzkin [37], Fan [23])**.** The system $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ with $A, \boldsymbol{b}$ as above is an IIS if and only if $\operatorname{rank}(A) = m - 1$ and $\exists\, \boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{y} > 0$, such that $\boldsymbol{y}A = \boldsymbol{0}$ and $\boldsymbol{y}\boldsymbol{b} < 0$.

The rank condition obviously implies that $m \leq n + 1$.

Now let $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ be an infeasible system which is not necessarily an IIS. The following result relates the IISs of Σ to the vertices of a given *alternative polyhedron*. Recall that the *support* of a vector is the set of indices of its nonzero components.

Theorem 2.2 (Gleeson and Ryan [26])**.** Let $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ be an infeasible system with $A, \boldsymbol{b}$ as above. Then the IISs of Σ are in one-to-one correspondence with the vertices of the polyhedron

$$P := \{\boldsymbol{y} \in \mathbb{R}^m \ :\ \boldsymbol{y}A = \boldsymbol{0},\ \boldsymbol{y}\boldsymbol{b} = -1,\ \boldsymbol{y} \geq \boldsymbol{0}\}\,.$$

In particular, the nonzero components of any vertex of P index an IIS.

See [40] for this statement that slightly extends the original result.

Theorem 2.2 can also be stated in terms of rays [40] and elementary vectors [27].

Definition 2.3. An *elementary vector* of a subspace $L \subseteq \mathbb{R}^m$ is a nonzero vector $\boldsymbol{y}$ that has minimal support (when expressed with respect to the standard basis of $\mathbb{R}^m$). In other words, if $\boldsymbol{x} \in L$ and $\operatorname{supp}(\boldsymbol{x}) \subset \operatorname{supp}(\boldsymbol{y})$ then $\boldsymbol{x} = \boldsymbol{0}$, where $\operatorname{supp}(\boldsymbol{y})$ denotes the support of $\boldsymbol{y}$.

Corollary 2.4 (Greenberg [27])**.** Let $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ be an infeasible system with A and $\boldsymbol{b}$ as above. Then $S \subseteq [m]$ corresponds to an IIS of Σ if and only if there exists an elementary vector $\boldsymbol{y}$ in the subspace $L := \{\boldsymbol{y} \in \mathbb{R}^m : \boldsymbol{y}A = \boldsymbol{0}\}$ with $\boldsymbol{y}\boldsymbol{b} < 0$ and $\boldsymbol{y} \geq \boldsymbol{0}$ such that $S = \operatorname{supp}(\boldsymbol{y})$.

The following result establishes an interesting geometric property of the polyhedra obtained by deleting any inequality from an IIS.

Theorem 2.5 (Motzkin [37])**.** Let $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ be an IIS and let $\sigma \in \Sigma$ be an arbitrary inequality of Σ. Then the polyhedron corresponding to $\Sigma \setminus \sigma$, i.e., the subsystem obtained by removal of σ, is an affine convex cone.

2.1. IIS simplex decomposition

We provide here a new geometric characterization of IISs with at least two inequalities, that is $m \geq 2$. For $A \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, let $A^i := A_{[m]\setminus\{i\}}$ and $\boldsymbol{b}^i := \boldsymbol{b}_{[m]\setminus\{i\}}$ denote the $(m-1) \times n$ submatrix and, respectively, the $(m-1)$-dimensional vector obtained by removing the ith row of A and ith component of $\boldsymbol{b}$. The following result strengthens the necessity of Theorem 2.1.

Lemma 2.6. Let $\{A\boldsymbol{x} \leq \boldsymbol{b}\}$ be an IIS. Then A^i has linearly independent rows, for all $1 \leq i \leq m$; i.e., $\operatorname{rank}(A^i) = m - 1$.

Proof. According to Theorem 2.1, there exists a $\boldsymbol{y} > \boldsymbol{0}$ such that $\boldsymbol{y}A = \boldsymbol{0}$ and $\boldsymbol{y}\boldsymbol{b} = -1$ (by scaling $\boldsymbol{y}\boldsymbol{b} < 0$). Suppose some proper subset of rows is linearly dependent; i.e., $\exists \boldsymbol{z}$, such that $\boldsymbol{z}A = \boldsymbol{0}$, $\boldsymbol{z}\boldsymbol{b} \geq 0$ (without loss of generality) and some $z_k = 0$.

If some component $z_i > 0$, consider $(\boldsymbol{y} - \epsilon\boldsymbol{z})A = 0$, $(\boldsymbol{y} - \epsilon\boldsymbol{z})\boldsymbol{b} \leq -1$, where $\epsilon = \min\{y_i/z_i \ : \ 1 \leq i \leq m, \ z_i > 0\} > 0$ (and $\boldsymbol{y}$ is as above). Then $\boldsymbol{y} - \epsilon\boldsymbol{z} \geq 0$, at least one additional component of $\boldsymbol{y} - \epsilon\boldsymbol{z}$ is 0, and the Farkas Lemma contradicts minimality of the system ($\boldsymbol{y} - \epsilon\boldsymbol{z}$ fulfills the requirements).

If all $z_i \leq 0$, then $-\boldsymbol{z} \geq \boldsymbol{0}$, $-\boldsymbol{z}A = \boldsymbol{0}$ and $-\boldsymbol{z}\boldsymbol{b} \leq 0$; so setting $\boldsymbol{y} = -\boldsymbol{z}$ in the Farkas Lemma leads to a contradiction of minimality, provided $-\boldsymbol{z}\boldsymbol{b} < 0$. If $-\boldsymbol{z}\boldsymbol{b} = 0$, then $(\boldsymbol{y} + \epsilon\boldsymbol{z})A = \boldsymbol{0}$, $(\boldsymbol{y} + \epsilon\boldsymbol{z})\boldsymbol{b} = -1$, with $\epsilon = \min\{y_i/(-z_i) \ : \ 1 \leq i \leq m, \ -z_i > 0\}$ leads to a contradiction as above. □

It is interesting to note that this lemma together with Theorem 2.1 imply that an infeasible system $\{A\boldsymbol{x} \leq \boldsymbol{b}\}$ is an IIS if and only if $\operatorname{rank}(A^i) = m - 1$ for all i, $1 \leq i \leq m$.

We then have the following *simplex decomposition* result for IISs.

Theorem 2.7. *The system $\{A\boldsymbol{x} \leq \boldsymbol{b}\}$ is an IIS if and only if $\{A\boldsymbol{x} = \boldsymbol{b}\}$ is infeasible and $\{\boldsymbol{x} \in \mathbb{R}^n \ : \ A\boldsymbol{x} \geq \boldsymbol{b}\} = L + Q$, where L is the lineality subspace $\{\boldsymbol{x} \in \mathbb{R}^n \ : \ A\boldsymbol{x} = \boldsymbol{0}\}$ and Q is an $(m-1)$-simplex with vertices determined by maximal proper subsystems of $\{A\boldsymbol{x} = \boldsymbol{b}\}$; namely, each vertex of Q is a solution for a subsystem $\{A^i\boldsymbol{x} = \boldsymbol{b}^i\}$, $1 \leq i \leq m$.*

Proof. ($\Rightarrow$) The system $\{A\boldsymbol{x} = \boldsymbol{b}\}$ is obviously infeasible. To see the feasibility of $\{A\boldsymbol{x} \geq \boldsymbol{b}\}$, delete constraint $\boldsymbol{a}^i\boldsymbol{x} \geq b_i$ to get the equality system $\{A^i\boldsymbol{x} = \boldsymbol{b}^i\}$. By Lemma 2.6, this system has a solution, say $\boldsymbol{x}^i$, and we must have $\boldsymbol{a}^i\boldsymbol{x}^i > b_i$, else $\boldsymbol{x}^i$ satisfies $\{A\boldsymbol{x} \leq \boldsymbol{b}\}$. Applying the polyhedral resolution theorem, $P := \{\boldsymbol{x} \in \mathbb{R}^n \ : \ A\boldsymbol{x} \geq \boldsymbol{b}\} \neq \emptyset$ can be written as $P = K + Q$, where $K = \{\boldsymbol{x} \in \mathbb{R}^n : A\boldsymbol{x} \geq \boldsymbol{0}\}$ is its recession cone and $Q \subseteq P$ is a polytope generated by representatives of its minimal nonempty faces.

If $\boldsymbol{x}$ satisfies $A\boldsymbol{x} \geq \boldsymbol{0}$ and $\boldsymbol{a}^i\boldsymbol{x} > 0$ for row $\boldsymbol{a}^i$ then $\boldsymbol{x}^i - \epsilon\boldsymbol{x}$ satisfies $A(\boldsymbol{x}^i - \epsilon\boldsymbol{x}) \leq \boldsymbol{b}$ for sufficiently large $\epsilon > 0$ and the original system $\{A\boldsymbol{x} \leq \boldsymbol{b}\}$ would be feasible. Therefore we must have that each $\boldsymbol{a}^i\boldsymbol{x} = 0$ for $1 \leq i \leq m$, $\boldsymbol{x} \in K$ and we get that in fact $K = L := \{\boldsymbol{x} \in \mathbb{R}^n \ : \ A\boldsymbol{x} = \boldsymbol{0}\}$.

For Q, minimal nonempty faces of P are given by changing a maximal set of inequalities into equalities (all but one relation). Thus the vectors $\boldsymbol{x}^i$ obtained by solving $\{A^i\boldsymbol{x} = \boldsymbol{b}^i\}$ determine Q; i.e., $Q = \operatorname{conv}\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m\}$. For $A \in \mathbb{R}^{m \times n}$, Q is the $(m-1)$-simplex generated by the m points $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m$. To see that the $\boldsymbol{x}^i$ generate an $(m-1)$-simplex, we must only show that they are affinely independent. But if $\boldsymbol{x}^i$ is affinely dependent on the other $\boldsymbol{x}^j$, then $\boldsymbol{x}^i = \sum_{j \neq i} \lambda_j \boldsymbol{x}^j$ with $\sum_{j \neq i} \lambda_j = 1$. Thus we have $\boldsymbol{a}^i\boldsymbol{x}^i > b_i$, but also $\boldsymbol{a}^i\boldsymbol{x}^i = \boldsymbol{a}^i(\sum_{j \neq i} \lambda_j \boldsymbol{x}^j) = \sum_{j \neq i} \lambda_j(\boldsymbol{a}^i\boldsymbol{x}^j) = \sum_{j \neq i} \lambda_j b_i = b_i$, which is a contradiction.

($\Leftarrow$) If the system $\{A\boldsymbol{x} \leq \boldsymbol{b}\}$ is infeasible, then the minimality is obvious, because the simplex conditions on Q imply that every proper subsystem has an equality solution.

To show that $\{A\boldsymbol{x} \leq \boldsymbol{b}\}$ is infeasible, assume for the sake of contradiction that $\hat{\boldsymbol{x}} \in \{\boldsymbol{x} \in \mathbb{R}^n \ : \ A\boldsymbol{x} \leq \boldsymbol{b}\} \neq \emptyset$ and $\hat{\boldsymbol{x}}$ satisfies a maximal number of these relations at equality. Since $A\boldsymbol{x} = \boldsymbol{b}$ is assumed to be infeasible, we have $A\hat{\boldsymbol{x}} \neq \boldsymbol{b}$, i.e., there exists $i \in [m]$ with $\boldsymbol{a}^i\hat{\boldsymbol{x}} < b_i$. Let $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m$ be the vertices of Q, where $\boldsymbol{x}^i$ is a solution of $\{A^i\boldsymbol{x} = \boldsymbol{b}^i\}$ for $i = 1, \ldots, m$. Similarly,

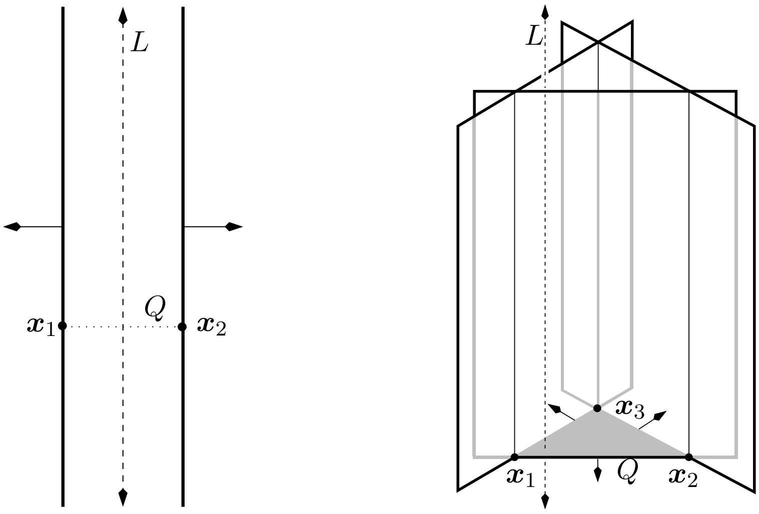


Figure 1: Illustrations of Theorem 2.7 in dimensions $n = 2$ and $n = 3$. The IISs corresponding to $A\boldsymbol{x} \leq \boldsymbol{b}$ are indicated by the halfspaces with arrows pointing inward. If these are turned around the resulting polyhedron can be written as the sum of a simplex Q (indicated by the dotted segment and grey area, respectively) and a lineality space L (indicated by the dashed lines).

the above assumption together with the fact that $Q \subseteq \{\boldsymbol{x} \, : \, A\boldsymbol{x} \geq \boldsymbol{b}\}$ implies that $\boldsymbol{a}^i\boldsymbol{x}^i > b_i$. Thus we can take $\lambda = (\boldsymbol{a}^i\boldsymbol{x}^i - b_i)/(\boldsymbol{a}^i\boldsymbol{x}^i - \boldsymbol{a}^i\hat{\boldsymbol{x}})$ and have $0 < \lambda < 1$, so that $\boldsymbol{a}^i(\lambda\hat{\boldsymbol{x}} + (1-\lambda)\boldsymbol{x}^i) = b_i$. But then at $\lambda\hat{\boldsymbol{x}} + (1-\lambda)\boldsymbol{x}^i$ more relations of $\{A\boldsymbol{x} \leq \boldsymbol{b}\}$ hold at equality than at $\hat{\boldsymbol{x}}$, contradicting the choice of $\hat{\boldsymbol{x}}$. □

According to the above proof, we can take among all possible solutions $\boldsymbol{x}^i$ of the corresponding subsystems $\{A^i\boldsymbol{x} = \boldsymbol{b}^i\}$, for $1 \leq i \leq m$, the representatives of the minimal nonempty faces of $\{A\boldsymbol{x} \leq \boldsymbol{b}\}$ that lie in the orthogonal linear subspace $L^\perp$; i.e., $Q \subset L^\perp$. By Lemma 2.6, we know that $\{\boldsymbol{x} \in \mathbb{R}^n \, : \, A^i\boldsymbol{x} = \boldsymbol{b}^i\} = \boldsymbol{x}^i + L$, where L is the lineality space of the original linear system $\{A\boldsymbol{x} \geq \boldsymbol{b}\}$. However, any choice of $\boldsymbol{x}^i$ would do (see Figure 1). It is worth noting that Theorem 2.7 handles the following special cases.

- If $m = 1$, then the system $\{A^1\boldsymbol{x} \leq \boldsymbol{b}^1\}$ is empty and hence has a solution. Consider for instance $\{A\boldsymbol{x} \leq \boldsymbol{b}\} = \{\boldsymbol{0}\boldsymbol{x} \leq -1\}$, then $L = \{\boldsymbol{x} \in \mathbb{R}^n \, : \, \boldsymbol{0}\boldsymbol{x} = 0\} = \mathbb{R}^n$ and $\{\boldsymbol{x} \in \mathbb{R}^n \, : \, \boldsymbol{0}\boldsymbol{x} \geq -1\} = \mathbb{R}^n + \{0\} = L + Q = L$.
- If $m = n+1$, then A has $n+1$ rows. Assuming A to be of full column rank, $L = \{\boldsymbol{x} \in \mathbb{R}^n \, : \, A\boldsymbol{x} = 0\} = \{0\}$, $Q = \operatorname{conv}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n+1}\}$ is an n-simplex and $\{\boldsymbol{x} \in \mathbb{R}^n \, : \, A\boldsymbol{x} \geq \boldsymbol{b}\} = \{0\} + Q$.

2.2. Minimum cardinality IISs

We now consider the complexity status of the following problem for which heuristics have been proposed in [20, 22, 39, 40].

Min IIS: *Given an infeasible system* $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ *as above, find a minimum cardinality IIS.*

To settle the issue left open in [20, 22, 28, 40], we prove that Min IIS is not only NP-hard to solve optimally but also hard to approximate. Note that, where DTIME($T(m)$) denotes the class of problems solvable in deterministic time $T(m)$, the assumption NP $\not\subseteq$ DTIME($m^{\text{polylog}(m)}$) is stronger than NP $\neq$ P, but it is also believed to be extremely likely. Since polylog(m) denotes any polynomial in $\log(m)$, the assumption amounts to stating that all problems in NP cannot be solved in quasi-polynomial time. Results that hold under such an assumption are often referred to as *almost NP-hard*.

Theorem 2.8. Assuming P $\neq$ NP, no polynomial-time algorithm is guaranteed to yield an IIS whose cardinality is at most c times larger than the minimum one, for any constant $c \geq 1$. Assuming NP $\not\subseteq$ DTIME($m^{\text{polylog}(m)}$), Min IIS cannot be approximated within a factor $2^{\log^{1-\varepsilon}(m)}$, for any $\varepsilon > 0$, where m is the number of inequalities.

Proof. We proceed by reduction from the following problem: Given a feasible linear system $D\boldsymbol{z} = \boldsymbol{d}$, with $D \in \mathbb{R}^{m' \times n'}$ and $\boldsymbol{d} \in \mathbb{R}^{m'}$, find a solution $\boldsymbol{z}$ satisfying all equations with as few nonzero components as possible. In [5] this problem is proved to be (almost) NP-hard to approximate within the same type of factors, but with m replaced by the number of variables n. Note that the above nonconstant factor grows faster than any polylogarithmic function, but slower than any polynomial function.

For each instance of the latter problem which has an optimal solution containing s nonzero components, we construct a particular instance of Min IIS with a minimum cardinality IIS containing $s+1$ inequalities. Given any instance (D, $\boldsymbol{d}$), consider the system

$$\begin{bmatrix} D & -D & -\boldsymbol{d} \end{bmatrix} \begin{pmatrix} \boldsymbol{z}^+ \\ \boldsymbol{z}^- \\ z_0 \end{pmatrix} = \boldsymbol{0}, \ \begin{bmatrix} \boldsymbol{0}^{\mathrm{T}} & \boldsymbol{0}^{\mathrm{T}} & -1 \end{bmatrix} \begin{pmatrix} \boldsymbol{z}^+ \\ \boldsymbol{z}^- \\ z_0 \end{pmatrix} < 0, \ \boldsymbol{z}^+, \boldsymbol{z}^- \geq \boldsymbol{0}, \ z_0 \geq 0. \quad (1)$$

Since the strict inequality implies $z_0 > 0$, the system $D\boldsymbol{z} = \boldsymbol{d}$ has a solution with s nonzero components if and only if (1) has one with $s+1$ nonzero components. Now, applying Corollary 2.4, (1) has such a solution if and only if the system

$$\begin{pmatrix} D^{\mathrm{T}} \\ -D^{\mathrm{T}} \\ -\boldsymbol{d}^{\mathrm{T}} \end{pmatrix} \boldsymbol{x} \leq \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ -1 \end{pmatrix} \quad (2)$$

has an IIS of cardinality $s+1$. Since (2) is the alternative system of (1), the Farkas Lemma implies that exactly one of these is feasible; as (1) is feasible, (2) must be infeasible. Thus (2) is a particular instance of Min IIS with $m = 2n' + 1$ inequalities in $n = m'$ variables.

Given that the polynomial-time reduction preserves the objective function modulo an additive unit constant, we obtain the same type of nonapproximability factors for Min IIS. □

Note that for the similar (but not directly related) problem of determining minimum witnesses of infeasibility in network flows, NP-hardness is established in [1].

3. IIS-hypergraphs

Although in the previous section the focus was on single IISs, we have seen in the introduction that the complementary version of Max FS, in which one aims at minimizing the number of inequalities that must be deleted to make a given infeasible system feasible, can be viewed as the problem of covering all its IISs with a minimum number of inequalities. Assuming the IISs are known, the entire combinatorial structure of a Max FS instance can thus be represented by an appropriate hypergraph containing one node per inequality and one edge for each IIS.

Let $H = (V, \mathcal{E})$ be a finite hypergraph with node set V and edge set $\mathcal{E} \subseteq 2^V$. All hypergraphs in this paper will be finite. H is called a *clutter* hypergraph, if no set of $\mathcal{E}$ contains any other set of $\mathcal{E}$, i.e., $\mathcal{E}$ is a *clutter*.

A hypergraph $H = (V, \mathcal{E})$ is *isomorphic* to a hypergraph $H' = (V', \mathcal{E}')$ if there exists a bijection $\pi : V \to V'$ and a bijection $\tau : \mathcal{E} \to \mathcal{E}'$ such that

$$\tau(E) = \{\pi(v) \ : \ v \in E\} \quad \text{for all } E \in \mathcal{E}.$$

This relation is denoted by $H \cong H'$.

In this section let K denote either the field $\mathbb{Q}$, $\mathbb{A}$, or $\mathbb{R}$. Recall that $\mathbb{A}$ denotes the real algebraic numbers, namely all real numbers that are roots of polynomials with integer coefficients.

Definition 3.1. A hypergraph $H = (V, \mathcal{E})$, with $m := |V|$, is an *IIS-hypergraph* (over K) if there exists an infeasible linear system $\Sigma = \{A\boldsymbol{x} \leq \boldsymbol{b}\}$, with $A \in K^{m \times n}$ (for some n) and $\boldsymbol{b} \in K^m$, such that H is isomorphic to the clutter hypergraph $\mathcal{H}(\Sigma) := ([m], \mathcal{I})$, where the i-th inequality of Σ is identified with i and $\mathcal{I}$ is the set of IISs of Σ.

In the above definition, infeasibility is meant with respect to $\mathbb{R}$.

Investigations of the structure of IIS-hypergraphs (over $\mathbb{R}$) were initiated by [44, 45]. IIS-hypergraphs (with no trivial IISs of cardinality 1) turn out to be *bicolorable*, i.e., their nodes can be partitioned into two subsets so that neither subset contains an edge. Furthermore, IIS-hypergraphs do not share many properties with other known classes of hypergraphs generalizing bipartite graphs. See, for instance, the figure in [45] summarizing how IIS-hypergraphs fit into Berge's hierarchy. Note, however, that there is more structure for IIS-hypergraphs than simply bicolorability, as there will generally exist many different bipartitions into two feasible subsystems [27, 44].

According to hypergraph terminology, Min IIS Cover amounts to finding a minimum cardinality *transversal*, i.e., a subset of nodes having nonempty intersection with every edge. Clearly, the problem can also be viewed as that of finding a maximum *stable* set in IIS-hypergraphs. The special structure of IIS-hypergraphs accounts for the fact a minimum transversal (maximum stable set) can be found in polynomial time in the size of the hypergraph if the corresponding alternative polyhedron is nondegenerate (a subclass of uniform hypergraphs) [45], while the problem is NP-hard even for simple graphs, i.e., for 2-uniform hypergraphs.

In this section we first introduce some terminology and discuss a property of IIS-hypergraphs which is needed in Section 4 to investigate facets of the feasible subsystem polytope. In Subsection 3.2, the same property is used to settle the complexity status of the problem of recognizing whether a given hypergraph is an IIS-hypergraph.

3.1. Connection between IIS-hypergraphs and vertex-facet incidences of polyhedra

Theorem 2.2 provides a connection between the combinatorial structure of the IISs of any given infeasible system (i.e., its IIS-hypergraph) and the vertex-facet incidences of its alternative polyhedron. To formalize this connection, we need the following concepts related to finite hypergraphs.

Let $H = (V, \mathcal{E})$ be a hypergraph. For $E \in \mathcal{E}$ define $\overline{E} := V \setminus E$ to obtain the *complement hypergraph* $\overline{H} := (V, \overline{\mathcal{E}})$, where $\overline{\mathcal{E}} = \{\overline{E} \ : \ E \in \mathcal{E}\}$.

Definition 3.2 (see [11])**.** For each node $v \in V$, $S_v := \{E \in \mathcal{E} \ : \ v \in E\}$ denotes the set of all edges of H which contain v. Then $H^* := \{\mathcal{E}, \mathcal{E}^*\}$, with the edges of H as nodes and $\mathcal{E}^* := \{S_v \ : \ v \in V\}$ as edges, is the *dual hypergraph* of H.

It is easily verified that $H^{**} \cong H$ and $(\overline{E})^* \cong \overline{(E^*)}$ for every edge E of H.

Definition 3.3. Let P be a pointed polyhedron with vertex set V_P. Let $F_1, \dots, F_m$ be the facets of P and let $\mathcal{F}_i := \{v \in V_P \ : \ v \in F_i\}$ be the vertex set of facet F_i, for $1 \leq i \leq m$. Then define $\mathcal{H}(P) := (V_P, \{\mathcal{F}_1, \dots, \mathcal{F}_m\})$. A hypergraph $H = (V, \mathcal{E})$ is a *vertex-facet incidence hypergraph* of P if H is isomorphic to $\mathcal{H}(P)$.

Now we have the following relation:

Lemma 3.4. Let $H = (V, \mathcal{E})$ be a finite IIS-hypergraph (over K) and $\overline{H^*}$ be a clutter hypergraph. Let $\Sigma : A\boldsymbol{x} \leq \boldsymbol{b}$, with $A \in K^{m \times n}$ and $\boldsymbol{b} \in K^m$, be any infeasible system such that $\mathcal{H}(\Sigma) \cong H$. Then $\overline{H^*}$ is a vertex-facet incidence hypergraph of the alternative polyhedron corresponding to Σ.

Proof. Denote by $\mathcal{I}$ the set of IISs of the given Σ. According to Theorem 2.2, the elements of $\mathcal{I}$ are in one-to-one correspondence with the supports of the vertices of the alternative polyhedron

$$P = \{\boldsymbol{y} \in \mathbb{R}^m \ : \ A^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{0}, \ \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} = -1, \ \boldsymbol{y} \geq \boldsymbol{0}\}.$$

Identify V with $[m]$ (the set of inequalities of Σ) so that $\mathcal{E} = \mathcal{I}$. Let $E \in \mathcal{E}$ correspond to an IIS and $\boldsymbol{v}$ be the vertex of P associated with E. The complement of the support of $\boldsymbol{v}$ is $\overline{E}$, and it determines which faces defined by $y_j = 0$, $1 \leq j \leq m$, are satisfied by $\boldsymbol{v}$ with equality, i.e., which of these faces contain $\boldsymbol{v}$. This means that each set $\overline{E} \in \overline{\mathcal{E}}$ gives the set of all faces containing a specific vertex.

By definition, each set in $\overline{\mathcal{E}^*}$ coincides with the vertex set of a face defined by $y_j = 0$ for some $1 \leq j \leq m$. Furthermore, each facet of P must be defined by $y_j = 0$ for some $1 \leq j \leq m$. Since $\overline{\mathcal{E}^*}$ is a clutter, no vertex set of the faces defined by $y_j = 0$ contains another. Altogether this implies that each $y_j = 0$ defines a facet of P. Thus $\overline{H^*}$ is a vertex-facet incidence hypergraph of P. □

It is worth noting that the reverse direction of the previous lemma also holds.

Lemma 3.5. Let $H = (V, \mathcal{E})$ be a vertex-facet incidence hypergraph of a polyhedron P (with a description over K) which is not a cone. Then $\overline{H^*}$ is an IIS-hypergraph (over K).

For completeness, the proof is given in the Appendix.

Note the slight asymmetry between the assumptions of Lemma 3.4 and Lemma 3.5, which is due to the fact that vertex-facet incidences cannot capture all information about the face lattice of unbounded polyhedra (see the comments at the end of Section 3). Restricting attention to hypergraphs H such that $\overline{H^*}$ is a clutter hypergraph yields the following result.

Corollary 3.6. Let $H = (V, \mathcal{E})$ be a finite hypergraph and $\overline{H^*}$ be a clutter hypergraph. Then H is an IIS-hypergraph if and only if $\overline{H^*}$ is a vertex-facet incidence hypergraph of a polyhedron.

Proof. For IIS-hypergraphs, Lemma 3.4 guarantees the "if"-direction. If $\overline{H^*}$ is a vertex-facet incidence hypergraph of a polyhedron P and it is a clutter hypergraph then P cannot be a cone. Thus by Lemma 3.5, H is an IIS-hypergraph. □

3.2. IIS-hypergraph recognition

In this subsection we address the interesting problem of recognizing IIS-hypergraphs.

IIS-hypergraph Recognition problem *over K: Given a hypergraph H, is H an IIS-hypergraph over K?*

The *face lattice* of a polytope P is its set of faces, ordered by inclusion, with the meet defined by intersection. It is well-known (see, e.g., [49]) that the face lattice of P has a rank function $r(\cdot)$, satisfying $r(F) = \dim F + 1$ for every face F, and is both atomic and coatomic. Two polytopes $P \subset \mathbb{R}^p$ and $Q \subset \mathbb{R}^q$ are *affinely equivalent* (denoted by $P \cong Q$) if there exists an affine map $\phi : \mathbb{R}^p \to \mathbb{R}^q$, which establishes a one-to-one correspondence between points in P and Q. Two polytopes with isomorphic face lattices are *combinatorially equivalent*. For the definitions of poset and (face) lattice we again refer the reader to [49].

We prove NP-hardness of IIS-hypergraph recognition by polynomial-time reduction from the following decision problem.

Steinitz problem *over K: Given a lattice $\mathcal{L}$, does there exist a polytope $P \subset \mathbb{R}^d$ (for some d) with vertices in K^d whose face lattice is isomorphic to $\mathcal{L}$?*

If the answer is affirmative, $\mathcal{L}$ is *realizable* as a polytope. In this case d can be assumed to be the dimension of $\mathcal{L}$. See [15] for related material. We need a special lattice construction arising from hypergraphs.

Let $H = (V, \mathcal{E})$ be a hypergraph. Define the poset $\mathcal{L}(H)$ as the set of all intersections of sets in $\mathcal{E}$, ordered by set inclusion. Furthermore, adjoin a maximal element $\hat{1}$. Clearly, $\mathcal{L}(H)$ is bounded and has a meet (defined by intersection); hence it is a lattice. Note that the size of $\mathcal{L}(H)$ can be exponential in the size of H. If H is a vertex-facet incidence hypergraph of a polytope P then $\mathcal{L}$ is isomorphic to the face lattice of P. This follows from the fact that all faces are determined by their vertex sets or by the facets they are contained in.

Conversely, let $\mathcal{L}$ be an arbitrary ranked, atomic, and coatomic lattice. Let V be the set of atoms of $\mathcal{L}$. For each coatom F, let $E_F := \{v \in V : v$ is below F in $\mathcal{L}\}$. Then define the hypergraph $\mathcal{H}(\mathcal{L}) := (V, \{E_F : F$ coatom of $\mathcal{L}\})$. Note that, since $\mathcal{L}$ is atomic, $\mathcal{H}(\mathcal{L})$ is a clutter hypergraph by construction. If $\mathcal{L}$ is the face lattice of a polytope, then $\mathcal{H}(\mathcal{L})$ is a vertex-facet incidence hypergraph.

Theorem 3.7. For $K \in \{\mathbb{Q}, \mathbb{A}, \mathbb{R}\}$, there is a polynomial-time reduction from the Steinitz problem (over K) to the IIS-hypergraph Recognition problem (over K).

Proof. We show that for any instance of the Steinitz problem, given by an arbitrary lattice $\mathcal{L}$, we can construct in polynomial time a special instance of the latter problem, given by a clutter hypergraph H, such that the answer to the first instance is affirmative if and only if the answer to the second instance is affirmative.

If $\mathcal{L}$ is ranked, atomic, and coatomic, take $H = \overline{\mathcal{H}(\mathcal{L})^*}$. Note that these properties of $\mathcal{L}$ can be checked (Test 1) and H can be constructed in polynomial time in the size of $\mathcal{L}$, namely the number of elements. If any of these properties fail, let H be any hypergraph which is not an IIS-hypergraph, e.g., take $H = (\{1,2,3\}, \{\{1,2\}, \{2,3\}, \{1,3\}\})$.

In [32] it is proved that, if H is a vertex-facet incidence hypergraph of a d-dimensional polyhedron P, there exists a number $\widetilde{\chi} = \widetilde{\chi}(H) \in \mathbb{Z}$, namely the *reduced Euler characteristic* of the order complex of $\mathcal{L}(H)$ (see e.g. [12]) such that $\widetilde{\chi} = (-1)^{d-1}$ if P is bounded while $\widetilde{\chi} = 0$ if P is unbounded. Moroever, $\widetilde{\chi}$ can be computed in polynomial time in the size of $\mathcal{L}(H)$. Note that this result implies that no unbounded polyhedron and polytope can have isomorphic vertex-facet incidence hypergraphs.

Since $\widetilde{\chi}(\overline{H^*})$ can be computed in polynomial time in the size of $\mathcal{L}(\overline{H^*})$, which equals the size of $\mathcal{L}$. If $\widetilde{\chi}(\overline{H^*}) = 0$ (Test 2), then replace H by any hypergraph which is not an IIS-hypergraph.

The resulting H is the input to the IIS-hypergraph Recognition problem. Assume that the answer to the IIS-hypergraph Recognition of H is affirmative, i.e., H is an IIS-hypergraph. As noted above, the atomicity of $\mathcal{L}$ implies that $\overline{H^*}$ is a clutter hypergraph. By Lemma 3.4, $\overline{H^*}$ is a vertex-facet incidence hypergraph of some polyhedron P.

First assume that P is a polytope. By construction, $\mathcal{L}$ is isomorphic to $\mathcal{L}(\overline{H^*}) = \mathcal{L}(\mathcal{H}(\mathcal{L}))$. Since P is a polytope, $\mathcal{L}(\overline{H^*})$ is isomorphic to the face lattice of P and hence so is $\mathcal{L}$, i.e., the answer to the Steinitz problem for $\mathcal{L}$ is affirmative.

Now assume P is an unbounded polyhedron. Then $\overline{H^*}$ is a vertex-facet incidence hypergraph of an unbounded polyhedron and, according to the above-mentioned result, we have $\widetilde{\chi}(\overline{H^*}) = 0$. But in this case we replaced the input by an instance which is not an IIS-hypergraph; this is a contradiction.

Conversely assume that the answer to the Steinitz problem for $\mathcal{L}$ is affirmative. Then there exists a polytope P such that $\mathcal{L}$ is isomorphic to the face lattice of P and hence, by construction, $\overline{H^*}$ is a vertex-facet incidence hypergraph of P. Now P is not a cone unless $P = \{0\}$, a case which can be easily identified and discarded. By applying Lemma 3.5 to $\overline{H^*}$, it follows that H is an IIS-hypergraph.

Note that since $\mathcal{L}$ is ranked, atomic, and coatomic, it has necessarily passed Test 1. Furthermore, by the above-mentioned result $\widetilde{\chi}(\overline{H^*}) = \pm 1$, which implies that it also passed Test 2. Thus, the answer to the IIS-hypergraph Recognition question for H is affirmative. □

Given polynomials $f_1, \dots, f_r$, $g_1, \dots, g_s$, $h_1, \dots, h_t \in \mathbb{Z}[x_1, \dots, x_\ell]$, the problem to decide whether the polynomial system $f_1 = 0, \dots, f_r = 0$, $g_1 \geq 0, \dots, g_s \geq 0$, $h_1 > 0, \dots, h_t > 0$ has a solution in $K^\ell = \mathbb{A}^\ell$ is called the *Existential Theory of the Reals* (ETR). ETR is polynomial-time equivalent to the Steinitz problem for 4-polytopes over $\mathbb{A}$, see [42]. All polytopes realizable over $\mathbb{R}$, are realizable over $\mathbb{A}$. Moreover, ETR is polynomial-time equivalent to the Steinitz problem for d-Polytopes with $d+4$ vertices over $\mathbb{A}$ [36]. Since ETR is easily verified to be NP-hard [13], the same is valid for the general Steinitz problem (over $\mathbb{A}$) and for the IIS-hypergraph recognition problem.

According to Theorem 2.7 of [15], for $K = \mathbb{Q}$ or $\mathbb{A}$, deciding whether an arbitrary polynomial $f \in \mathbb{Z}[x_1, \dots, x_\ell]$ has zeros in K^ℓ, where ℓ is a positive integer, is equivalent to solving the Steinitz problem for K. For $K = \mathbb{Q}$, it is not even clear whether the Steinitz problem (and therefore the IIS-hypergraph Recognition) is decidable, since finding roots in $K = \mathbb{Q}$ of a single polynomial $f \in \mathbb{Z}[x_1, \dots, x_\ell]$ is the unsolved rational version of Hilbert's 10th problem. By the quantifier elimination result of Tarski, the problem is decidable for $K = \mathbb{A}$. Note that, unlike $\mathbb{R}$, $\mathbb{A}$ admits a finite representation. For $K = \mathbb{A}$, it is unknown whether the Steinitz problem is in NP. See [14, 35] and references therein for this and related issues.

Finally it is worth noting that to establish the reverse direction of Theorem 3.7 one would need to provide an appropriate input (a lattice) to the Steinitz problem. This task appears to be difficult to achieve because we need to consider the case of unbounded polyhedra. In fact, as shown in [32], it is in general impossible to reconstruct the face lattice of an unbounded polyhedron P given a vertex-facet incidence hypergraph H of P, even when H is a clutter hypergraph.

4. Feasible Subsystem (FS) Polytope

An *independence system* $(E, \mathcal{I})$ is defined by a finite ground set E and a collection of subsets $\mathcal{I} \subseteq 2^E$ such that $I \in \mathcal{I}$ and $J \subset I$ imply $J \in \mathcal{I}$. The subsets of E that (do not) belong to $\mathcal{I}$ are the so-called *independent* (*dependent*) sets. An independence system can be defined by its collection of *independent sets* $\mathcal{I}$ or, equivalently, by the collection $\mathcal{C}$ of all minimal dependent subsets of E; i.e., any dependent subset each of whose proper subsets are independent. To any independence system $(E, \mathcal{I})$ with the collection of *circuits* $\mathcal{C}$, we can associate the polytope

$$P(\mathcal{I}) = \operatorname{conv}\{\boldsymbol{y} \in \{0,1\}^{|E|} \ : \ \boldsymbol{y} \text{ is the incidence vector of an } I \in \mathcal{I}\},$$

which will also be denoted by $P(\mathcal{C})$.

Now consider an infeasible system $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ with no single inequality that is trivially infeasible. Let $[m] = \{1, \ldots, m\}$ be the set of indices of the inequalities in Σ. If $\mathcal{I}$ denotes the set of all feasible subsystems of Σ, $([m], \mathcal{I})$ is clearly an independence system and its set of circuits $\mathcal{C}$ corresponds to the set of all IISs. We denote by $P_{FS}(\Sigma)$ the *Feasible Subsystem polytope*, defined as the convex hull of all the incidence vectors of feasible subsystems.

Before investigating this polytope, let us recall some definitions and facts regarding general independence system polytopes. The *rank function* is defined by $r(S) = \max\{|I| \ : \ I \subseteq S, \ I \in \mathcal{I}\}$ for all $S \subseteq E$. For any $S \subseteq E$, the *rank inequality* for S is $\sum_{e \in S} y_e \leq r(S)$, which is clearly valid for $P(\mathcal{I})$. A subset $S \subseteq E$ is *closed* if $r(S \cup \{t\}) \geq r(S) + 1$ for all $t \in E - S$ and *nonseparable* if $r(S) < r(T) + r(S - T)$ for all $T \subset S$, $T \neq \emptyset$. For any set $S \subseteq E$, S must be closed and nonseparable for the corresponding rank inequality to define a facet of $P(\mathcal{I})$. These conditions generally are only necessary, but sufficient conditions can be stated using the following concept [33]. For $S \subseteq E$, the *critical graph* $G_S(\mathcal{I}) = (S, F)$ is defined as follows: $(e, e') \in F$, for e, $e' \in S$, if and only if there exists an independent set I such that $I \subseteq S$, $|I| = r(S)$ and $e \in I$, $e' \notin I$, $I - e + e' \in \mathcal{I}$. It is shown in [33] that if S is a closed subset of E and the critical graph $G_S(\mathcal{I})$ of $\mathcal{I}$ on S is connected, then the corresponding rank inequality induces a facet of the polytope $P(\mathcal{I})$. (See also the references in [16].)

We now turn to the feasible subsystem polytope. According to well-known facts about independence system polytopes, $P_{FS}(\Sigma)$ is full-dimensional if and only if there are no trivially infeasible inequalities in Σ. Moreover, the inequalities $y_i \geq 0$ are facet defining for all $1 \leq i \leq m$, and it is easy to verify that for each i the inequality $y_i \leq 1$ defines a facet of $P_{FS}(\Sigma)$ if and only if there is no IIS of cardinality 2 that includes the ith inequality of Σ.

4.1. Rank facets arising from IISs

In fact, Parker [39] began an investigation of the polytope associated to the Min IIS Cover problem, considering it as a special case of the general set covering polytope (see also references in [16]). Since there is a simple correspondence between set covering polytopes and the associated independence system polytopes [33], the results in [39] can be translated so that they apply to $P_{FS}(\Sigma)$.

From now on, we assume that all IISs are nontrivial, i.e., they are of cardinality greater or equal to two. Let S be an arbitrary IIS of Σ, with $A_S\boldsymbol{x} \leq \boldsymbol{b}_S$ its associated subsystem. Then the rank inequality

$$\sum_{i \in S} y_i \leq r(S) = |S| - 1$$

is called an *IIS-inequality*. Because the corresponding covering inequality $\sum_{i \in S} y_i \geq 1$ is proved to be facet defining in [39], we have:

Theorem 4.1. Every IIS-inequality defines a (rank) facet of $P_{FS}(\Sigma)$.

We give a geometric proof (based on the above-mentioned sufficient conditions [33] and our IIS simplex decomposition result) in the following, which is simpler than that of [39] and which provides additional insight into the IIS structure.

Proof. It is easy to verify that IIS-inequalities are valid for $P_{FS}(\Sigma)$. Since the critical graph corresponding to any IIS is clearly connected (in fact, a complete graph), we just need to show that the index set of every IIS is closed.
a) First consider the case of maximal IISs defined by subset $S \subseteq E$, i.e., with $|S| = n+1$, where E is the index set of the entire system Σ.

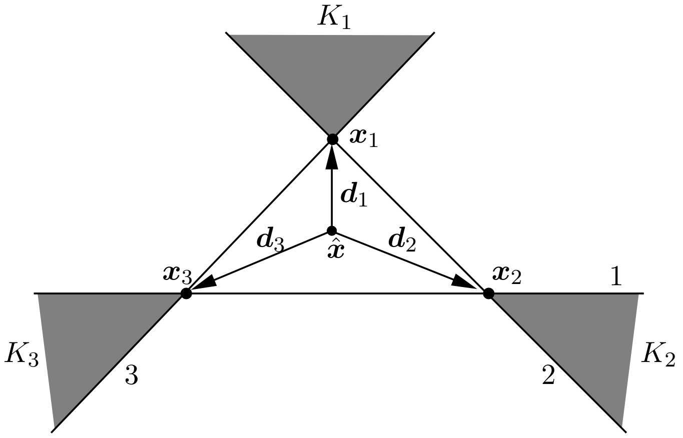


Figure 2: Illustration of the proof of Theorem 4.1.

For each $i \in S$, consider the unique $\boldsymbol{x}^i = A^{-1}_{S\setminus\{i\}}\boldsymbol{b}_{S\setminus\{i\}}$. By the proof of Theorem 2.7, we know that $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{n+1}$ are affinely independent. If we define $\boldsymbol{d}_i := (\boldsymbol{x}^i - \hat{\boldsymbol{x}})$ for all i, $1 \le i \le n+1$, where $\hat{\boldsymbol{x}} := \frac{1}{n+1}\sum_{i=1}^{n+1}\boldsymbol{x}^i$ is the barycenter of the $\boldsymbol{x}^i$'s, then $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{n+1}$ are also affinely independent. Clearly $\sum_{i=1}^{n+1}\boldsymbol{d}_i = \boldsymbol{0}$ and the $\boldsymbol{d}_i$'s generate $\mathbb{R}^n$. Since each $\boldsymbol{x}^i$ satisfies exactly n of the $n+1$ inequalities in $A_S\boldsymbol{x} \le \boldsymbol{b}$ with equality and for the ith one $\boldsymbol{a}^i\boldsymbol{x}^i > b_i$ (otherwise S would be feasible), we have $\hat{\boldsymbol{x}} \in \{\boldsymbol{x} \in \mathbb{R}^n : A_S\boldsymbol{x} \ge \boldsymbol{b}_S\}$. In other words, $\hat{\boldsymbol{x}}$ satisfies the reversed inequalities of the IIS. In fact, $\hat{\boldsymbol{x}}$ is an interior point of the above "reversed" polyhedron.

According to Theorem 2.5, deleting any inequality from an IIS yields a feasible subsystem that defines an affine cone. For maximal IISs, we have $n+1$ affine cones $K_i := \boldsymbol{x}^i + K'_i$, where $K'_i = \{\boldsymbol{x} \in \mathbb{R}^n : A_{S\setminus\{i\}}\boldsymbol{x} \le \boldsymbol{0}\}$ for $1 \le i \le n+1$. Note that the ray generated by $\boldsymbol{d}_i$ passing through $\boldsymbol{x}^i$, i.e., $R_i := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} = \boldsymbol{x}^i + \alpha\boldsymbol{d}_i,\ \alpha \ge 0\}$, is contained in K_i because we have

$$A_{S\setminus\{i\}}(\alpha\boldsymbol{d}_i) = \alpha A_{S\setminus\{i\}}(\boldsymbol{x}^i - \hat{\boldsymbol{x}}) = \alpha(\boldsymbol{b}_{S\setminus\{i\}} - A_{S\setminus\{i\}}\hat{\boldsymbol{x}}) \le \boldsymbol{0},$$

where we used the fact that $A_{S\setminus\{i\}}\hat{\boldsymbol{x}} \ge \boldsymbol{b}_{S\setminus\{i\}}$. To show that the maximal IIS defined by S is closed, we consider an arbitrary inequality $\tilde{\boldsymbol{a}}\boldsymbol{x} \le \tilde{b}$ with $\tilde{\boldsymbol{a}} \ne \boldsymbol{0}$ and verify that $H := \{\boldsymbol{x} \in \mathbb{R}^n : \tilde{\boldsymbol{a}}\boldsymbol{x} \le \tilde{b}\}$ has a nonempty intersection with at least one of the K_i's, $1 \le i \le n+1$. This implies, in particular, that for any inequality index $t \in E - S$ we have $\operatorname{rank}(S \cup \{t\}) = \operatorname{rank}(S) + 1 = n+1$, which means that the IIS under consideration is closed.

Since $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{n+1}$ generate $\mathbb{R}^n$ and $\sum_{i=1}^{n+1} \boldsymbol{d}_i = \boldsymbol{0}$, we have

$$\sum_{i=1}^{n+1} \tilde{\boldsymbol{a}}\boldsymbol{d}_i = \tilde{\boldsymbol{a}}(\sum_{i=1}^{n+1} \boldsymbol{d}_i) = 0$$

and therefore $\tilde{\boldsymbol{a}} \neq \boldsymbol{0}$ implies that we cannot have $\tilde{\boldsymbol{a}}\boldsymbol{d}_i = 0 \; \forall i,\; 1 \leq i \leq n+1$. Thus there exists at least one i, such that $\tilde{\boldsymbol{a}}\boldsymbol{d}_i < 0$. But this implies that $R_i \cap H \neq \emptyset$. In other words, $K_i \cap H \neq \emptyset$ and this proves the theorem for maximal IISs.

b) The result can be easily extended to non-maximal IISs, i.e., with $|S| < n+1$. From Theorem 2.7 we know that $P := \{\boldsymbol{x} \in \mathbb{R}^n \;:\; A_S\boldsymbol{x} \geq \boldsymbol{b}_S\} = L + Q$ with $Q \subseteq L^{\perp}$. Since P is full-dimensional (the barycenter of Q is an interior point), $n = \dim P = \dim L + \dim Q$ and $\dim Q = \operatorname{rank}(A_S) = |S| - 1 < n$ imply that $\dim L \geq 1$.

Two cases can arise:
i) If the above-mentioned $\tilde{\boldsymbol{a}}$ belongs to the linear hull of the rows of A_S denoted by $\operatorname{lin}(\{\boldsymbol{a}^i \;:\; i \in S\}) = L^{\perp}$, then since $\dim L^{\perp} = \dim Q$, we can apply the above result to $L^{\perp}$.
ii) If $\tilde{\boldsymbol{a}} \notin \operatorname{lin}(\{\boldsymbol{a}^i \;:\; i \in S\}) = L^{\perp}$, then the projection of $H^{=} := \{\boldsymbol{x} \in \mathbb{R}^n \;:\; \tilde{\boldsymbol{a}}\boldsymbol{x} = \tilde{b}\}$ onto $L^{\perp}$ yields all of $L^{\perp}$ and therefore $H = \{\boldsymbol{x} \in \mathbb{R}^n \;:\; \tilde{\boldsymbol{a}}\boldsymbol{x} \leq \tilde{b}\}$ must have a nonempty intersection with all the cones corresponding to the maximal consistent subsystems of $\{A_S\boldsymbol{x} \leq \boldsymbol{b}_S\}$. □

It is worth emphasizing that closedness of every IIS makes the feasible subsystem polytope quite special among all independence system polyhedra, since the circuits of a general independence system need not be closed. For example, consider the independent system defined by stable sets of nodes in a simple graph; here the circuits correspond to the edges of the graph and it is clear that these circuits are not necessarily closed (it suffices to consider any K_3 in the graph).

We now turn to the **IIS-inequality Separation problem**, which is defined as follows:

Given an infeasible system Σ *and an arbitrary vector* $\boldsymbol{y} \in \mathbb{R}^m$*, show that* $\boldsymbol{y}$ *satisfies all IIS-inequalities or find at least one violated by* $\boldsymbol{y}$*.*

In view of the trivial valid inequalities, we can assume that $\boldsymbol{y} \in [0,1]^m$. Moreover, we may assume with no loss of generality, that the nonzero components of $\boldsymbol{y}$ correspond to an infeasible subsystem of Σ.

Proposition 4.2. The separation problem for IIS-inequalities is NP-hard.

Proof. We proceed by polynomial-time reduction from the decision version of the Min IIS problem, which is NP-hard according to Theorem 2.8. Given an infeasible system $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ with m inequalities, n variables and a positive integer K with $1 \leq K \leq n+1$, does it have an IIS of cardinality at most K?

Let $(A, \boldsymbol{b})$ and K define an arbitrary instance of the above decision problem. Consider the particular instance of the separation problem given by the same infeasible system together with the vector $\boldsymbol{y}$ such that $y_i = 1 - 1/(K+1)$ for all i, $1 \leq i \leq m$.

Suppose that Σ has an IIS of cardinality at most K which is indexed by the set S. Then the corresponding IIS-inequality $\sum_{i \in S} y_i \leq |S| - 1$ is violated by the vector $\boldsymbol{y}$ because

$$\sum_{i \in S} y_i = \sum_{i \in S} (1 - \frac{1}{K+1}) = |S| - \frac{|S|}{K+1} > |S| - 1,$$

where the strict inequality is implied by $|S| \leq K$. Thus the vector $\boldsymbol{y}$ can be separated from $P_{FS}(\Sigma)$.

Conversely, if there exists an IIS-inequality violated by $\boldsymbol{y}$, then

$$\sum_{i \in S} y_i = |S| - \frac{|S|}{(K+1)} > |S| - 1$$

implies that the cardinality of the IIS defined by S is at most K.

Therefore, the original infeasible system Σ contains an IIS of cardinality at most K if and only if some IIS-inequality is violated by the given vector $\boldsymbol{y}$. □

4.2. Rank facets arising from generalized antiwebs

In [33] the concept of generalized antiwebs, which generalize cliques, odd holes and antiholes to independence systems, is introduced. Necessary and sufficient conditions are also established for the corresponding rank inequalities to define facets of the associated independence system polytope.

Let m, t, q be integers such that $2 \leq q \leq t \leq m$, let $E = \{e_0, \ldots, e_{m-1}\}$ be a finite set, and define for each $i \in M := \{0, \ldots, m-1\}$ the subset $E^i = \{e_i, \ldots, e_{i+t-1}\}$ (where the indices are taken modulo m) formed by t consecutive elements of E. An (m, t, q)-*generalized antiweb* on E is the independence system having the following family of subsets of E as circuits:

$$\mathcal{AW}(m,t,q) = \{C \subseteq E \ : \ C \subseteq E^i \ \text{ for some } \ i \in M, \ |C| = q\}.$$

Define $P(\mathcal{AW}(m,t,q))$ to be the polytope of the independence system defined by $\mathcal{AW}(m,t,q)$ and $\mathcal{AW}(m,t) := \mathcal{AW}(m,t,t)$. Note that the case $t = q = 1$ would correspond to m trivially infeasible inequalities, e.g., $\mathbf{0}\,\boldsymbol{x} \leq -1$.

As mentioned in [33], $\mathcal{AW}(m,t,q)$ corresponds to *generalized cliques* when $m = t$, to *generalized odd holes* when $q = t$ and t does not divide m, and to *generalized antiholes* when $m = qt + 1$.

In this section we determine under which circumstances generalized antiwebs give rise to rank facets of the form $\sum_{i \in S} y_i \leq r(S)$ of $P_{FS}(\Sigma)$. Defining the hypergraph $\mathcal{H}(\mathcal{AW}(m,t,q)) := (E, \mathcal{AW}(m,t,q))$, the first question is: for which values of m, t, and q is $\mathcal{H}(\mathcal{AW}(m,t,q))$ an IIS-hypergraph?

Lemma 4.3. If $\mathcal{H}(\mathcal{AW}(m,t,q))$ is an IIS-hypergraph then $t = q$.

Proof. Suppose that $q < t$ holds, and consider E^1, an arbitrary circuit $C \in \mathcal{AW}(m,t,q)$ with $C \subseteq E^1$, and an arbitrary element $e \in E^1 \backslash C$. By definition of $\mathcal{AW}(m,t,q)$, any cardinality q subset of E^1 is a circuit. This must be true in particular for all subsets containing e and $q-1$ elements of C. But then C cannot be closed because $r(C \cup \{e\}) = r(C)$ and thus we have a contradiction to the fact that all IISs are closed (consequence of Theorem 4.1). □

To provide a characterization of IIS-hypergraphs arising from generalized antiwebs, we need the following result that is proved using topological arguments.

Proposition 4.4 (Joswig, Kaibel, Pfetsch, Ziegler [32])**.** Let $1 < k < m$ be integers. Then $\mathcal{H}(\mathcal{AW}(m,k))$ is a vertex-facet incidence hypergraph of a polyhedron P if and only if P is a simplex or a polygon.

Together with Lemma 3.4 and Lemma 4.3 we obtain:

Proposition 4.5. $\mathcal{H}(\mathcal{AW}(m,t,q))$ is an IIS-hypergraph if and only if $t = q$ and

(1) $t = m$ or
(2) $t = m - 2$.

Proof. Lemma 4.3 implies that necessarily $t = q$. Now assume that $H := \mathcal{H}(\mathcal{AW}(m,t))$ is an IIS-hypergraph. If $t = m$, we have a single IIS of size m. Therefore assume $t < m$.

Since $t < m$, $\overline{H^*}$ is a clutter hypergraph and hence, by Lemma 3.4, $\overline{H^*}$ is a vertex-facet incidence hypergraph of a polyhedron P. We have that $\overline{\mathcal{AW}(m,t)} \cong \mathcal{AW}(m,k)$ with $k := m - t > 0$ and $\mathcal{H}(\mathcal{AW}(m,k))^* \cong \mathcal{H}(\mathcal{AW}(m,k))$. Hence $\mathcal{H}(\mathcal{AW}(m,k))$ is a vertex-facet incidence hypergraph of P. Since $2 \leq t < m$ we have $0 < k < m - 1$. Furthermore $k > 1$ because $\mathcal{H}(\mathcal{AW}(m,1))$ can only be a vertex-facet hypergraph if $m = k = 1$, and this case is excluded by $1 < t < m$.

By Proposition 4.4, P is a polygon; i.e., $k = 2$ $(t = m-2)$. Note that the case of a simplex $(k = m - 1)$ cannot arise. Clearly, examples of infeasible inequality systems exist for all possible values of the above parameters. This proves sufficiency. □

This proposition implies that only two types of generalized antiwebs can arise as induced hypergraph of IIS-hypergraphs. In particular, the only generalized cliques that can occur are those with $t = m$, namely those corresponding to single IISs. For generalized odd holes the only cases that can arise are those with $t = m - 2$. Finally, all generalized antiholes are ruled out since $m = tq + 1 \Leftrightarrow m = (m-2)^2 + 1$, which is never satisfied.

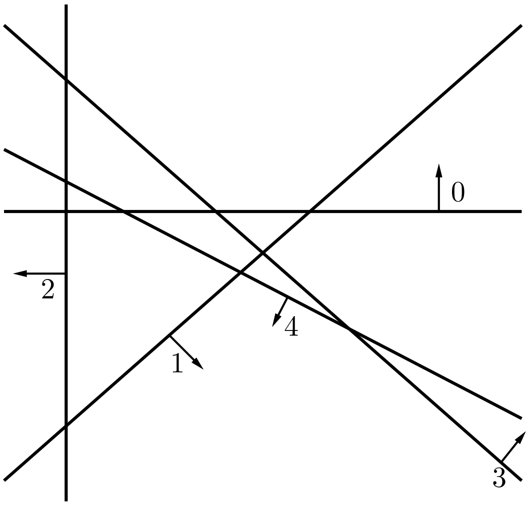
To determine in which cases facets arise from generalized antiwebs, we need the two following results.

Lemma 4.6 (Laurent [33])**.** The valid inequality $\sum_{e \in E} y_e \leq \lfloor m(q-1)/t \rfloor$ (rank inequality) arising from a generalized antiweb defines a facet of the independence system polytope $P(\mathcal{AW}(m,t,q))$ if and only if $t = m$ or t does not divide $m(q-1)$.

Note that the right hand side of the above inequality is the rank of the independence system defined by $\mathcal{AW}(m,t,q)$ (see [33]).

Let $\mathcal{C}$ be the set of circuits of an independence system $\mathcal{I}$ over the ground set $[m]$. For any $S \subseteq [m]$, let $\mathcal{C}_S = \{C \in \mathcal{C} \,:\, C \subseteq S\}$ denote the family of circuits of $\mathcal{I}$ induced on S.

Lemma 4.7 (Laurent [33])**.** The rank inequality $\sum_{e \in S} y_e \leq r(S)$ induces a facet of $P(\mathcal{C})$ if and only if S is closed and it induces a facet of $P(\mathcal{C}_S)$.



$$\begin{array}{c|ccccc|} & e_0 & e_1 & e_2 & e_3 & e_4 \\ \hline E_0 & 1 & 1 & 1 & & \\ E_1 & & 1 & 1 & 1 & \\ E_2 & & & 1 & 1 & 1 \\ E_3 & 1 & & & 1 & 1 \\ E_4 & 1 & 1 & & & 1 \\ \hline \end{array}$$

$$\begin{array}{c|ccccc|} & E_0 & E_1 & E_2 & E_3 & E_4 \\ \hline S_0 & 1 & & & 1 & 1 \\ S_1 & 1 & 1 & & & 1 \\ S_2 & 1 & 1 & 1 & & \\ S_3 & & 1 & 1 & 1 & \\ S_4 & & & 1 & 1 & 1 \\ \hline \end{array}$$

Figure 3: Left: an infeasible linear inequality system, whose IISs $\{0,1,2\}$, $\{1,2,3\}$, $\{2,3,4\}$, $\{3,4,0\}$, and $\{4,0,1\}$ form a generalized antiweb $\mathcal{AW}(5,3)$. **Top right**: incidence matrix of $\mathcal{H}(\mathcal{AW}(5,3))$ according to the notation of Section 3. **Bottom right**: incidence matrix of the dual hypergraph $\mathcal{H}(\mathcal{AW}(5,3))^*$. This matrix is the transpose of the above matrix. Clearly, the incidence matrix of the complement hypergraph is a vertex-facet incidence matrix of a polygon.

Altogether we obtain the following characterization of the rank facets of $P_{FS}(\Sigma)$ that can be induced by generalized antiwebs.

Theorem 4.8. Let Σ be an infeasible inequality system with m inequalities and $\mathcal{C}$ be the IISs of Σ. Let $S \subseteq [m]$ and assume $\mathcal{C}_S = \mathcal{AW}(|S|,t)$ for some $2 \leq t \leq |S|$. The rank inequality

$$\sum_{e\in S} y_e \leq \left\lfloor \frac{|S|(q-1)}{t} \right\rfloor \tag{3}$$

defines a facet of $P_{FS}(\Sigma)$ if and only if $t = q$ and one of the following holds

(1) $t = |S|$ (IIS-inequality)

(2) S is closed, $t = |S| - 2$ and $t \neq 2$.

Proof. By Proposition 4.5, there are only two cases in which $\mathcal{AW}(|S|,t)$ can arise as an induced hypergraph of an IIS-hypergraph (in both of them necessarily $t = q$).

i) Case $t = |S|$: $\mathcal{AW}(|S|,t)$ consists of a single circuit (IIS). Since Theorem 4.1 implies that S is closed, this gives (together with Lemma 4.7) another proof that the rank facets arising from IISs define facets.

ii) Case $t = |S| - 2$: By Lemma 4.6, inequality (3) defines a facet for $P(\mathcal{AW}(|S|,t))$ if and only if t does not divide $|S|(t-1) = (t+2)(t-1) = t^2+t-2$. Clearly this can only be the case if $t = 1$ (which is not feasible) or $t = 2$. Therefore by Lemma 4.7, inequality (3) defines a facet of $P_{FS}(\Sigma)$ if and only if S is closed and $t \neq 2$.

This proves the theorem. □

Example 4.9. Figure 3 shows an infeasible system with $m = 5$ inequalities in dimension $n = 2$ (see also [41]). Its IISs form an $\mathcal{AW}(5,3)$. The inequalities are indexed by $0, 1, 2, 3, 4$. In the corresponding $P_{FS}(\Sigma)$ polytope the variables are numbered likewise. Its full description is given by the following facets:

- Trivial bounds: $0 \leq y_i \leq 1$ for $0 \leq i \leq 4$.
- The IIS-inequalities: $\sum_{i\in S} y_i \leq 2$ for $S = \{0,1,2\}$, $\{1,2,3\}$, $\{2,3,4\}$, $\{3,4,0\}$, $\{4,0,1\}$.
- The rank inequality $y_0 + y_1 + y_2 + y_3 + y_4 \leq 3$ arising from the unique generalized antiweb.

5. Concluding Remarks

A question that naturally arises is whether our results are also valid for more general (mixed) linear systems with equality as well as inequality relations. Since any equation $\boldsymbol{ax} = b$ can be substituted by the pair of inequalities $\boldsymbol{ax} \leq b$ and $-\boldsymbol{ax} \leq -b$, any generalized Max FS instance I with m_1 equations and m_2 inequalities can obviously be reduced to a usual Max FS instance I' with $2m_1 + m_2$ inequalities, in which one aims at maximizing the number of such pairs of inequalities that can be simultaneously satisfied. Clearly, since any vector $\boldsymbol{x}$ satisfies at least one inequality out of each pair, an optimal solution of I contains m^* linear relations if and only if an optimal solution of I' contains $m^* + m_1$ inequalities. Thus, from a computational point of view, generalized instances of Max FS with mixed systems can be dealt with a polyhedral approach based, among others, on the facet-defining inequalities discussed in this paper. Not all of the above results, however, can be easily generalized to mixed systems. In particular, it is still open whether the simplex decomposition characterization (Theorem 2.7) can be extended. On the other hand, the complexity results regarding Min IIS (Theorem 2.8) and the IIS-hypergraph Recognition problem (Theorem 3.7) obviously hold for this generalized class of instances. Note also that generalized versions of the alternative polyhedron result (Theorem 2.2) for general mixed systems or mixed systems (LPs) where all inequalities are nonnegativity constraints are given in [40].

In this paper we have investigated structural and algorithmic properties of IISs, IIS-hypergraphs, and of the feasible subsystem polytope $P_{FS}(\Sigma)$. On the structural and geometric side, we have: provided a new characterization of IISs, given a new proof of the fact that all IISs are closed, and shown that only two very specific types of generalized antiwebs (generalized cliques and odd holes) can arise as induced hypergraphs of an IIS-hypergraph. In particular, the only generalized cliques that can occur are those corresponding to single IISs. The above results imply that the feasible subsystem polytope $P_{FS}(\Sigma)$ admits only a very limited type of rank facets induced by generalized antiwebs. This is in sharp contrast with other known independence system polytopes related to graphical problems, such as the maximum cardinality stable set problem in a graph, for which a wealth of such rank facets have been extensively studied. On the algorithmic side, we have established that: finding smallest cardinality IISs is very hard to approximate, IIS-hypergraph recognition is NP-hard and IIS rank facets cannot be separated in polynomial time, unless P = NP.

Interesting open questions include: What is the computational complexity of separating inequalities arising from generalized antiwebs? Do other P_{FS}-specific rank facets exist? Does the polytope P_{FS} admit higher order facets besides the ones studied in [9] with $0, 1, 2$ coefficients?

Acknowledgements

The authors would like to thank Michael Joswig, Volker Kaibel and Günter M. Ziegler for helpful discussions. We are also grateful to the anonymous referees for useful comments that improved the paper.

Appendix

To prove Lemma 3.5 of Section 3.1, we first need to verify the following.

Claim. Let P be a d-dimensional pointed polyhedron which has a description over K and is not a polyhedral cone. Let m be the number of facets. Then there exists a polyhedron

$$P' = \left\{ \boldsymbol{y} \in \mathbb{R}^m \;:\; A^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{0},\ \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} = -1,\ \boldsymbol{y} \geq \boldsymbol{0} \right\},$$

where $A \in K^{m\times(m-d-1)}$ and all inequalities $y_j \geq 0$, $1 \leq j \leq m$, define facets, which is affinely (and hence combinatorially) equivalent to P.

Proof. By projection onto the affine hull of P we can assume, without loss of generality, that P is full-dimensional. Moreover, it can be represented as $P = \{\boldsymbol{x} \in \mathbb{R}^d \;:\; C\boldsymbol{x} \leq \boldsymbol{c}\}$. Since P has a minimal description over K, $C \in K^{m\times d}$ and each inequality defines a facet. The resulting polyhedron is affinely equivalent to P and can be represented as:

$$\left\{ \boldsymbol{x} \in \mathbb{R}^d \;\middle|\; \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \boldsymbol{x} \leq \begin{pmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{pmatrix} \right\},$$

where C_1 is a full-rank $d\times d$ matrix (P is pointed), C_2 is an $(m-d)\times d$ matrix, $\boldsymbol{c}_1 \in K^d$, and $\boldsymbol{c}_2 \in K^{m-d}$. Now apply the (bijective) affine transformation $\boldsymbol{x} \mapsto C_1^{-1}(\boldsymbol{c}_1 - \boldsymbol{u})$, where $\boldsymbol{u} := \boldsymbol{c}_1 - C_1\boldsymbol{x} \in \mathbb{R}^d$ and get:

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} C_1^{-1}(\boldsymbol{c}_1 - \boldsymbol{u}) \leq \begin{pmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{pmatrix} \;\Leftrightarrow\; \begin{pmatrix} -I \\ -C_2C_1^{-1} \end{pmatrix} \boldsymbol{u} \leq \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{c}_2 - C_2C_1^{-1}\boldsymbol{c}_1 \end{pmatrix}.$$

Setting $\boldsymbol{c}' := \boldsymbol{c}_2 - C_2C_1^{-1}\boldsymbol{c}_1$ and $C' := -C_2C_1^{-1} \in K^{(m-d)\times d}$ gives

$$P \cong \{\boldsymbol{u} \in \mathbb{R}^d \;:\; C'\boldsymbol{u} \leq \boldsymbol{c}',\ \boldsymbol{u} \geq \boldsymbol{0}\}.$$

Clearly, all inequalities define facets. The usual introduction of slack variables $\boldsymbol{s} \in \mathbb{R}^{m-d}$ yields

$$P \cong \left\{ (\boldsymbol{u}, \boldsymbol{s}) \in \mathbb{R}^d \times \mathbb{R}^{m-d} \;:\; C'\boldsymbol{u} + I\boldsymbol{s} = \boldsymbol{c}',\ \boldsymbol{u} \geq \boldsymbol{0},\ \boldsymbol{s} \geq \boldsymbol{0} \right\},$$

in which all inequalities still define facets and the matrix $[C'\ I]$ has size $(m-d)\times m$.

Since P is not a cone, we must have $\boldsymbol{c}' \neq \boldsymbol{0}$. Therefore $\boldsymbol{c}'$ has at least one nonzero component; assume it is the last one. By adding multiples of the last row to the other rows of $[C' \; I \,|\, \boldsymbol{c}']$, we can eliminate all other nonzero components of $\boldsymbol{c}'$. The resulting system with matrix $[A' \; A'']$ and right hand side $(0, \dots, 0, \alpha)^{\mathrm{T}}$, with $\alpha \neq 0$, is clearly affinely equivalent. We denote by A^{T} the matrix $[A' \; A'']$ without the last row and by $\boldsymbol{b}^{\mathrm{T}}$ the last row of $[A' \; A'']$ divided by $-\alpha$ (in order to scale the right hand side to -1). Then $A \in K^{m \times (m-d-1)}$, $\boldsymbol{b} \in K^m$ and

$$P \cong P' := \left\{ \boldsymbol{y} \in \mathbb{R}^m \; : \; A^{\mathrm{T}} \boldsymbol{y} = \boldsymbol{0}, \; \boldsymbol{b}^{\mathrm{T}} \boldsymbol{y} = -1, \; \boldsymbol{y} \geq \boldsymbol{0} \right\},$$

where each inequality $\boldsymbol{y}_j \geq 0$ defines a facet for $j = 1, \dots, m$. Since only affine transformations were applied, P' is affinely equivalent to P. □

Proof of Lemma 3.5. According to the claim, there exists a polyhedron P' affinely equivalent to P, where

$$P' = \left\{ \boldsymbol{y} \in \mathbb{R}^m \; : \; A^{\mathrm{T}} \boldsymbol{y} = \boldsymbol{0}, \; \boldsymbol{b}^{\mathrm{T}} \boldsymbol{y} = -1, \; \boldsymbol{y} \geq \boldsymbol{0} \right\}.$$

Each face of P' defined by $y_j = 0$ is a facet, $1 \leq j \leq m$. Now V corresponds to the vertices of P and hence P'. If one identifies V with the set of vertices of P', then each set of $\mathcal{E}$ is the vertex set of a facet of P'. Moreover, each set $E^* \in \mathcal{E}^*$ is the set of facets which contain a specific vertex $\boldsymbol{v}$ of P'. If we identify $[m]$ with the set of facets, $\overline{E^*}$ is the support of $\boldsymbol{v}$. Thus, by Theorem 2.2, $\{A\boldsymbol{x} \leq \boldsymbol{b}\}$ is an infeasible system whose IISs correspond bijectively to the sets in $\overline{\mathcal{E}^*}$. □

References

[1] C. C. Aggarwal, R. K. Ahuja, J. Hao, and J. B. Orlin, *Diagnosing infeasibilities in network flow problems*, Math. Programming **81** (1998), pp. 263–280.

[2] E. Amaldi, *From finding maximum feasible subsystems of linear systems to feedforward neural network design*, PhD thesis, Dep. of Mathematics, EPF-Lausanne, October 1994.

[3] E. Amaldi, P. Belotti, and R. Hauser, *Randomized relaxation methods for the maximum feasible subsystem problem*, in Proc. 11th International Conference on Integer Programming and Combinatorial Optimization (IPCO), Berlin, M. Jünger and V. Kaibel, eds., LNCS 3509, Springer-Verlag, Berlin Heidelberg, 2005, pp. 249–264.

[4] E. Amaldi and V. Kann, *The complexity and approximability of finding maximum feasible subsystems of linear relations*, Theoret. Comput. Sci. **147** (1995), pp. 181–210.

[5] E. Amaldi and V. Kann, *On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems*, Theoret. Comput. Sci. **209** (1998), pp. 237–260.

[6] E. Amaldi and M. Mattavelli, *The MIN PFS problem and piecewise linear model estimation*, Discrete Appl. Math. **118** (2002), pp. 115–143.

[7] E. Amaldi, M. E. Pfetsch, and L. E. Trotter, Jr., *Some structural and algorithmic properties of the maximum feasible subsystem problem*, in Proceedings of the 10th Integer Programming and Combinatorial Optimization conference (IPCO'99), G. Cornuéjols, R. Burkard, and G. Woeginger, eds., Springer-Verlag, 1999, pp. 45–59. Lecture Notes in Comput. Sci. 1610.

[8] S. Arora, L. Babai, J. Stern, and Z. Sweedyk, *The hardness of approximate optima in lattices, codes, and systems of linear equations*, J. Comput. Syst. Sci. **54**, no. 2 (1997), pp. 317–331.

[9] E. Balas and S. M. Ng, *On the set covering polytope: All the facets with coefficients in {0,1,2}*, Math. Programming **43** (1989), pp. 57–69.

[10] K. P. Bennett and E. Bredensteiner, *A parametric optimization method for machine learning*, INFORMS J. Comput. **9** (1997), pp. 311–318.

[11] C. Berge, *Graphs and Hypergraphs*, North-Holland, 2nd ed., 1976.

[12] A. Björner, *Topological methods*, in "Handbook of Combinatorics," Vol. II, R. Graham, M. Grötschel, and L. Lovász, eds., North-Holland, 1995, pp. 1819–1872.

[13] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. M. Ziegler, *Oriented Matroids*, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2nd ed., 1999.

[14] L. Blum, F. Cucker, M. Shub, and S. Smale, *Complexity and Real Computation*, Springer-Verlag, 1997.

[15] J. Bokowski and B. Sturmfels, *Computational Synthetic Geometry*, no. 1355 in Lecture Notes in Math., Springer-Verlag, 1989.

[16] S. Ceria, P. Nobili, and A. Sassano, *Set covering problem*, in Annotated Bibliographies in Combinatorial Optimization, M. Dell'Amico, F. Maffioli, and S. Martello, eds., John Wiley, 1997, ch. 23.

[17] N. Chakravarti, *Some results concerning post-infeasibility analysis*, Eur. J. Oper. Res. **73** (1994), pp. 139–143.

[18] J. W. Chinneck, *Computer codes for the analysis of infeasible linear programs*, J. Oper. Res. Soc. **47** (1996), pp. 61–72.

[19] J. W. Chinneck, *An effective polynomial-time heuristic for the minimum-cardinality IIS set-covering problem*, Ann. Math. Artificial Intelligence **17** (1996), pp. 127–144.

[20] J. W. Chinneck, *Finding a useful subset of constraints for analysis in an infeasible linear program*, INFORMS J. Comput. **9**, no. 2 (1997), pp. 164–174.

[21] J. W. Chinneck, *Fast heuristics for the maximum feasible subsystem problem*, INFORMS J. Comput. **13**, no. 3 (2001), pp. 210–223.

[22] J. W. Chinneck and E. Dravnieks, *Locating minimal infeasible constraint sets in linear programs*, ORSA J. Comput. **3** (1991), pp. 157–168.

[23] K. Fan, *On systems of linear inequalities*, in Linear Inequalities and Related Systems, H. W. Kuhn and A. W. Tucker, eds., no. 38 in Ann. of Math. Stud., Princeton University Press, NJ, 1956, pp. 99–156.

[24] M. Frean, *A "thermal" perceptron learning rule*, Neural Comput. **4**, no. 6 (1992), pp. 946–957.

[25] M. R. Garey and D. S. Johnson, *Computers and Intractability: A guide to the theory of NP-completeness*, W. H. Freeman and Company, San Francisco, 1979.

[26] J. Gleeson and J. Ryan, *Identifying minimally infeasible subsystems of inequalities*, ORSA J. Comput. **2**, no. 1 (1990), pp. 61–63.

[27] H. J. Greenberg, *Consistency, redundancy, and implied equalities in linear systems*, Ann. Math. Artificial Intelligence **17** (1996), pp. 37–83.

[28] H. J. Greenberg and F. H. Murphy, *Approaches to diagnosing infeasible linear programs*, ORSA J. Comput. **3** (1991), pp. 253–261.

[29] R. Greer, *Trees and Hills: Methodology for Maximizing Functions of Systems of Linear Relations*, Ann. Discrete Math. 22, Elsevier science publishing company, Amsterdam, 1984.

[30] J. Håstad, *Some optimal inapproximability results*, J. of ACM **48** (2001), pp. 798–859.

[31] D. S. Johnson and F. P. Preparata, *The densest hemisphere problem*, Theoret. Comput. Sci. **6** (1978), pp. 93–107.

[32] M. Joswig, V. Kaibel, M. E. Pfetsch, and G. M. Ziegler, *Vertex-facet incidences of unbounded polyhedra*, Adv. Geom. **1**, no. 1 (2001), pp. 23–36.

[33] M. Laurent, *A generalization of antiwebs to independence systems and their canonical facets*, Math. Programming **45** (1989), pp. 97–108.

[34] O. L. Mangasarian, *Misclassification minimization*, J. Global Optim. **5**, no. 4 (1994), pp. 309–323.

[35] B. Mishra, *Computational real algebraic geometry*, in Handbook of Discrete and Computational Geometry, J. Goodman and J. O'Rouke, eds., CRC Press, 1997, ch. 29.

[36] N. E. Mnëv, *The universality theorems on the classification problem of configuration varieties and convex polytopes varieties*, in Topology and Geometry – Rohlin Seminar, O. Y. Viro, ed., no. 1346 in Lecture Notes in Math., Springer-Verlag, 1988, pp. 527–543.

[37] T. S. Motzkin, *Beiträge zur Theorie der Linearen Ungleichungen*, PhD thesis, University of Basel, 1933.

[38] Netlib Repository. available at http://www.netlib.org.

[39] M. Parker, *A set covering approach to infeasibility analysis of linear programming problems and related issues*, PhD thesis, Dep. of Mathematics, University of Colorado at Denver, 1995.

[40] M. Parker and J. Ryan, *Finding the minimum weight IIS cover of an infeasible system of linear inequalities*, Ann. Math. Artificial Intelligence **17** (1996), pp. 107–126.

[41] M. E. Pfetsch, *Examples of generalized antiweb facets.* Electronic Geometry Models, No. 2000.09.029, available at http://www.eg-models.de, 2000.

[42] J. Richter-Gebert, *Realization Spaces of Polytopes*, no. 1643 in Lecture Notes in Math., Springer-Verlag, 1996.

[43] F. Rossi, A. Sassano, and S. Smriglio, *Models and algorithms for terrestrial digital broadcasting*, Ann. Oper. Res. **107** (2001), pp. 267–283.

[44] J. Ryan, *Transversals of IIS-hypergraphs*, Congr. Numer. **81** (1991), pp. 17–22.

[45] J. Ryan, *IIS-hypergraphs*, SIAM J. Discrete Math. **9** (1996), pp. 643–653.

[46] J. Sankaran, *A note on resolving infeasibility in linear programs by constraint relaxation*, Oper. Res. Lett. **13** (1993), pp. 19–20.

[47] M. Tamiz, S. Mardle, and D. Jones, *Detecting IIS in infeasible linear programmes using techniques from goal programming*, Comput. Oper. Res. **23**, no. 2 (1996), pp. 113–119.

[48] J. N. M. van Loon, *Irreducibly inconsistent systems of linear inequalities*, Eur. J. Oper. Res. **8** (1981), pp. 282–288.

[49] G. M. Ziegler, *Lectures on Polytopes*, Springer-Verlag, New York, 1995. Revised edition 1998.

Paper 6

Branch-And-Cut for the Maximum Feasible Subsystem Problem

Marc E. Pfetsch
Branch-And-Cut for the Maximum Feasible Subsystem Problem[1]
SIAM J. Optimization **19** (2008), no. 1, pp. 21–38

Abstract. This paper presents a branch-and-cut algorithm for the NP-hard maximum feasible subsystem problem: For a given infeasible linear inequality system, determine a feasible subsystem containing as many inequalities as possible. The complementary problem, where one has to remove as few inequalities as possible in order to make the system feasible, can be formulated as a set covering problem. The rows of this formulation correspond to irreducible infeasible subsystems, which can be exponentially many. It turns out that the main issue of a branch-and-cut algorithm for Max FS is to efficiently find such infeasible subsystems. We present three heuristics for the corresponding NP-hard separation problem and discuss cutting planes from the literature, such as set covering cuts of Balas and Ng, Gomory cuts, and $\{0, \frac{1}{2}\}$-cuts. We furthermore compare a heuristic of Chinneck and a simple greedy algorithm. The main contribution of this paper is an extensive computational study on a variety of instances arising in a number of applications.

1. Introduction

In the *maximum feasible subsystem problem* (Max FS), we are given an infeasible linear inequality system $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$, with $A \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m$, and have to find a feasible subsystem containing as many inequalities as

[1] Supported by the DFG Research Center Matheon in Berlin

possible. This NP-hard combinatorial optimization problem has a number of interesting applications in a wide range of fields, for instance, in linear programming [29, 31, 36], statistical discriminant analysis and machine learning [4, 19, 43], telecommunications [54], and computational biology [61]. Additional applications and a survey can be found in [4] and [5], respectively.

The complementary problem of MAX FS amounts to removing as few inequalities of Σ as possible so that the resulting system is feasible. To achieve feasibility, one has to remove at least one inequality from each *irreducible infeasible subsystem* (IIS), i.e., an infeasible subsystem of Σ for which every proper subsystem is feasible. Introducing a binary variable y_i for each inequality of Σ, the complementary problem can be formulated as a set covering problem and is therefore called MIN IIS COVER:

$$\begin{array}{lll} \min & \sum_{i=1}^{m} y_i & \\ \text{s.t.} & \sum_{i\in I} y_i \geq 1 & \text{for all IISs } I \\ & \boldsymbol{y} \in \{0,1\}^m. & \end{array} \tag{1}$$

Since the number of IISs can be exponential in the size of the system Σ (see Chakravarti [28] and Pfetsch [53]), IISs have to be generated dynamically in order to solve this formulation of MIN IIS COVER efficiently.

Clearly, the set of all inequalities not contained in a solution of MAX FS form a solution of MIN IIS COVER and vice versa. Hence, these two problems are equivalent when solving to optimality and are both strongly NP-hard, see Johnson and Preparata [39], Sankaran [58], and Chakravarti [28]. In terms of approximability, however, they differ: MAX FS does not admit a polynomial-time approximation scheme, unless P = NP, but there exists a 2-approximation, see Amaldi and Kann [9]. MIN IIS COVER is harder to approximate: Unless P = NP, it cannot be approximated within any constant factor, see Amaldi and Kann [10].

In this paper, we present a branch-and-cut approach for MAX FS via formulation (1) for MIN IIS COVER. A key issue of this approach is to find violated *IIS-inequalities*, i.e., the inequalities arising from IISs in (1). The corresponding separation problem is NP-hard, and we present three heuristics for it (see Section 3.2). Two of these methods either generate a feasible solution for MIN IIS COVER or a (hopefully violated) IIS-inequality. As long as no feasible solution has been generated, the process is iterated, which often produces many useful IIS-inequalities. The additional benefit are reasonably good primal solutions, which can be improved by a simple greedy algorithm. This combination leads to an effective primal heuristic. Additionally, we examine the application of inequalities of Balas and Ng [18] for set covering problems, $\{0, \frac{1}{2}\}$-cuts, and Gomory cuts.

The emphasis of this paper is on an extensive computational study of the branch-and-cut implementation. Our aim is to show the potential and the limits of such an approach by performing tests on three problem sets: random infeasible inequalities systems (Section 4.2), problems arising in digital video broadcasting (Section 4.3), and classification problems (Section 4.4).

The theoretical foundation for our approach appears in Amaldi, Pfetsch, and Trotter [12], where algorithmic and geometric questions concerning IISs are studied and the feasible subsystem polytope is investigated. (The polyhedral results carry over to the polytope for Min IIS Cover by a simple affine transformation.) The work presented here is an improved version of part of the author's Ph.D. thesis [53].

In the literature so far, only two exact approaches towards Min IIS Cover appeared. Parker and Ryan [52] discuss an iterative approach that generates IISs in each step and then solves an integer program. This approach turns out to be impractical for harder instances. Codato and Fischetti [33] present a branch-and-cut algorithm for Min IIS Cover in a more general context. We discuss these approaches in more detail in the next section. Our algorithm improves upon both methods and is currently the best available exact approach (see Section 4).

The outline of this paper is as follows. In Section 2 we review solution approaches for Max FS. In Section 3 we describe the main ingredients of our branch-and-cut implementation. We discuss a way to check the feasibility of solutions for Min IIS Cover, three methods to separate IIS-inequalities, primal heuristics, preprocessing, branching, inequalities by Balas and Ng, and further used cutting planes. In Section 5 we extensively test the implementation on the above mentioned problem sets. We close with some conclusions in Section 5.

We use the following notation. We define $[n] := \{1, \dots, n\}$ for $n \in \mathbb{N}$ and typeset vectors in bold font. For a set $S \subseteq [n]$ and a vector $\boldsymbol{x} \in \mathbb{R}^n$, define

$$\boldsymbol{x}(S) = \sum_{i \in S} x_i \,.$$

The *support* of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is $\operatorname{supp}(\boldsymbol{x}) := \{i \in [n] \;:\; x_i \neq 0\}$. By $\mathbb{1}$ we denote a vector of all ones of appropriate dimension.

2. Alternative Solution Approaches

In this section we give a short overview of solution approaches for Max FS and Min IIS Cover.

In the context of linear programming, attention was first devoted to the problem of identifying IISs with a small and possibly minimum number of inequalities (see Greenberg and Murphy [36]; Chinneck [30]; Chinneck and Dravnieks [32]). The goal is to help the modeler resolve infeasibility of large linear programs. Since minimum cardinality covers of IISs reveal essential information about infeasibility of the model and are often smaller than IISs, emphasis has shifted towards their identification. Chinneck [29, 31] developed heuristics for Max FS/Min IIS Cover and provided computational results, see Section 4.4. These heuristics are extended greedy algorithms.

For the application of Min IIS Cover to classification problems (see Section 4.4), several heuristics were proposed, based on nonlinear programming formulations of Max FS (Bennett and Bredensteiner [19]; Bennett and Mangasarian [20]; Mangasarian [43]).

An exact integer programming approach for Min IIS Cover appeared in Parker [51] and Parker and Ryan [52]. Their idea is to consider the formulation in (1) with a partial list of IISs. If there exist IISs that are not covered by a solution to this formulation, they are added and the process is iterated. Otherwise, an optimal solution to MIN IIS COVER is found. Parker and Ryan discuss several methods to generate IISs at each step and consider heuristics for solving the set covering problem (only the last instance has to be solved exactly).

We reimplemented a basic version of their algorithm, where the set covering problems are solved to optimality. This implementation turned out to be inferior to our branch-and-cut implementation: it could not solve instances within one hour, solved by our branch-and-cut approach within a few minutes. We therefore refrained from performing further experiments.

There is a straightforward mixed integer programming formulation for MIN IIS COVER containing a binary variable with a "big-M" for each of the inequalities of Σ, so that an inequality is relaxed when the corresponding binary variable is 1. This formulation has the typical numerical problems of big-M formulations and is in general inefficient for MAX FS, see Parker [51]. If there are fixed bounds on the variables, however, one can obtain a tight formulation. This leads to a quite efficient approach, see Rossi, Sassano, and Smriglio [54] and Codato and Fischetti [33]. In fact, Codato and Fischetti proposed a general way of removing the "big-M" from this type of formulations and apply it to classification instances. In this context, it leads to the formulation (1) and their solution method is in fact a branch-and-cut method for MIN IIS COVER, independent from our approach. Computational results show that their approach is faster compared to the big-M formulation. In Section 4.4 we compare our implementation with their approach.

Versions of the classical *relaxation method* of Agmon [3] and Motzkin and Schoenberg [47] for solving linear inequality systems can be applied to minimize the sum of violations in infeasible linear inequality systems. Randomized variants of this method were proposed by Amaldi [4] to solve MAX FS. Amaldi and Hauser [8] and Amaldi, Belotti, and Hauser [6] establish probabilistic convergence guarantees to an optimal solution of MAX FS under appropriate conditions. Computational results for digital video broadcasting data, classification instances, and huge systems arising in computational biology are given in [6].

Amaldi, Bruglieri, and Casale [7] propose a two-step heuristic in which first a linearization of an exact bilinear formulation of MAX FS is used to derive a feasible subsystem. In the second step, a reduced problem is solved to optimality in order to identify inequalities that can be added to the first system while preserving feasibility. This turns out to be competitive with respect to the method of Codato and Fischetti and CPLEX applied to the "big-M" formulation for the whole system.

3. Ingredients for Branch-and-Cut

In the following we assume that the reader is familiar with the branch-and-cut approach. More information can be found in Nemhauser and Wolsey [48], Padberg and Rinaldi [50], Thienel [60], and Caprara and Fischetti [26]. A description and computational study of Gomory cuts is given in Balas, Ceria, Cornuéjols, and Natraj [17].

Recall that we are given the infeasible system $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$, where $A \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$. Depending on the application, *mandatory* variable bounds can be present, i.e., these bounds may not be removed for obtaining a feasible system (see Sections 4.3 and 4.4). This can easily be dealt with in the branch-and-cut approach. Furthermore, weighted versions of Min IIS Cover are easy to handle, too.

Without loss of generality we can restrict attention to inequality systems in the form of Σ: Clearly, bounds on variables and "greater or equal" inequalities can be transformed to this format. Equations can be replaced by a pair of opposing inequalities. Since any point satisfies at least one inequality out of each pair, an optimal solution to the new instance contains $m^* + m_E$ inequalities if and only if an optimal solution to the original instance with m^* linear relations exists; here m_E is the number of equations. Thus, from a computational point of view, it suffices to handle systems in the form of Σ. Polyhedral results for the two cases, however, may differ, see [12, 53] for more information.

To simplify notation, we identify an inequality of Σ with its index. Then $S(\Sigma) := [m]$ is the set of constraints of Σ. With this notation, $I \subseteq S(\Sigma)$ is an IIS of Σ if and only if all proper subsets of I are feasible. We call a set $C \subseteq S(\Sigma)$ an *IIS-cover* if it intersects every IIS of Σ.

In the rest of this section we give a more detailed account of the main aspects of our implementation: the recognition problem for IIS-covers, the separation problem of IIS-inequalities, pool handling, primal heuristics, preprocessing, branching, and further cutting planes.

3.1. Recognition Problem for IIS-Covers

We consider the following fundamental problem: Given a subset $C \subseteq S(\Sigma)$, check whether it is an IIS-cover and if this is not the case generate a witness, i.e., an IIS which is not covered. Our approach is based on the following theorem.

Theorem 3.1 (Gleeson and Ryan [35]). Let $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ be an infeasible system. Then the IISs of Σ are in one-to-one correspondence with the supports of the vertices of the polyhedron

$$P(\Sigma) := \{\, \boldsymbol{y} \in \mathbb{R}^m \;:\; \boldsymbol{y}^{\mathrm{T}} A = \boldsymbol{0},\ \boldsymbol{y}^{\mathrm{T}} \boldsymbol{b} = -1,\ \boldsymbol{y} \geq \boldsymbol{0} \,\}.$$

Note that the vertices of $P(\Sigma)$ are uniquely defined by their supports. This theorem is strongly related to the Farkas lemma, which states that $P(\Sigma) \neq \varnothing$ if and only if Σ is infeasible, see e.g. Schrijver [59]. The polyhedron $P(\Sigma)$ is called the *alternative polyhedron* of Σ.

To apply Theorem 3.1, we define for $S \subseteq S(\Sigma)$ the polyhedron

$$P_S(\Sigma) := \{\, \boldsymbol{y} \in P(\Sigma) \;:\; y_i = 0,\ i \in S \,\},$$

which might be empty. We need the following fact:

Lemma 3.2 (Parker and Ryan [52]). The set $C \subseteq S(\Sigma)$ is an IIS-cover if and only if $P_C(\Sigma) = \varnothing$.

Proof. The system defining $P(\Sigma)$, in which all variables indexed by C are removed, has no solution if and only if $P_C(\Sigma) = \varnothing$. By the Farkas lemma, the former is the case if and only if Σ with inequalities indexed by C removed is feasible, i.e., C is an IIS-cover. □

Recognizing whether $C \subseteq S(\Sigma)$ is an IIS-cover is then easy: In the case of $P_C(\Sigma) = \varnothing$, by Lemma 3.2, C is an IIS-cover. Otherwise, let $\boldsymbol{v}$ be a vertex of $P_C(\Sigma)$. Then $\operatorname{supp}(\boldsymbol{v}) \cap C = \varnothing$, which shows that $\operatorname{supp}(\boldsymbol{v})$ is an IIS that is uncovered (by Theorem 3.1). This provides a polynomial time algorithm for the problem, since finding a vertex of a polyhedron can be done in polynomial time, see Grötschel, Lovász, and Schrijver [37]. Note that by Theorem 3.1 and Lemma 3.2, $P_C(\Sigma)$ always has a vertex if it is nonempty.

This recognition test in fact suffices for a rudimentary branch-and-cut algorithm, since we can now test feasibility of a vector $\boldsymbol{y} \in \{0,1\}^m$ for (1) by testing whether $\operatorname{supp}(\boldsymbol{y})$ is an IIS-cover.

3.2. Separation of IIS-Inequalities

IIS-inequalities play a prominent role in the formulation (1) for Min IIS Cover. In fact, it can be shown that the inequality arising from the IIS I defines a facet of the polytope

$$P_{IISC} = \operatorname{conv}\{\, \boldsymbol{y} \in \{0,1\}^m \;:\; y(S) \geq 1 \text{ for all IISs } S \,\},$$

as long as $|I| > 1$, see Amaldi, Pfetsch, and Trotter [12]. Therefore, the following *separation problem for IIS-inequalities* is of crucial importance: Given a vector $\boldsymbol{y}^* \in [0,1]^m$, check whether there exists an IIS I so that its corresponding inequality is violated by $\boldsymbol{y}^*$, i.e., $\boldsymbol{y}^*(I) < 1$. The recognition problem for IIS-covers is a special case, where $\boldsymbol{y}^*$ is the incidence vector of the set to be tested. In the general case, however, we have the following.

Proposition 3.3 (Amaldi, Pfetsch, and Trotter [12])**.** The separation problem for IIS-inequalities is NP-hard.

In this section, we therefore present three heuristics for the separation problem. All of these heuristics may fail to produce a violated IIS-inequality.

The heuristics build on the following reformulation of the separation problem: Compute

$$\lambda := \min\{\, \boldsymbol{y}^*(S) \;:\; S = \operatorname{supp}(\boldsymbol{v}),\ \boldsymbol{v} \text{ vertex of } P(\Sigma) \,\}. \tag{2}$$

If $\lambda < 1$, by Theorem 3.1, $\operatorname{supp}(\boldsymbol{v})$ provides an IIS whose IIS-inequality is violated; otherwise no such IIS exists (we define $\lambda = \infty$ if $P(\Sigma) = \varnothing$).

3.2.1. Method 1: "Single"

The first quite intuitive idea to separate an IIS-inequality, already used by Parker and Ryan [52], is to approximate (2) by the following LP:

$$\min\{\, (\boldsymbol{y}^*)^{\mathrm{T}} \boldsymbol{p} \;:\; \boldsymbol{p} \in P(\Sigma) \,\}.$$

A vertex solution provides an IIS, whose corresponding inequality is not necessarily violated, but in practice it often is.

This method only generates one IIS at a time. We also experimented with solving the above LP by the simplex algorithm and then testing whether the support of each vertex on the path to the optimum is an IIS whose inequality is violated. In our experiments this variant was inefficient and will not be considered further.

3.2.2. Method 2: "Extend"

We extend method 1 as follows. Let S be the support of $\boldsymbol{y}^*$. Applying Lemma 3.2, we can check whether S is an IIS-cover by finding a vertex solution of

$$\min\{\, (\boldsymbol{y}^*)^{\mathrm{T}}\boldsymbol{p} \;:\; \boldsymbol{p} \in P_S(\Sigma)\,\},$$

if there exists one. If the LP is feasible, the result gives us a vertex which corresponds to an IIS, otherwise we found an IIS-cover, i.e., a primal solution for Min IIS Cover.

This approach can be iterated when S is not an IIS-cover. Let I be the IIS obtained in this case. We enlarge S greedily by an element of I and iterate. In our implementation, we choose an element of I that is contained in the maximal number of IISs we have found so far. At termination this yields an IIS-cover. This procedure is related to a primal heuristic proposed by Ryan [57].

The IISs found by this approach have several nice properties. First, the new IISs are different from all IISs that were known before the run, if the current solution $\boldsymbol{y}^*$ of the LP-relaxation satisfies $y^*(I) \geq 1$ for each previously found IIS I. This follows since at least one element of each I is contained in S, and hence I cannot be generated again. Second, the corresponding inequalities are always violated, since they have empty intersection with $S \supseteq \operatorname{supp}(\boldsymbol{y}^*)$, i.e., $\boldsymbol{y}^*(I) = 0 < 1$ for each produced IIS I. Third, by construction of the set S, the generated IISs are pairwise different.

This method turns out to be quite effective for generating many violated IIS-inequalities. Furthermore, we obtain a primal solution in each run, which can be improved to very good solutions, see Section 3.4. When the current LP-relaxation contains many cuts, however, the support of $\boldsymbol{y}^*$ tends to be large and often is already an IIS-cover or close to one, and the method cannot produce new IISs; this often happens in the deeper regions of the branch-and-bound tree. This might even be desirable, since this saves time for high depths. Nevertheless, this situation can be changed as indicated by the next method.

3.2.3. Method 3: "Round"

The idea of method 2 can be further extended by using the fact that an arbitrary set S can be used at the start. In the extension, we choose $\alpha \in [0,1]$ and initially let $S := \{i : y_i^* \geq \alpha\}$. In the implementation we start with $\alpha = 0.1$ and then increase α by 0.1 until S is not an IIS-cover (in this case the above procedure is started). We terminate with a failure if α exceeds 0.6.

The fact that S is smaller for larger α has two effects: First, the number of steps needed to greedily obtain an IIS-cover is larger and hence the number of generated IISs is increased. Second, the method also computes IISs in the deeper regions of the tree.

Again, in each step an IIS is generated, which is not covered by S, except in the last step where we obtain an IIS-cover. In contrast to method "extend", the generated IISs are not necessarily new and their corresponding inequalities may not be violated by $\boldsymbol{y}^*$.

3.3. Pool for IIS-Inequalities

The above three methods tend to produce many IISs, which we store in a pool. It turned out that the best performance of the algorithm is achieved by checking the pool for violated inequalities in *every* node of the tree. Of course, the pool should be as small as possible without losing important inequalities. Therefore, the pool is equipped with an aging mechanism which removes IISs whose inequality has not been active for some time.

The computational results presented in Section 4 indicate that only a small fraction of the total number of IISs needs to be generated by our branch-and-cut implementation; indeed, for larger problems there are far too many IISs to be enumerated completely, cf. Table 2 in Section 4.2. Hence, the size of the pool can be relatively small.

3.4. Primal Heuristics

Chinneck [31] proposed a greedy heuristic for Min IIS Cover, which we use as an initial primal heuristic. The basic tool is a so-called *elastic LP* in which the inequalities $\Sigma : \{A\boldsymbol{x} \leq \boldsymbol{b}\}$ are relaxed by adding slack variables and the sum of violations is minimized:

$$\begin{aligned} \min\ & \mathbb{1}^{\mathrm{T}}\boldsymbol{s} \\ & A\boldsymbol{x} - \boldsymbol{s} \leq \boldsymbol{b} \\ & \boldsymbol{s} \geq 0. \end{aligned}$$

Starting with $S = \varnothing$, in each iteration $S \subseteq S(\Sigma)$ is enlarged by an inequality that yields the largest drop in the elastic LP objective, if its objective coefficient is set to 0. The method stops once the objective is 0, i.e., S is a Min IIS Cover. To speed up the solution, in each iteration only inequalities from a candidate set are checked. Chinneck proposes a measure based on the violation and dual variables to generate the candidate set. We refer to [31] for the details.

For a heuristic running in the tree, we use a primal heuristic that greedily decreases the size of a given IIS-cover until a minimal one is obtained. We start this heuristic from IIS-covers produced by the separation methods in Section 3.2, if available (otherwise we use a simple rounding heuristic). We start with C being an IIS-cover to be improved. We consider each element from C in the order of increasing fractional value of the current LP-solution $\boldsymbol{y}^*$. We remove an element if the remaining set is an IIS-cover (which is checked by the method in Section 3.1).

3.5. Preprocessing

In a preprocessing step we search for small IISs. Such small IISs are of interest since their corresponding IIS-inequalities provide "strong" cuts and are hard to find by other methods.

We first check for IISs of cardinality one, e.g., $\mathbf{0}\boldsymbol{x} \leq -1$. Then we check for IISs that involve one inequality and bounds on the variables (if present). Such IISs often occur when variable bounds are mandatory, see e.g. Section 4.4. In this case, a single inequality might be infeasible with the bounds and counts as an IIS. Furthermore, we look for IISs of cardinality two, which are easy to find by comparing their normal vectors and right hand sides. Identifying other types of IISs would require higher computational effort.

3.6. Branching

As a branching rule, we apply *reliability branching*, introduced by Achterberg, Koch, and Martin [2]. It performs strong branching on a subset of the variables, which are chosen based on their so-called pseudo costs during branching. If in strong branching one of the child nodes turns out to be infeasible, the corresponding variable is fixed to the complementary value; if both children are infeasible the current node can be pruned.

We also experimented with constraint branching rules. For instance, we used the well-known rule of Ryan and Foster [56]. This rule was superior to a simple variable branching, but inferior to reliability branching both in terms of computation time and the number of branch-and-bound nodes. We therefore selected reliability branching for all tests.

3.7. Inequalities for Set Covering

Many facet-defining inequalities for the set covering polytope have been investigated, see Ceria, Nobili, and Sassano [27] and Borndörfer [22]. However, few (problem-specific) polynomial time separable inequalities for set covering are known. For many classes of inequalities the complexity status is unknown, but is likely to be NP-hard.

We experimented with the aggregated cycle cuts of Borndörfer and Weismantel [23, 24]. Unfortunately, on our test problems their separation heuristic almost never found a violated inequality. It remains as an interesting open problem to identify problem specific inequalities for Min IIS Cover.

A class of inequalities for set covering that we use in our implementation were proposed by Balas and Ng [18]. To describe these inequalities, consider the set covering polytope $P_{SC}(D) = \operatorname{conv}\{\boldsymbol{y} \in \{0,1\}^m \ : \ D\boldsymbol{y} \geq \mathbb{1}\}$, where $D = (d_{ij}) \in \{0,1\}^{k \times m}$. Assume $\boldsymbol{a}\boldsymbol{y} \geq \beta$, with $\boldsymbol{a} \in \mathbb{Z}^m$ and $\beta \in \mathbb{Z}$, defines a facet of $P_{SC}(D)$. It is well known that if $\beta > 0$ then $\boldsymbol{a} \geq \mathbf{0}$, and if $\beta = 1$ then $\boldsymbol{a}$ is a row of D (see, e.g., [18]).

Balas and Ng showed that for every facet defining inequality $\boldsymbol{a}\boldsymbol{y} \geq 2$ with $\boldsymbol{a} \in \mathbb{Z}^n$, there exists a set $S \subseteq [k]$ such that $\boldsymbol{a} = \boldsymbol{a}^S$, where

$$a_j^S = \begin{cases} 0 & \text{if } d_{ij} = 0 \text{ for all } i \in S, \\ 2 & \text{if } d_{ij} = 1 \text{ for all } i \in S, \\ 1 & \text{otherwise} \end{cases} \qquad \text{for } j = 1, \dots, m.$$

These inequalities can also be obtained by a Chvátal-Gomory rounding procedure. Furthermore, Balas and Ng discuss conditions under which $\boldsymbol{a}^S\boldsymbol{y} \geq 2$ defines a facet of $P_{SC}(D)$.

The separation problem for these inequalities is NP-hard, see Amaldi and Pfetsch [11]. However, when the size of S is fixed, the separation problem can be solved in polynomial time by enumeration. In our implementation we enumerate sets S of *cardinality three* and check whether the inequalities $\boldsymbol{a}^S\boldsymbol{y} \geq 2$ are violated by the current LP-solution. Note that sets S of cardinality two are uninteresting, since in this case $\boldsymbol{a}^S\boldsymbol{x} \geq 2$ is the sum of two IIS-inequalities and hence is never violated, if the IIS-inequalities are satisfied.

Additionally, we try to strengthen these cuts: If an inequality is violated, we greedily enlarge the set S as long as the violation of the resulting inequality increases. See Section 4 for computational results.

3.8. General Purpose Inequalities

In our computational experiments we used Gomory cuts as implemented in SCIP; see the books of Nemhauser and Wolsey [48] or Schrijver [59] for a description.

We further used $\{0, \frac{1}{2}\}$-cuts introduced by Caprara and Fischetti [25]. Codato and Fischetti [33] identified these cuts as important for solving MIN IIS COVER. We implemented these cuts along the lines of Hansen, Labbé, and Schindl [38]. See also Andreello, Caprara, and Fischetti [13] for a computational study of $\{0, \frac{1}{2}\}$-cuts. Note that in our implementation $\{0, \frac{1}{2}\}$-cuts are only produced for set covering and nonnegativity inequalities; in particular, they do not depend on $\{0, \frac{1}{2}\}$-cuts produced earlier.

We also experimented with mixed integer rounding cuts (CMIR) (see Marchand and Wolsey [44]) and strengthened Chvátal-Gomory cuts (see Letchford and Lodi [42]) as they are implemented in SCIP. The results were, however, discouraging and we therefore do not present them.

4. Computational Results

In this section we discuss computational results of our branch-and-cut implementation for MIN IIS COVER. The algorithm was implemented in `C++` and uses version 0.90 of the framework SCIP (Solving Constraint Integer Programs) by Achterberg [1]. CPLEX 10.11 is used as the basic LP solver. The computations were performed on a 3.4 GHz Pentium 4 machine with 3 GB of main memory and 1 GB cache running Linux. All instances used in the following can be obtained from the web page [45].

We use best-first search as a node selection scheme and the branching rule explained in Section 3.6. All separation routines are called only every tenth level of the tree, except that the pool of IIS-inequalities is checked in every node of the tree. In nodes in which cuts are separated, we proceed until no more violated cuts can be found. SCIP chooses among the generated cuts according to an orthogonality measure, see for instance Andreello, Caprara, and Fischetti [13]. We perform reduced cost fixing at every node of the tree.

Before presenting computational results, we want to discuss the influence of the limited precision used for solving LPs. The basic question that has to be repeatedly answered in our context is whether a given system is infeasible or not. Today's LP solvers are tuned towards quickly finding an optimal solution of a feasible LP. Sometimes their bases are not really optimal, but this only has a negligible effect on the objective function value, see Koch [41]. When checking infeasibility, however, small errors can lead to completely wrong decisions. The answer depends on the particular instance, the solution method of the LP solver, its parameters, e.g. the precision (usually around 10^{-6}), and often also the preprocessing and starting basis. Being aware of the possibility that we might produce wrong results, as a safeguard, we confirmed that the final solution is really an IIS-cover for the original system.

Currently, using exact LP solvers, like the ones included in lrs [15] or cdd [34] is computationally too expensive. In the future, codes that use dynamically adjusted precision might help, see Applegate, Cook, Dash, and Espinoza [14].

4.1. The Netlib Problems

The Netlib library [49] contains a well known set of 29 infeasible linear inequality systems. We do not report results on these data since these instances all can be solved within seconds, except for numerical difficulties with the problem `gran`. They were also solved to optimality by Parker [51] and Parker and Ryan [52]; for more computational results on these problems see Chinneck [31] and Pfetsch [53].

4.2. Random Problems

We consider random inequality systems to compare different cut strategies in the branch-and-cut implementation. We used difficult random instances that nevertheless can be solved within approximately one hour of computation time. In contrast, the instances discussed in the following sections vary highly in size and complexity: most are either solved within seconds or cannot be solved to optimality in reasonable time.

The infeasible random inequality systems are generated follows: Each coefficient and the right hand side was chosen to be a random integer in the range -100 to 100. We generated five instances for each of the combinations $(5, 100)$, $(10, 80)$, $(15, 80)$, $(20, 90)$, $(25, 90)$, where the first component is the dimension n of the space and the second one is the number m of inequalities. Each system turned out to be infeasible (this almost always happens as soon as $m > 2 \cdot n$, see Motzkin [46]) and is almost completely dense. Note that all the instances in the following sections are dense as well.

The alternative polyhedra in Theorem 3.1 of these random systems are nondegenerate with high probability. It is currently unknown, whether Max FS and Min IIS Cover restricted to such systems are NP-hard.

We first compare the three different strategies to separate IIS-inequalities of Section 3.2. Table 1 provides a comparison of methods “single” (Section 3.2.1), “extend” (Section 3.2.2), and “round” (Section 3.2.3). Columns “nodes” give the average number of nodes in the branch-and-bound tree, “time” are the average CPU times in seconds, and “IISs” give the average number of IISs found during the optimization; here averages are taken over the five instances of each size. To eliminate the influence of primal heuristics we initialized all runs with the optimal solution.

Table 1: Results of the branch-and-cut algorithm on *random inequality systems* for different IIS separation strategies. The numbers are averages over five instances of each size. The last line gives the averages over each column.

			single			extend			round	
n	m	nodes	time	IISs	nodes	time	IISs	nodes	time	IISs
5	100	70473.0	1050.64	8781.0	120371.4	1808.71	5281.4	16913.8	564.44	11034.8
10	80	167970.8	1226.45	10298.4	174302.6	1689.26	8450.4	79086.6	996.51	14491.8
15	80	214004.0	1509.72	53419.8	255933.0	1984.60	44825.8	106119.0	1465.16	62151.0
20	90	50029.0	276.05	22354.8	59117.8	337.11	15869.0	28699.0	317.22	23418.0
25	90	169868.2	1185.81	99728.6	243568.6	1534.17	80400.4	77147.0	1235.41	155331.4
	$\varnothing$:	134469.0	1049.73	38916.5	170658.7	1470.77	30965.4	61593.1	915.75	53285.4

Table 2: The number of IISs found by method "round" for random problems and the total number of IISs.

n	m	found	total
5	30	11	1986
5	40	101	44816
5	50	520	204833
5	60	526	614853
5	70	453	1818718

Among the three IIS-inequality separation versions method "round" outperforms methods "single" and "extend" in the number of nodes and in the total computation time, although method "single" is sometimes a bit faster. Method "round" also generates the highest number of IISs. Based on this result, we decided to use method "round" in the following experiments.

Table 2 shows the total number of IISs and the number of IISs found by method "round" for small random instances generated in the same manner as above. By Theorem 3.1, the IISs correspond to vertices of the alternative polyhedron. We enumerated the vertices with lrs [15]. Since the alternative polyhedra are nondegenerate, the IISs can be generated in time polynomial in the input and output size, see Avis and Fukuda [16]. Note that for general polyhedra this is not possible, unless P = NP, see Khachiyan et al. [40].

We could not enumerate or count the IISs for larger instances. From Table 2, however, it can be expected that the total number of IISs for the instances used in Table 1 is much higher. We conclude that the branch-and-cut implementation only needs a small part of the total set of IISs (the number of IISs for instance $(5, 70)$ is two orders of magnitudes larger than the average number of IISs found by any of the variants in Table 1).

Table 3 lists computational results for all combinations of method "round" with Balas/Ng cuts (BaNg), Gomory cuts (Gom.), and $\{0, \frac{1}{2}\}$-cuts. The values are averages over all 25 instances. Column "root" gives the dual bound after the root node. The last three columns list the number of cuts found for the respective methods. Again, we initialize the algorithms with the optimal solution. All cuts are separated every 10 levels of the tree.

The studied combinations on average reduce the number of nodes with respect to the method "round" alone; the best combination in this respect are Balas/Ng and Gomory cuts. Furthermore, all combinations, except

Table 3: Results of the branch-and-cut algorithm on *random inequality systems* for different cut generation strategies; all variant use method "round" as a basis. Given are the average values over all 25 instances.

type	nodes	time	root	# BaNg	# Gom.	#$\{0, \frac{1}{2}\}$
round	61593.1	915.75	6.54	0.0	0.0	0.0
BaNg	58796.4	1054.39	6.80	6134.0	0.0	0.0
Gom.	58434.7	1164.56	7.00	0.0	10440.6	0.0
$\{0, \frac{1}{2}\}$	61479.1	957.37	6.54	0.0	0.0	43.0
BaNg & Gom.	57911.9	1298.49	7.22	6955.3	10234.8	0.0
BaNg & $\{0, \frac{1}{2}\}$	60197.0	1080.89	6.78	5738.6	0.0	31.0
Gom. & $\{0, \frac{1}{2}\}$	58852.8	1158.42	7.01	0.0	10441.2	56.8
all	60092.7	1365.63	7.19	6699.5	10335.6	46.2

Table 4: Results of method "round" for random instances with $m = 80$ inequalities. Column "Opt" gives the average optimal solution values. All entries are averages over five instances.

n	nodes	time	IISs	root	opt
5	2029.4	32.26	3527.8	12.23	21.8
10	79086.6	996.51	14491.8	6.88	15.8
15	106119.0	1465.16	62151.0	4.56	11.8
20	7408.0	56.18	5743.4	2.69	5.8
25	16472.6	132.79	20884.0	2.43	6.8
∅:	42223.1	536.58	21359.6	5.76	12.4

$\{0, \frac{1}{2}\}$-cuts, improve the root dual bound with respect to the basic version. The studied methods, however, increase the CPU time needed. The main slowdown comes from the fact that the intermediate LPs become harder to solve. The corresponding separation times are acceptable, however: the average separation times for the version that uses all three methods are: 1.8% (BaNg), 17.0% (Gomory), 1.0% ($\{0, \frac{1}{2}\}$). We conclude that the basic version "round" alone is fastest on random systems.

Table 4 shows average results for method "round" on random instances with $m = 80$ inequalities. It can be observed that the optimal values of the random problems tend to decrease when increasing the dimension. This often makes the problems more tractable. But of course, the solution of the intermediate LPs over the alternative polyhedron is more time consuming.

4.3. Digital Video Broadcasting Problems

In this section we present results for problems arising in an application of Max FS in telecommunications, which is described by Rossi, Sassano, and Smriglio [54]. Here, to plan the digital video broadcasting (DVB) network of Italy, transmitters have to be placed and their emission frequency and power have to be chosen as to maximize the area coverage, subject to quality constraints. A subproblem of this can be modeled as a linear inequality system. Interference of the signals leads to areas where the digital signal cannot be received, resulting in an infeasible system. Maximizing the total weight of satisfied inequalities then amounts to maximize the area coverage.

Table 5: Results for the DVB instances in Section 4.3 with method "round". The Column labeled "[6]" lists the names of the instances as used in Amaldi et al. [6].

name	[6]	m	nodes	time	IISs	root	dual	best	gap
dvb1	dvb2	1044	503	103.6	3064	166.4	174.0	174	0.0
mfs_UHF_P4_1	dvb1	642	1	2.3	86	104.0	104.0	104	0.0
mfs_UHF_P4_3	dvb3	1717	539	599.72	5414	174.2	183.0	183	0.0
mfs_UHF_P4_4	–	1174	68049	196514.41	1002912	90.3	115.2	124	7.6

Linearizing the model leads to numerically challenging problems. The coefficients take values between 10^{-11} and 10^{11}, and the resulting LPs are very instable. We tackled the problems by scaling the original instances before starting the branch-and-cut algorithm. This helps, but nevertheless leaves hard problems. Without scaling, however, the algorithm terminated early with a completely wrong solution.

We could compute optimal solutions for the smallest instances used in Amaldi, Belotti, and Hauser [6] and Amaldi, Bruglieri, and Casale [7], see Table 5. Here, column "dual" gives the final lower bound, "best" denotes the value of the best primal solution obtained (i.e. the primal bound), and "gap" is the gap between the dual bound and primal bound in percent, computed as (best − dual)/dual · 100.0. The dimension of these instances is always 487 and the variable bounds ($0 \leq \boldsymbol{x} \leq 1$) are mandatory. We separate $\{0, \frac{1}{2}\}$-cuts every 10th level of the tree. Our primal heuristic of Section 3.4 is run every 40th level. Note that these instances can be solved faster using the "big-M" formulation (resulting in the same optimal solution values), see [6, 7].

4.4. Classification Problems

One of the historically first applications of Min IIS Cover is the design of linear classifiers, see Amaldi [4], Mangasarian [43], Bennett and Bredensteiner [19], and Rubin [55].

In this application, one is given m points $\boldsymbol{p}_1, \dots, \boldsymbol{p}_m$ in $\mathbb{R}^N$, each belonging to one of two possible *classes* P_1 and P_2, i.e., P_1 and P_2 partition the set $\{\boldsymbol{p}_1, \dots, \boldsymbol{p}_m\}$. Each of the N components of the points stores a measurement of an attribute (or feature) relevant for the concrete application. The goal is to strictly separate these points in $\mathbb{R}^N$ by an oriented hyperplane defined by $\boldsymbol{ax} \leq \beta$, with $\boldsymbol{a} \in \mathbb{R}^N$ and $\beta \in \mathbb{R}$. The points in P_1 should satisfy the inequality $\boldsymbol{ax} < \beta$ and the points in P_2 should satisfy $\boldsymbol{ax} > \beta$. Hence, we are looking for $(\boldsymbol{a}, \beta) \in \mathbb{R}^n$, with $n := N + 1$ so that the number of misclassified points

$$|\{\boldsymbol{p} \in P_1 \ : \ \boldsymbol{ap} \geq \beta\}| + |\{\boldsymbol{p} \in P_2 \ : \ \boldsymbol{ap} \leq \beta\}|$$

is minimized. This minimization is performed in order to maximize the chance that a new point can be correctly classified. Note that with this formulation points in $\{\boldsymbol{x} \ : \ \boldsymbol{ax} = \beta\}$ are counted twice (the models can be modified to eliminate this).

In the following we will discuss two equivalent ways to model this problem via Min IIS Cover and present computational results for different datasets. In the first model no bounds on the variables are present, while in the second all variables are bounded except one.

Table 6: Characteristics of the *classification instances.* Column "N" lists the number of attributes. The column labeled $m^\star$ gives the number of original data sets and m the number of data sets remaining after removing incomplete ones. The right most column gives additional notes, e.g., the name of the instance in the UCI database.

name	N	m	$m^\star$	Notes
breast-cancer	9	683	699	breast-cancer-wisconsin
bupa	6	345	345	liver-disorders
echo	8	61	132	echocardiogram
glass	9	214	214	type 2 vs. others
heart	13	297	303	heart-disease (Cleveland)
ionosphere	34	351	351	
iris.1	4	150	150	Versicolor vs. others
iris.2	4	150	150	Virginica vs. others
new-thyroid	5	215	215	normal vs. others
pima	8	768	768	Pima-indians-diabetes
tic-tac-toe	9	958	958	
wpbc	32	194	198	Wisconsin breast-cancer database

For the first model we use variables $(\boldsymbol{a}, \beta) \in \mathbb{R}^n$ and the following inequalities

$$\boldsymbol{pa} - \beta \begin{cases} < 0 & \text{if } \boldsymbol{p} \in P_1 \\ > 0 & \text{if } \boldsymbol{p} \in P_2 \end{cases} \qquad \text{for each } \boldsymbol{p} \in \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_m\}.$$

Since $(\boldsymbol{a}, \beta)$ are unbounded we can scale them to obtain

$$\boldsymbol{pa} - \beta \begin{cases} \leq -1 & \text{if } \boldsymbol{p} \in P_1 \\ \geq 1 & \text{if } \boldsymbol{p} \in P_2 \end{cases} \qquad \text{for each } \boldsymbol{p} \in \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_m\}.$$

Of course, any other positive value instead of 1 can be taken in order to obtain a numerically more stable system.

The second model is due to Rubin [55]. It uses variables $\boldsymbol{a} \in \mathbb{R}^N$ and β, $\gamma \in \mathbb{R}$ in the following system:

$$\begin{aligned} \boldsymbol{pa} - \beta + \gamma \leq 0 \quad & \text{if } \boldsymbol{p} \in P_1 \\ \boldsymbol{pa} - \beta - \gamma \geq 0 \quad & \text{if } \boldsymbol{p} \in P_2 \\ -\mathbb{1} \leq \boldsymbol{a} \leq \mathbb{1} & \\ \gamma \geq 0.001. & \end{aligned}$$

Hence, the coefficients of the normal vector $\boldsymbol{a}$ are bounded to lie within the interval $[-1, 1]$, while β is unbounded. Of course, the lower bound 0.001 for γ can be replaced by any positive number. For instances arising from this model the variable bounds are mandatory.

Note that in both models it might happen that the systems are feasible, i.e., the points are completely separable (in which case we only need to solve one linear program).

In our first test we use the first model and classification data from the UCI Repository of Machine Learning Databases (Blake and Merz [21]). The problem characteristics are given in Table 6. For some instances we had to remove incomplete data sets. A complete description of the instances is

Table 7: Results of the branch-and-cut algorithm for the *classification instances*.

name	nodes	time	IISs	root	dual	best	gap	Chi
breast-cancer	313	2.88	359	7.2	11.0	11	0.0	11
bupa	9669	18000.11	179562	43.2	59.6	83	39.3	83
echo	2	0.05	89	6.0	6.0	6	0.0	6
glass	36859	18000.00	99833	18.5	32.7	36	10.0	41
heart	51274	18000.02	122000	12.8	23.5	29	23.6	30
ionosphere	2465	38.59	3967	2.4	6.0	6	0.0	6
iris.1	845	12.45	623	19.1	25.0	25	0.0	25
iris.2	1	0.01	2	0.0	1.0	1	0.0	1
new-thyroid	2	0.09	147	11.0	11.0	11	0.0	11
pima	1522	18000.18	64166	68.2	75.6	148	95.7	148
tic-tac-toe	50691	5167.03	19850	60.9	86.0	86	0.0	93
wpbc	56657	18000.00	739494	3.5	8.7	13	48.7	13

available at the UCI Repository. Most of these twelve instances are also used by Chinneck [31] for testing his heuristic for MAX FS/MIN IIS COVER.

Table 7 lists the results of the branch-and-cut implementation on these instances with method “round” of Section 3.2.3. The computation time was limited to *five hours* (18000 sec.). The columns have the same meaning as in Sections 4.2 and 4.3.

Column “Chi” gives results obtained by the heuristic of Chinneck, see Section 3.4; its running times are negligible and therefore not listed. Our implementation found the same solutions as Chinneck [31], except for the instances `glass` and `wpbc`, for which Chinneck obtained solutions of size 39 and 10, respectively. Our primal heuristic described in Section 3.4 is run every tenth level. It could improve the initial solutions for models `glass`, `heart`, and `tic-tac-toe`. We conclude that the heuristic of Chinneck generates very good starting solutions, while our primal heuristic sometimes helps to find better solutions.

The results of Table 7 show that most instances are quite hard to solve and about half of them could not be solved within the time bound of five hours. Because of their size, only few nodes could be processed.

We also conducted experiments with the same data but, using the second model instead of the first. Intuitively this should result in better numerical properties of the LPs that have to be solved during the algorithm. The results are, however, comparable to the ones shown in Table 7, and we therefore do not present them here.

Table 8 compares the gaps of the different cut strategies. The table only displays instances for which the optimal solutions could not be found within five hours. It turns out that all variants find the same final primal solutions, although at different times during the computation. Note that this actually compares the interplay of cutting strategies and our primal heuristic. On the average, the smallest gaps are produced by taking Gomory cuts, then method “round”, Gomory and $\{0, \frac{1}{2}\}$-cuts, $\{0, \frac{1}{2}\}$-cuts alone, Balas/Ng cuts, Balas/Ng cuts and Gomory cuts, Balas/Ng cuts and $\{0, \frac{1}{2}\}$-cuts, and finally all cuts together. The main reason why all cuts together produce the worst results (on average) is that this combination could explore the fewest number

Table 8: *Classification problems*: Comparison of the *gaps* of different variants of cutting planes. Only instances for which a positive gap after five hours remains are shown. The notation is as in Table 3. The last line contains the averages over each column.

name	round	BaNg	Gom.	$\{0, \frac{1}{2}\}$	BaNg Gom.	BaNg $\{0, \frac{1}{2}\}$	Gom. $\{0, \frac{1}{2}\}$	all
bupa	39.3	46.7	40.0	41.9	44.0	45.0	41.5	45.5
glass	10.0	12.7	10.0	10.6	12.2	12.7	9.8	12.4
heart	23.6	23.8	22.6	24.2	25.4	25.2	27.8	26.0
pima	95.7	101.8	95.0	98.6	103.2	101.4	94.1	105.4
wpbc	48.7	47.8	44.6	49.8	49.0	49.0	45.6	50.5
∅:	43.5	46.6	42.4	45.0	46.7	46.7	43.8	48.0

of nodes. We conclude that the additional cutting planes do not yield a big improvement over method "round" alone. Although Gomory cuts produce the smallest gaps, the studied cutting planes do not seem to be crucial to solve these instances.

Our second test set consists of data from Codato and Fischetti [33] and uses the second model. The data again originate from the UCI Repository of Machine Learning Databases, but are preprocessed in way we could not reconstruct. Hence, the results for these instances and the instances of Table 6 may not be comparable (there are three instances which seem to arise from the same original data: `breast-cancer` ↔ `breast-cancer-2`, `iris.1` ↔ `iris-150`, `wpbc` ↔ `WPBC194`). Instances `Breast-Cancer-2` and `Breast-Cancer-400` seem to be different to the ones used in Codato and Fischetti [33].

Table 9 shows the results of method "round" on these instances. The notation is as in Table 3. Note that here the dimension is $n = N + 2$, because we use the second model. Most of the instances could be solved within a few seconds. This is the first time that the complete set could be solved to optimality: no optimal solution to the harder instances (`Flags-169`, `Horse-colic-185`, `Horse-colic-253`, and `Solar-flare-1066`) was previously available. Our implementation solves all instances except these four in under a minute. Although we worked on a faster computer, it seems therefore fair to say that our code considerably improves upon the results of Codato and Fischetti [33].

5. Conclusions

In this paper we described a branch-and-cut implementation for the Max FS/Min IIS Cover problem, which is the best exact method currently available. The findings of the extensive computational results can roughly be summarized as follows: With respect to the implementation, the best cutting plane strategy is to find as many (violated) IIS-inequalities as possible. Additionally applying Balas/Ng, Gomory, or $\{0, \frac{1}{2}\}$-cuts does not significantly help to improve the performance: On random instances they do not improve the running time, but usually help to reduce the number of nodes. Gomory cuts only slightly help to reduce the gaps for classification instances and the other cuts do not improve the gap.

Table 9: *Classification Problems*: Results of the branch-and-cut algorithm for the problems of Codato and Fischetti with method “round”.

name	n	m	nodes	time	IISs	root	opt
Balloons-76	7	76	1	0.02	59	10.0	10
BCW-367	12	367	110	0.97	252	5.5	8
BCW-683	12	683	71	1.70	235	6.8	10
Breast-Cancer-2	11	683	352	2.21	322	7.0	11
Breast-Cancer-400	20	400	2	0.08	116	24.0	24
Bridges-132	14	132	299	3.44	1563	20.2	23
BusVan-437	20	437	237	1.72	353	3.0	6
BusVan-445	20	445	605	5.53	750	3.3	8
BusVan-447	20	447	2334	37.65	4187	4.4	10
BV-OS-282	20	282	214	1.39	338	3.0	6
BV-OS-376	20	376	969	12.03	1361	4.2	9
Chorales-107	8	107	951	9.57	1187	21.4	27
Chorales-116	8	116	1022	19.85	1981	17.2	24
Chorales-134	8	134	1198	50.99	4008	20.8	30
Credit-300	17	300	13	0.93	222	5.9	8
Flag-169	31	169	7621	209.63	17276	3.5	9
Glass-163	12	163	15	0.64	158	10.9	13
Horse-Colic-151	28	151	231	2.25	540	2.2	5
Horse-Colic-185	28	183	69155	886.10	61414	3.6	10
Horse-Colic-253	28	253	273389	7938.84	308862	4.8	13
House-Votes84-435	18	435	56	0.68	200	4.0	6
Iris-150	7	150	1017	6.58	1011	11.7	18
Lymphography-142	20	142	21	0.24	131	2.9	5
Mech-analysis-107	10	107	1	0.04	83	7.0	7
Mech-analysis-137	9	137	757	5.83	890	11.6	18
Mech-analysis-152	10	152	900	32.05	3042	13.0	21
Monks-tr-115	8	115	917	16.24	1570	20.9	27
Monks-tr-122	8	122	4	0.45	267	11.2	13
Monks-tr-124	8	124	489	5.91	1187	18.1	24
Opel-Saab-76	20	76	1111	9.28	1756	2.9	7
Opel-Saab-80	20	80	241	2.01	512	3.0	6
Opel-Saab-83	20	83	2113	25.05	3904	3.2	8
Opel-Saab-84	20	84	572	7.06	1318	3.3	7
Pb-gr-txt-198	12	198	147	1.09	267	7.7	11
Pb-hl-pict-277	12	277	178	1.61	314	6.7	10
Pb-pict-txt-444	12	444	2	0.12	79	7.0	7
Postoperative-88	10	88	1	0.12	209	16.0	16
Solar-flare-323	14	323	3	0.71	478	37.2	38
Solar-flare-1066	14	1066	2292	787.64	14960	227.3	243
Water-treat-206	40	206	41	1.43	204	1.7	4
Water-treat-213	40	213	288	8.04	845	2.2	5
WPBC-194	36	194	172	3.21	468	2.2	5

With respect to the problem data, the considered instances vary highly in their properties and difficulty. Depending on the particular data, quite large instances can be solved to optimality, but there are also relatively small instances which turn out to be extremely hard to solve. As shown by the DVB problems, one has to be careful with numerically instable instances.

An interesting open issue is the existence of problem specific cutting planes and whether they can be efficiently separated. Another question is whether other valid inequalities for the set covering problem could be helpful to improve the performance of the implementation.

Acknowledgments

The author thanks Tobias Achterberg for help with the SCIP implementation and Edoardo Amaldi and Les Trotter for helpful discussions. Furthermore, he thanks Edoardo Amaldi and Pietro Belotti for providing the DVB instances of Section 4.3, and Gianni Codato and Matteo Fischetti for the data used in Section 4.4.

References

[1] T. Achterberg, *SCIP – A framework to integrate constraint and mixed integer programming*, Report 04-19, Zuse Institute Berlin, 2004. Available online at http://www.zib.de/Publications/abstracts/ZR-04-19/.

[2] T. Achterberg, T. Koch, and A. Martin, *Branching rules revisited*, Oper. Res. Lett. **33**, no. 1 (2005), pp. 42–54.

[3] S. Agmon, *The relaxation method for linear inequalities*, Can. J. Math. **6** (1954), pp. 382–392.

[4] E. Amaldi, *From Finding Maximum Feasible Subsystems of Linear Systems to Feedforward Neural Network Design*, PhD thesis, EPF-Lausanne, 1994.

[5] E. Amaldi, *The maximum feasible subsystem problem and some applications*, in Modelli e Algoritmi per l'Ottimizzazione di Sistemi Complessi, A. Agnetis and G. D. Pillo, eds., Pitagora Editrice, Bologna, 2003, pp. 31–69.

[6] E. Amaldi, P. Belotti, and R. Hauser, *Randomized relaxation methods for the maximum feasible subsystem problem*, in Proc. 11th International Conference on Integer Programming and Combinatorial Optimization (IPCO), Berlin, M. Jünger and V. Kaibel, eds., LNCS 3509, Springer-Verlag, Berlin Heidelberg, 2005, pp. 249–264.

[7] E. Amaldi, M. Bruglieri, and G. Casale, *A two-phase relaxation-based heuristic for the maximum feasible subsystem problem*, Computers and Operations Research (2007). To appear.

[8] E. Amaldi and R. Hauser, *Boundedness theorems for the relaxation method*, Math. Oper. Res. **30**, no. 4 (2005), pp. 1–17.

[9] E. Amaldi and V. Kann, *The complexity and approximability of finding maximum feasible subsystems of linear relations*, Theor. Comput. Sci. **147**, no. 1–2 (1995), pp. 181–210.

[10] E. Amaldi and V. Kann, *On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems*, Theor. Comput. Sci. **209**, no. 1–2 (1998), pp. 237–260.

[11] E. Amaldi and M. E. Pfetsch, *Separation problems for set covering*, 2005. Manuscript.

[12] E. Amaldi, M. E. Pfetsch, and L. E. Trotter, Jr., *On the maximum feasible subsystem problem, IISs, and IIS-hypergraphs*, Math. Program. **95**, no. 3 (2003), pp. 533–554.

[13] G. Andreello, A. Caprara, and M. Fischetti, *Embedding cuts in a branch&cut framework: a computational study with* $\{0,\frac{1}{2}\}$*-cuts*, INFORMS J. Comput. **19**, no. 2 (2007), pp. 229–238.

[14] D. L. Applegate, W. Cook, S. Dash, and D. G. Espinoza, *Exact solutions to linear programming problems*, Oper. Res. Lett. (2007). To appear.

[15] D. Avis, lrs *home page.* Available at: http://cgm.cs.mcgill.ca/~avis/C/lrs.html.

[16] D. Avis and K. Fukuda, *Reverse search for enumeration*, Discrete Appl. Math. **65**, no. 1–3 (1996), pp. 21–46.

[17] E. Balas, S. Ceria, G. Cornuéjols, and N. Natraj, *Gomory cuts revisited*, Oper. Res. Lett. **19**, no. 1 (1996), pp. 1–9.

[18] E. Balas and S. M. Ng, *On the set covering polytope: I. All the facets with coefficients in* $\{0, 1, 2\}$, Math. Program. **43**, no. 1 (1989), pp. 57–69.

[19] K. P. Bennett and E. J. Bredensteiner, *A parametric optimization method for machine learning*, INFORMS J. Comput. **9**, no. 3 (1997), pp. 311–318.

[20] K. P. Bennett and O. L. Mangasarian, *Neural network training via linear programming*, in Advances in optimization and parallel computing, P. M. Pardalos, ed., North-Holland, Amsterdam, 1992, pp. 56–67.

[21] C. L. Blake and C. J. Merz, *UCI repository of machine learning databases*, 1998. Available at http://www.ics.uci.edu/~mlearn/MLRepository.html.

[22] R. Borndörfer, *Aspects of Set Packing, Partitioning, and Covering*, PhD thesis, TU Berlin, 1998.

[23] R. Borndörfer and R. Weismantel, *Set packing relaxations of some integer programs*, Math. Program. **88** (2000), pp. 425–450.

[24] R. Borndörfer and R. Weismantel, *Discrete relaxations of combinatorial programs*, Discrete Appl. Math. **112**, no. 1–3 (2001), pp. 11–26.

[25] A. Caprara and M. Fischetti, $\{0, \frac{1}{2}\}$*-Chvátal-Gomory cuts*, Math. Prog. **74**, no. 3 (1996), pp. 221–235.

[26] A. Caprara and M. Fischetti, *Branch-and-cut algorithms*, in Annotated Bibliographies in Combinatorial Optimization, M. Dell'Amico, F. Maffioli, and S. Martello, eds., John Wiley & Sons, Chichester, 1997, ch. 4, pp. 45–63.

[27] S. Ceria, P. Nobili, and A. Sassano, *Set covering problem*, in Annotated Bibliographies in Combinatorial Optimization, M. Dell'Amico, F. Maffioli, and S. Martello, eds., John Wiley & Sons, Chichester, 1997, ch. 23, pp. 415–428.

[28] N. Chakravarti, *Some results concerning post-infeasibility analysis*, Eur. J. Oper. Res. **73** (1994), pp. 139–143.

[29] J. W. Chinneck, *An effective polynomial-time heuristic for the minimum-cardinality IIS set-covering problem*, Ann. Math. Artif. Intell. **17**, no. 1–2 (1996), pp. 127–144.

[30] J. W. Chinneck, *Finding a useful subset of constraints for analysis in an infeasible linear program*, INFORMS J. Comput. **9**, no. 2 (1997), pp. 164–174.

[31] J. W. Chinneck, *Fast heuristics for the maximum feasible subsystem problem*, INFORMS J. Comput. **13**, no. 3 (2001), pp. 210–223.

[32] J. W. Chinneck and E. W. Dravnieks, *Locating minimal infeasible constraint sets in linear programs*, ORSA J. Comput. **3**, no. 2 (1991), pp. 157–168.

[33] G. Codato and M. Fischetti, *Combinatorial Benders' cuts*, in Proc. 10th International Conference on Integer Programming and Combinatorial Optimization (IPCO), New York, D. Bienstock and G. Nemhauser, eds., LNCS 3064, Springer-Verlag, Berlin Heidelberg, 2004, pp. 178–195.

[34] K. Fukuda, cdd *home page.* Available at: http://www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html.

[35] J. Gleeson and J. Ryan, *Identifying minimally infeasible subsystems of inequalities*, ORSA J. Comput. **2**, no. 1 (1990), pp. 61–63.

[36] H. J. Greenberg and F. H. Murphy, *Approaches to diagnosing infeasible linear programs*, ORSA J. Comput. **3**, no. 3 (1991), pp. 253–261.

[37] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Algorithms and Combinatorics 2, Springer-Verlag, Heidelberg, 2nd ed., 1993.

[38] P. Hansen, M. Labbé, and D. Schindl, *Set covering and packing formulations of graph coloring: algorithms and first polyhedral results*, tech. report, GERAD, 2005.

[39] D. S. Johnson and F. P. Preparata, *The densest hemisphere problem*, Theor. Comput. Sci. **6** (1978), pp. 93–107.

[40] L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, and V. Gurvich, *Generating all vertices of a polyhedron is hard*, in Proc. of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, ACM Press, 2006, pp. 758–765.

[41] T. Koch, *The final* Netlib-*LP results*, Oper. Res. Lett. **32**, no. 2 (2004), pp. 138–142.

[42] A. N. Letchford and A. Lodi, *Strengthening Chvátal-Gomory cuts and Gomory fractional cuts*, Oper. Res. Lett. **30**, no. 2 (2002), pp. 74–82.

[43] O. L. Mangasarian, *Misclassification minimization*, J. Glob. Optim. **5**, no. 4 (1994), pp. 309–323.

[44] H. Marchand and L. Wolsey, *Aggregation and mixed integer rounding to solve mips*, Operations Research **49**, no. 3 (2001), pp. 363–371.

[45] Maximum Feasible Subsystem Home Page. Available online at: http://www.elet.polimi.it/res/maxfs/.

[46] T. S. Motzkin, *The probability of solvability of linear inequalities*, in Selected papers, D. Cantor, B. Gordon, and B. Rothschild, eds., Contemporary Mathematicians, Birkhäuser, Boston Basel Stuttgart, 1983, pp. 116–120.

[47] T. S. Motzkin and I. J. Schoenberg, *The relaxation method for linear inequalities*, Can. J. Math. **6** (1954), pp. 393–404.

[48] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, 1988.

[49] Netlib Repository. available at http://www.netlib.org.

[50] M. Padberg and G. Rinaldi, *A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems*, SIAM Rev. **33**, no. 1 (1991), pp. 60–100.

[51] M. Parker, *A Set Covering Approach to Infeasibility Analysis of Linear Programming Problems and Related Issues*, PhD thesis, University of Colorado at Denver, 1995.

[52] M. Parker and J. Ryan, *Finding the minimum weight IIS cover of an infeasible system of linear inequalities*, Ann. Math. Artif. Intell. **17**, no. 1–2 (1996), pp. 107–126.

[53] M. E. Pfetsch, *The Maximum Feasible Subsystem Problem and Vertex-Facet Incidence of Polyhedra*, PhD thesis, TU Berlin, 2002.

[54] F. Rossi, A. Sassano, and S. Smriglio, *Models and algorithms for terrestrial digital broadcasting*, Annals of Operations Research **107** (2001), pp. 267–283.

[55] Rubin, *Solving mixed integer classification problems by decomposition*, Ann. Oper. Res. **74** (1997), pp. 51–64.

[56] D. M. Ryan and B. A. Foster, *An integer programming approach to scheduling*, in Computer scheduling of public transport: Urban passenger vehicle and crew scheduling, A. Wren, ed., North-Holland, Amsterdam, 1981.

[57] J. Ryan, *Transversals of IIS-hypergraphs*, in Proc. 22nd Southeast Conf. on Combinatorics, Graph Theory, and Computing, Baton Rouge, Congr. Numerantium 81, 1991, pp. 17–22.

[58] J. K. Sankaran, *A note on resolving infeasibility in linear programs by constraint relaxation*, Oper. Res. Letters **13** (1993), pp. 19–20.

[59] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, Chichester, 1986.

[60] S. Thienel, *ABACUS – A Branch-And-CUt System*, PhD thesis, Universität zu Köln, 1995.

[61] M. Wagner, J. Meller, and R. Elber, *Solving huge linear programming problems for the design of protein folding potentials*, Math. Program. **101** (2004), pp. 301–318.