



HAL
open science

Some contributions to mathematical programming and combinatorial optimization

José Neto

► **To cite this version:**

José Neto. Some contributions to mathematical programming and combinatorial optimization. Computer Science [cs]. Université Evry-Val d'Essonne/Université Paris Saclay, 2019. tel-04261549

HAL Id: tel-04261549

<https://hal.science/tel-04261549>

Submitted on 27 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License



Université d'Evry Val d'Essonne

Habilitation à diriger des recherches

Spécialité : informatique

présentée et soutenue publiquement par

José NETO

le 18 juin 2019

Some contributions to mathematical programming and combinatorial optimization

Rapporteurs

BAIOU Mourad, CNRS, Université Clermont II
FORTZ Bernard, Université Libre de Bruxelles
LISSER Abdel, Université Paris Sud

Jury

ANGEL Eric, Université d'Evry
BEN-AMEUR Walid, Télécom SudParis
CORNAZ Denis, Université Paris Dauphine
MEUNIER Frédéric, Ecole Nationale des Ponts et Chaussées
PICOULEAU Christophe, Conservatoire National des Arts et Métiers

Acknowledgments

I wish to thank all my family and friends for their constant care and support.

More specially, relating to this work, I am much indebted to all the teachers and professors I could meet for their education. Among them, and with respect to the subject of this manuscript, I am particularly grateful to Philippe Mahey for having introduced to me convex analysis and optimization, and to Ridha Mahjoub who introduced to me polyhedral combinatorics and combinatorial optimization.

Naturally, I wish thank Walid Ben-Ameur for having led me to learn about many advanced topics since a very long time ago; for having shared so many ideas and works, for his constant kindness, support and patience.

Some investigations gave me the opportunity to meet people abroad; exchanging and developing ideas with them has been really stimulating; among them namely, Miguel Anjos, Andreas Bley, Michele Conforti, Mauro Dell'Amico, Claudio Gentile, Brigitte Jaumard, Giovanni Rinaldi, Paolo Ventura, Roland Wessäly.

I am also grateful to all my colleagues from the RS2M Department at Télécom SudParis, and in particular to its Director, Djamel Zeghlache for his undefectible confidence and support.

Last but not least, there are many people who have been carrying out many tasks and helpful support work all along the way: Corinne Le Roux, Brigitte Arousseau, Christine Jost and Arnaud Luiz - a great team - at the library from Télécom SudParis, and also Valérie Mateus from the secretary of the RS2M Department. Thank you all very much for your constant helpfulness and kindness.

Short content overview

Consider the optimization (i.e. maximization or minimization) of a real valued function f , defined on a subset of \mathbb{R}^n , subject to some set of constraints on the variables. The present document gathers and summarizes part of the work I could carry out on such optimization problems when the objective function is either linear or quadratic and when the constraints set on the variables are linear and/or integrity constraints.

In the first part, methods are presented for dealing with general families of mathematical programs: (mixed-integer) linear programs and unconstrained binary quadratic programs, namely. Chapter 1 is concerned with different types of cutting-plane methods for (mixed-integer) linear programs. In Chapter 2 we present a combinatorial algorithm to solve a particular family of unconstrained binary quadratic programs in polynomial time. General unconstrained quadratic binary programs are then considered in Chapter 3, for which we show how information related to the spectrum of the matrix defining the objective may be used to derive bounds on the optimal objective value.

In the second part, we present our contributions w.r.t. several classical combinatorial optimization problems. They namely contain studies on descriptions of polytopes related to the maximum cut problem when the sizes of the shores are fixed (Chapter 4). Given an edge weighted graph, this problem consists in identifying a node subset S of prescribed size, such that the sum of the weights of the edges having exactly one endpoint in S is maximized. Another very classical problem in combinatorial optimization is the one which consists in identifying a minimum cardinality dominating set. Recall that given a graph, a dominating set is a node subset S such that each node not in S has at least one neighbor in S . Polyhedral investigations have been carried out w.r.t. the weighted version of the problem and for different generalizations of the domination concept (Chapter 5). A graph partitioning problem called the *k-separator problem* is then dealt with from different perspectives (complexity, approximations and polyhedral descriptions) in Chapter 6. Basically it consists in identifying a node subset in a graph whose removal induces a graph having connected components of bounded size. Finally, in Chapter 7, we present several features of an original orientation problem in graphs, including the identification of polynomial time solvable cases, several formulations and connections with the maximum cut problem.

Contents

Contents	v
I Some developments in mathematical programming	1
1 Cutting-planes in linear programming	3
1.1 Dealing with large size LPs via cutting-plane algorithms	3
1.2 The <i>In/Out</i> algorithm: a new mixed cutting-plane scheme	6
1.3 A multiple-points cutting-plane scheme	10
1.4 A simple finite cutting-plane algorithm for (mixed-) integer LPs	18
1.5 Some perspectives	21
2 A recursive algorithm for some unconstrained $(-1, 1)$-quadratic optimization problems	23
2.1 Introduction	23
2.2 Properties of optimal solutions for particular instances of (QP)	24
2.3 Determining cells in an arrangement of n hyperplanes	25
2.4 Polynomially solvable cases for problem (QP)	29
2.5 Some perspectives	31
3 Spectral bounds for unconstrained $(-1, 1)$-quadratic optimization problems	33
3.1 Introduction	33
3.2 Computing bounds for problem (QP)	34
3.3 Improvement over a classical semidefinite bound	38
3.4 Applications to the weighted maximum cut problem	39
3.5 Further extensions for the weighted maximum k -cut problem	41
3.6 Some elements on computational experiments	44
3.7 Some perspectives	44
II Graph optimization problems and related polyhedral structures	47
4 Uniform cut polytopes	49
4.1 Introduction	49
4.2 Properties of facet defining inequalities and dimension	51
4.3 A walk through some families of facet-defining inequalities	52
4.4 On the diameter of cut polytopes	56
4.5 Some perspectives	59
5 Generalized domination concepts	61
5.1 Introduction	61
5.2 On the polyhedral structure of \mathcal{T}_G^f	64
5.3 On the polyhedral structure of \mathcal{D}_G^f	66
5.4 On the polyhedral structure of \mathcal{W}_G^f	67
5.5 Some perspectives	70

6	The k-separator problem	73
6.1	Introduction	73
6.2	Polynomially solvable cases	75
6.3	Polyhedral results	78
6.4	Approximations	81
6.5	Some perspectives	83
7	Orientation problems and connections to $max-cut$	85
7.1	Introduction and notation	85
7.2	A sketch of new complexity results on MAXIM	86
7.3	MIP formulations for MAXIM	87
7.4	New formulations for unweighted $max-cut$	90
7.5	Some perspectives	92
A	Curriculum vitae	93
B	Selected publications	101

Part I

Some developments in mathematical programming

Chapter 1

Cutting-planes in linear programming

Nowadays, cutting-plane methods constitute a classical approach when dealing with *difficult* (possibly mixed integer) linear programs (LPs). The term *difficult* employed here may refer to different kinds of situations. A first situation is when a complete description of the set of feasible solutions is known, but its size (in terms of the number of variables and/or constraints involved) is too important to be given directly as input to an LP solver (due to huge memory requirements and/or computation times). Another classical situation (not exclusive from the former) is when the feasible region is not known explicitly but some procedure is known that is able to generate inequalities that are satisfied by all the feasible solutions (details are given hereafter).

In this chapter, we give a synthesis of some of our works that could lead to improvements over classical cutting plane schemes and/or provide some additional insights on existing ones. After we describe the setting and briefly review cutting plane methods from the literature in Section 1.1, we introduce original cutting plane schemes in the following three sections.

Sections 1.2 and 1.3 are a joint work with Prof. Ben-Ameur.

1.1 Dealing with large size LPs via cutting-plane algorithms

In this introductory section, we briefly describe the overall setting and review the main types of cutting-plane algorithms present in the literature: *boundary point methods*, *interior point methods*, *center point methods* and *mixed methods*. Throughout this and the next chapters, the reader is assumed to be familiar with linear programming, see, e.g., [66, 181, 194, 205].

A linear program consists of a linear objective function to be maximized or minimized, subject to linear constraints. It may be expressed as follows.

$$(PB) \begin{cases} \max & c^T x \\ s.t. & x \in S, \end{cases}$$

with $c \in \mathbb{R}^n$, n is a positive integer, x denotes the vector of variables and S is a polyhedron representing the set of all feasible solutions (also called the *feasible region*): $S = \{x \in \mathbb{R}^n : Ax \leq b\}$, where $A \in \mathbb{R}^{m \times n}$, for some positive integer m .

LPs provide a suitable framework for modeling a wide diversity of optimization problems stemming from many applications. Difficulties for solving formulations such as (PB) may arise, e.g., when the description of S comprises a huge number of constraints (and/or variables), or (not exclusively) when the set S is known implicitly by means of an *exact separation oracle* [121], that is, a procedure solving the following *exact separation problem*.

Definition. *Exact separation problem.*

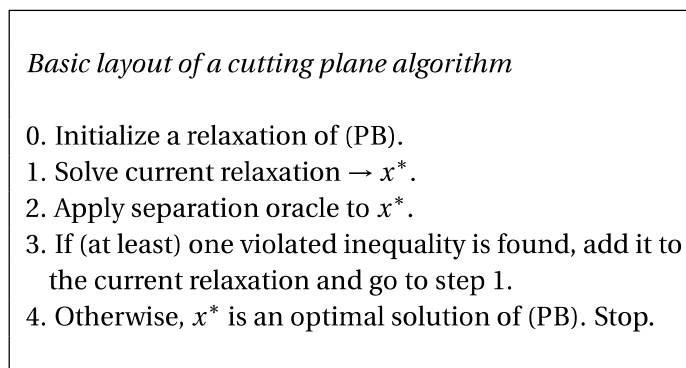
Given a vector $y \in \mathbb{R}^n$, determine whether $y \in S$ and, if not, find a vector $(a, b) \in \mathbb{R}^{n+1}$ such that the inequality $a^T x \leq b$ is valid for S and violated by y (i.e. $a^T y > b$).

In fact - as is the case with NP-hard problems [111] - another common situation is when a complete formulation of S is not even known, but only some restricted set of inequalities defining a relaxation of (PB), i.e. a linear program differing from (PB) in the feasible region that is enlarged to some set $S' \supseteq S$.

In this chapter and for our purposes, we shall only consider feasible regions described by exact separation oracles. (Note that, trivially, if a complete formulation of S is known, the situation can be reduced to that of an implicit description by means of an exact separation oracle.)

Presently, a common way of dealing with difficult instances of LPs as described before, rests on the iterative resolution of a relaxation that is enriched with inequalities stemming from a separation oracle. The basic functioning of a cutting plane algorithm may be summarized as follows (see also Figure 1.1). Iteratively, a linear relaxation of problem (PB) is solved and a separation oracle is applied to the solution found. Then, two cases follow. If a violated inequality is returned by the oracle, it is added to the current linear relaxation that will be solved anew and the whole process is reiterated. Otherwise, the solution found is feasible, and thus optimal for the original problem, so that the resolution stops.

Figure 1.1 – Description of a cutting plane algorithm



In fact, as we shall see next, more involved procedures -still making use of an exact separation algorithm- appear in the literature. We review them shortly, classifying them into four families depending on the nature of the point that is given as input to the separation oracle.

Boundary point methods

In boundary point cutting-plane algorithms, the point that is given as input to the separation oracle is an optimal solution of the current linear relaxation, that is located on the boundary of its feasible region (or, more precisely, at an extreme point, if one makes use of a simplex type algorithm). Early works fitting into this framework notably include Dantzig, Fulkerson and Johnson's algorithm for the TSP (Traveling Salesman Problem) [76], Chvátal's [64] and Gomory's algorithm [118] for integer linear programs, Benders' decomposition [37, 112] for mixed problems, Kelley's [141] or Cheney and Goldstein's [58] methods for more general convex problems.

Since then, many works have appeared aiming at improving the convergence of the former methods. These notably include Magnanti and Wong's method [161] for Bender's decomposition, stabilization techniques [153, 217] for convex problems, among which bundle algorithms [107].

Interior point methods

Interior point cutting plane algorithms give as input to the separation oracle a solution of the current relaxation, provided by an interior point algorithm (see, e.g., [160]). So, this point is interior to the feasible region of the current relaxation. Though experimentally interior point algorithms seem rather limited with respect to the size of the problems they can deal with (by comparison with the simplex algorithm), they have been very important for establishing fundamental theoretical

results (such as the polynomial-time equivalence between optimization and separation [121]). For more information on the use of interior point methods within cutting-plane algorithms, see, e.g., [170–174].

Center point methods

The first cutting plane methods involving the notion of center seem to have been proposed by Levin [154], Elzinga and Moore [96] (see also Betrò [41] for some improvements of the latter). Center point methods substantially differ from the former (boundary/interior point methods) in the following aspects:

- the problem that is solved iteratively is not a relaxation of the original problem. It rather consists in minimizing a convex potential function over a convex feasible region containing the intersection of S with a set of the form $\{x \in \mathbb{R}^n : c^T x \leq V\}$ for some value $V \in \mathbb{R}$, i.e. the set of points satisfying some inequality w.r.t. the objective value.
- the result from the separation process with the center (the solution of the problem involving the potential function mentioned above) as input is either an *optimality cut* (i.e. a constraint with respect to its objective value) for the case that the center belongs to S , or a *feasibility cut* (i.e. a linear inequality valid for S and violated by the center) otherwise.

So, w.r.t. the general presentation of cutting plane methods given earlier (Figure 1.1 and before), S is to be interpreted here as the set of optimal solutions of the original problem. Note that the ellipsoid method by Khachiyan [142], which is the first polynomial time algorithm known to solve LPs, belongs to this family of methods. For further information and a state of the art on analytical center methods, the reader may consult, e.g., [115, 220].

Mixed methods

The last family of cutting plane methods we consider and that we will call *mixed* methods, consists intuitively in combining somehow the former approaches by making use of an interior point and a boundary point in the separation process. Such an approach appears in a work due to Veinott [214] for solving convex programs, making use of linear relaxations. Assuming a point x_{in} in the interior of the feasible region is known, and that x_{out} is an optimal solution of the current relaxation, a point x_{bd} which belongs to the line segment $[x_{in}, x_{out}]$ and to (or is close to) the boundary of the feasible region is determined (see Figure 1.2). The generated inequality (denoted by C in Figure 1.2) then corresponds to (or is somehow close to) a supporting hyperplane of S at x_{bd} . A drawback of this methods lies in the computational work needed to compute x_{bd} .

Before we conclude this section let us mention that, by adopting a *dual perspective*, the former approaches may be useful in the context of column (rather than constraint) generation in order to solve LPs involving a large number of columns. For a more in-depth discussion on these two frameworks and their connections, see [33].

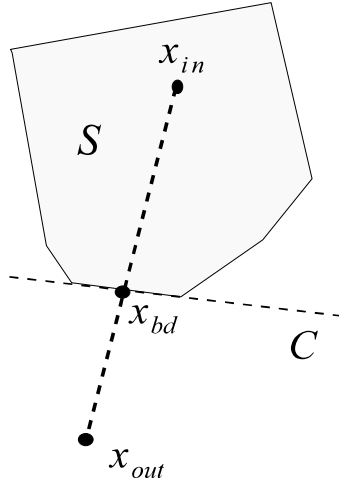


Figure 1.2 – Illustration of Veinott's method

1.2 The *In/Out* algorithm: a new mixed cutting-plane scheme

In this section we present the *In/Out* algorithm: a new cutting-plane method fitting into the family of the *mixed methods* introduced earlier. After we describe the method and present some of its properties, we summarize results from our computational experiments. Details about the work from this section can be found in [33].

1.2.1 Description

Consider the problem (PB) introduced in Section 1.1: $\max\{c^T x : x \in S\}$, and assume an exact separation oracle is known for the set S . The *In/Out* algorithm, which may be interpreted as a generalization of Veinott's method [214] proceeds as follows for the separation step. Assume that the current relaxation of (PB) that is used is given by $\max\{c^T x : x \in P\}$, with $P \supseteq S$. Three points are considered:

- $x_{in} \in S$: a feasible solution of the original problem (PB) (differently from Veinott's method, this point may change in the course of the algorithm),
- $x_{out} \in P$: an optimal solution of the current relaxation, and
- x_{sep} : a point which is a convex combination of x_{in} and x_{out} :

$$x_{sep} = \alpha x_{out} + (1 - \alpha)x_{in},$$

with $\alpha \in]0, 1]$, a parameter whose value may change in the course of the algorithm.

It is the point x_{sep} that is given as input to the separation oracle. This point will also be called *separation point* in what follows. Then, there are two cases left in the separation process:

1. Case $x_{sep} \notin S$: a violated inequality is returned by the separation oracle. This inequality is added to the current relaxation.
2. Case $x_{sep} \in S$: x_{in} is updated with x_{sep} .

The iterations are stopped when the relative gap between the objective values of x_{in} and x_{out} falls below some given precision parameter value ϵ . Algorithm 1 hereafter describes the overall procedure.

Algorithm 1. *In/Out* cutting-plane algorithm

Step 0.

- (a) Define a polyhedral relaxation $P \supseteq S$.
- (b) Find $x_{in} \in S$.
- (c) Compute x_{out} such that $c^T x_{out} = \max\{c^T x : x \in P\}$.

Step 1.

- (a) Select $\alpha \in]0, 1]$.
- (b) Set $x_{sep} := \alpha x_{out} + (1 - \alpha)x_{in}$.
- (c) Give x_{sep} as input for the separation oracle.
- (d) If a violated inequality is found, then
 - Add a violated inequality to the description of P .
 - Compute x_{out} such that $c^T x_{out} = \max\{c^T x : x \in P\}$.
- (e) Else
 - Set $x_{in} := x_{sep}$.

End_If;

Step 2.

If $|c^T x_{in} - c^T x_{out}| \leq \epsilon |c^T x_{in}|$ then
 Stop.

Else

Go to 'Step 1'.

End_If;

END.

Note the flexibility of the algorithm w.r.t. the method used to compute x_{out} (in the steps 0.c and 1.d). In fact (see [33] for induced modifications in the rest of the algorithm) we may allow x_{out} to be an approximate solution or to be obtained by an interior point method. These features may be potentially attractive w.r.t. computation times.

The *In/Out* algorithm may also be adapted for the case when an exact separation oracle is not available. In such a case the point x_{in} should not be modified when no violated inequality is found: rather, the value of α should be increased to compute another separation point. The stopping criterion should also be changed to handle the inexactness of the separation oracle.

1.2.2 Some convergence properties

Convergence properties of the *In/Out* algorithm have been studied in a context more general than linear programming: convex optimization. To present some of these results, we may assume in this section that the original problem to be solved is of the form (PB) with S a convex set (in lieu of a polyhedron) for which an exact separation oracle is known. (Definition of an exact separation oracle for a convex set is the same as the one given in Section 1.1 for a polyhedron).

Assume that the number of constraints that can be generated is finite. Then, since any constraint cannot be generated twice, the number of iterations for which an inequality is generated is finite. In order to conclude on the convergence of the *In/Out* algorithm, we must also consider the iterations for which no inequality is generated (or equivalently, when the point x_{in} is modified). Let $\alpha_k \in]0, 1]$ denote the value of the parameter α at iteration k . Assume the convex optimization problem to be solved has a finite optimal solution and that the precision parameter ϵ is set with value 0. Then, a simple sufficient condition ensuring the convergence of the *In/Out* algorithm is as follows.

Proposition 1.2.1. [33] *Assuming that the number of constraints that can be generated is finite and that the series $\sum_k \alpha_k$ diverges, then the points x_{in} and x_{out} of the *In/Out* algorithm both converge to an optimal solution. \square*

Now, let a convex optimization problem be of the form below:

$$\begin{cases} \min & c^T x \\ \text{s.t.} & g_i(x) \leq 0, i \in \{1, \dots, p\}, \\ & x \in \mathbb{R}^n, \end{cases}$$

where the functions g_i are continuously differentiable, strictly convex, p is a positive integer and $c \neq 0$. Among the first proposed methods to deal with such problems, Kelley's cutting-plane algorithm [141] relies on the use of linear inequalities induced by the convexity of the functions g_i of the form:

$$g_i(x_{sep}) + \nabla g_i^T(x_{sep})(x - x_{sep}) \leq 0, \quad (1.1)$$

where $\nabla g_i(x_{sep})$ denotes a subgradient vector of the function g_i at x_{sep} .

Consider now the use of inequalities (1.1) within the framework of the *In/Out* algorithm. Let $x_{out}^k, x_{sep}^k, x_{in}^k$ denote the points used at iteration k (see Algorithm 1 above). Further assume a compact set Q containing the set of optimal solutions is known. We may assume Q is a polyhedron and that all the points x_{out}^k belong to Q . The inequality added to the linear relaxation at iteration k is the one corresponding to the function g_j with $j = \arg \max \{g_i(x_{sep}^k) \mid i \in \{1, \dots, p\}\}$. The global convergence of Kelley's method (see, e.g., [159]) suggests a similar property for the *In/Out* algorithm. This is indeed the case under some assumptions, as stated by the next proposition, where α_{min} stands for a given constant value satisfying $0 < \alpha_{min} < 1$.

Proposition 1.2.2. [33] *If $\epsilon = 0$, the functions g_i are continuously differentiable, strictly convex and if there exists $K \in \mathbb{N}$ such that $\alpha_k \geq \alpha_{min}$, for all $k \geq K$, then the sequence $(x_{out}^k)_{k \in \mathbb{N}}$ converges to an optimal solution.*

The only cases when the sequence (x_{in}^k) may fail to converge to an optimal solution is when $\alpha_k = 1$ over an infinite number of iterations. Excluding such situations we deduce the next result, in which α_{max} stands for a given value satisfying $0 < \alpha_{min} \leq \alpha_{max} < 1$.

Corollary 1.2.3. [33] *If the functions g_i are continuously differentiable, strictly convex, and if there exists K such that $0 < \alpha_{min} \leq \alpha_k \leq \alpha_{max} < 1$, for all $k \geq K$, then the sequences (x_{out}^k) and (x_{in}^k) converge to an optimal solution.*

The last corollary may explain some observations stemming from computational experiments that we carried out [33]: for some problem instances we observed that a very little change in the value of the parameter α (for example $\alpha = 0.99$ by comparison with $\alpha = 1$) could greatly impact the performance of the *In/Out* algorithm. Also note that when $\alpha = 1$ the algorithm stops only when x_{out} becomes feasible. By contrast, choosing $\alpha_{max} < 1$, the algorithm may terminate when the gap between the objective values of x_{out} and x_{in} becomes lower than some given precision ϵ .

1.2.3 Applications

In this section we report different types of problems on which the *In/Out* algorithm has been evaluated: survivable network design, multicommodity flow problems, and randomly generated LPs. After their presentation we briefly sketch the main observations resulting from our experiments (see [33] for details).

Survivable network design

Let a network be represented by an undirected graph $G = (V, E)$ where V denotes the set of nodes and E the set of edges. Let K denote a set of demands: for each $k \in K$ some amount of traffic d_k should circulate between two specific nodes. The demands are satisfied by using the capacities available on the links of the network. Let x_e denote the maximum amount of traffic which may circulate on link $e \in E$. Here, the link capacities $(x_e)_{e \in E}$ are variables, whereas the demands $(d_k)_{k \in K}$ are part of the data. The unit capacity installation cost on edge $e \in E$ is c_e . A traffic demand may

be satisfied using several paths. In addition, a requirement w.r.t. the network survivability must be taken into account, and can be formulated as follows. Let R denote a set of possible states of the network, each state $r \in R$ being characterized by a subset of edges $E_r \subseteq E$ that can be used. It is then required that the installed capacities allow the satisfaction of all demands for all the states in R . The objective is to minimize the total cost of the capacities to be installed while satisfying the latter survivability requirement. The problem can be formulated under the form (PB) (see Section 1.1) where S stands for a polyhedron for which an exact separation oracle can be designed (that generates semimetric inequalities [136, 189] by solving auxiliary linear programs).

Multicommodity network flows

A second family of problem we considered are Multicommodity Min Cost Flow (MMCF) problems (see, e.g., [1, 149, 169]), which were solved using a column generation approach.

Given a graph $G = (V, E)$, a set K of demands, unit flow costs $(s_e^k)_{e \in E, k \in K}$ and edge capacities $(c_e)_{e \in E}$, the objective is to find a minimum cost flow satisfying all the demands and all the capacity constraints. It may be formulated as follows.

$$(MP) \begin{cases} \min \sum_{k \in K} \sum_{j^k \in P_k} s_{j^k} f_{j^k} \\ s.t. \\ \sum_{j^k \in P_k} f_{j^k} = d_k, \quad \forall k \in K, & (\pi_k) \\ \sum_{k \in K} \sum_{j^k \ni e} f_{j^k} \leq c_e, \quad \forall e \in E, & (v_e) \\ f_{j^k} \geq 0, \quad \forall k \in K, j^k \in P_k, \end{cases}$$

with the following notation:

- P_k : set of paths joining the extremities of the demand k ,
- $s_{j^k} = \sum_{e \in j^k} s_e^k$: unit flow cost on path $j^k \in P_k$,
- f_{j^k} : flow on path $j^k \in P_k$ to satisfy the demand k ,
- $(\pi_k)_{k \in K}, (v_e)_{e \in E}$: dual variables associated with the first and second type of constraints of formulation (MP).

In order to deal with a potentially very large number of columns, methods based on a Dantzig-Wolfe decomposition are classically used. A restriction (w.r.t. the set of columns) of the master problem (MP) is initialized with a set of columns P^0 such that the feasible region is nonempty. Then, iteratively, the current restricted master problem (RMP) is solved and the dual optimal values are used to generate new columns. For the case of (MMCF) problems this column generation step reduces to finding shortest paths. So, differently from the presentation of Section 1.1, in that case we apply the *In/Out* algorithm in the dual space of the current restricted master problem.

Random linear programs

A third type of problem instances used to evaluate the *In/Out* approach were randomly generated LPs of the following form.

$$\begin{cases} \min c^T x \\ s.t. \\ A \cdot x \geq \vec{1}, \\ x \in \mathbb{R}_+^n, \end{cases}$$

where $\vec{1}$ denotes an all ones vector of appropriate dimension. The entries of the matrix A are real numbers (with no sign restriction) randomly generated, so that the feasible region is nonempty. The coefficients of the objective function are nonnegative random real numbers. The matrix A is dense, with no peculiar structure, and its number of rows is always much larger than that of columns.

Main observations

For the different problems described above, different variants of separation oracles have been implemented, summarizing: one variant generating quickly *weak* inequalities and another one generating *strong* inequalities but requiring more important computation times. Here the terms *weak* and *strong* denote the fact that when considering the basic classical cutting plane scheme (i.e. setting $\alpha = 1$), the *strong* variant converges more rapidly to an optimal solution than the *weak version*. Main observations from the experiments may be synthesized as follows.

Consider firstly the case when a *strong* separation oracle is used. Decreasing the value of α to low values (below 0.5) tends to augment the total computation time (multiple relaxations are solved which do not lead to the generation of violated inequalities) by comparison with the classical scheme ($\alpha = 1$). On the positive side, when decreasing α below 1, the total number of generated constraints tends to decrease (which contributes to reduce the time spent on solving the relaxation at each iteration), so that, for values of α from 0.99 down to ≈ 0.8 , improvements on the total computation time could be observed in many cases.

Consider now the case when a *weak* separation oracle is used. Spending less time on separation at each iteration can make it attractive for sufficiently large instances, together with values strictly lower than 1 for the parameter α . In particular, it may even become competitive with the *strong* variant (considering any value of α for the *strong* variant): total computation times could be more than halved for some instances of network design problems (by comparison with the best results obtained using the *strong* oracle).

We also observed that tiny modifications of the parameter α (considering in particular the cases $\alpha = 0.99$ and $\alpha = 1$), could greatly impact the global performance (both w.r.t. the total computation time and the total number of generated inequalities). Thus far, we do not have any rule of thumb to determine for some given problem the best value for this parameter. On the whole, a value of α below but close to 1 (about 0.8) seems to be a robust choice. Further investigations may be directed towards an efficient setting of this parameter (possibly dynamic, i.e. changing in the course of the algorithm).

1.3 A multiple-points cutting-plane scheme

This section is dedicated to *multiple-points separation*: a generalization of classical constraint generation algorithms. Basically, it consists in finding an hyperplane separating a polyhedron from a given set of points X . So the cutting plane schemes we dealt with in the preceding sections correspond to the particular case when the set X is reduced to a single point (i.e. $|X| = 1$).

Related works appear in the literature, in particular in the field of analytical geometry dealing with *classification* or *discrimination* problems [15, 164–166, 203]. In such problems, the objective consists in finding a discriminating function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, separating two point sets A and B contained in \mathbb{R}^n , i.e. such that $f(x) > 0$, for all $x \in A$ and $f(x) < 0$, for all $x \in B$. This function may correspond to an affine function, a polynomial, etc. (see, e.g., Section 8.6 in [54]).

Here, we restrict ourselves to linear separation (i.e. the case when f is an affine function), one difference with classical classification problems being that the separation is to be done here between some given set of points and a polyhedron S which is described by an exact separation oracle. Another difference is our objective: generate an inequality which is valid for S (and, if possible, facet-defining), whereas in classification problems there is some *equivalence* in the way the points are considered with an objective that is often related to the misclassified points, see [38, 50].

The purpose of our investigations, summarized hereafter, were twofold: better understand the complexity of different problems inherent to the multiple-points separation framework, and evaluate the computational performance of such an approach by carrying out experiments on different family of optimization problems.

The rest of this section is organized as follows. After we discuss some particular features inherent to multiple-points separation and establish its polynomial equivalence with a classical single

separation procedure, we present a generic algorithm in order to generate constraints, making use of several separation points. Then, we briefly sketch our complexity results w.r.t. diverse problems related to this kind of separation. Finally, we shortly comment on the computational experiments we carried out and the obtained results. Further details on the work reported in this section can be found in [32].

1.3.1 Polynomial equivalence of single- and multiple-points separations

In what follows, let X^{sep} stand for the set of points that we aim at separating from a polyhedron S with some hyperplane.

In the classical (i.e. single-point) separation process, if we know an explicit formulation of S , (i.e. a finite set of linear inequalities and/or equations), then an obvious way to answer the exact separation problem (see Section 1.1) consists in computing the violation of the given separation point w.r.t. each constraint in the formulation. Such an approach fails, in general, if we want to find an hyperplane separating several points from S . For, assume that we wish to separate the set of points $\{x^1, x^2\}$ from the region S as illustrated in Figure 1.3. Note that, in this example, none of

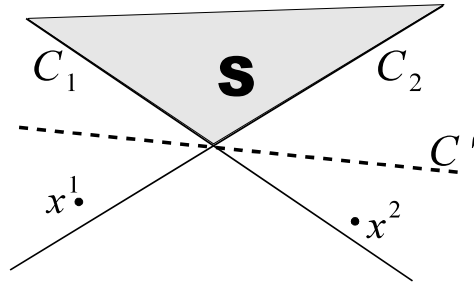


Figure 1.3 – Using the polar to generate a constraint separating several points from S .

the points, x^1 nor x^2 , belongs to the feasible region S since each of them violates a constraint that is valid for S : C_1 and C_2 respectively. Also, the single constraint C_1 (or C_2) cannot separate both x^1 and x^2 from S . However, in the present case, this could be achieved by a constraint such as C' (dashed line), which corresponds to a convex combination of C_1 and C_2 (and thus, is valid for S). This illustrates the necessity of working with the polar of S for separating several points from this region. In general, obviously, it is not always possible to generate an inequality separating some set of points X^{sep} from S , even in the case when $S \cap X^{sep} \neq \emptyset$: this precisely occurs if the intersection $conv(X^{sep}) \cap S$ is nonempty.

From the discussion above, it may seem much more difficult to proceed to multiple-points separation by comparison with single-point separation. In fact, relying on the fundamental ground-work by Grötschel et al. [121], we could prove that multiple-points separation has a particularly desirable feature w.r.t. the single-point counterpart: both are polynomially equivalent. We shall report hereafter the proof of this result since it provides some interesting insights, and the algorithmic structure of a procedure to perform a separation making use of several points easily follows. So let us now mention more precisely the results obtained, starting with the formal definition of the *exact* (or *strong*) *multiple-points separation problem*, denoted [MSEP], and two important formal assumptions to be used in our proofs.

Definition. *Exact multiple-points separation problem [MSEP].*

Given a point set $X^{sep} = \{x^1, \dots, x^p\}$, find

- (i) a vector in $conv(X^{sep}) \cap S$, or
- (ii) a valid inequality for S , that is violated by all the points in X^{sep} .

Assumption 1. S is a polyhedron in \mathbb{R}^n with facet complexity at most ϕ , i.e. there exists a system of inequalities with rational coefficients admitting S for solution set and such that the encoding length

of each inequality is at most ϕ . Let $S = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, with $A \in \mathbb{R}^{m \times n}$, $m \in \mathbb{Z}_+$, $b \in \mathbb{R}^m$ stand for such a formulation.

Assumption 2. Each point in X^{sep} has an encoding length which is upper bounded by a polynomial in n and ϕ . Also $p = |X^{sep}|$ is upper bounded by some polynomial in n and ϕ .

In order to find a constraint separating all the points in X^{sep} from S for the case when $\text{conv}(X^{sep}) \cap S = \emptyset$, we introduce the following LP whose feasible region corresponds to the valid inequalities for S , and w.r.t. which each point in X^{sep} has a violation larger than or equal to some value $\epsilon > 0$.

$$(PSEP_\epsilon) \begin{cases} \min & 0 \\ \text{s.t.} & \sum_{i=1}^m \lambda_i (a_i x^l - b_i) \geq \epsilon, \quad l \in \{1, 2, \dots, p\}, \quad (u_l) \\ & \lambda_i \geq 0, \quad i \in \{1, 2, \dots, m\}, \end{cases}$$

where a_i denotes the i -th row of the matrix A introduced in Assumption 1. After multiplying the objective function by the scalar $\frac{1}{\epsilon}$, the dual problem writes as follows.

$$(DSEP) \begin{cases} z_{DSEP}^* = \max & \sum_{l=1}^p u_l \\ \text{s.t.} & \sum_{l=1}^p u_l (a_i x^l - b_i) \leq 0, \quad i \in \{1, 2, \dots, m\}, \quad (\lambda_i) \\ & u_l \geq 0, \quad l \in \{1, 2, \dots, p\}. \end{cases}$$

Proposition 1.3.1. [32] *There exists an hyperplane separating all the points in X^{sep} from S if and only if $z_{DSEP}^* = 0$.*

By Proposition 1.3.1, it follows that part (ii) of problem [MSEP] can be reduced to solving (DSEP). This is used to establish the next result on the complexity of multiple-points separation.

The notion of *polynomiality* we use refers to polynomiality in the parameters n , ϕ and the time used by the separation oracle. Note in particular that this is independent from the parameter m introduced in Assumption 1. In what follows, by *oracle-polynomial time* we mean a duration (for our purposes we may assume this is an integer number of time units), that is upper bounded by a polynomial depending only on n and ϕ , assuming each call to the separation oracle takes one unit of time, and the same for each elementary operation.

Proposition 1.3.2. [32] *Under the assumptions 1 and 2, if S is given by an exact (single-point) separation oracle, then the exact multiple-points separation problem [MSEP] can be solved in oracle-polynomial time.*

Proof. Assume S is given by a strong single-point separation oracle. Let F_{MDSEP} stand for the feasible set of the problem (MDSEP) introduced above. From now on, we work with an implicit description of F_{MDSEP} by means of a strong separation oracle. The latter can be derived from the separation oracle for S . To be more precise, if the separation oracle is called with input $u = (0, \dots, 0)^T$, it returns that $u \in F_{MDSEP}$. Else, for a given point $u \in \mathbb{R}_+^p \setminus \{0\}$, the known strong separation oracle for S is called with the vector $\sum_{l=1}^p u_l x^l / \sum_{l=1}^p u_l$. If an inequality is returned by the latter oracle, it can be used to separate u , else $u \in F_{MDSEP}$. This separation oracle can be used to solve (MDSEP) by traditional constraint generation. Then, it follows from the polynomial equivalence between optimization and separation established by Grötschel et al. [121], that (MDSEP) can be solved in oracle-polynomial time. From Proposition 1, we have two cases: either $z_{MDSEP}^* = 0$ or $z_{MDSEP}^* = 1$. Let u^* stand for an optimal solution.

If $z_{MDSEP}^* = 1$, then, from Proposition 1, we know that there exists no valid inequality cutting off all the points in X^{sep} .

Consider now the case $z_{MDSEP}^* = 0$. Note that F_{MDSEP} has facet-complexity bounded above by a polynomial in n and ϕ . This follows from Assumptions 1 and 2, the fact that a representation of F_{MDSEP} can be derived from a formulation of S whose inequalities have encoding length bounded by ϕ , and the following property: $\langle x^T y \rangle \leq \langle x \rangle + \langle y \rangle$, for all $x, y \in \mathbb{R}^n$, where $\langle x \rangle$ denotes

the encoding length of x . Define now a basic optimum standard dual solution to be an optimum dual solution corresponding to a standard representation of the primal feasible set. Since F_{MDSEP} is given by a strong separation algorithm, it follows that a basic optimum standard dual solution $(\lambda_1^*, \lambda_2^*, \dots, \lambda_N^*)^T$ can be found in oracle-polynomial time (see Theorem 6.5.14 in [121]), together with the corresponding constraints $(g^1, h^1), \dots, (g^N, h^N)$ arising in a standard formulation of F_{MDSEP} . For our purposes, we consider that equality constraints of the form $g^T x = h$ are represented by two inequalities with coefficients corresponding to the row vectors (g, h) and $(-g, -h)$, so that the dual values λ^* are nonnegative. Note that the dual solution found λ^* satisfies the following relation denoted by (R^*) : $(1, \dots, 1, 0) = \sum_{i=1}^N \lambda_i^* (g_1^i, g_2^i, \dots, g_p^i, h^i)$. In fact, we namely have $h^i = 0$, for all $i \in \{1, 2, \dots, N\}$, since for that case $F_{\text{MDSEP}} = \{0\}$. Furthermore, given a valid inequality $(g, h) \in \mathbb{R}^{p+1}$ for F_{MDSEP} , a valid inequality for S from which (g, h) is derived, can be computed in oracle-polynomial time. Consider for this the polyhedron

$$Q' = \left\{ (\pi, \pi_0) \in \mathbb{R}^{n+1} : (\pi, \pi_0) \text{ is valid for } S; g_j \leq \pi x^j - \pi_0, j \in \{1, \dots, p\} \right\}.$$

Since (g, h) is valid for F_{MDSEP} , it is equivalent to or dominated by a constraint derived from a linear combination of constraints valid for F_{MDSEP} , i.e. there exist nonnegative coefficients $(\gamma_l)_{l=1}^p$ (for nonnegativity constraints), and $(\beta_i)_{i=1}^m$ (for constraints with coefficient vectors $(s^i, 0)$ derived from inequalities valid for S) such that (g, h) is dominated by $(-\sum_{l=1}^p \gamma_l e_l + \sum_{i=1}^m \beta_i s^i, 0)$, where e_i stands for the i -th unit vector. The latter implies $g_j \leq \sum_{i=1}^m \beta_i (s^i)_j$, for all $j \in \{1, 2, \dots, p\}$ (in case all the coefficients β are equal to zero we may consider the valid inequality $\vec{0}^T x \leq 0$ for S , (where $\vec{0}$ stands for the zero vector) so that we may assume that there always exists at least one positive coefficient β). It follows that Q' is nonempty, and from Assumptions 1 and 2, Q' has facet-complexity bounded above by a polynomial in n and ϕ . Also, a strong separation algorithm for Q' can easily be derived from a strong violation algorithm for S . Then, since the latter can be solved in oracle-polynomial time (by Theorem 6.4.9 in [121]), it follows that a point in Q' can be computed in oracle-polynomial time (by Theorem 6.4.1 in [121]). Hence, for any (g^i, h^i) , $i = 1, 2, \dots, N$, a constraint (π^i, π_0^i) that is valid for S and from which (g^i, h^i) is derived can be computed in oracle-polynomial time. Finally, it follows from the relation (R^*) that the inequality $\sum_{i=1}^N \lambda_i^* \pi^i x \leq \sum_{i=1}^N \lambda_i^* \pi_0^i$, which is valid for S , separates all the points in X^{sep} (since it corresponds to a feasible solution for $(\text{PSEP})_{\epsilon=1}$). \square

1.3.2 A generic multiple-points separation procedure

After we discuss on the two situations which may occur in the multiple-points separation (i.e. $\text{conv}(X^{\text{sep}}) \cap S$ is empty or not) a general multiple-points separation scheme is introduced.

Case $\text{conv}(X^{\text{sep}}) \cap S = \emptyset$.

In case $\text{conv}(X^{\text{sep}}) \cap S = \emptyset$, (according to the *linear discrimination alternative*, see Section 8.6.1 in [54]) there exists a hyperplane separating all the points in X^{sep} from S and consequently the feasible region of problem (PSEP_ϵ) is nonempty (for all $\epsilon > 0$). This is illustrated by Figure 3.2(a).

Case $\text{conv}(X^{\text{sep}}) \cap S \neq \emptyset$.

When it is not possible to separate all the points in X^{sep} (as illustrated by Figure 3.2(b)), one way to proceed may consist in iteratively removing points from X^{sep} before resuming separation with remaining points. For example, consider the case when X^{sep} contains an optimal solution x^* for the current relaxation. Then, unless x^* is feasible (and thus optimal) for the original problem, applying the former idea by keeping x^* in X^{sep} , will always lead to the generation of a constraint.

In addition, when $\text{conv}(X^{\text{sep}}) \cap S \neq \emptyset$, following the methodology adopted for the proof of Proposition 1.3.2 (see [32]), we can obtain a feasible solution of the original problem that can be used to update a primal bound on the optimal objective value.

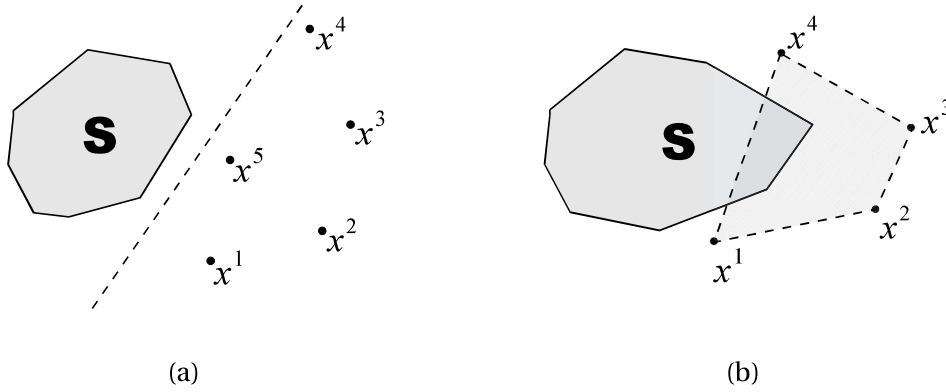


Figure 1.4 – Illustration of the cases: (a) $\text{conv}(X^{sep}) \cap S = \emptyset$ and (b) $\text{conv}(X^{sep}) \cap S \neq \emptyset$

Another way to proceed, instead of removing points from X^{sep} , could consist in looking for valid constraints separating subsets of points of X^{sep} . For example, assuming all the points in X^{sep} lie outside the feasible region, it is then possible to decompose X^{sep} as the union of subsets of points $Y^j \subseteq X^{sep}$ (non-necessarily disjoint) satisfying:

- $S \cap \text{conv}(Y^j) = \emptyset$,
- $S \cap \text{conv}(Y^j \cup \{z\}) \neq \emptyset, \forall z \in (X^{sep} \setminus Y^j)$.

Such a decomposition is illustrated by Figure 1.5. We provide later (Section 1.3.3) complexity results related to this kind of procedures.

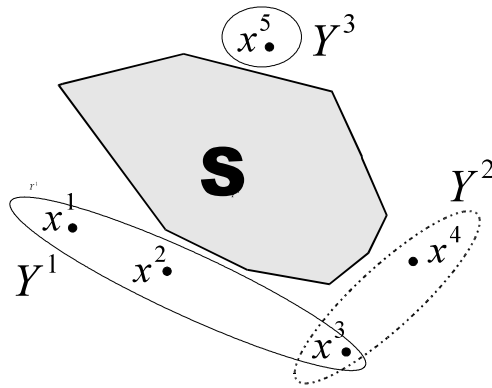


Figure 1.5 – Using several sets of infeasible points to generate several constraints

Algorithmic structure

Let x^* denote an optimal solution of the current relaxation of problem (PB). We shall not give here an accurate definition for the points contained in X^{sep} in order to present the method in its full generality. However, we will make the assumption that the optimal solution of the current relaxation x^* always belongs to this set.

The algorithm only differs from classical constraint generation procedures at the separation step. Assuming a set of points X^{sep} is given, the separation procedure looks for a constraint that is violated by all those points. In case none is found, (i.e. $\text{conv}(X^{sep}) \cap S \neq \emptyset$) with $|X^{sep}| > 1$, points different from x^* are removed from X^{sep} and separation is resumed on the set of remaining points. Otherwise, if the separation procedure does not find any violated inequality with $X^{sep} = \{x^*\}$, then the algorithm stops with x^* as optimal solution for the original problem. In case the separation

procedure finds an inequality separating all the points in X^{sep} from the feasible region, it is added to the current relaxation that will be solved anew and the whole process is iterated.

Algorithm 2. Generic multiple-points separation (GMPS)

1. Solve the current relaxation $\rightarrow x^*$;
 2. Update X^{sep} ($x^* \in X^{sep}$) ;
 3. While no inequality separating all the points in X^{sep} is found and x^* is not optimum do:
 - 3.1. Find an inequality separating X^{sep} from S ;
 - 3.2. If an inequality has been found, then
 - Add this inequality to the current relaxation. Go to step 1 ;
 - 3.3. Else if $|X^{sep}| > 1$ then
 - Remove from X^{sep} a subset of points Q with $x^* \notin Q$,
 $X^{sep} := X^{sep} \setminus Q$;
 - 3.4. Else
 - x^* is an optimal solution for the original problem ;
- End_if;
End_while;
END.
-

Remark. If the point x^* is not included in X^{sep} , one cannot guarantee that the solution of the current iteration will change in the next iteration. Anyway, different algorithmic schemes may be elaborated with point sets X^{sep} not including x^* initially.

Remark. For the case when the multiple-points separation procedure fails to determine an inequality separating all the points in X^{sep} , one may generate several inequalities, each separating a subset of points of X^{sep} .

Two elements of the procedure (GMPS) remain to be described before proceeding to an implementation: an accurate definition of the point set X^{sep} that is used at each iteration, and of the points which may be iteratively removed from X^{sep} . We have not yet investigated deeply these parts, but started dealing with theoretical matters w.r.t. the complexity of problems related to these two points.

1.3.3 Complexity results

The problems considered in this section are related to the way the set of points X^{sep} can be dealt with in step 3 of procedure (GMPS). Throughout this section we assume that multiple-points separation can be done in polynomial-time. We start with some useful definitions.

Definition. A set of points $X = \{x^1, x^2, \dots, x^p\}$ is said to be *separable* if there exists a constraint $a^T x \leq b$ separating all the points in X from the feasible region S , i.e. $a^T x^i > b$, for all $i \in \{1, \dots, p\}$. Otherwise this set is said to be *non-separable*.

Definition. Given a set of non-separable points $X = \{x^1, x^2, \dots, x^p\}$, a subset $Y \subseteq X$ is said to be *maximal separable* if Y is separable and, for all $x \in (X \setminus Y)$, $Y \cup \{x\}$ is non-separable.

Definition. Given a set of points $X = \{x^1, x^2, \dots, x^p\}$, a subset $Y \subseteq X$ is said to be *minimal non-separable* if $|Y| \geq 2$, Y is non separable, and for all $x \in Y$, $Y \setminus \{x\}$ is separable.

Next, complexity results dealing with problems related to maximal separable sets are reported. These are followed by their *non-separable* counterparts.

Maximal separable set with maximum cardinality

Given a non-separable point set X^{sep} , a possible criterion for evaluating the quality of the constraint generated could be the number of points of X^{sep} that this constraint separates. The decision version of the problem which consists in looking for a constraint separating a maximum number of points contained in X^{sep} can be formulated as follows.

[SEP_MAX]

Instance: a set of p points in $\mathbb{R}^n : X^{sep} = \{x^1, x^2, \dots, x^p\}$, an integer $K \leq p$.

Question: Does there exist a separable set $Y \subseteq X^{sep}$ satisfying $|Y| \geq K$?

With a reduction from the stable set problem in a graph [111], we could prove the following result on the complexity status of [SEP_MAX].

Proposition 1.3.3. [32] *Problem [SEP_MAX] is \mathcal{NP} -complete.*

Minimum cardinality maximal separable set

Depending on the used separation procedure, handling several points may induce non negligible additional computations. One may then be interested in looking for a *good* constraint in the sense that it would separate a maximal separable set, but with the latter having minimum cardinality to limit computation times. W.r.t. the decision version of the corresponding problem denoted by [SEP_MIN], we could establish the next result, by reduction from the stable dominating set problem in a graph [111].

Proposition 1.3.4. [32] *Problem [SEP_MIN] is \mathcal{NP} -complete.*

Correspondence between a minimum partition into separable sets and the minimum number of constraints for separation

Given a set of points X^{sep} , all of which lying out of the feasible region, in order to limit the number of constraints in the current relaxation, one may be interested in the minimum number of constraints needed to separate all the points in X^{sep} . We could show this problem corresponds to determining a minimum cardinality partition of X^{sep} into separable sets.

Proposition 1.3.5. [32] *The minimum cardinality of a partition of X^{sep} into separable sets is equal to the minimum number of constraints which are necessary to separate all the points contained in X^{sep} .*

Also, we can show that the problem which consists in partitioning a node set of a graph into k stable sets (the so-called *graph k -colourability* or *chromatic number*) is polynomially reducible to the problem of partitioning a set of infeasible points into k separable sets. And since the decision problem *graph k -colourability* is \mathcal{NP} -complete [111], it follows that the minimum cardinality partition of a set of infeasible points into separable sets is \mathcal{NP} -hard.

Minimum cardinality minimal non-separable set

We start with the decision version of the problem which consists in determining a minimum cardinality non-separable set.

[NON_SEP_MIN]

Instance: a set of p points $X^{sep} = \{x^1, x^2, \dots, x^p\}$ in \mathbb{R}^n , a positive integer $K \leq p$.

Question: Does there exist a minimal non-separable set $Y \subseteq X^{sep}$ with $|Y| \leq K$?

By reduction from the minimum cardinality dominating set problem in a graph [111], we deduce the next result.

Proposition 1.3.6. [32] *Problem [NON_SEP_MIN] is \mathcal{NP} -complete.*

Maximum cardinality minimal non-separable set

W.r.t. the complexity of problem [NON_SEP_MAX] - whose formulation differs from [NON_SEP_MIN] only in the question, replacing $|Y| \leq K$ by $|Y| \geq K$ - we have the following result (obtained by reduction from the maximum cardinality minimal dominating set problem in a graph [59]).

Proposition 1.3.7. [32] *Problem [NON_SEP_MAX] is \mathcal{NP} -complete.*

Partition into non-separable sets

The partition of a set of non-separable points into several non-separable sets could be used to obtain several primal bounds. The corresponding decision problem may be formulated as follows. [NON_SEP_PART]

Instance: a set of p points $X^{sep} = \{x^1, x^2, \dots, x^p\}$, a positive integer $K \leq p$.

Question: Does there exist a partition of X^{sep} into $k \geq K$ non-separable sets?

The next result follows by reduction from the *domatic number* problem [111].

Proposition 1.3.8. [32] *Problem [NON_SEP_PART] is \mathcal{NP} -complete.*

1.3.4 An overview of computational experiments

Computational experiments have been performed on LPs of the following form.

$$(P) \begin{cases} \max & c^T x \\ \text{s.t.} & Ax \leq b, \\ & x \in \mathbb{R}^n, \end{cases}$$

where the feasible region $S = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ is assumed to be bounded, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $m \gg n$. The multiple-points separation procedure evaluated, denoted by l -RELAX consists in defining X^{sep} as the set of the l last optimal solutions of the relaxations solved. Experiments have been carried out on randomly generated LPs and survivable network design problems (such as the ones used to evaluate the *In/Out* approach we described before) for different values of the parameter l . The reader is referred to [32] for more details.

Randomly generated linear programs

We summarize hereafter the main observations from our experiments on random LPs.

- The final number of constraints of the relaxations supplying an optimal solution for the original problem stand rather close for the whole set of separation procedures evaluated (including the *classical* single-point separation).
- However, the total number of iterations may vary considerably, depending on the number of points used in the separation process: handling more points tends to decrease the total time spent on solving relaxations. To illustrate this, for an instance corresponding to a formulation with 600 variables and 10000 constraints, the number of iterations was more than divided by six between a single-point separation and the variant 50-RELAX. However, handling more points in the set X^{set} induces a longer time spent on the separation at each iteration. And if the number of points used becomes too high, the total computation time increases.
- The obtained results suggest an increasing interest for multiple-points separation for an increasing size of the instances. For example, the total computation times have been more than halved for the largest instances solved (considering the procedure 20-RELAX by comparison with single-point separation).

Survivable network design

We now briefly sketch the main observations from experiments on survivable network design instances.

- Results tend to display an increasing interest for multiple-points separation for problems with an increasing number of states (and a fixed number of demands).
- If the number of states becomes too small, the performance of the single-point separation scheme tends to dominate that of multiple-point procedures.
- On the contrary, for a sufficiently high number of states, a reduction of the total computation time could be observed, through a decrease of the total number of iterations due to the generation of more constraints by the separation procedure. For example, the total number of iterations is more than divided by 6 between the cases of single-point separation and ∞ -RELAX (which consists in making use of all previous optimal solutions of the relaxations solved) for an instance with 40 nodes, 200 edges, 110 demands and 30 states.

1.4 A simple finite cutting-plane algorithm for (mixed-) integer LPs

In this section we present a cutting plane algorithm in order to solve integer linear programs in a finite number of iterations, under the assumption that the feasible region is bounded. The method also applies to the resolution of mixed-integer linear programs in a finite number of iterations, under the additional assumption that the optimal objective value is integral. This work has been published in *Operations Research Letters* [183].

1.4.1 General introduction and related work

For an up-to-date and much broader review of cutting plane algorithms for (mixed-)integer programming (and some related implementation issues), the reader may consult, e.g., [87]. The first cutting plane algorithm to solve integer linear programs with a proof of finite convergence was proposed by Gomory in the 1950's [117, 118, 181]. Since then, other finitely convergent cutting plane methods have emerged in the literature, e.g., [5, 26, 53, 191]. Next, we present a simple such algorithm relying on the generation of special constraints that are iteratively added to a relaxation of the original integer program formulated as a lexicographic optimization problem. A particular feature of the method by comparison with Gomory's original algorithm [117, 118] is that it does not require knowledge of the *LP-tableau* to generate cutting-planes: only the values of the components of the optimal solution found for a relaxation of the problem are used at each iteration. The optimal LP-basis was also used in the algorithm introduced later by Bell [26]. There, the cutting-planes used essentially rely on the integrality of the optimal objective value and they are iteratively applied to linear relaxations of the problem having an objective function that may be modified by the procedure. As is the case in Orlin's algorithm [191] the method presented here involves the iterative resolution of a lexicographic optimization problem. However it differs from the former in that it may be applied to mixed-integer programs under the assumptions mentioned hereafter (instead of 0/1-mixed integer programs) and it does not involve additional auxiliary variables in the objective function of the lexicographic formulation. As some of the methods just mentioned above, the algorithm presented here may be viewed as a particular enumerative process and a counterpart to its simplicity may be that the polyhedral structure of the feasible region of the problem is not exploited for a potentially better efficiency.

Before we present the method we introduce some notation. Given two vectors $x, y \in \mathbb{R}^{n+1}$, the vector $x = (x_0, x_1, \dots, x_n)$ is said to be *lexicographically lower* than the vector $y = (y_0, y_1, \dots, y_n)$ and we write $x <_L y$ if there exists an integer $k \in \{0, 1, \dots, n\}$ with $x_k < y_k$ and $x_i = y_i$ for all $i < k$. And similarly, we write $x \leq_L y$ if $x <_L y$ or $x = y$.

Given a set $S \subseteq \mathbb{R}^{n+1}$, consider the following problem

$$(PBLEX) \text{ lexmax } \{z_0, z_1, \dots, z_n : z \in S\}$$

which consists in finding a lexicographically maximum vector in S .

Any integer linear program in the following form

$$(IP) \max\{c^T x : x \in P \cap \mathbb{Z}^n\}$$

with $c \in \mathbb{R}^n$ and where P stands for a polyhedron in \mathbb{R}^n , may be expressed as a problem of the type (PBLEX), setting $z_0 := c^T x$, $z_i := x_i$, for all $i \in \{1, 2, \dots, n\}$ and $S := \mathbb{R} \times (P \cap \mathbb{Z}^n)$.

In what follows, we consider solving a problem of the form (IP) under the assumptions that the objective function is defined by an integer vector: $c \in \mathbb{Z}^n$, and that the feasible region P is bounded: $P \subseteq [0, d]^n$, for some positive integer d .

1.4.2 Cutting-plane algorithm

In this section we describe the proposed method after we introduce the cutting planes used by the latter.

Proposition 1.4.1. *Let $z = (z_0, z_1, \dots, z_n) \in \mathbb{R} \times [0, d]^n$. Then, the following inequalities are satisfied by any integer vector $x \in \mathbb{Z} \times ([0, d] \cap \mathbb{Z})^n$ which is lexicographically lower than z (i.e. $x \leq_L z$)*

$$x_i + \sum_{j=0}^{i-1} a_{i-j}(x_j - \lceil z_j \rceil) \leq \lfloor z_i \rfloor, \quad (1.2)$$

for all $i \in \{0, 1, \dots, n\}$ with $a_1 = d$ and $a_k = d(1 + \sum_{j=1}^{k-1} a_j)$ for any integer $k \in \{2, 3, \dots, n\}$.

Proof. Let $y \in \mathbb{Z} \times ([0, d] \cap \mathbb{Z})^n$ satisfying $y \leq_L z$. The proposition is obvious if $y = z$ or for the index value $i = 0$ (i.e. $y_0 \leq \lfloor z_0 \rfloor$).

So assume now $y \neq z$ and $k \geq 1$ with $k = \arg \min_j \{y_j < z_j : j = 0, 1, \dots, n\}$. Notice that the definition of k and the assumption $y \leq_L z$ imply $y_q = z_q$, for all $q \in \{0, 1, \dots, k-1\}$. Thus the inequality (1.2) is trivially satisfied for all $i \in \{0, \dots, k\}$.

Consider the left hand-side of the inequality (1.2) for the case $i > k$. We have for the vector y :

$$\begin{aligned} y_i + \sum_{j=0}^{i-1} a_{i-j}(y_j - \lceil z_j \rceil) &= y_i + \sum_{j=k}^{i-1} a_{i-j}(y_j - \lceil z_j \rceil) \\ &\leq y_i - a_{i-k} + \sum_{j=k+1}^{i-1} a_{i-j}(y_j - \lceil z_j \rceil) \\ &\leq y_i - a_{i-k} + d \sum_{j=1}^{i-k-1} a_j = y_i - d \\ &\leq 0 \\ &\leq \lfloor z_i \rfloor. \end{aligned}$$

The first equation follows from the observation mentioned above that $y_q = z_q$, for all $q \in \{0, \dots, k-1\}$. The first inequality follows from the definition of the index k which implies $y_k - \lceil z_k \rceil \leq -1$. The second inequality is due to the fact that y_j and z_j belong to the interval $[0, d]$ for all $j \in \{1, \dots, n\}$. \square

In what follows, the inequality (1.2) corresponding to some particular index value $i \in \{0, 1, \dots, n\}$ will be denoted $(1.2)_i$. We now present an algorithm to solve (IP) under the assumptions mentioned above. Also, for simplicity in the presentation, we consider the case when S is nonempty, i.e. the polyhedron P contains at least one integral point. The case when S is empty may be handled by the routine which solves the relaxation of the lexicographical problem at each iteration (Step 1 in Algorithm 3 below), and if this occurs the whole procedure is stopped.

Algorithm 3.

Step 0. $S^0 := \{(x_0, x') \in \mathbb{R}^{n+1} : x' = (x_1, \dots, x_n) \in P, x_0 = c^T x'\}$.
 $t := 0$.

Step 1. Solve $\text{lexmax } \{z_0, z_1, \dots, z_n : z \in S^t\}$.

Step 2. If $z_i \in \mathbb{Z}$, for all $i \in \{1, \dots, n\}$ then

Stop. An optimal solution for (IP) is (z_1, \dots, z_n) .

Else

$k := \text{argmin}_j \{z_j \notin \mathbb{Z} : j = 0, \dots, n\}$.

$S^{t+1} := \{x \in S^t : x \text{ satisfies (1.2)}_k\}$.

$t := t + 1$.

Go to Step 1.

End_if.

END.

One may notice that the key feature of the inequalities of the form (1) that are used in Step 2 of the Algorithm 3 is that they allow to separate the current fractional solution (z_0, z_1, \dots, z_n) while not cutting off any lexicographically smaller integral solution. In the next section we provide some further specifications on the set of vectors that are contained in S^t and are not cut off by the generated constraint (Lemma 1.4.3).

1.4.3 Finiteness

We now prove that the Algorithm 3 solves (IP) in a finite number of iterations. Given a vector $(y_0, y_1, \dots, y_n) \in \mathbb{R} \times [0, d]^n$, $d \in \mathbb{Z}_+$, let $\alpha(y)$ denote the integral vector defined as follows: $\alpha(y)_i = \lfloor y_i \rfloor$ if $i \leq k$ and $\alpha(y)_i = d$ otherwise, for all $i \in \{0, 1, \dots, n\}$, with $k := \text{argmin}_j \{y_j \notin \mathbb{Z} : j = 0, \dots, n\}$. So $\alpha(y)$ is the lexicographically largest integral vector in $\mathbb{R} \times [0, d]^n$ that is lexicographically smaller than y .

Lemma 1.4.2. *Let $\bar{z} = (\bar{z}_0, \dots, \bar{z}_n)$ be an optimal solution of the problem $\text{lexmax } \{x_0, x_1, \dots, x_n : x' = (x_1, \dots, x_n) \in P \text{ and } x_0 = c^T x'\}$ where P stands for a polyhedron such that $P \subseteq [0, d]^n$. Let $\bar{x} = (\bar{x}_0, \dots, \bar{x}_n)$ be an optimal solution of the problem $\text{lexmax } \{x_0, x_1, \dots, x_n : x' = (x_1, \dots, x_n) \in P' \text{ and } x_0 = c^T x'\}$ with $P' = P \cap \mathbb{Z}^n$. Then the following inequalities hold*

$$\left(d \sum_{i=1}^n \min(0, c_i), 0, \dots, 0 \right) \leq_L \bar{x} \leq_L \alpha(\bar{z}).$$

Proof. The case when \bar{z} is integral is trivial. So let us consider the case when \bar{z} is not integral. The first inequality (on the left) stems from the bounds on the variables x_1, \dots, x_n . The other inequality is due to the fact that \bar{z} solves a relaxation of the problem solved by \bar{x} , thus implying $\bar{x} \leq_L \bar{z}$, and since \bar{x} is integral we have $\bar{x} \leq_L \alpha(\bar{z})$. \square

Lemma 1.4.3. *Let $Q \subseteq \mathbb{R} \times [0, d]^n$ denote a polyhedron and let $z = (z_0, z_1, \dots, z_n) \in \mathbb{R} \times [0, d]^n$ denote a vector such that $Q \subseteq \{x \in \mathbb{R}^{n+1} : x \leq_L z\}$. Then the following inclusion holds*

$$\{x \in Q : x \text{ satisfies (1.2)}_k\} \subseteq \{x \in Q : x \leq_L \alpha(z)\},$$

with $k := \text{argmin}_j \{z_j \notin \mathbb{Z} : j = 0, 1, \dots, n\}$.

Proof. Let $y \in Q$ and assume $\alpha(z) <_L y$. From the assumption $Q \subseteq \{x \in \mathbb{R}^{n+1} : x \leq_L z\}$ we have $y_i = z_i$ for all $i \in \{1, 2, \dots, k-1\}$. So the inequality $\alpha(z) <_L y$ implies $y_k > \lfloor z_k \rfloor$ and y violates the inequality (1.2)_k. \square

Lemma 1.4.4. *Let z^t denote an optimal solution of the problem $\text{lexmax}\{z_0, z_1, \dots, z_n : z \in S^t\}$ (that is solved at iteration t in Step 1 of Algorithm 3). Then, if z^t and z^{t+1} are not integral vectors, the following inequality holds: $\alpha(z^{t+1}) <_L \alpha(z^t)$.*

Proof. Since the vectors z^t and z^{t+1} are not integral we have $\alpha(z^q) <_L z^q$ for $q = t, t+1$. Also, from the definition of the set S^{t+1} in Step 2 of the Algorithm 3, the Lemma 1.4.3 and the fact that z^{t+1} is not integral, we deduce $z^{t+1} <_L \alpha(z^t)$. So we get $\alpha(z^{t+1}) <_L z^{t+1} <_L \alpha(z^t) <_L z^t$, and the proposition follows. \square

Theorem 1.4.5. *Algorithm 3 finds an optimal solution of the problem $\max\{c^T x : x \in P \cap \mathbb{Z}^n\}$ (with $c \in \mathbb{Z}^n, P \subseteq [0, d]^n$) in a finite number of iterations.*

Proof. Let \bar{x} stand for an optimal solution of the problem $\text{lexmax}\{z_0, z_1, \dots, z_n : z' = (z_1, z_2, \dots, z_n) \in P \cap \mathbb{Z}^n \text{ and } z_0 = c^T z'\}$. From Lemma 4.2.1 ($d \sum_{i=1}^n \min(0, c_i), 0, \dots, 0 \leq_L \bar{x} \leq_L \alpha(\bar{z}^t)$) for each iteration t , where \bar{z}^t stands for an optimal solution of the problem that is solved in Step 1 of Algorithm 3 at iteration t . Since the number of integer vectors that are lexicographically between $(d \sum_{i=1}^n \min(0, c_i), 0, \dots, 0)$ and $\alpha(\bar{z}^t)$ is finite and since from Lemma 4.2.2 we have $\alpha(z^{t+1}) <_L \alpha(z^t)$ whenever z^{t+1} is not an integer vector, we get that an integer solution is found in a finite number of iterations. From Lemma 4.2.1, the number of iterations is $\mathcal{O}((d+1)^n C)$ with $C = 1 + d \sum_{i=1}^n |c_i|$. \square

Similarly to Gomory's original cutting-plane algorithm the method presented may be used to solve mixed-integer programs (for a mixed-integer program with n integer and p continuous variables one may use a formulation in which the variables are ordered such that x_0 corresponds to the objective value, then x_1, x_2, \dots, x_n correspond to the integer variables and x_{n+1}, \dots, x_{n+p} to the continuous ones, see e.g., [117, 181]) under the assumptions that the optimal objective value is integral, bounded and that the integer variables are bounded.

1.5 Some perspectives

In this chapter, we presented different developments brought to cutting plane methods. The *In/Out* algorithm and multiple-points approach generalize classical cutting-plane methods for LPs in the separation process, while the last presented method may be viewed as a simplification of Gomory's original algorithm for some mixed-integer programs.

For all the presented methods, further research may be directed towards much deeper computational experiments, evaluating them to solve different families of problems and compare them with other approaches.

More particularly w.r.t. the *In/Out* algorithm, it would be interesting to study the impact of a dynamic setting of the parameter α defining the separation point at each iteration, and (ideally) be able to propose an efficient and robust general rule.

W.r.t. the multiple-points approach, so far, only very restricted choices for the set of separation points have been used. Then, a natural research direction is the identification of sets of separation points leading to the generation of strong inequalities or to the derivation of interesting bounds on the optimal objective value (making use of the iterations when the separation fails).

Another potential future work may consist in adjusting the *In/Out* and multiple-points approaches for dealing with integer programs.

Chapter 2

A recursive algorithm for some unconstrained $(-1, 1)$ -quadratic optimization problems

In this chapter we present a combinatorial algorithm to solve a special family of unconstrained quadratic optimization problems with variables taking their values in the set $\{-1, 1\}$. This work, joint with Prof. Ben-Ameur, allowed us to identify a family of instances of such problems that can be solved in polynomial time, slightly extending previous results from the literature. Further details on this approach, including some computational experiments can be obtained in our publication [36].

2.1 Introduction

Consider a quadratic function $q : \mathbb{R}^n \rightarrow \mathbb{R}$ given by: $q(x) = x^T Q x$, with $Q \in \mathbb{R}^{n \times n}$. An unconstrained $(-1, 1)$ -quadratic optimization problem can be expressed as follows:

$$(QP) Z^* = \min \{q(x) \mid x \in \{-1, 1\}^n\},$$

where $\{-1, 1\}^n$ denotes the set of n -dimensional vectors with entries in the set $\{-1, 1\}$. We assume here that the matrix Q is symmetric and given by its spectrum, i.e. the set of its eigenvalues and the associated unit and pairwise orthogonal eigenvectors. For our purposes, we further assume that all the values occurring in the spectrum, i.e. eigenvalues and entries of eigenvectors are rational.

Problem (QP) is a classical combinatorial optimization problem with many applications, e.g., in statistical physics and circuit design [23, 119, 197]. It is well-known that any $(0, 1)$ -quadratic problem expressed as: $\min \{x^T A x + c^T x \mid x \in \{0, 1\}^n\}$, $A \in \mathbb{R}^{n \times n}$, $c \in \mathbb{R}^n$, can be formulated in the form of problem (QP) (with dimension $n + 1$ instead of n) and conversely [78, 123].

Problem (QP) is known to be NP-hard in general [139]. Previously to our work [36], the following cases were known to be solvable in polynomial time:

1. when the matrix Q has nonpositive off-diagonal entries only [196],
2. when the matrix Q has fixed rank and the continuous relaxation of (QP) has an optimal solution in $\{-1, 1\}^n$ [4, 56].

We could extend the known polynomially solvable cases of (QP) to when the matrix Q has fixed rank and the number of positive diagonal entries is $\mathcal{O}(\log(n))$. To be more precise, we introduced a new polynomial-time recursive algorithm for solving instances of problem (QP) corresponding to this case and could proceed to some computational experiments.

In Section 2.2 we introduce properties that are verified by all optimal solutions of (QP). A geometrical interpretation of these properties is then presented in Section 2.3 where we also describe

a recursive procedure for determining the cells of an arrangement of hyperplanes. This is the basis of the proposed algorithm for solving the afore mentioned polynomially solvable cases of (QP) which is introduced in Section 2.4. We finally draw some perspectives in Section 2.5.

2.2 Properties of optimal solutions for particular instances of (QP)

Let us firstly introduce some notation to be used hereafter. The rank of matrix Q is denoted by $p \leq n$. The p non-zero eigenvalues of Q are denoted by $\lambda_1(Q) \leq \lambda_2(Q) \leq \dots \leq \lambda_p(Q)$ (or more simply $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_p$ when clear from the context) and the corresponding unit (in Euclidean norm) and pairwise orthogonal eigenvectors: v_1, \dots, v_p . The j th entry of the vector v_i is denoted by v_{ij} . Given some set of vectors $a_1, \dots, a_q \in \mathbb{R}^n$, $q \in \mathbb{N}$, $\text{Lin}(a_1, \dots, a_q)$ represents the subspace spanned by these vectors.

In this section we shall make the following assumptions on the matrix Q :

(i) Q has rank $p \leq n$,

(ii) Q has nonpositive diagonal entries only, and

(iii) Q is given by its set of rational eigenvalues and eigenvectors: $Q = \sum_{i=1}^p \lambda_i v_i v_i^T$.

Consider a vector $y \in \{-1, 1\}^n$ satisfying $\sum_{i=1}^p \lambda_i \alpha_i v_{ij} > 0$ and $y_j = 1$, for some index $j \in \{1, \dots, n\}$, with $\alpha_i = y^T v_i$, $\forall i \in \{1, \dots, n\}$. Let $\bar{y}^j \in \{-1, 1\}^n$ denote the vector differing from y only at the j th entry: $(\bar{y}^j)_j = -1$ and set $\bar{\alpha}_i = (\bar{y}^j)^T v_i$, $i \in \{1, \dots, n\}$. Then we have $\bar{\alpha}_i = \alpha_i - 2v_{ij}$. From the latter we deduce: $(\bar{y}^j)^T Q (\bar{y}^j) = \sum_{i=1}^p \lambda_i \bar{\alpha}_i^2 = \sum_{i=1}^p \lambda_i \alpha_i^2 + 4 \sum_{i=1}^p \lambda_i v_{ij}^2 - 4 \sum_{i=1}^p \lambda_i \alpha_i v_{ij} < \sum_{i=1}^p \lambda_i \alpha_i^2 = y^T Q y$, where the last inequality follows from $\sum_{i=1}^p \lambda_i \alpha_i v_{ij} > 0$ and $\sum_{i=1}^p \lambda_i v_{ij}^2 = Q_{jj} \leq 0$ (assumption (ii)).

Hence we have the following proposition.

Proposition 2.2.1. *Any optimal solution y^* to the problem $\min_{y \in \{-1, 1\}^n} y^T Q y$ satisfies the following implications:*

$$\sum_{i=1}^p \lambda_i \alpha_i v_{ij} > 0 \Rightarrow y_j^* = -1 \quad (2.1)$$

$$\sum_{i=1}^p \lambda_i \alpha_i v_{ij} < 0 \Rightarrow y_j^* = 1. \quad (2.2)$$

Implications of Proposition 2.2.1 can also be seen to correspond to first order optimality conditions. If we now assume that, for some index $j \in \{1, \dots, n\}$, an optimal solution y satisfies: $\sum_{i=1}^p \lambda_i \alpha_i v_{ij} = 0$, with $y_j = \pm 1$ and consider the vector $\bar{y}^j \in \{-1, 1\}^n$ that is obtained by multiplying the j th entry of vector y by -1 . Then we have: $\sum_{i=1}^p \lambda_i \bar{\alpha}_i^2 = \sum_{i=1}^p \lambda_i \alpha_i^2 + 4 \sum_{i=1}^p \lambda_i v_{ij}^2$. The optimality of y namely implies $\sum_{i=1}^p \lambda_i v_{ij}^2 = Q_{jj} = 0$, i.e. the j th entry of the diagonal of the matrix Q is zero. In case $v_{ij} = 0$, for all $i \in \{1, \dots, p\}$, then row j can be ignored (i.e. an optimal solution y can take value $y_j = \pm 1$ and changing the sign of its j th entry we get another optimal solution). So w.l.o.g., we shall assume hereafter that for each index $j \in \{1, \dots, n\}$ there exists at least one index $i \in \{1, \dots, p\}$ with $v_{ij} \neq 0$. To deal with this situation one can proceed as follows. Assume that in the former representation of the matrix Q the eigenvalues λ_i and all the entries of the eigenvectors v_i , $i \in \{1, \dots, n\}$ are expressed with a common denominator D . Let $\epsilon > 0$ and consider the matrix $Q_\epsilon = \sum_{i=1}^p (\lambda_i - \epsilon) v_i v_i^T$. In other words, $(Q_\epsilon)_{jj} = Q_{jj} - \epsilon \sum_{i=1}^p v_{ij}^2$. By the assumption above, Q_ϵ has no zero diagonal entry, so that the case $\sum_{i=1}^p \lambda_i \alpha_i v_{ij} = 0$ cannot occur for any optimal solution y of the problem $\min_{x \in \{-1, 1\}^n} \{x^T Q_\epsilon x\}$.

Proposition 2.2.2. [36] *Setting $y_\epsilon := \text{Argmin}_{x \in \{-1, 1\}^n} (x^T Q_\epsilon x)$ and $y^* := \text{Argmin}_{x \in \{-1, 1\}^n} (x^T Q x)$, then, for $\epsilon > 0$ sufficiently small, the equation $(y^*)^T Q y^* = y_\epsilon^T Q y_\epsilon$ holds.*

Proof. Let $V \in \mathbb{R}^{n \times p}$ denote the matrix whose columns correspond to the vectors v_1, \dots, v_p . From the definition of y_ϵ and y^* we have: $y_\epsilon^T Q_\epsilon y_\epsilon \leq (y^*)^T Q_\epsilon y^*$ and $y_\epsilon^T Q y_\epsilon \geq (y^*)^T Q y^*$ (recall that $Q_\epsilon = Q - \epsilon V V^T$). This leads to: $-\epsilon \left((y^*)^T V V^T y^* - y_\epsilon^T V V^T y_\epsilon \right) \geq y_\epsilon^T Q y_\epsilon - (y^*)^T Q y^* \geq 0$. Using the common denominator D of the entries of vectors v_i and eigenvalues λ_i , $y_\epsilon^T Q y_\epsilon - (y^*)^T Q y^*$ is an integer multiple of $1/D^3$ while $\left((y^*)^T V V^T y^* - y_\epsilon^T V V^T y_\epsilon \right)$ is an integer multiple of $1/D^2$. Let M denote the sum in absolute value of all the numerators of the entries of the matrix $V V^T$. Then taking for ϵ the value $\frac{1}{4MD}$, we get that necessarily $0 \leq y_\epsilon^T Q y_\epsilon - (y^*)^T Q y^* < \frac{1}{D^3}$ implying that $(y^*)^T Q y^* = y_\epsilon^T Q_\epsilon y_\epsilon$, i.e. the vectors y_ϵ and y^* are optimal solutions for problem (QP). Note that ϵ can be encoded polynomially in the size of the instance that is defined by the representation mentioned in point (iii) above. \square

From Propositions 2.1 and 2.2 it follows that in order to find an optimal solution of problem (QP) (potentially considering the matrix Q_ϵ instead of Q as mentioned above), it suffices to enumerate over all vectors $y \in \{-1, 1\}^n$ for which there exists a vector $\alpha \in \mathbb{R}^p$ such that $y_j = -\text{sign}(\sum_{i=1}^p \lambda_i \alpha_i v_{ij})$ (or equivalently $y_j = \text{sign}(\sum_{i=1}^p \lambda_i \alpha_i v_{ij})$, see hereafter), $\sum_{i=1}^p \lambda_i \alpha_i v_{ij} \neq 0$, for all $j \in \{1, \dots, n\}$, with $\text{sign}(x) = 1$ if $x > 0$ and -1 if $x < 0$. In the next section we investigate on finding such a set of vectors.

2.3 Determining cells in an arrangement of n hyperplanes

Let $v_1, \dots, v_p \in \mathbb{R}^n$ denote p independent vectors. Let $V \in \mathbb{R}^{n \times p}$ denote the matrix whose columns correspond to the vectors v_1, \dots, v_p and V_i the i th row of V . From this set of vectors we define n hyperplanes in \mathbb{R}^p : $H_j = \{\alpha \in \mathbb{R}^p \mid V_j \cdot \alpha = 0\}$ with $j \in \{1, \dots, n\}$. Then we can notice that there is a one-to-one correspondence between the set of vectors in $\{-1, 1\}^n$ for which there exists a vector $\alpha \in \mathbb{R}^p$ such that $y_j = \text{sign}(\sum_{i=1}^p \alpha_i v_{ij})$, with $\sum_{i=1}^p \alpha_i v_{ij} \neq 0$, for all $j \in \{1, \dots, n\}$ and the cells (i.e. the full dimensional regions) in \mathbb{R}^p of the hyperplane arrangement $\mathcal{A}(H)$ that is defined by the family of hyperplanes $(H_j)_{j=1}^n$. To see this, just interpret the sign vector y as the position vector of the corresponding cell c w.r.t. an orientation of the space by the vector V_j : cell c is *above* hyperplane H_j iff $y_j > 0$ and *under* otherwise.

For a general arrangement in \mathbb{R}^p that is defined by n hyperplanes (see, e.g., [93, 223] for further elements on arrangements), the number of cells is upper bounded by $\sum_{i=0}^p \binom{n}{i}$ (which is in $\mathcal{O}(n^p)$). (For a proof, we refer the reader, e.g., to Lemma 1.2 in [93]). In our case, since all the hyperplanes considered contain the origin (i.e. the arrangement is *central*), this number reduces to $\mathcal{O}(n^{p-1})$ (see Section 1.7 in [93]).

Two main approaches seem to emerge in the literature for determining the cells in a central arrangement:

- One is based on the incremental algorithm [93, 94]. Here the basic idea consists in iteratively adding an hyperplane to the current arrangement. At each iteration all the regions intersected by the currently added hyperplane are determined and the arrangement is updated accordingly. This algorithm has time complexity $\mathcal{O}(n^{p-1})$. The main disadvantage is a large memory requirement: the space required is proportional to the output size.
- Another approach is based on the reverse search algorithm [16]. Although it has higher time complexity (in $\mathcal{O}(n \text{ LP}(n, p) C)$ where C denotes the number of cells in $\mathcal{A}(H)$ and $\text{LP}(n, p)$ is the time needed to solve a linear program with n inequalities and p variables) it seems to be of more practical interest because of a reduced space complexity that is polynomially bounded in the input size [101].

We describe hereafter a simple procedure with time complexity lying between these two approaches in order to compute a set of vectors in $\{-1, 1\}^n$ corresponding to a description of a set containing the cells of the arrangement $\mathcal{A}(H)$. Space complexity is shown to be polynomially

bounded by the output size. Then one may ask for the interest of this method by comparison with the former ones. To our view it is 2-fold:

1. it is easy to understand and implement,
2. computationally, by using proper data structures (introduced later) we could solve instances of the same magnitude as the ones reported in [101], without parallelization and efficient computation times.

A vector $y \in \{-1, 1\}^n$ will correspond to the polyhedron that is defined by $\{\alpha \in \mathbb{R}^p \mid V_i \cdot \alpha \leq 0, \forall i \in I^- \text{ and } V_i \cdot \alpha \geq 0, \forall i \in I^+\}$, with $I^+ = \{i \in \{1, \dots, n\} \mid y_i = 1\}$ and $I^- = \{1, \dots, n\} \setminus I^+$. So recall that the cells in $\mathcal{A}(H)$ exactly correspond to the full dimensional polyhedra of the latter type. For our purposes, a set of vectors describing the cells is represented in the following way: we have an initial vector $y^0 \in \{-1, 1\}^n$ and \mathcal{L} : a chained list of sets of indices Ind , where Ind denotes a set of indices in $\{1, \dots, n\}$ corresponding to variables y_i that are multiplied by -1 . Starting with y^0 and considering the first subset of indices Ind in the list \mathcal{L} we get a vector $\bar{y} \in \{-1, 1\}^n$ such that $Ind = \{i \in \{1, \dots, n\} \mid y_i \neq \bar{y}_i\}$. Then, replacing y^0 by \bar{y} and iterating this way over all the elements in \mathcal{L} , we get a set of vectors in $\{-1, 1\}^n$ describing polyhedra in the arrangement $\mathcal{A}(H)$ as mentioned above.

Next, we introduce two procedures for determining the cells of an arrangement of n hyperplanes in \mathbb{R}^p : they will correspond to the cases $p = 2$ and $p \geq 2$, respectively. In order to simplify the presentation, we will assume that for each call of these procedures, the input matrix V as introduced before contains no zero nor collinear rows or columns. Such cases can be easily handled:

- a zero row or column in the original matrix V can be ignored. The coefficient associated with a zero row can be set with any value ± 1 .
- among a set of collinear columns just keep a single one.
- among a set of collinear rows just keep a "representative" one in the system and remove all the others after their indices and associated scalar signs w.r.t. the "representative" row have been stored.

We start introducing the procedure for computing a list containing a representation of all the cells in $\mathcal{A}(H)$ for the case $p = 2$.

The case $p = 2$

The procedure is based on the observation that since the arrangement considered here is central, if a vector $y \in \{-1, 1\}^n$ represents a cell, the same holds for the vector $-y$. It follows that in order to determine all the vectors associated with cells it suffices to know all the ones on one side of some hyperplane, since all the others can be obtained by the componentwise opposite vectors.

Assume that we are given two vectors $v_1, v_2 \in \mathbb{R}^n$ and that we want to determine the full dimensional polyhedra in the arrangement $\mathcal{A}(H)$ they define. From the afore mentioned observation, we may firstly find all the full dimensional polyhedra that are *above* the hyperplane H_1 , by considering the arrangement that is defined by H_2, \dots, H_n in the affine subspace $H'_1 = \{x \in \mathbb{R}^2 \mid V_1 \alpha = 1\}$ which is of dimension 1: hence the cells of this arrangement are intervals whose extremities are either unbounded or correspond to an intersection $H'_1 \cap H_j$ for some $j \in \{2, \dots, n\}$.

These intersections may be easily determined: using the equation defining H'_1 and assuming w.l.o.g. that $v_{21} \neq 0$ (possibly by permuting rows and/or columns), we may eliminate variable α_2 from the system of equations $V_j \alpha = 0, j \in \{2, \dots, n\}$, thus obtaining a system $\bar{V}_j \alpha_1 = b_j$, with $(\bar{V}_j, b_j) \in \mathbb{R}^2, j \in \{2, \dots, n\}$. Notice that the assumption on the non-collinearity of the rows namely implies $\bar{V}_j \neq 0, \forall j$, so that the set $F = \left\{ \frac{b_j}{\bar{V}_j} \mid j \in \{2, \dots, n\} \right\}$ is properly defined. Note furthermore that since all the hyperplanes H_2, \dots, H_n pass through the origin, all the values $\frac{b_j}{\bar{V}_j}$ are distinct.

Assume now that we order the values in F : $\frac{b_{i_2}}{\bar{V}_{i_2}} < \dots < \frac{b_{i_n}}{\bar{V}_{i_n}}$, with $i_2, \dots, i_n \in \{2, \dots, n\}$. These values define n intervals in \mathbb{R} : $\left] -\infty, \frac{b_{i_2}}{\bar{V}_{i_2}} \left[, \dots, \left] \frac{b_{i_n}}{\bar{V}_{i_n}}, +\infty \left[$. And note in addition that any cell of $\mathcal{A}(H)$ above H_1 contains a point $\alpha \in \mathbb{R}^2$ with α_1 belonging to any such interval. Let I denote any one of these intervals. For any value $\alpha_1 \in I$ we have either $\bar{V}_j \alpha_1 > b_j$ or $\bar{V}_j \alpha_1 < b_j$ (for any $j \in \{2, \dots, n\}$) and these order relations remain invariant in the interval I . This means that any such interval I corresponds to a cell of the arrangement considered in H'_1 and it can be seen that the converse also holds.

The whole procedure is described in Pseudo-code 1. We have just explained above the two first steps of the procedure.

Step 3 of Pseudo-code 1 sets the values of the integral vector y^0 corresponding to the cell of $\mathcal{A}(H)$ whose intersection with H'_1 corresponds to points $\alpha \in \mathbb{R}^2$ with $\alpha_1 \in \left] -\infty, \frac{b_{i_2}}{\bar{V}_{i_2}} \left[$. Then increasing α_1 , the integral vector denoting the cell of $\mathcal{A}(H)$ whose intersection with H'_1 corresponds to vectors $\alpha \in \mathbb{R}^2$ with $\alpha_1 \in \left] \frac{b_{i_2}}{\bar{V}_{i_2}}, \frac{b_{i_3}}{\bar{V}_{i_3}} \left[$ is determined by simply changing the i_2 -th component of y^0 . This is applied iteratively in order to determine all the cells intersecting H'_1 . In order to get all the other cells of $\mathcal{A}(H)$ it is then left to add to the integral vectors in $\{-1, 1\}^n$ currently represented the opposite of each (which is the last part of step 4 when adding to \mathcal{L} the set of indices $\{1, \dots, n\}$ and the subsets that were already present in \mathcal{L} before this last insertion, in reverse order).

Pseudo-code 1. Determination of the cells of the arrangement $\mathcal{A}(H)$ for $p = 2$.

Input: $v_1, v_2 \in \mathbb{R}^n$, defining a system of equations as presented above (no zero nor collinear rows/columns) and w.l.o.g. such that $v_{21} \neq 0$.

Output: (y^0, \mathcal{L}) : a representation of a set of vectors in $\{-1, 1\}^n$ containing a representation of all the cells of the arrangement defined by v_1, v_2 .

1. Express each equation $V_j \cdot \alpha = 0$, $j \in \{2, \dots, n\}$ in the affine subspace $H'_1 = \{\alpha \in \mathbb{R}^2 \mid V_1 \cdot \alpha = 1\}$, eliminating variable α_2 .

We get $\bar{V}_j \cdot \alpha_1 = b_j$, $j \in \{2, \dots, n\}$, $b_j \in \mathbb{R}$.

2. Compute the set $\mathcal{F} = \left\{ \left(\frac{b_j}{\bar{V}_j}, j \right) \mid j = 2, \dots, n \right\}$ and order its elements by increasing value $\frac{b_j}{\bar{V}_j}$;

3. Set the initial vector y^0 corresponding to the interval $\alpha_1 \in \left] -\infty, \min_j \frac{b_j}{\bar{V}_j} \left[$:

Set $y_1^0 := 1$;

Set $y_j^0 := -\text{sgn}(\bar{V}_j)$, $\forall j \in \{2, \dots, n\}$;

Initialize \mathcal{L} with the ordered indices in \mathcal{F} ;

4. Add $\{1, \dots, n\}$ to the end of the list \mathcal{L} . Then add all the elements of \mathcal{L} except the last one (i.e. $\{1, \dots, n\}$) in reverse order at its end.

END.

Example. We illustrate Pseudo-code 1 on a small example with $p = 2$ and $n = 3$. The original system of equations is as follows:

$$\begin{cases} \alpha_2 = 0 \\ \alpha_1 = 0 \\ \alpha_1 - \alpha_2 = 0 \end{cases}$$

The situation is depicted in Figure 1.

There is no zero nor collinear rows in the original system of equations $\mathcal{S}_{3,2} = \{V_j \cdot \alpha = 0, j \in \{1, 2, 3\}, \alpha \in \mathbb{R}^2\}$. Eliminating variable α_2 by considering the equation $\alpha_2 = 1$ corresponding to the hyperplane represented with a dotted line in Figure 2.1, we obtain the vector $y^0 = (1, -1, -1)$. The set \mathcal{F} is given by: $\{(0, \{2\}), (1, \{3\})\}$. The output list we get is therefore: $\mathcal{L} = (\{2\}, \{3\}, \{1, 2, 3\}, \{3\}, \{2\})$. Hence the set of vectors corresponding to cells and represented by (y^0, \mathcal{L}) is $\{(1, -1, -1), (1, 1, -1), (1, 1, 1), (-1, -1, -1), (-1, -1, 1), (-1, 1, 1)\}$. So the first half of vectors represented correspond to

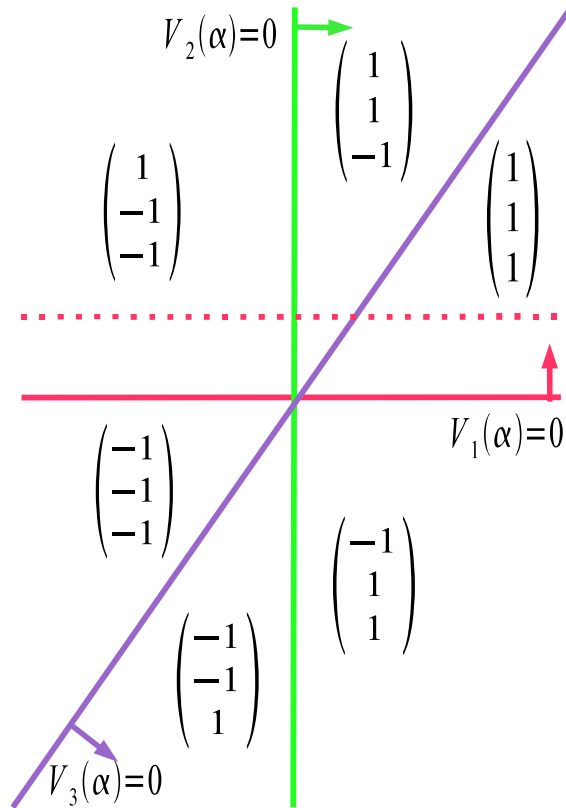


Figure 2.1 – Illustration for Example 1

the cells of the arrangement intersecting the hyperplane $H'_1 = \{\alpha \in \mathbb{R}^2 \mid V_1 \cdot \alpha = 1\}$ (represented with a dotted line in Figure 2.1), whereas the second half corresponds to the cells intersecting $H''_1 = \{\alpha \in \mathbb{R}^2 \mid V_1 \cdot \alpha = -1\}$.

The case $p \geq 2$

Next we introduce the procedure for the general case $p \geq 2$. As for the case $p = 2$, in order to simplify the presentation we do not consider in the given pseudo-code the case when a zero row or collinear rows/columns occur in the input matrix V .

The basic principle lying behind Pseudo-code 2 can be expressed as follows. Given some integer $q \in \{1, \dots, n\}$, let $\mathcal{A}^q(H)$ denote the arrangement in the subspace $\{\alpha \in \mathbb{R}^p \mid V_q \alpha = 0\}$ that is defined by the hyperplanes (in \mathbb{R}^{p-1}) $\{H_j \cap H_q \mid j \in \{1, \dots, n\}, j \neq q\}$. Any cell of $\mathcal{A}^q(H)$ (which is a region of dimension $p - 1$) corresponds to a facet of exactly two cells c_1 and c_2 of $\mathcal{A}(H)$. Since each cell of $\mathcal{A}(H)$ intersects at least one hyperplane H_q , it is clear that by varying the value of q , all cells will be generated at least once.

Pseudo-code 2. Determination of the cells of the arrangement $\mathcal{A}(H)$ for $p \geq 2$.

Input: $v_1, \dots, v_p \in \mathbb{R}^n$

Output: (y^0, \mathcal{L}_p) : a representation of a set of vectors in $\{-1, 1\}^n$ containing a representation of all the cells of the arrangement defined by v_1, \dots, v_p .

1. If $p = 2$ then

 Apply pseudo-code 1 $\rightarrow (y^0, \mathcal{L}_p)$ and return ;

2. Else

 For i from 1 to n do

 Express the system of equations $\{V_j \alpha = 0, j \in \{1, \dots, n\}, j \neq i, \alpha \in \mathbb{R}^p\}$ in the subspace

$H_i = \{\alpha \in \mathbb{R}^p \mid V_i \alpha = 0\}$ and let $\{\bar{V} \bar{\alpha} = 0, \bar{\alpha} \in \mathbb{R}^{p-1}\}$ be the resulting system after row i and some column have been removed ($\bar{V} \in \mathbb{R}^{(n-1) \times (p-1)}$).

 Make a recursive call to pseudo-code 2 with input parameters the columns of $\bar{V} \rightarrow (\bar{y}_i^0, \bar{\mathcal{L}}_i)$;

 Express \bar{y}_i^0 in \mathbb{R}^n by adding one entry set to 1 (corresponding to index i) removed from V to get \bar{V} above) using the equation defining H_i ;

 For each subset of indices Ind that is represented in $\bar{\mathcal{L}}_i$ do

 Increment by one all the indices in Ind that are greater than or equal to i ;

 Insert $\{i\}$ before Ind ;

 End_for ;

 Insert $\{i\}$ in the last position of the list $\bar{\mathcal{L}}_i$;

 End_for ;

3. For i from 1 to $n - 1$ do

 Add to $\bar{\mathcal{L}}_i$ in the last position the set Ind where Ind denotes the set of entries of \bar{y}_{i+1}^0 that are different from those of the last element represented by $(\bar{y}_i^0, \bar{\mathcal{L}}_i)$

 End_for ;

4. Set $y^0 := \bar{y}_1^0$ and concatenate the lists $\bar{\mathcal{L}}_1, \dots, \bar{\mathcal{L}}_n$ (in this order) $\rightarrow \mathcal{L}_p$;

 End_if ;

END.

Let us state more precisely what is done in step 2. Each equation of the system $\bar{V} \bar{\alpha} = 0$ defines an hyperplane of the subspace H_i . Then, the recursive calls result in a description $(\bar{y}_i^0, \bar{\mathcal{L}}_i)$ of the cells in H_i that are induced by the hyperplanes $H_i \cap H_j$, $j \neq i$. The addition of the singleton i after each element in the list representing some cell c in H_i , leads to the generation of one cell above and another one under H_i in the original space, each having c as a facet. Also, and since the description of the cells in H_i corresponds to a set of vectors in \mathbb{R}^{n-1} we have to consider one additional entry corresponding to the i th component of the description vectors of the cells in the original space \mathbb{R}^n . Since the j th row of the system $\bar{V} \bar{\alpha} = 0$ corresponds to the j th (resp. $(j + 1)$ th) hyperplane of the original system $V \alpha = 0$ if $j < i$ (resp. $j \geq i$), indices greater than or equal to i in the list $\bar{\mathcal{L}}_i$ have to be incremented by one.

2.4 Polynomially solvable cases for problem (QP)

Recall that given $Q = \sum_{i=1}^p \lambda_i v_i v_i^T$ satisfying the assumptions introduced at the beginning of Section 2.2, an optimal solution of problem (QP) corresponds to a cell of the central arrangement $\mathcal{A}(H)$ that is defined by the vectors $(\lambda_i v_i)_{i=1}^p$. Using this feature, the Pseudo-code 3 describes how to compute Z^* . For shortness, we do not mention the code for returning an optimal solution in the output since this is trivial and does not affect the complexity results to be given next.

Pseudo-code 3. Solving (QP)

Input: a matrix (given by its spectral decomposition) $Q = \sum_{i=1}^p \lambda_i v_i v_i^T$

Output: the value Z^*

1. Apply Pseudo-code 2 with input the vectors $(\lambda_i v_i)_{i=1}^p \rightarrow (y^0, \mathcal{L})$;
 2. Set $y := y^0$;
 3. Compute $\alpha_k := (y^0)^T v_k, \forall k \in \{1, \dots, p\}$;
 4. Set $\bar{Z} := \sum_{k=1}^p \lambda_k \alpha_k^2$;
 5. For each index set Ind that is represented in \mathcal{L} do
 - For each index $j \in Ind$ do
 - $y_j := -y_j$;
 - $\alpha_k := \alpha_k + 2y_j v_{kj}, \forall k \in \{1, \dots, p\}$;
 - End_for;
 - Set $Z := \sum_{k=1}^p \lambda_k \alpha_k^2$;
 - If $Z < \bar{Z}$ then set $\bar{Z} := Z$; End_if;
 - End_For;
 - Return \bar{Z} ;
- END.
-

Remark. The approach presented here for solving (QP) is suitable for parallelization. Also Pseudo-code 2 can be sequentially applied on the projection of the original system on each hyperplane $H_i, i \in \{1, \dots, n-1\}$ and the $n-1$ lists of integer vectors stemming from these calls could be processed sequentially between each such call and removed immediately, thus reducing memory requirements.

We next show that for fixed p , problem (QP) with a matrix Q satisfying the conditions introduced in Subsection 2.2 can be solved in strongly polynomial time. We start giving a preliminary result on the size of the output from recursive calls to Pseudo-code 2.

Lemma 2.4.1. [36] For $p \geq 2$, the total number of indices (i.e. the sum of the cardinalities of all the sets of indices) that are present in any list \mathcal{L}_i in step 4 of Pseudo-code 2 is $\mathcal{O}(n^{p-1})$.

Proof. We give a proof by induction on p .

Initialization: $p = 2$. It can be easily shown that the total number of indices (with multiplicities) that are contained in the output list of any call to Pseudo-code 1 is upper bounded by $3n$ and is thus $\mathcal{O}(n)$.

Now assume $p > 2$ and that the proposition holds until $p-1$ instead of p . In step 2 of Pseudo-code 2 the number of indices contained in each list $\mathcal{L}_i, i \in \{1, \dots, n-1\}$ increases by $|\mathcal{L}_i| + 1$ (where $|\mathcal{L}_i|$ stands for the total number of sets of indices in \mathcal{L}_i), which is $\mathcal{O}(n^{p-2})$ by the induction assumption. In step 4, by concatenating the lists $\mathcal{L}_1, \dots, \mathcal{L}_{n-1}$ results in a list containing $\mathcal{O}(n^{p-1})$ indices. \square

Proposition 2.4.2. [36] For fixed $p \geq 2$, if the matrix Q (given by its nonzero eigenvalues and associated eigenvectors) has rank at most p and nonpositive diagonal entries only, then problem (QP) can be solved in time $\mathcal{O}(n^{p-1} \log(n))$.

Proof. The proof we give is by induction on p .

Initialization: $p = 2$. In Pseudo-code 1 the steps 1 and 3 can be performed in time $\mathcal{O}(n)$, while step 2 can be done in time $\mathcal{O}(n \log(n))$: For each index $j \in \{2, \dots, n\}$, we compute $\frac{b_j}{v_j}$ and then order these values. The latter takes time $\mathcal{O}(n \log(n))$. Step 4 can be performed in time $\mathcal{O}(n)$, which follows from Lemma 2.4.1. Considering now Pseudo-code 3, trivially the steps 2, 3 and 4 can be performed

in time $\mathcal{O}(n)$ while using Lemma 2.4.1 we can see that step 5 can be done in time $\mathcal{O}(n^{p-1})$ (p is here equal to 2).

Now assume $p > 2$ and that the proposition holds until $p-1$ instead of p . Let us firstly consider the steps involved in Pseudo-code 2. By the induction assumption, in step 2 each recursive call takes $\mathcal{O}(n^{p-2} \log(n))$ time. In addition, from Lemma 2.4.1 the total number of indices contained in any list \mathcal{L}_i is $\mathcal{O}(n^{p-2})$. It follows that all the other steps of Pseudo-code 2 but also those of Pseudo-code 3 can be performed in time $\mathcal{O}(n^{p-1} \log(n))$.

From the former we get that the number of elementary arithmetic operations is polynomially bounded. It is then left to show that the size of the numbers occurring in the course of the algorithm is polynomially bounded by the size of the input. As to the numbers occurring in the systems considered in step 2 of Pseudo-code 2, the latter can be shown analogously to the proof of polynomiality of the Gaussian elimination method [95]. In fact, there are at most $p-1$ consecutive projections that are performed when our procedure is applied (since the value of p decreases after each projection). This is why the numbers involved in the algorithm remain under control (remember that p is fixed). \square

In case the matrix Q has positive diagonal entries, let I_{pos} stand for the set of indices in $\{1, \dots, n\}$ corresponding to positive diagonal entries in Q . Then both implications (1) – (2) may not hold for indices $j \in I_{pos}$. To deal with this situation $2^{|I_{pos}|-1}$ calls to Pseudo-code 3 would suffice: for each combination of values in $\{-1, 1\}$ for the entries with index in I_{pos} (except one, since by a symmetry argument we can take any fixed entry with index in I_{pos} to take value 1), previously to each call, fix the corresponding entries of the vectors to be considered in Pseudo-code 3. Then we get the following result.

Theorem 2.4.3. [36] *For a fixed integer $p \geq 2$, if the matrix Q (given by its nonzero eigenvalues and associated eigenvectors) has rank at most p and $\mathcal{O}(\log(n))$ positive diagonal entries, then problem (QP) can be solved in strongly polynomial time.*

Çela, Klinz and Meyer [56] independently showed that problem (QP) can be solved in polynomial time if, for p fixed, the matrix Q has rank at most p and all the optimal solutions of the continuous relaxation are in $\{-1, 1\}^n$. The latter case namely contains the instances of (QP) with Q having rank p and all diagonal entries *strictly* negative (see Proposition 2.2.1 and the discussion following it).

Detailed computational experiments are reported in [36]. Summarizing, the recursive approach presented above appears to be more efficient than reverse search w.r.t. computation times. The main drawback of our method w.r.t. the former are important memory requirements which limit the size of the instances to which the method applies (instances with $n \leq 200$ for $p = 3$, $n \leq 30$ for $p = 6, \dots$ see [36] for details).

2.5 Some perspectives

We introduced a new approach for solving in polynomial time unconstrained quadratic optimization problems of the form: $\min \{x^T Q x \mid x \in \{-1, 1\}^n\}$, with Q a symmetric matrix with fixed rank and whose number of positive diagonal entries is $\mathcal{O}(\log(n))$. The recursive procedure presented here can be a valuable approach on some instances by comparison with a reverse search w.r.t. computation times. However, for the time being, due to important memory requirements, it could only be applied to rather small instances.

Further computational studies could be carried out, involving a parallelization of the code as mentioned in Section 2.4 in order to deal with larger instances.

Also, one may look for other families of quadratic optimization problems that can be solved in polynomial time (possibly containing the one described in this chapter?).

Chapter 3

Spectral bounds for unconstrained $(-1, 1)$ -quadratic optimization problems

In this chapter we review works we carried out dealing with the problem which consists in optimizing a quadratic function, subject to variables taking their values in the set $\{-1, 1\}$, and some extensions. Differently from the last chapter, we shall not make any particular assumption on the matrix defining the objective, so that the recursive method we previously introduced cannot be applied in general to solve the problem exactly. We shall see, however, that it can be used in different ways to get approximations on the optimal objective value for general instances. After we formally introduce a new family of bounds (Sections 3.1 and 3.2), we mention some connections with a semidefinite bound from the literature in Section 3.3. Then we draw our attention to the spectral bounds more thoroughly when applied to the weighted maximum cut problem (Section 3.4) and its generalization: the weighted maximum k -cut problem (Section 3.5). We then briefly sketch some observations from the computational experiments that could be done so far (Section 3.6), before we terminate with some perspectives in Section 3.7.

The work presented in this chapter is joint with Prof. Ben-Ameur [34–36], except for Section 3.5 which stems from a joint work with Prof. Anjos [6].

3.1 Introduction

Just like in Chapter 2, we shall focus here on unconstrained $(-1, 1)$ -quadratic optimization problems expressed as follows:

$$(QP) Z^* = \min\{q(x) \mid x \in \{-1, 1\}^n\}.$$

where $q: \mathbb{R}^n \rightarrow \mathbb{R}$ stands for a quadratic function given by: $q(x) = x^T Q x$, with Q representing an $n \times n$ rational and symmetric matrix (the assumption of being symmetric is with no loss of generality). We refer to the beginning of Chapter 2 for further references on this problem. Differently from the last chapter, however, we shall not consider particular conditions on Q (so that problem (QP) may not be solvable in polynomial time assuming $P \neq NP$), and we are now interested in computing bounds on the optimal objective value Z^* of (QP) (rather than solving it exactly).

Different methods for computing bounds for problems such as (QP) have been proposed in the literature. An early reference is Hammer and Rubin's paper [124] in which the authors proposed a method convexifying the objective function by making use of the smallest eigenvalue of the matrix Q . This approach has then been generalized and improved by many people (see, e.g., [44, 85, 86, 198] leading to bounds equivalent to the ones obtained by a semidefinite formulation presented in [114]. Further improvements over the latter have been introduced by Malik et al. [163].

Let the eigenvalues of the matrix Q be denoted by $\lambda_1(Q) \leq \lambda_2(Q) \leq \dots \leq \lambda_n(Q)$ (or more simply $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ when clear from the context). The corresponding unit (in Euclidean norm) and pairwise orthogonal eigenvectors are denoted by v_1, \dots, v_n . For our purposes, we shall assume in the rest of the chapter, unless otherwise stated, that all the eigenvalues and eigenvectors

are given and rational. The j th entry of the vector v_i is denoted by v_{ij} . Given some set of vectors $a_1, \dots, a_q \in \mathbb{R}^n$, $q \in \mathbb{N}$, let $\text{Lin}(a_1, \dots, a_q)$ represent the subspace spanned by these vectors. Given some vector $y \in \{-1, 1\}^n$, $\text{dist}(y, \text{Lin}(v_1, \dots, v_p))$ stands for the Euclidean distance between the vector y and $\text{Lin}(v_1, \dots, v_p)$, i.e. $\text{dist}(y, \text{Lin}(v_1, \dots, v_p)) = \|y - y_p\|_2$ where y_p stands for the orthogonal projection of y onto $\text{Lin}(v_1, \dots, v_p)$ and $\|\cdot\|_2$ represents the Euclidean norm. Given some index $j \in \{1, \dots, n\}$, d_j denotes the distance between the set $\{-1, 1\}^n$ and the subspace that is spanned by the eigenvectors v_1, \dots, v_j , i.e. $\min\{\text{dist}(y, \text{Lin}(v_1, \dots, v_j)) \mid y \in \{-1, 1\}^n\}$. Notice that d_j depends on a particular spectral decomposition of the matrix Q when there is an eigenvalue with multiplicity greater than one: considering different orders for the eigenvectors associated with the same eigenvalue generally leads to different values of d_j . Analogously, \bar{d}_j will denote the distance between the set $\{-1, 1\}^n$ and $\text{Lin}(v_j, \dots, v_n)$.

By using the property $x_i^2 = 1, \forall i \in \{1, \dots, n\}$ for any vector $x \in \{-1, 1\}^n$, we notice that the set of optimal solutions of the problem (QP) remains unchanged if diagonal entries of the matrix Q are modified. More precisely, let $u \in \mathbb{R}^n$, $\text{Diag}(u) \in \mathbb{R}^{n \times n}$ stand for the matrix with diagonal u and all the other entries set equal to zero, and denote by $(\text{QP})_u$ the $(-1, 1)$ -quadratic problem: $\bar{Z} = \min\{x^T(Q + \text{Diag}(u))x \mid x \in \{-1, 1\}^n\}$. Then trivially, we have: $Z^* = \bar{Z} - \sum_{i=1}^n u_i$. However altering the diagonal entries of the matrix Q generally changes its spectrum, i.e. eigenvalues and eigenvectors. The bounds that we introduce hereafter rely on the spectrum of the matrix Q , and applying them to the matrix $Q + \text{Diag}(u)$ instead of Q , we can still derive bounds that are valid for the original problem. Naturally they depend on the vector u that is used and a still open question consists in determining the *best* such vector.

In order to simplify the presentation we consider Q (rather than $Q + \text{Diag}(u)$) as an input matrix for which we compute bounds for the corresponding problem (QP), since for the case we use $Q + \text{Diag}(u)$ it is then trivial to derive bounds for the original problem. In the next section, we present some methods for computing spectral bounds on the optimal objective of (QP).

3.2 Computing bounds for problem (QP)

In this section we introduce three different ways of computing bounds for problem (QP), all using the eigenvalues and eigenvectors of the matrix Q (possibly with modified diagonal entries). The basic spectral bounds of Section 3.2.1 have an expression involving eigenvalues among $(\lambda_i)_{i=1}^n$ and distances among $(d_i)_{i=1}^{n-1}$. Another method for computing bounds from a substitution of the matrix Q by a sum of particular matrices is proposed in Section 3.2.2. Basic idea here is to replace the original problem (QP) by several instances each satisfying Proposition 2.4.2. A different approach is undertaken in Section 3.2.3 where the original matrix Q is replaced by a single matrix satisfying Proposition 2.4.2 and whose spectrum differs from Q in some subset of eigenvalues.

3.2.1 Basic spectral bounds

Using the smallest eigenvalue of the matrix Q the following lower bound trivially holds: $Z^* \geq \lambda_1 n$. This bound can be strengthened by using the whole spectrum of the matrix Q .

Proposition 3.2.1. [35] *The following inequality holds: $Z^* \geq \lambda_1 n + \sum_{j=1}^{n-1} d_j^2 (\lambda_{j+1} - \lambda_j)$.*

Proof. Consider a vector $y \in \{-1, 1\}^n$, and its expression in a basis of eigenvectors: $y = \sum_{i=1}^n \alpha_i v_i$. Then we namely have: $y^T y = n = \sum_{i=1}^n \alpha_i^2$. Also from the definition of the distances d_j , $j \in \{1, \dots, n\}$, the following inequality holds: $d_j^2 \leq \sum_{i=j+1}^n \alpha_i^2$.

We have $y^T Q y = \sum_{i=1}^n \alpha_i^2 \lambda_i = \lambda_1 n + \sum_{i=2}^n (\lambda_i - \lambda_1) \alpha_i^2$. Now, from the last inequality mentioned above with $j = 1$ we get: $\alpha_2^2 \geq d_1^2 - \sum_{i=3}^n \alpha_i^2$. Hence we have:

$$y^T Q y \geq \lambda_1 n + (\lambda_2 - \lambda_1) d_1^2 - (\lambda_2 - \lambda_1) \sum_{i=3}^n \alpha_i^2 + \sum_{i=3}^n (\lambda_i - \lambda_1) \alpha_i^2,$$

$$\Leftrightarrow y^T Q y \geq \lambda_1 n + (\lambda_2 - \lambda_1) d_1^2 + \sum_{i=3}^n (\lambda_i - \lambda_2) \alpha_i^2,$$

and, inductively (by using analogously the inequality $\alpha_j^2 \geq d_{j-1}^2 - \sum_{i=j+1}^n \alpha_i^2$), we get:

$$y^T Q y \geq \lambda_1 n + \sum_{i=1}^{n-1} (\lambda_{i+1} - \lambda_i) d_i^2. \quad \square$$

In the same way that we derived lower bounds on the optimal objective value, upper bounds can be obtained. Using the largest eigenvalue λ_n of the matrix Q we get: $Z^* \leq n\lambda_n$. A better upper bound using the whole spectrum is as follows.

Proposition 3.2.2. [35] *The following inequality holds: $Z^* \leq \lambda_n n + \sum_{j=1}^{n-1} \bar{d}_{j+1}^2 (\lambda_j - \lambda_{j+1})$.*

Thus the results from Propositions 3.2.2 and 3.2.1 lead to the *spectral gap*:

$$(\lambda_n - \lambda_1) n - \sum_{j=1}^{n-1} (\lambda_{j+1} - \lambda_j) (d_j^2 + \bar{d}_{j+1}^2). \quad (3.1)$$

Let Z_c^* stand for the optimal objective value of the following (continuous) relaxation of problem (QP): $\min \{x^T Q x \mid x \in [-1, 1]^n\}$ denoted (QPC) in what follows. In the particular case when the matrix Q has at least one negative eigenvalue, then by computing the objective value of a properly scaled eigenvector associated with a negative eigenvalue, we get the simple upper bound given hereafter.

Proposition 3.2.3. [35] *If the matrix Q has at least one negative eigenvalue then this upper bound holds:*

$$Z_c^* \leq \min \left\{ \frac{\lambda_q}{\|v_q\|_\infty^2} \mid \lambda_q < 0 \right\}, \quad (3.2)$$

with $\|v_q\|_\infty = \max_{i \in \{1, \dots, n\}} |v_{qi}|$.

An upper bound for problem (QP) can be obtained similarly.

Proposition 3.2.4. [35] *The following upper bound holds for problem (QP):*

$$Z^* \leq \min \left\{ \frac{\lambda'_q}{\|v'_q\|_\infty^2} \mid \lambda'_q < 0 \right\} + \sum_{i=1}^n \bar{q}_i, \quad (3.3)$$

where $\lambda'_1, \lambda'_2, \dots, \lambda'_n$ denote the eigenvalues of the matrix $Q - \text{Diag}(\bar{q})$, v'_1, v'_2, \dots, v'_n are corresponding unit eigenvectors and $\bar{q} = (\bar{q}_1, \dots, \bar{q}_n)$ stands for the diagonal of the matrix Q .

A geometric view to the spectral bound of Proposition 3.2.1 in a special case is provided by the following result.

Proposition 3.2.5. [35] *If a vector $y \in \{-1, 1\}^n$ satisfies: $\text{dist}(y, \text{Lin}(v_1, \dots, v_k)) = d_k$ for all indices $k \in \{1, \dots, n-1\}$ such that $\lambda_{k+1} > \lambda_k$ then y is an optimal solution of problem (QP).*

From Proposition 3.2.5 and its proof, it follows that the spectral bound from Proposition 3.2.1 coincides with the optimal objective value Z^* iff there exists a vector y verifying the assumptions formulated there. We mention hereafter an easy special case for which the spectral bound coincides with Z^* .

Corollary 3.2.6. [35] *If the matrix Q has at most two distinct eigenvalues then the spectral bound of Proposition 3.2.1 coincides with the optimal objective value Z^* .*

The following Proposition shows that it is generally NP-hard to compute the distances involved in the expression of the bound from Proposition 3.2.1.

Proposition 3.2.7. [34] *Computing the distance d_{n-1} is NP-hard.*

Proof. We may consider the following decision problem (\mathcal{D}) that is related to d_{n-1} : given unit and pairwise orthogonal vectors v_1, v_2, \dots, v_n , is it true that $d_{n-1} = 0$? (Or equivalently, does there exist a vector y in $\{-1, 1\}^n$ such that $y^T v_n = 0$?)

We show this problem is NP-complete by reduction from the partition problem (also called *number partitioning problem*) which consists in determining whether some given set of positive numbers a_1, a_2, \dots, a_n can be partitioned into two subsets S_1 and S_2 , such that the sum of the

numbers in S_1 equals the sum of the numbers in S_2 . The partition problem is a well-known NP-complete problem and its optimization version which consists in finding a partition (S_1, S_2) of the numbers such that the difference between the sum of the elements in S_1 and the sum of the elements in S_2 is minimized, is NP-hard.

Let us consider the squared matrix of order n having all the entries equal to zero. All the eigenvalues are equal to zero. We can consider that the vector $a \in \mathbb{R}^n$ having a_i for i th entry, is (up to a multiplicative factor) the eigenvector corresponding to λ_n : $v_n = \frac{a}{\|a\|}$. Let v_1, v_2, \dots, v_n denote any set of unit and pairwise orthogonal vectors in \mathbb{R}^n that are also orthogonal to v_n . Then, if there were a polynomial algorithm for (\mathcal{D}) , one could determine whether there exists a vector $y \in \{-1, 1\}^n$ which is orthogonal to a . Thus the partition problem could be solved in polynomial time. \square

Despite the fact that it is generally difficult to compute all the distances $(d_j)_{j \in [1, n-1]}$, some of them can be computed in polynomial time. We can easily get the following expression for the squared distance $d_p^2 = n + \min_{y \in \{-1, 1\}^n} y^T (-\sum_{i=1}^p v_i v_i^T) y$. Since the matrix that is given by $-\sum_{i=1}^p v_i v_i^T$ satisfies the conditions of Theorem 2.4.3, we get the following result.

Corollary 3.2.8. [35] For fixed p , the terms d_1^2, \dots, d_p^2 can be computed in polynomial time.

3.2.2 Spectral bounds by decomposing the matrix Q

Recall that v_1, \dots, v_n denote the unit and pairwise orthogonal eigenvectors of the matrix $Q \in \mathbb{R}^{n \times n}$ and $\lambda_1 \leq \dots \leq \lambda_n$ the corresponding eigenvalues. Consider some set of r matrices $(Q_q)_{q=1}^r$, $r \in \mathbb{N}$, $Q_q \in \mathbb{R}^{n \times n}$ of the following form: $Q_q = \alpha_q \sum_{i \in I_q} v_i v_i^T$, with $I_q \subseteq \{1, \dots, n\}$ and $\alpha_q \in \mathbb{R}$ a constant associated to I_q , $q = 1, 2, \dots, r$, such that $\sum_{q: i \in I_q} \alpha_q \leq \lambda_i$, for all $i \in \{1, \dots, n\}$.

Then for any vector $y \in \mathbb{R}^n$ we have $y^T Q y \geq \sum_q y^T Q_q y$. $(y^T Q y = \sum_{i=1}^n \lambda_i (y^T v_i)^2 \geq \sum_{i=1}^n \sum_{q: i \in I_q} \alpha_q (y^T v_i)^2 = \sum_q \alpha_q \sum_{i \in I_q} (y^T v_i)^2 = \sum_q y^T Q_q y)$, thus leading to the lower bound:

$$Z^* \geq \sum_q \min_{y \in \{-1, 1\}^n} y^T Q_q y. \quad (3.4)$$

An observation to be made here is that a subproblem $\min_{y \in \{-1, 1\}^n} y^T Q_q y$ is *easy*, in the sense of polynomial time solvability, when:

- (i) $|I_q| \leq p$ and $\alpha_q < 0$ $\left(\min_{y \in \{-1, 1\}^n} y^T Q_q y = \alpha_q \left(n - \text{dist}(\{-1, 1\}^n, \text{Lin}(\{v_i\}_{i \in I_q})^2) \right) \right)$
- (ii) $|I_q| \geq n - p$ and $\alpha_q > 0$ $\left(\min_{y \in \{-1, 1\}^n} y^T Q_q y = \alpha_q \text{dist}(\{-1, 1\}^n, \text{Lin}(\{v_i\}_{i \notin I_q})^2) \right)$

for some fixed integer p , since both cases reduce to solving a problem satisfying the conditions of Theorem 2.4.3. (In case (i) the matrix Q_q has rank at most p and nonpositive diagonal entries only. For case (ii) note that for any vector $y \in \{-1, 1\}^n$ we have $y^T Q y = n\alpha_q + y^T Q'_q y$ with $Q'_q = -\alpha_q \sum_{i \in I_q} v_i v_i^T$: a matrix with rank at most p and nonpositive diagonal entries only).

We shall now consider the problem that consists in looking for the best bound that can be obtained from formula (3.4) by using matrices Q_q fitting into the cases (i), (ii) for some prescribed value p . This value is introduced in order to limit the computations, and trivially the larger p is, the better the bound one may expect to obtain by using (3.4), considering all potential candidate matrices Q_q .

Let us introduce some additional notation. We define $S_p = \{I \subseteq \{1, \dots, n\} \mid 1 \leq |I| \leq p\}$, and set $\bar{I} = \{1, \dots, n\} \setminus I$, $d_I^2 = \text{dist}(\{-1, 1\}^n, \text{Lin}(\{v_i\}_{i \in I}))^2$. Given some matrix $Q_q = \alpha_q \sum_{i \in I_q} v_i v_i^T$, $I_q \in S_p$, $\alpha_q \in \mathbb{R}_-$, we have $\min_{y \in \{-1, 1\}^n} y^T Q_q y = \alpha_q (n - d_{I_q}^2)$. Then, in order to get the best bound in (3.4) over all possible decompositions involving matrices Q_q as described above the following program can be solved:

$$(LP1) \begin{cases} \max \sum_{I \in S_p} (\alpha_{\bar{I}} d_{\bar{I}}^2 - \alpha_I (n - d_I^2)) \\ s.t. \\ \sum_{I \in S_p} (\alpha_{\bar{I}} \chi^{\bar{I}} - \alpha_I \chi^I) \leq \lambda, \\ \alpha_I, \alpha_{\bar{I}} \in \mathbb{R}_+, \forall I \in S_p, \end{cases}$$

where λ denotes the column vector $(\lambda_1, \dots, \lambda_n)^T$, $\chi^I \in \{0, 1\}^n$ stands for the incidence vector of the set I (and inequalities for vectors are taken componentwise).

We may notice that (LP1) has the same objective value as the following formulation with one variable added:

$$(LP1') \begin{cases} \max \sum_{I \in S_p} (\alpha_{\bar{I}} d_I^2 - \alpha_I (n - d_I^2)) - n\theta \\ s.t. \\ \sum_{I \in S_p} (\alpha_{\bar{I}} \chi^{\bar{I}} - \alpha_I \chi^I) - \theta \bar{1} \leq \lambda, \\ \alpha_I, \alpha_{\bar{I}} \in \mathbb{R}_+, \forall I \in S_p, \theta \in \mathbb{R}_+, \end{cases}$$

where $\bar{1}$ stands for the n -dimensional all ones vector. Considering any feasible solutions (α, θ) of (LP1') with $\theta > 0$, then by decreasing α_I and increasing $\alpha_{\bar{I}}$ of an amount θ we get a feasible solution for (LP1) with an identical objective value.

Now, if we consider a feasible solution (α, θ) of (LP1') with $\alpha_I > 0$ for some set I , then by increasing $\alpha_{\bar{I}}$ and θ by an amount α_I and setting α_I to value zero we get another feasible solution with the same objective value. This observation leads to the following equivalent reformulation of (LP1):

$$(DECOMP-LP) \begin{cases} \max \sum_{I \in S_p} \alpha_{\bar{I}} d_I^2 - n\theta \\ s.t. \\ \sum_{I \in S_p} \alpha_{\bar{I}} \chi^{\bar{I}} - \theta \bar{1} \leq \lambda \\ \theta \in \mathbb{R}_+, \alpha_{\bar{I}} \in \mathbb{R}_+, \forall I \in S_p, \end{cases}$$

(i.e. all former variables of the form α_I are now somehow implicitly considered through the single variable θ). For p fixed, since the set S_p is of polynomial cardinality (in $\mathcal{O}(n^p)$), it follows that (DECOMP-LP) is of polynomial size and hence that it can be solved in polynomial time. However, considering the fact that its total number of columns grows exponentially with p , it may be convenient to solve it using a column generation procedure (see, e.g., [158] and the references therein). (We did so for our computational experiments.)

3.2.3 Spectral bounds by restricted eigenvalue selection

Theorem 2.4.3 may suggest computing upper and lower bounds for general $(-1, 1)$ -quadratic optimization problems as follows.

For some fixed integer p , $1 \leq p \leq n-1$, select a set I of $n-p$ indices in $\{1, \dots, n\}$, set $\bar{I} = \{1, \dots, n\} \setminus I$, $\lambda_{\min} := \min_{i \in I} \lambda_i$, $\lambda_{\max} := \max_{i \in I} \lambda_i$ and define $Q_{\bar{I}} \in \mathbb{Q}^{n \times n}$ as the matrix given by $Q_{\bar{I}} = \sum_{i \in \bar{I}} \lambda_i v_i v_i^T$. Then, setting $Q^{\inf} := Q_{\bar{I}} + \lambda_{\min} \sum_{i \in I} v_i v_i^T$ and $Q^{\sup} := Q_{\bar{I}} + \lambda_{\max} \sum_{i \in I} v_i v_i^T$, for any vector $y \in \{-1, 1\}^n$, we have: $y^T Q^{\inf} y \leq y^T Q y \leq y^T Q^{\sup} y$. Note also that $y^T Q^{\inf} y = y^T Q_{\bar{I}}^{\inf} y + n\lambda_{\min}$ with $Q_{\bar{I}}^{\inf} = \sum_{i \in \bar{I}} (\lambda_i - \lambda_{\min}) v_i v_i^T$ and that the matrix $Q_{\bar{I}}^{\inf}$ has rank at most p . In case all the entries on the diagonal of the matrix $Q_{\bar{I}}^{\inf}$ are nonpositive, then it follows from Theorem 2.4.3 that the lower bound $\min_{y \in \{-1, 1\}^n} y^T Q^{\inf} y$ can be computed in polynomial time. An analogous result holds for an upper bound, by considering the matrix Q^{\sup} .

Note that, differently from the spectral bounds introduced above, both the upper and lower bounds introduced here converge to the optimal objective value Z^* when the value of the parameter p increases.

Remark. We may notice that for any value of the parameter p , the bounds provided by the latter approach by taking $I = \{p+1, \dots, n\}$ improve over the restriction of the basic bound to the first p terms in the sum involving the distances (see Proposition 3.2.1). To see this, considering the bound obtained using Q^{\inf} with $I = \{p+1, \dots, n\}$, for any vector $y \in \{-1, 1\}^n$ we have:

$$\begin{aligned} y^T Q^{\inf} y &= \sum_{i=1}^p \lambda_i (v_i^T y)^2 + \lambda_{p+1} \sum_{i=p+1}^n (v_i^T y)^2 \\ &= \lambda_1 n - \lambda_1 \sum_{k=2}^n (v_k^T y)^2 + \sum_{k=2}^p \lambda_k (y^T v_k)^2 + \lambda_{p+1} \sum_{k=p+1}^n (v_k^T y)^2 \\ &= \lambda_1 n - \lambda_1 \sum_{k=2}^n (v_k^T y)^2 + (\sum_{k=2}^p \lambda_k \sum_{i=k}^n (y^T v_i)^2) - (\sum_{k=2}^p \lambda_k \sum_{i=k+1}^n (y^T v_i)^2) \\ &\quad + \lambda_{p+1} \sum_{k=p+1}^n (v_k^T y)^2 \\ &= n\lambda_1 + \sum_{i \in \bar{I}} (\lambda_{i+1} - \lambda_i) (\sum_{k=i+1}^n (v_k^T y)^2) \\ &\geq n\lambda_1 + \sum_{i=1}^p (\lambda_{i+1} - \lambda_i) d_i^2 \end{aligned}$$

where the last inequality follows from the fact that $d_i^2 \leq \sum_{k=i+1}^n (v_k^T y)^2 = \text{dist}(y, \text{Lin}(v_1, \dots, v_i))^2$, for any $y \in \{-1, 1\}^n$.

Let (QP^{inf}) denote the problem $\min_{y \in \{-1, 1\}^n} y^T Q y$ and Z^{inf} stand for its optimal objective value. The following result indicates some cases when the bound Z^{inf} is exact.

Proposition 3.2.9. [35] *If $\lambda_{\min} = \min_{i \in I} \lambda_i$, $I_{\min} = \{j \in I \mid \lambda_j = \lambda_{\min}\}$, then the lower bound Z^{inf} is exact iff the set of optimal solutions of (QP^{inf}) is contained in $\text{Lin}(\{v_i \mid i \in \bar{I} \cup I_{\min}\})$.*

3.3 Improvement over a classical semidefinite bound

Given any vector $u \in \mathbb{R}^n$, analogously to Proposition 3.2.1, we can show:

$$Z^* \geq \lambda'_1 n - \sum_{i=1}^n u_i + \sum_{j=1}^{n-1} (d'_j)^2 (\lambda'_{j+1} - \lambda'_j) \quad (3.5)$$

with $d'_j = \min \{ \text{dist}(y, \text{Lin}(v'_1, \dots, v'_j)) \mid y \in \{-1, 1\}^n \}$, v'_1, \dots, v'_n denote unit and pairwise orthogonal vectors of the matrix $Q + \text{Diag}(u)$ and $\lambda'_1 \leq \dots \leq \lambda'_n$ stand for the corresponding eigenvalues.

Notice that replacing u by $u + \xi \vec{1}$, $\xi \in \mathbb{R}$, leaves the right-hand side of the last inequality unchanged (since the eigenvectors remain unchanged and each eigenvalue is increased by an amount ξ), so that any bound provided by the last formula can always be obtained by considering either of the two following restrictions:

- i) $\lambda'_1 = 0$, or
- ii) $\sum_{i=1}^n u_i = 0$.

Trivially the following lower bound holds for the problem (QP):

$$Z^* \geq \lambda'_1 n - \sum_{i=1}^n u_i. \quad (3.6)$$

A vector u leading to the best bound that can be derived from this formula can be obtained by solving the following problem:

$$(P1) \begin{cases} \max n \lambda_1(Q + \text{Diag}(u)) - \sum_{i=1}^n u_i \\ u \in \mathbb{R}^n, \end{cases}$$

which can be reformulated as the following semidefinite program:

$$(P2) \begin{cases} \max n \lambda_{\min} - \sum_{i=1}^n u_i \\ Q + \text{Diag}(u) \geq \lambda_{\min} I, \\ u \in \mathbb{R}^n, \lambda_{\min} \in \mathbb{R}, \end{cases}$$

where I denotes the identity matrix with order n . (To see that (P2) is equivalent to (P1), note that since the constraint in (P2) expresses that the matrix $Q + \text{Diag}(u) - \lambda_{\min} I$ must be positive semidefinite, it follows that in any optimal solution λ_{\min} will correspond to the smallest eigenvalue of the matrix $Q + \text{Diag}(u)$). Now, considering the restriction $\lambda_{\min} = 0$ in the last formulation (which does not change the optimal objective value), the dual problem can be expressed as follows:

$$(P3) \begin{cases} \min \text{Trace}(Y^T Q) \\ \text{diag}(Y) = \vec{1}, \\ Y \geq 0, \\ Y \in \mathcal{S}_n, \end{cases}$$

where \mathcal{S}_n denotes the set of symmetric matrices with order n . It is easy to see that Slater's conditions are satisfied (see, e.g., [208]), so that the optimal objective values of problems (P2) and (P3)

coincide. Then the formula (3.6) can be seen to provide a classical semidefinite bound for (QP). It hence follows that if u^* stands for an optimal solution of problem (P2), then the basic spectral bound stemming from formula (3.5) by setting $u := u^*$ is better than, or equal to this semidefinite bound. Note that for $u = u^*$ and $r := \operatorname{argmax}_j \{\lambda'_j \mid \lambda'_j = 0\}$ the restriction of the formula (3.5) to the first r terms in the last sum coincides with the bound introduced by Malik et al. [163].

3.4 Applications to the weighted maximum cut problem

Given an undirected edge-weighted and simple graph $G = (V, E)$ with node set $V = \{1, \dots, n\}$, and edge weights $(w_e)_{e \in E}$, the (weighted) maximum cut problem can be formulated as follows:

$$(MC) \begin{cases} w^* = \max \frac{1}{2} \sum_{1 \leq i < j \leq n} w_{ij} - \frac{1}{4} y^T W y \\ y \in \{-1, 1\}^n, \end{cases}$$

where W stands for the weight matrix of the graph G , i.e. for the entry corresponding to the i th row and j th column of the matrix W we have: $W_{ij} = w_{ij}$ if $ij \in E$ and 0 otherwise. The particular case when $w_{ij} = 1$ if $ij \in E$ and 0 otherwise will be called *unweighted*.

In this section, we consider with some more details the application of the *basic* spectral bounds introduced in Section 3 to the maximum cut problem. The reader is referred to Section 4 for a general introduction to this problem and linear programming approaches to solve it.

In their seminal paper [114], Goemans and Williamson introduced a 0.87856-approximation algorithm for the maximum cut problem with nonnegative weights only. It is based on the following semidefinite relaxation of the problem:

$$(MC_SDP) \begin{cases} \max \frac{1}{2} \sum_{1 \leq i < j \leq n} w_{ij} - \frac{1}{2} \sum_{1 \leq i < j \leq n} w_{ij} y_i y_j \\ Y_{ii} = 1, \forall i, \\ Y \geq 0, \\ Y \in \mathcal{S}_n. \end{cases}$$

Let Z_{GW}^* stand for the optimal objective value of (MC_SDP). It is proved in [199] that the eigenvalue bound of [85] is equal to Z_{GW}^* . It is known that the integrality gap (i.e. the ratio of the upper bound Z_{GW}^* to the optimum weight) of the formulation (MC_SDP) can approach 1.139 ($\approx \frac{1}{0.87856}$) arbitrarily closely as the size of the graph increases [98]. The semidefinite relaxation (MC_SDP) is strengthened, for example in [103, 131, 201], and in [7] by using a lift-and-project technique.

Considering the matrix $W + \operatorname{Diag}(u)$ for any vector $u \in \mathbb{R}^n$, and resuming the notation given in Section 3.3 with $Q := W$, the spectral bound corresponding to formula (3.5) leads to the following upper bound for the optimal objective value of the formulation (MC) (see [34]).

$$(MC_BD)_u \quad w^* \leq \frac{1}{2} \sum_{i < j} w_{ij} + \frac{1}{4} \sum_{i=1}^n u_i - \lambda_1 \frac{n}{4} - \frac{1}{4} \sum_{i=1}^{n-1} (d'_i)^2 (\lambda'_{i+1} - \lambda'_i) \quad (3.7)$$

Notice that considering the formulation (P3) introduced in Section 3.3 (setting $Q := W$), we can derive the classical SDP relaxation of the maximum-cut problem [86, 198], namely used by Goemans and Williamson [114] in their approximation algorithm. Note also that, from the observation mentioned in Section 3.3 concerning the invariance of the optimal objective value of the formulation (P1), by translation along the all-ones vector solutions, we can easily see that we can equivalently work with the Laplacian matrix L of the graph ($L = \operatorname{Diag}(d) - W$, with $d_i = \sum_{j=1}^n w_{ij}$) instead of the weight matrix, i.e. the eigenspaces of the matrices $W + \operatorname{Diag}(u^*)$ (for some optimal solution u^* of (P1)) and $L + \operatorname{Diag}(z^*)$ coincide, where z^* denotes an optimal solution of the following eigenvalue optimization problem introduced by Mohar and Poljak [17]:

$$(LMC) \begin{cases} \min \frac{n}{4} \lambda_n(L + \operatorname{Diag}(z)) \\ s.t. \\ \sum_{i=1}^n z_i = 0, \\ z \in \mathbb{R}^n. \end{cases}$$

(Given z^* , it can be easily shown that an optimal solution of (P1) is given by the vector $\bar{u} = -z^* - d$). In particular all the properties that Delorme and Poljak [85, 86] established relatively to the eigenspace associated with the largest eigenvalue of $L + \text{Diag}(z^*)$ apply in our case as well (i.e. relatively to the eigenspace associated with the smallest eigenvalue of $W + \text{Diag}(u^*)$). Among others, their work directly leads to the following result. (An automorphism of a graph $G = (V, E)$ is a permutation π of the vertices such that $(i, j) \in E \iff (\pi(i), \pi(j)) \in E$. Two vertices $i, j \in V$ belong to the same orbit of G iff $\pi(i) = j$ for some π in the automorphism group of the graph G .)

Proposition 3.4.1. [86, 198] *Let u^* stand for an optimal solution of the formulation (P1) with $Q := W$. If the vertices i and j belong to the same orbit of the graph, then $u_i^* = u_j^*$. In particular for the case of vertex-transitive graphs we have $u^* = \alpha \vec{1}$, $\alpha \in \mathbb{R}$.*

We mention hereafter a simple result that may be useful in some particular cases, in order to get a lower bound on some of the distances involved in the expressions of the spectral bounds introduced above.

Lemma 3.4.2. [35] *The following relation holds:*

$$\min \{ \text{dist}^2(y, \text{Lin}(\vec{1})^\perp) \mid y \in \{-1, 1\}^n \} = \begin{cases} \frac{1}{n} & \text{if } n \text{ odd,} \\ 0 & \text{otherwise.} \end{cases} \quad (3.8)$$

Proof. Let $y \in \{-1, 1\}^n$ with exactly p entries having the same value, $p \in \{1, \dots, n\}$. W.l.o.g. assume $y_1 = y_2 = \dots = y_p = 1$ and $y_{p+1} = \dots = y_n = -1$. We are looking for a vector $u \in (\text{Lin}(\vec{1}))^\perp$ that is closest in Euclidean distance to y .

Note that if n is even, setting $p = \frac{n}{2}$ the vector y defined above solves the problem:

$$\text{dist}^2(y, \text{Lin}(\vec{1})^\perp) = 0.$$

Assuming now n is odd, and noting that $\text{dist}^2(y, \text{Lin}(\vec{1})^\perp) = \frac{|\sum_{i=1}^n y_i|^2}{n} = \frac{|2p-n|^2}{n}$ we see that the minimum is obtained with $p \in \{\lfloor \frac{n}{2} \rfloor, \lceil \frac{n}{2} \rceil\}$, and the result follows. \square

A notable consequence of Lemma 3.4.2 is given by the following proposition that could be used in some particular cases so as to get a positive lower bound on some distances d_k .

Corollary 3.4.3. [35] *Assume n is odd and $\vec{1}$ is an eigenvector of Q with associated eigenvalue λ_k . Then the following inequality holds: $d_i^2 \geq \frac{1}{n}$, $\forall i \in \{1, \dots, k-1\}$.*

In what follows we will call a bound *exact* whenever it coincides with the optimal objective value of the maximum cut problem w^* .

Proposition 3.4.4. [35] *The bound $(\text{MC_BD})_{u=0}$ is exact for complete graphs (unweighted case).*

Notice that from Proposition 3.4.1 vertex-transitive and regular graphs with odd order are suitable for the application of Proposition 3.4.3. This way, though the exact distance values may be unknown, a potential improvement of the semidefinite bound may still be obtained using this information. Another relevant case for which the spectral bound is exact is the graph C_5 (i.e. cycle of length 5).

Proposition 3.4.5. *The spectral bound $(\text{MC_BD})_{u=0}$ is exact for C_5 .*

The Proposition 3.4.5 definitely draws a strong contrast between the spectral and classical semidefinite bounds. Indeed, for C_5 the ratio between the optimal objective value of the classical semidefinite relaxation and the optimum of the maximum cut problem is about ≈ 1.131 . (It was even conjectured for some time that C_5 would give the worst case ratio for the semidefinite bound in the unweighted case, which was however disproved later by Feige and Schechtman [98].)

More generally for odd cycles, the following proposition gives an upper bound (generally weaker than the exact value of the spectral bound but) strictly improving over the upper bound provided by the classical SDP relaxation (and that is given by: $\frac{n}{2} (1 + \cos(\frac{\pi}{n}))$).

Proposition 3.4.6. [35] *The spectral bound $(MC_BD)_{u=0}$ for an odd cycle C_n is lower than or equal to the following quantity:*

$$\frac{n}{2} \left(1 + \cos\left(\frac{\pi}{n}\right) \right) - \frac{1}{n} \sin\left(\frac{\pi}{n}\right) \sum_{j=1}^{(n-1)/2} \sin\left(\frac{2j\pi}{n}\right). \quad (3.9)$$

Another interesting particular case is given hereafter illustrating the fact that the lower bound on w^* that may be derived from formula (3.2), i.e.

$$w^* \geq \frac{1}{2} \sum_{i < j} w_{ij} - \frac{1}{4} \min_{\lambda_q < 0} \left\{ \frac{\lambda_q}{\|v_q\|_\infty^2} \right\}, \quad (3.10)$$

where the minimum is taken over all the negative eigenvalues of the matrix W , may coincide with the optimal objective value.

Proposition 3.4.7. [35] *The lower bound (3.10) is exact in the unweighted case for wheels with an even number of spokes.*

3.5 Further extensions for the weighted maximum k -cut problem

In this section we introduce spectral bounds for the (*weighted*) *maximum k -cut problem* (or *max k -cut*) which generalizes the maximum cut problem, where k denotes an integer larger than one. The objective is to find a partition of the node set into k subsets so as to maximize the sum of the weights of the edges having endpoints in different subsets. (So maximum cut is nothing but max 2-cut.)

Among the main developed approaches for dealing with this problem, some works [188, 213] aimed at deriving bounds on the optimal objective value by making use of the spectrum of the Laplacian or of the weighted adjacency matrix. We summarize hereafter a recent work investigating further this line of research. Despite the fact that (differently from the maximum cut problem) for $k \geq 3$ max k -cut cannot be formulated as an unconstrained $(-1, 1)$ -quadratic optimization problem, we shall see, however that the approach developed above to get spectral bounds may be simply extended for this problem.

In what follows, we make use of (and not recall all) the notation introduced in the former sections of this chapter. Let us just mention some additional one and recall some elements to avoid any ambiguity. Given a positive integer q , let $[q]$ stand for the set of integers $\{1, 2, \dots, q\}$. Given any partition (V_1, V_2, \dots, V_k) of V into k subsets V_1, V_2, \dots, V_k (some of which may be empty), the k -cut defined by this partition is the set $\delta(V_1, V_2, \dots, V_k)$ of edges in E having their endpoints in different sets of the partition, and the weight of the k -cut is the sum of the weights of the edges it contains. The maximum weight of a k -cut in G is denoted w_k^* . Given two disjoint node subsets A, B , let $w[A, B]$ denote the sum of the weights of the edges having one endpoint in A and the other in B : $w[A, B] = \sum_{(i,j) \in A \times B: ij \in E} w_{ij}$. Similarly, $w[A]$ denotes the sum of the weights of the edges with both endpoints in A : $w[A] = \sum_{\substack{(i,j) \in A^2: \\ ij \in E, i < j}} w_{ij}$. Recall that $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ denote the eigenvalues of

W and v_1, v_2, \dots, v_n the corresponding unit and pairwise orthogonal eigenvectors. The maximum eigenvalue of the weighted Laplacian matrix is denoted by $\lambda_n(L)$.

For the particular case when $k = 2$, Mohar and Poljak [17] proved the inequality $w_2^* \leq \frac{n}{4} \lambda_n(L)$. More recently, van Dam and Sotirov [213] proved the following upper bound on w_k^* , still making use of the largest eigenvalue of the Laplacian and providing in the same reference several graphs for which this bound is tight together with some comparisons with other bounds stemming from semidefinite formulations.

Theorem 3.5.1. [213]

$$w_k^* \leq \frac{n(k-1)}{2k} \lambda_n(L). \quad (3.11)$$

Also recently, Nikiforov [188] introduced an upper bound for the maximum cardinality of a k -cut in G (i.e., the maximum k -cut problem with $w_e = 1, \forall e \in E$) that is easily extended to the weighted case and can be expressed as follows.

Theorem 3.5.2. [188]

$$w_k^* \leq \frac{k-1}{k} \left(w[V] - \frac{\lambda_1 n}{2} \right). \quad (3.12)$$

As noted in [188], the bounds from Theorems 3.5.1 and 3.5.2 are equivalent for regular graphs but different in general.

In what follows we extend the result in Theorem 3.5.2 by using the full spectral information of W , following the approach described in Section 3.4 for the maximum cut problem.

3.5.1 Spectral bounds

Given $r \in \mathbb{R} \setminus \{0, 1\}$, let $d_{j,r}$ denote the distance between the set of vectors $\{r, 1\}^n$ and the subspace $\text{Lin}(v_1, v_2, \dots, v_j)$ that is generated by the first j eigenvectors of W :

$$d_{j,r} = \min \{ \|z - y\| : z \in \{r, 1\}^n, y \in \text{Lin}(v_1, v_2, \dots, v_j) \}. \quad (3.13)$$

Theorem 3.5.3. [6] For any $r \in \mathbb{R} \setminus \{0, 1\}$,

$$w_k^* \leq \frac{1}{2(r-1)^2} \left((r^2 + k - 1)(2w[V] - \lambda_1 n) - k \sum_{l \in [n-1]} (\lambda_{l+1} - \lambda_l) d_{l,r}^2 \right). \quad (3.14)$$

Proof. Let (V_1, V_2, \dots, V_k) denote a partition of V corresponding to an optimal solution of the maximum k -cut problem. For all $i \in [k]$, let the vector $y^i \in \{r, 1\}^n$ be defined as follows: $y_l^i = r$ if $l \in V_i$ and 1 otherwise. We have:

$$\begin{aligned} (y^i)^T W y^i &= 2r^2 w[V_i] + 2 \sum_{j \in [k] \setminus \{i\}} w[V_j] + 2r \sum_{j \in [k] \setminus \{i\}} w[V_i, V_j] + \\ &\quad 2 \sum_{\substack{(j,l) \in ([k] \setminus \{i\})^2 \\ j < l}} w[V_j, V_l]. \end{aligned} \quad (3.15)$$

Let us now compute the sum of each term occurring in the right-hand-side of (3.15) over all $i \in [k]$.

$$\begin{aligned} \sum_{i \in [k]} 2r^2 w[V_i] &= 2r^2 (w[V] - w_k^*), \\ \sum_{i \in [k]} 2 \sum_{j \in [k] \setminus \{i\}} w[V_j] &= 2(k-1) (w[V] - w_k^*), \\ \sum_{i \in [k]} 2r \sum_{j \in [k] \setminus \{i\}} w[V_i, V_j] &= 4r w_k^*, \\ \sum_{i \in [k]} 2 \sum_{\substack{(j,l) \in ([k] \setminus \{i\})^2 \\ j < l}} w[V_j, V_l] &= 2(k-2) w_k^*. \end{aligned}$$

Thus, we deduce

$$\sum_{i \in [k]} (y^i)^T W y^i = 2w_k^* (-r^2 + 2r - 1) + 2w[V] (r^2 + k - 1). \quad (3.16)$$

We now derive a lower bound on $(y^i)^T W y^i$ making use of the spectrum of W . First, we mention some preliminary properties. Note that since W is symmetric we may assume (v_1, v_2, \dots, v_n) forms an orthonormal basis, and consider the expression of y^i in this basis: $y^i = \sum_{l \in [n]} \alpha_l v_l$ with $\alpha \in \mathbb{R}^n$. Then, we have $\|y^i\|^2 = \sum_{l \in [n]} \alpha_l^2 = n + |V_i|(r^2 - 1)$. From the definition of the distance defined above we deduce $d_{j,r}^2 \leq \sum_{l=j+1}^n \alpha_l^2, \forall j \in [n-1]$. Thus, we have

$$\begin{aligned} (y^i)^T W y^i &= \sum_{l \in [n]} \lambda_l \alpha_l^2 \\ &= \lambda_1 (n + |V_i|(r^2 - 1) - \sum_{l=2}^n \alpha_l^2) + \sum_{l=2}^n \lambda_l \alpha_l^2 \\ &= \lambda_1 (n + |V_i|(r^2 - 1)) + \sum_{l=2}^n (\lambda_l - \lambda_1) \alpha_l^2. \end{aligned}$$

Then, iteratively making use of the inequality $\alpha_j^2 \geq d_{j-1,r}^2 - \sum_{l=j+1}^n \alpha_l^2$ for $j = 2, \dots, n$, we deduce

$$(y^i)^T W y^i \geq \lambda_1 (n + |V_i|(r^2 - 1)) + \sum_{l \in [n-1]} (\lambda_{l+1} - \lambda_l) d_{l,r}^2.$$

Summing up these inequalities for all $i \in [k]$ we obtain

$$\sum_{i \in [k]} (y^i)^T W y^i \geq \lambda_1 n (k + r^2 - 1) + k \left(\sum_{l \in [n-1]} (\lambda_{l+1} - \lambda_l) d_{l,r}^2 \right). \quad (3.17)$$

Combining (3.16) and (3.17), the result follows. \square

Remark. Enforcing the value ‘1’ among the two possible values for the components of the vectors used in the definition of the distances (3.13) is done only for simplicity of the presentation. We are basically interested in the distance between $\text{Lin}(v_1, v_2, \dots, v_j)$ and a set of vectors whose components are restricted to take any of two nonzero values. If we denote by d_{j,r_1,r_2} the distance between $\text{Lin}(v_1, v_2, \dots, v_j)$ and the set of vectors $\{r_1, r_2\}^n$ with $(r_1, r_2) \in (\mathbb{R} \setminus \{0\})^2$, then $d_{j,r_1,r_2} = |r_1| d_{j, \frac{r_2}{r_1}}$, $\forall j \in [n]$, and the results obtained by using such vectors are equivalent to the ones presented.

Note that all the terms occurring in the last sum of the inequality (3.14) are nonnegative, so that even after removing from the right-hand side some or all of the terms involved in this sum, the expression obtained still provides an upper bound on w_k^* .

In view of the bound (3.14) on w_k^* , one may ask for the best choice for the parameter r . If we consider the truncated bound obtained from (3.14) by removing the last sum, it is straightforward to check that the ratio $\frac{r^2+k-1}{(r-1)^2}$ is minimized for $r = 1 - k$, which is the value used by Nikiforov [188] and leads to Theorem 3.5.2. One may ask for the best choice of r when considering the whole expression of the bound in (3.14). Preliminary computational experiments show that other values of r may lead to strictly better bounds than Theorems 3.5.1 and 3.5.2, depending on the instance.

The approach for proving Theorem 3.5.3 can also be used to obtain lower bounds on the weight of any k -cut. Let l_k^* denote the minimum weight of a k -cut in G and let $\bar{d}_{j,r}$ denote the distance between the set of vectors $\{r, 1\}^n$ and the subspace $\text{Lin}(v_j, v_{j+1}, \dots, v_n)$ that is generated by the last $n - j + 1$ eigenvectors of W :

$$\bar{d}_{j,r} = \min \{ \|z - y\| : z \in \{r, 1\}^n, y \in \text{Lin}(v_j, v_{j+1}, \dots, v_n) \}. \quad (3.18)$$

Theorem 3.5.4. [6]

$$l_k^* \geq \frac{1}{2(r-1)^2} \left((r^2 + k - 1)(2w[V] - \lambda_n n) + k \sum_{l \in [n-1]} (\lambda_{l+1} - \lambda_l) \bar{d}_{l+1,r}^2 \right). \quad (3.19)$$

Theorems 3.5.3 and 3.5.4 lead to the definition of the *spectral bound gap* s_k , which is the difference between the upper and lower spectral bounds:

$$s_k = \frac{1}{2(r-1)^2} \left[(r^2 + k - 1) n (\lambda_n - \lambda_1) - k \sum_{l \in [n-1]} (\lambda_{l+1} - \lambda_l) (\bar{d}_{l+1,r}^2 + d_{l,r}^2) \right].$$

3.5.2 On some particular cases

Computing the distances $(d_{j,r})_{j=1}^{n-1}$ involved in the expression of the bound (3.14) is NP-hard in general (see Proposition 4.4 in [34]). In this section we specialize Theorem 3.5.3 for the particular case when $\vec{1}$ is an eigenvector of W . This is notably the case when considering the max k -cut problem in regular graphs with unit edge weights. Its expression does not involve distances and leads to an upper bound on w_k^* that is lower than or equal to the bounds of Theorems 3.5.1 and 3.5.2.

We start with an auxiliary result on the minimum squared distance between any vector in $\{1, r\}^n$ and the subspace in \mathbb{R}^n that is orthogonal to $\text{Lin}(\vec{1}_n)$, denoted by $\text{Lin}(\vec{1})^\perp$.

Proposition 3.5.5. [6]

$$\min \{ \|y - z\|^2 : y \in \{1, r\}^n, z \in \text{Lin}(\vec{1}_n)^\perp \} = \begin{cases} n & \text{if } r \geq 1, \\ nr^2 & \text{if } 0 < r < 1, \\ \min\left(\frac{(s+r-1)^2}{n}, \frac{s^2}{n}\right) & \text{otherwise,} \end{cases}$$

with $n \equiv s \pmod{1-r}$, $0 \leq s < 1-r$, for the case when $r < 0$.

Taking $r = 1 - k$, using Theorem 3.5.3 and Proposition 3.5.5 together with the fact that $d_{j,r} \geq d_{j+1,r}$, for all $j \in [n - 1]$ leads to the next result.

Corollary 3.5.6. [6] *If G is a complete graph and W is its weighted adjacency matrix, then*

$$w_k^* \leq \frac{1}{2k} \left((k-1)n^2 - \min((s-k)^2, s^2) \right), \quad (3.20)$$

with $n \equiv s \pmod k$, $0 \leq s < k$.

Corollary 3.5.6 gives an infinite class of graphs (complete graphs such that $\min((s-k)^2, s^2) > 0$) where Theorem 3.5.3 strictly improves on Theorem 3.5.2. The bound (3.20) has also the feature of coinciding with the optimal objective value of max k -cut for some cases. Before we give an example, let us firstly mention the next result w.r.t. the number of edges in Turán's graphs, i.e. complete k -partite graphs for integer $k \geq 2$ whose partition sets differ in cardinality by at most one.

Proposition 3.5.7. *The number of edges of a k -partite Turán graph with n vertices is*

$$\frac{1}{2k} \left((k-1)n^2 + s^2 - sk \right),$$

with $n \equiv s \pmod k$, $0 \leq s < k$.

By Turán's theorem [212] and Proposition 3.5.7, the maximum cardinality of a k -cut in the complete graph K_n is $\frac{1}{2k} \left((k-1)n^2 + s^2 - sk \right)$, with $n \equiv s \pmod k$, $0 \leq s < k$. This coincides with the bound (3.20) if $n \equiv 0 \pmod k$ or $n \equiv \frac{k}{2} \pmod k$ and k is even. For $k = 2$ and unit edge weights it follows that the bound (3.20) coincides with w_k^* for all complete graphs (see also Proposition 3.4.4 in [35]), whereas this fails for the bounds of Theorems 3.5.1 and 3.5.2 for complete graphs having an odd number of vertices.

3.6 Some elements on computational experiments

Some computational experiments involving the spectral bounds introduced in Section 3.2 have been carried out on small weighted maximum cut problem instances (with up to 20 nodes), see [35] for details. From these, it appears that, in order to observe improvements (by comparison with the classical semidefinite bound), it is important to make use of modified diagonal entries for the weight matrix (as mentioned in Section 3.3 by solving the semidefinite program (P2)). This observation holds for all the bounds that have been presented in Section 3.2. For a fixed value of the parameter p , the bound provided by the *decomposition* approach tends to be better than the one obtained by *restricted eigenvalue selection*. The quality of the bounds given by the *basic* approach when restricting the last sum in formula (3.5) to the first p terms (so that the distances computed to evaluate the bound involve matrices of rank at most p) and the one given by the *decomposition* approach tend to be comparable, slightly better for the *decomposition* approach. Note however that the latter seems computationally viable only for very moderate values of the parameter p and that the computational effort grows quickly with this parameter.

3.7 Some perspectives

In this chapter we introduced new families of bounds for unconstrained $(-1, 1)$ -quadratic optimization problems and considered more closely their application to the max-cut and further extensions to the max k -cut problems.

Present joint work with Prof. Anjos is being dedicated to computational experiments on spectral bounds for max k -cut. This research works covers two main features. One is that we wish to deal with larger instances than the ones considered in our former experiments for the max-cut

problem (up to 20 nodes). To achieve this aim a different approach for computing the distances will be carried out: convert the problem which consists in computing the distances into a maximum cut problem and apply Rendl et al.'s method [201] method (a branch-and-cut type algorithm relying namely on the use of strengthened semidefinite relaxations). The second and natural feature of this research is to study the parameterizations (i.e. the setting of diagonal entries for the weight matrix, as mentioned in Section 3.3 and of the parameter r used in the definition of the distances as presented in Section 3.5) leading to the best results.

Another research perspective is to look for other possible extensions of the techniques we used to derive spectral bounds. We namely started to obtain results for the graph partitioning problem in a graph, which consists in partitioning the node set into subsets of prescribed sizes, so as to minimize or maximize the sum of the weights of the edges having their endpoints in the same subset.

In a probably longer term, and depending on the results obtained along the different research directions we just mentioned above, it may be worth studying the potential use of such spectral bounds within the framework of branch-and-bound or branch-and-cut type algorithms.

Part II

Graph optimization problems and related polyhedral structures

Chapter 4

Uniform cut polytopes

In this chapter, we draw our attention to the polyhedral structure of uniform cut polytopes. The latter correspond to the convex hull of the incidence vectors of all the cuts that are defined by a node subset of given cardinality. We gather and summarize selected results from our three papers [182, 184, 185]. The layout is as follows. After we formally introduce this family of polytopes together with some related works from the literature, we sketch our contributions w.r.t. linear descriptions (providing several new families of facet-defining inequalities) and studies on the diameter of such polytopes.

4.1 Introduction

Let G denote an undirected graph with node set $V(G) = \{v_1, v_2, \dots, v_n\}$ and edge set $E(G) = \{e_1, e_2, \dots, e_m\}$. Given a node set $S \subseteq V(G)$, let $\delta_G(S)$ (or $\delta(S)$ when G is clear from the context) stand for the *cut* that is defined by S , i.e. the set of edges in $E(G)$ with exactly one endpoint in S : $\delta_G(S) = \{e \in E(G) : |e \cap S| = 1\}$. The node sets S and $V(G) \setminus S$ are called the *shores* or *sides* of the cut $\delta(S)$ and the *size* of the cut $\delta(S)$ is the cardinality of the node set of one shore: $|S|$ or $n - |S|$. If $0 < |S| < |V|$, the cut $\delta(S)$ is said to be *proper*. The set of edges in the graph $G = (V, E)$ with both endpoints in $S \subseteq V$ is denoted by $E_G(S)$. Given two sets A and B , the notation $A \Delta B$ represents their symmetric difference, i.e. $A \Delta B = (A \cup B) \setminus (A \cap B)$. For any node subsets $T, U \subseteq V$, $\delta(T) \Delta \delta(U) = \delta(T \Delta U)$, hence the collection of cuts is closed under taking symmetric differences (see for example [206]). A cut I is said to be *minimal* if there does not exist any cut $J \subsetneq I$.

Given a set F of edges in G , its incidence vector $\chi^F \in \mathbb{R}^{E(G)}$ is defined by $\chi_e^F = 1$ if $e \in F$ and 0 otherwise. Given a node $v \in V(G)$, $N(v)$ denotes the set of nodes in $V(G) \setminus \{v\}$ that are adjacent to v in G . Given $S \subseteq V$, $G[S]$ denotes the subgraph of G that is induced by the node set S .

The convex hull of the incidence vectors of all the cuts in G defines the *cut polytope* $\text{CUT}(G) = \text{conv}(\{\chi^{\delta_G(S)} : S \subseteq V\})$. Optimization problems over cut polytopes arise, e.g., in statistical physics [22, 80] or circuit design [119] (for further information on properties and applications of these families of polytopes the reader may consult, e.g., [88]). For an introduction to polytopes and related notions for which we shall not recall the definition, the reader may consult, e.g., the textbooks by Grünbaum [122] and Ziegler [223]. Let us just recall here few basic elements on polytopes. Let $P \neq \emptyset$ denote a polyhedron defined as follows: $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ with $(A, b) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m$, m and n stand for positive integers. A set F is a *face* of P if and only if $F \neq \emptyset$ and $F = \{x \in P : \bar{A}x = \bar{b}\}$, for some subsystem $\bar{A}x \leq \bar{b}$ of $Ax \leq b$. A face of P having dimension 0 (resp. 1, $\dim(P) - 1$) is also called a *vertex* (resp. an *edge*, a *facet*) of P . Two vertices of a polyhedron P are called *adjacent* or *neighboring* if they are both contained in an edge of P .

In this chapter we shall focus on restrictions of the cut polytope defined as follows. A *uniform cut polytope* is the convex hull of all cuts with a prescribed size. Given some integer k satisfying $k \leq \lfloor \frac{n}{2} \rfloor$, $\text{CUT}_k(G)$ denotes the convex hull of the incidence vectors of all the cuts for which one shore has cardinality k , i.e. $\text{CUT}_k(G) = \text{conv}(\{\chi^{\delta_G(S)} : S \subseteq V(G), |S| = k\})$, where, for some given subset of edges $F \subseteq E(G)$, $\chi^F \in \mathbb{R}^{E(G)}$ denotes the incidence vector of F (i.e. $\chi_e^F = 1$ if $e \in F$ and 0

otherwise). In the particular case when the graph G is K_n : the complete graph with order n , we shall use the notation V_n (resp. E_n) for the node set (resp. edge set) and write CUT_k^n in lieu of $CUT_k(G)$.

A motivation underlying our investigations on uniform cut polytopes is to develop knowledge with respect to the polyhedral structure of cut polyhedra with potential applications to solve difficult graph partitioning problems. Among the latter we can namely cite the maximum cut and equipartition problems. Both may be formulated by linear programs having for feasible region the cut polytope and the equipartition polytope (i.e. $CUT_{\lfloor \frac{n}{2} \rfloor}(G)$), respectively. Another related polytope is the *bisection cut polytope*: given nonnegative integer node weights $(w_v)_{v \in V(G)}$ and a nonnegative integer q , the bisection cut polytope $BC(G)$ is defined as the convex hull of the incidence vectors of all the cuts $\delta(S)$ satisfying the inequalities $\sum_{v \in S} w_v \leq q$ and $\sum_{v \in V(G) \setminus S} w_v \leq q$. For the particular case when all the node weights have value 1, $BC(G)$ coincides with the cut polytope if $q \geq \sum_{v \in V(G)} w_v$ and the equipartition polytope $CUT_{\lfloor \frac{n}{2} \rfloor}$ if $q = \lceil \frac{n}{2} \rceil$.

Given that optimizing a linear function over these polytopes is NP-hard in general [111], there is no *decent* complete descriptions for them, in the sense that the separation problem associated to these polytopes is NP-hard. This aroused many polyhedral studies (e.g., [3, 10, 24, 25, 70–72, 79, 81, 82, 88, 99, 100]) aiming at determining families of facet-defining inequalities for these polytopes (and some extensions) which could then be used in order to obtain tight relaxations in order to solve optimization problems involving them.

Obviously, as uniform cut polytopes are all contained in the cut polytope, any inequality that is valid for the cut polytope is also valid for all uniform cut polytopes (and naturally, the converse does not hold in general). However, in some cases, uniform cut polytopes may have a much simpler linear description, so that it may be valuable, when solving optimization problems on uniform cut polytopes (or potentially in order to solve an optimization problem on the cut polytope by decomposing it according to different sizes of the shores), to investigate more thoroughly their precise polyhedral structure. As an illustration, consider the Table 4.1 where we reported the number of facets of some small uniform cut polytopes CUT_k^n ($k = 1, 2, 3$), cut cones \widehat{CUT}^n and cut polytopes CUT_n , where \widehat{CUT}^n stands for the conic hull of the incidence vectors of cuts in K_n . The results concerning \widehat{CUT}^n and CUT^n are from [88], the others have been obtained using PORTA [61].

Table 4.1 – Number of facets of some small cut polyhedra

n	CUT^n	\widehat{CUT}^n	CUT_k^n			
			k			
			1	2	3	4
8	217,093,472	49,604,520	8	28	52,367	896
7	116,764	38,780	7	21	896	
6	368	210	6	15	10	
5	56	40	5	10		
4	16	12	4	3		

The polytope $CUT_{\lfloor \frac{n}{2} \rfloor}$ was studied by Conforti *et al.* [70, 71]. They introduced several classes of inequalities and showed that some are facet-defining. Their work was extended by de Souza and Laurent [82], who introduced several classes of facets arising from generalizations of an inequality from [70, 71] that is based on a cycle. Otherwise, we are presently not aware of other polyhedral studies dedicated to uniform cut polytopes such as the ones we carried out and summarize hereafter.

The rest of this chapter is organized as follows. We firstly introduce some preliminaries on the dimension together with some key properties which allowed us to prove the facet definiteness property for many families of inequalities. Then, we exhibit diverse families of facet-defining inequalities. They extend works by Conforti *et al.* [70, 71] (on the equipartition polytope) to uniform cut polytopes of the form CUT_k^n with $k < \lfloor \frac{n}{2} \rfloor$. Then, the polyhedral structure of uniform cut

polytopes is investigated from the perspective of their diameter according to two different definitions of the latter present in the literature. For more details and omitted proofs, the reader may consult [182, 184, 185].

4.2 Properties of facet defining inequalities and dimension

In this section, we report our results on the dimension of uniform cut polyhedra (and some of their natural extensions) together with some important lemmas that are the cornerstone in the proofs establishing that some families of inequalities are facet defining. We start with some useful notation.

For some vector indexed on the edges: $y \in \mathbb{R}^{E(G)}$, and two disjoint node subsets $S_1, S_2 \subseteq V(G)$, let $y[S_1 : S_2]$ denote the expression $\sum_{i,j \in E(G): i \in S_1, j \in S_2} y_{ij}$. Given a set F of edges in G and a vector $x \in \mathbb{R}^{E(G)}$, we let $x(F)$ denote $\sum_{e \in F} x_e$. The dimension of a polyhedron P (i.e. the maximum number of affinely independent points in P minus 1) is denoted by $\dim(P)$.

Lemma 4.2.1. [182] *Let G be a graph with n nodes, and let $\alpha^T x \leq \alpha_0$ be a valid inequality for $\text{CUT}_k(G)$. Let Y, Y', Z and Z' denote pairwise disjoint subsets of $V(G)$ such that $|Y| = |Z|$, $|Y'| = |Z'|$, and $|Y| + |Y'| = k$. Set $W = V(G) \setminus (Y \cup Y' \cup Z \cup Z')$. If the incidence vectors of the cuts $\delta(Z \cup Z')$, $\delta(Y \cup Y')$, $\delta(Z' \cup Y)$ and $\delta(Y' \cup Z)$ satisfy $\alpha^T x = \alpha_0$, then $\alpha[W : Z] = \alpha[W : Y]$.*

Proof. Set $\Delta_1 = \delta(Z \cup Z')$, $\Delta_2 = \delta(Y' \cup Y)$, $\Delta_3 = \delta(Z' \cup Y)$ and $\Delta_4 = \delta(Y' \cup Z)$. Since the incidence vectors of the cuts Δ_1 and Δ_3 satisfy $\alpha^T x = \alpha_0$, we have $0 = \alpha^T \chi^{\Delta_1} - \alpha^T \chi^{\Delta_3} = \alpha[Y' : Z'] + \alpha[Y' : Z] + \alpha[Y : Z'] + \alpha[Y : Z] + \alpha[W : Z'] + \alpha[W : Z] - \alpha[Y' : Z'] - \alpha[Y' : Y] - \alpha[Z : Z'] - \alpha[Z : Y] - \alpha[W : Z'] - \alpha[W : Y]$. Analogously, considering now the incidence vectors of the cuts Δ_2 and Δ_4 , we have $0 = \alpha^T \chi^{\Delta_2} - \alpha^T \chi^{\Delta_4} = \alpha[Y' : Z'] + \alpha[Y' : Z] + \alpha[Y' : W] + \alpha[Y : Z'] + \alpha[Y : Z] + \alpha[Y : W] - \alpha[Y' : Z'] - \alpha[Y' : Y] - \alpha[Y' : W] - \alpha[Z : Z'] - \alpha[Z : Y] - \alpha[Z : W]$.

Subtracting one equation from the other leads to: $\alpha[W : Z] = \alpha[W : Y]$. \square

Lemma 4.2.2. [182] *In the setting of Lemma 4.2.1, suppose also that $1 \leq k \leq \lfloor \frac{n-1}{2} \rfloor$. Fix z in Y' . For $w \in W$, let $Y'_w = (Y' \cup \{w\}) \setminus \{z\}$. In addition to the cuts listed in Lemma 4.2.1, if $\delta(Y \cup Y'_w)$ and $\delta(Z \cup Y'_w)$ also satisfy $\alpha^T x = \alpha_0$ for all $w \in W$, then $\alpha[z : Y] = \alpha[z : Z]$.*

Proof. Application of Lemma 4.2.1 directly leads to the following relations:

$$\alpha[Y : W] = \alpha[Z : W],$$

and

$$\alpha[Y : (W \cup \{z\}) \setminus \{w\}] = \alpha[Z : (W \cup \{z\}) \setminus \{w\}], w \in W.$$

Therefore,

$$\alpha[Y : z] - \alpha[Y : w] = \alpha[Z : z] - \alpha[Z : w], w \in W.$$

Now, summing the last equations over all $w \in W$ leads to

$$|W|\alpha[Y : z] = |W|\alpha[Z : z],$$

and the assertion follows. \square

We mention hereafter another important result reported in [70] that is useful for showing that some inequalities are facet defining.

Lemma 4.2.3. [70] *Let G be a graph and $k \in \{2, \dots, \lfloor \frac{n}{2} \rfloor\}$. Let $i, j, l, m \in V(G)$, $S = V(G) \setminus \{i, j, l, m\}$, and $(S_1 | S_2)$ be a partition of S with $|S_1| = k - 2$ and $|S_2| = n - k - 2$. Let $a^T x \leq b$ be a valid inequality for $\text{CUT}_k(G)$. If the incidence vectors of the cuts below*

$$\begin{aligned} & \delta(S_1 \cup \{i, j\}), & \delta(S_1 \cup \{l, m\}), & \delta(S_1 \cup \{i, l\}), \\ & \delta(S_1 \cup \{j, m\}), & \delta(S_1 \cup \{i, m\}), & \delta(S_1 \cup \{j, l\}), \end{aligned}$$

satisfy $a^T x = b$, then

$$a_{ij} + a_{lm} = a_{il} + a_{jm} = a_{im} + a_{jl}.$$

We now give the dimension of uniform cut polytopes for complete graphs: CUT_k^n with $k \in \{2, \dots, \lfloor \frac{n-1}{2} \rfloor\}$.

Proposition 4.2.4. [182] *If $k \in \{2, \dots, \lfloor \frac{n-1}{2} \rfloor\}$, then $\dim(\text{CUT}_k^n) = \binom{n}{2} - 1$.*

Proposition 4.2.5. [182] *For $n \geq 3$, $\dim(\text{CUT}_1^n) = n - 1$.*

Propositions 4.2.4, 4.2.5 and the results obtained by Conforti et al. [70] on the dimension of the equipartition polytope lead to the following proposition.

Proposition 4.2.6. [70, 182] *The following relations hold:*

$$\dim(\text{CUT}_k^n) = \begin{cases} \binom{n}{2} - n & \text{if } n \text{ is even and } k = \frac{n}{2}, \\ n - 1 & \text{if } k = 1, \\ \binom{n}{2} - 1 & \text{otherwise.} \end{cases}$$

Similarly, we can show that slight extensions of uniform cut polytopes are full-dimensional. Given some set of integers $Q \subseteq \{1, \dots, \lfloor \frac{n}{2} \rfloor\}$, and defining $\text{CUT}_Q(G) = \text{conv}(\{\chi^{\delta_G(S)} : S \subseteq V(G), |S| \in Q\})$, we have the following result.

Proposition 4.2.7. [182] *If Q is composed of pairwise distinct elements from $\{1, \dots, \lfloor \frac{n}{2} \rfloor\}$ and satisfies $|Q| \geq 2$, then for any graph G , the polytope $\text{CUT}_Q(G)$ is full-dimensional.*

4.3 A walk through some families of facet-defining inequalities

We list hereafter families of facet-defining inequalities for uniform cut polytopes. Their proofs heavily rely on the three lemmas from Section 4.2 and tend to be often technical and lengthy; they are thus omitted from the current document and can be retrieved from the given references.

4.3.1 Basic facet-defining inequalities

Proposition 4.3.1. [182] *Let uv be an edge in $E(G)$ that is not contained in any triangle of the graph G . Let G' be the subgraph of G induced by $V' = V(G) \setminus \{u, v\}$ and k be an integer with $3 \leq k \leq \lfloor \frac{n-1}{2} \rfloor$. Suppose that*

- G' is complete,
- every node in V' is adjacent to either node u or node v but not both,
- $n_u \geq n_v$ where n_u and n_v denote the number of nodes in V' that are adjacent to u and v respectively, and define $\bar{n}_v = \min(n_v, k - 1)$, $\bar{n}_u = \min(n_u, n - k - 1)$.

Then the following inequality defines a facet of $\text{CUT}_k(G)$:

$$\sum_{i,j \in E(G')} x_{ij} + 2 \sum_{w \in N(u) \cap V'} x_{uw} + 2 \sum_{w \in N(v) \cap V'} x_{vw} + (n - 1 - 2\bar{n}_u - 2\bar{n}_v)x_{uv} \leq k(n - k). \quad (4.1)$$

Proposition 4.3.2. [182] *Let $uv \in E$ and G' be the subgraph of G that is induced by $V' = V(G) \setminus \{u, v\}$. Then the inequality*

$$x_{uv} \geq 0 \quad (4.2)$$

defines a facet of $\text{CUT}_k(G)$ with $k \in \{3, \dots, \lfloor \frac{n-1}{2} \rfloor\}$ if and only if

- i) edge uv is not in any triangle, and
- ii) either G' is not a complete graph or there exists a node $w \in V'$ which is not adjacent to u nor v .

Proposition 4.3.3. [182] Let uv denote an edge of the graph G and let G' stand for the subgraph of G that is induced by $V' = V(G) \setminus \{u, v\}$. For an integer k with $3 \leq k \leq \lfloor \frac{n-1}{2} \rfloor$ the inequality

$$x_{uv} \leq 1 \quad (4.3)$$

defines a facet of $\text{CUT}_k(G)$ if and only if

- edge uv is not contained in any triangle, and
- G' is not a complete graph.

Proposition 4.3.4. [182] Let G be a graph with no isolated node. Let $v \in V(G)$, G' denote the subgraph of G that is induced by $V' = V(G) \setminus \{v\}$ and $\overline{G'}$ stand for the complementary graph of G' . Then, the inequalities

$$x(\delta(v)) \leq n - k, \quad (4.4)$$

and

$$x(\delta(v)) \geq k, \quad (4.5)$$

define facets of $\text{CUT}_k(G)$ with $3 \leq k \leq \lfloor \frac{n-1}{2} \rfloor$ if and only if node v is adjacent to every node in V' and $\overline{G'}$ has no bipartite component.

4.3.2 Facets from subgraphs

Let $\{e_1, \dots, e_{\binom{n}{2}}\}$ denote the set of edges of K_n and G be the subgraph of K_n that is induced by the edges $\{e_1, \dots, e_l\}$. We start with the following result that can be derived from Theorem 5.1 in [71]. It states that facet defining inequalities of the polytope $\text{CUT}_k(K_n)$ with support (i.e. the set of edges with nonzero coefficients) E' are also facet defining for the cut polytope $\text{CUT}_k(G)$ of any subgraph G containing E' , with $k \in \{2, \dots, \lfloor \frac{n-1}{2} \rfloor\}$. It follows that many facet defining inequalities presented later for $\text{CUT}_k(K_n)$ are valid not only for K_n but for more general graph classes.

Proposition 4.3.5. Let $a^T x \leq b$ denote a facet defining inequality for CUT_k^n , $k \in \{2, \dots, \lfloor \frac{n-1}{2} \rfloor\}$. Then for any graph G that is obtained from K_n by removing edges which do not belong to the support of the vector a , $a^T x \leq b$ is facet defining for $\text{CUT}_k(G)$.

Proposition 4.3.6. [182] Given a nonempty subgraph K_q of K_n with $\lfloor \frac{q}{2} \rfloor < k - 1$, the inequality

$$x(E(K_q)) \leq \left\lfloor \frac{q}{2} \right\rfloor \left\lceil \frac{q}{2} \right\rceil \quad (4.6)$$

defines a facet of CUT_k^n with $k \in \{3, \dots, \lfloor \frac{n-1}{2} \rfloor\}$ if and only if q is odd.

Definition. A bicycle h -wheel is a graph (W, E) consisting of a cycle C_h with length h and two nodes $(v, w) \in (W \setminus V(C_h))^2$ (also called *universal nodes*) that are adjacent to each other and to all the nodes of C_h .

A bicycle 5-wheel is represented in Figure 4.1 with dashed lines for edges incident to any of the universal nodes v, w .

Proposition 4.3.7. [182] Let W be a bicycle h -wheel, $h \geq 5$ odd, which is a subgraph of K_n . Then the inequality

$$x(E(W)) \leq 2h \quad (4.7)$$

defines a facet of CUT_k^n , $k \in \left\{ \frac{h+5}{2}, \dots, \lfloor \frac{n-1}{2} \rfloor \right\}$.

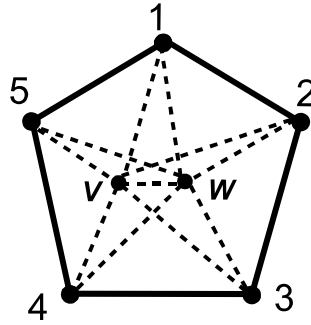


Figure 4.1 – A bicycle 5-wheel.

4.3.3 Nonboolean facets

Differently from the last two sections, we now introduce several facet-defining inequalities for the polytope CUT_k^n whose coefficients on the variables may take values outside the set $\{0, 1\}$, hence the name *nonboolean* facets.

Let $T \subset V(K_n)$, $R = V(K_n) \setminus T$, $t = |T| \geq 2$, t even, $k \in \{\frac{t}{2} + 2, \dots, \lfloor \frac{n-1}{2} \rfloor\}$, and consider the following inequality

$$\sum_{i \in R} \sum_{j \in T} x_{ij} + (n - 2k + 2) \sum_{ij \in T} x_{ij} \leq \frac{(n - 2k)}{4} t^2 + \frac{n}{2} t. \quad (4.8)$$

The support of an inequality of the form (4.8) is represented in Figure 4.2, where a dashed line between two node sets represents all the edges with exactly one endnode in each of these sets.

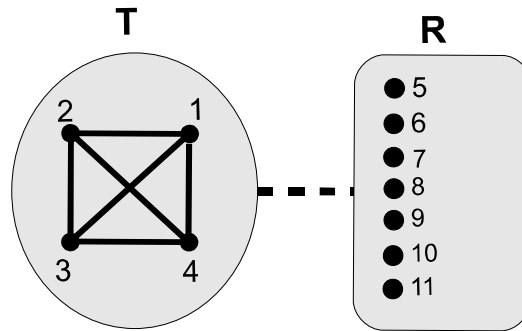


Figure 4.2 – An illustration of the support of an inequality (4.8) defining a facet for CUT_4^{11}

Proposition 4.3.8. [182] *Inequality (4.8) defines a facet of CUT_k^n .*

Proposition 4.3.9. [182] *Let K_q be an induced subgraph of K_n . Let w_i be a system of integer weights on the nodes of K_q such that $w_i \geq 1$, $\sum_{i=1}^q w_i = 2r + 1$, $r \geq 3$, $\sum_{i \in W} w_i \leq r - 1$, with $W = \{i \in V(K_q) : w_i \geq 2\}$, $k \geq r + 2$, $k \leq \lfloor \frac{n-1}{2} \rfloor$, $q \leq n - \sum_{i \in W} w_i + |W|$. Then the inequality*

$$\sum_{ij \in K_q} w_i w_j x_{ij} \leq r(r + 1) \quad (4.9)$$

induces a facet of CUT_k^n .

4.3.4 Homogeneous triangle inequalities

Proposition 4.3.10. [184] *The homogeneous triangle inequalities defined by:*

$$x_{ij} - x_{il} - x_{jl} \leq 0, \forall i, j, l \in \{1, \dots, n\} \text{ with } i < j \text{ and } i \neq l \neq j, \quad (4.10)$$

are facet-defining for uniform cut polytopes CUT_k^n , with $k \in \{3, \dots, \lfloor \frac{n-1}{2} \rfloor\}$.

4.3.5 Cycle inequalities

The next proposition extends to uniform cut polytopes a family of facet-defining inequalities for the equipartition polytope presented in [71].

Proposition 4.3.11. *Let $C = (V(C), E(C))$ denote a chordless cycle in the graph $G = (V, E)$ and let F be an odd set of edges in $E(C)$. Set $L := |E(C)| - \frac{|F|-1}{2}$ if $F \subset E(C)$ and $L := |V(C)| + 2$ if $F = E(C)$. If $k \geq L$ and G contains either two nonadjacent nodes in $V \setminus V(C)$ or a node in $V(C)$ not adjacent to a node in $V \setminus V(C)$, then the inequality*

$$\sum_{e \in F} x_e - \sum_{e \in E(C) \setminus F} x_e \leq |F| - 1 \quad (4.11)$$

is facet-defining for CUT_k^n .

4.3.6 (1 Stable set / 1 Clique) facets

Another family of facet-defining inequalities having the same support as (4.8) can be derived in a similar way.

Proposition 4.3.12. [184] *Let $T \subset V_n$, $R = V_n \setminus T$, $t = |T| \geq 2$, t even, $k \in \{\frac{t}{2} + 1, \dots, \lfloor \frac{n-1}{2} \rfloor\}$, then the inequality*

$$\sum_{i \in R} \sum_{j \in T} x_{ij} - (n - 2k - 2) \sum_{ij \in T} x_{ij} \geq -\frac{(n-2k)}{4} t^2 + \frac{n}{2} t \quad (4.12)$$

is facet-defining for CUT_k^n .

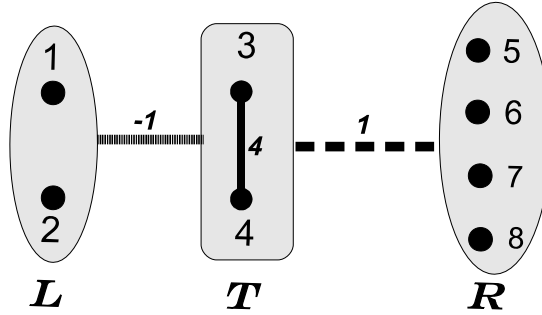
Remark. We may notice that the difference of the two inequalities (4.8)–(4.12) gives the inequality $x_{uv} \leq 1$ (with $T = \{u, v\}$), thus implying the latter is redundant in any description of CUT_k^n . A characterization of the graphs for which the trivial constraint $x_{uv} \leq 1$ is facet-defining for CUT_k^n is reported in [182] (see Proposition 3.3 in this reference).

4.3.7 (2 stable sets / 1 Clique) facets

Before we introduce the general form of the next family of facet-defining inequalities for uniform cut polytopes, let us firstly illustrate some aspects of their intrinsic structure on a small example. For, consider the uniform cut polytope CUT_3^8 , a partition (L, T, R) of the node set V_8 with $|L| = 2$, $|T| = 2$ and $|R| = 4$, and an edge weight function $(w_{uv})_{uv \in E_n}$ defined as follows:

$$w_{uv} = \begin{cases} 4 & \text{if } (u, v) \in T \times T, \\ 1 & \text{if } (u, v) \in T \times R, \\ -1 & \text{if } (u, v) \in T \times L, \\ 0 & \text{otherwise.} \end{cases}$$

The situation is depicted in Figure 4.3, where an edge between two node sets A and B represents the set of all the edges of the complete bipartite graph with bipartition (A, B) . The value reported close to an edge corresponds to the associated weight defined above. Edges with zero weight are not represented. For this particular example, we can easily check that the maximum weight of any cut (considering no constraint on the size) is 8, whereas it is 6 for cuts with size 3. (Given a cut $\delta(S)$, its weight $w(\delta(S))$ is defined as the sum of the weights of the edges it contains, i.e.

Figure 4.3 – An illustration of the support of a facet-defining inequality of type (4.13) for CUT_3^8


$w(\delta(S)) = \sum_{uv \in \delta(S)} w_{uv}$). Thus the inequality with a left-hand side corresponding to this weight function and a right-hand side with value 6: $w^t x \leq 6$, is valid for CUT_3^8 (whereas it is *not* for CUT_4^8 since it is violated by the incidence vector of the cut $\delta(R)$). More generally, we could obtain the next result.

Lemma 4.3.13. [184] *Let $k \geq 3$ denote an integer and (L, T, R) denote a partition of the vertex set V_n such that $|T| = t$ is a positive even integer, $|L| = l > k - \frac{t}{2} - 1$, $|R| = r > k - \frac{t}{2} + 1$ and $k > \frac{t}{2} + 1$. Then the following inequality is valid for CUT_k^n , $k \in \{\frac{t}{2} + 2, \dots, \lfloor \frac{n-1}{2} \rfloor\}$.*

$$\sum_{(u,w) \in T \times R} x_{uw} + \gamma_L \sum_{(u,v) \in T \times L} x_{uv} + \gamma_T \sum_{(u,z) \in T \times T: u \neq z} x_{uz} \leq \gamma_0, \quad (4.13)$$

with

$$\gamma_L = \frac{2r + t - 2k - 2}{-2n + 2r + t + 2k - 2}, \quad (4.14)$$

$$\gamma_T = \frac{(n - 2k + 2)(t - 2k - 2) + 4r}{-2n + 2r + t + 2k - 2}, \quad (4.15)$$

$$\gamma_0 = \frac{t^2}{4} (\gamma_T - 2\gamma_L) + \frac{t}{2} [(n - r)\gamma_L + r]. \quad (4.16)$$

Proposition 4.3.14. [184] *Inequality (4.13) defines a facet of CUT_k^n .*

4.4 On the diameter of cut polytopes

4.4.1 Introduction

Given a nonempty polytope P , the *1-skeleton* of P is the graph whose nodes correspond to the vertices of P , and having an edge joining two nodes iff (if and only if) the corresponding vertices of P are adjacent on P . Given two vertices v_1, v_2 of P , let $d(v_1, v_2)$ denote the length (w.r.t. the number of edges) of a shortest path between the nodes representing v_1 and v_2 in the 1-skeleton of P . The *diameter* of P , which is denoted by $\text{diameter}(P)$ is the maximum distance between any two vertices of P , i.e. $\max\{d(v_1, v_2) : v_1, v_2 \text{ are vertices of } P\}$.

The notion of diameter of a polyhedron presents, among others, some connections with linear programming and linear optimization methods that are based on the fact that if a linear program having for feasible region a pointed polyhedron has a finite optimal objective value, then there exists an optimal vertex solution. For example, consider the simplex algorithm applied to such a linear program: starting from a particular vertex v_0 of the feasible region F , this method consists in iteratively moving from a vertex (*the current basic feasible solution*) of F to an adjacent vertex

(with better or equal objective value), until an optimal vertex solution is found. Given a nonempty polytope P , any vertex v of P is the unique minimizer of some linear function over P . For such an objective function, if the simplex method is applied starting from a vertex v_0 satisfying $d(v, v_0) = \text{diameter}(P)$, then the number of iterations required is at least $\text{diameter}(P)$.

To our knowledge, the first study of the 1-skeleton of the cut polytope appears in the paper by Barahona and Mahjoub [25], where the following characterization of vertex adjacency on the cut polytope is provided (Theorem 4.1 in [25]).

Theorem 4.4.1. [25] *Let $G = (V, E)$ be a connected graph, and let I, J denote two cuts in G . Set $F = E \setminus (I \Delta J)$. Then χ^I and χ^J are adjacent on $\text{CUT}(G)$ iff the graph $H_{I \Delta J} = (V, F)$ has two connected components.*

The proof the authors give relies on the property that the extreme points χ^I and χ^J are adjacent on $\text{CUT}(G)$ iff there exists a vector c such that they are the only two vertices optimizing $\max\{c^T x : x \in \text{CUT}(G)\}$. Many of the results reported hereafter rely on this proof technique.

In the same reference [25], it is also established that the Hirsch property holds for $\text{CUT}(G)$. (A d -dimensional polytope P with f facets is said to satisfy the Hirsch property if its diameter verifies the inequality: $\text{diameter}(P) \leq f - d$). In fact, Naddef [177] proved later that this property holds for all $(0, 1)$ -polytopes, and thus also for $\text{CUT}(G)$ and the uniform cut polytopes. What also follows from Theorem 4.4.1 and its proof is that $\text{CUT}(G)$ is a so-called *combinatorial polyhedron* [167, 178], a family of polyhedra for which Matsui and Tamura established further different properties related to the Hirsch property [167]. Hereafter we mention part of our results w.r.t. the diameter of the cut polytope and uniform cut polytopes.

4.4.2 On the diameter of the cut polytope

For the particular case of complete graphs, Barahona and Mahjoub [25] showed that the cut polytope has diameter 1: this is a direct consequence of Theorem 4.4.1. For general graphs they provided the following upper bound on the diameter that is given by the next theorem (Theorem 4.3 in [25]). Before stating the result we give some additional terminology and notation. Let $G = (V, E)$ be an undirected graph containing an edge $\{u, v\}$. *Contracting edge $e = \{u, v\}$* is the operation of replacing u and v by a single node whose incident edges are the edges other than e that were incident to u or v . Formally, the edge contraction w.r.t. $\{u, v\}$ results in the graph $G' = (V', E')$ with $V' = V \setminus \{u, v\} \cup \{w\}$ and $E' = \{e \in E : |e \cap \{u, v\}| = 0\} \cup \{\{w, z\} : \{u, z\} \in E \text{ or } \{v, z\} \in E \text{ with } z \in V \setminus \{u, v\}\}$. Given a connected graph G and a cut C in G , let $T(C)$ denote the graph obtained from G by contracting edges not in C and replacing multiple edges by single edges. Let $m(C)$ be the number of edges of $T(C)$. Given two cuts I, J of G , let $d(I, J)$ stand for the distance in the 1-skeleton of $\text{CUT}(G)$ between the nodes corresponding to the incidence vectors of I and J . With a slight abuse of terminology, we will say that two cuts I and J are adjacent on $\text{CUT}(G)$ if $d(I, J) = 1$.

Theorem 4.4.2. [25] *If I and J are two cuts in any graph G , then $d(I, J) \leq m(I \Delta J)$.*

Given a subset of edges I of the graph G , let H_I denote the graph obtained from G by removing all the edges in I . Given a graph G , let $\kappa(G)$ denote the number of connected components in G . A generally better bound on the distance between the incidence vectors of two cuts in the 1-skeleton of $\text{CUT}(G)$ is as follows.

Proposition 4.4.3. [185] *Let G be a connected graph. If I and J are two cuts in G , then $d(I, J) \leq \kappa(H_{I \Delta J}) - 1$.*

Corollary 4.4.4. [185] *For any graph $G = (V, E)$ we have $\text{diameter}(\text{CUT}(G)) \leq |V| - \kappa(G)$.*

The upper bounds given by Theorem 4.4.2 and Corollary 4.4.4 on the diameter of the cut polytope are both tight for trees. However, for general graphs, both may strongly differ. We namely reported cases (complete bipartite graphs) for which the bound from Theorem 4.4.2 is quadratic in the number of vertices of the graph whereas the one from Corollary 4.4.4 is linear, see [185].

For any subgraph G' of G , we could prove that the 1-skeleton of $\text{CUT}(G')$ is a subgraph of the 1-skeleton of $\text{CUT}(G)$, leading to the next result.

Proposition 4.4.5. [185] *Let $G' = (V, E')$ be a connected subgraph of $G = (V, E)$. Then $\text{diameter}(\text{CUT}(G')) \geq \text{diameter}(\text{CUT}(G))$.*

A special graph family for which our investigations led to the exact value of the diameter of the cut polytope is that of complete bipartite graphs. Given two positive integers k_1, k_2 , let K_{k_1, k_2} represent the complete bipartite graph with node bipartition (V_1, V_2) such that $|V_i| = k_i$, for $i = 1, 2$.

Proposition 4.4.6. [185] *For any two integers $k_1, k_2 \geq 2$, $\text{diameter}(\text{CUT}(K_{k_1, k_2})) = 2$.*

4.4.3 On the diameter of uniform cut polytopes

Sufficient conditions for adjacency on uniform cut polytopes

Notice first that if two cuts $\delta(S_1), \delta(S_2)$ with $|S_1| = |S_2| = k \leq |V|/2$ are adjacent on $\text{CUT}(G)$, then they are adjacent also on $\text{CUT}_k(G)$, since there exists an objective function $c \in \mathbb{R}^{E(G)}$ that is optimized exactly by the closed line segment joining χ^{S_1} and χ^{S_2} on both polytopes. However, it is easy to see that the condition of Theorem 4.4.1 is not necessary for adjacency on uniform cut polytopes (see [185] for one example). The following proposition formulates a sufficient conditions for adjacency relations on uniform cut polytopes which may not hold in the cut polytope.

Proposition 4.4.7. [185] *Let $G = (V, E)$ be a graph and let (C_1, C_2, S, W) denote a partition of V such that $|C_1| = |C_2|$, $|C_1| + |S| = k \leq \lfloor \frac{|V|}{2} \rfloor$, and the graph $G' = (V, E \setminus \delta(C_1 \cup C_2))$ has three connected components. Then, if different, the cuts $\delta(C_1 \cup S)$ and $\delta(C_2 \cup S)$ are adjacent on $\text{CUT}_k(G)$ if one of the following two conditions holds.*

- (i) $G'[S \cup W]$ is connected and $G'[C_1 \cup C_2]$ has two connected components G'_1 and G'_2 such that $|V(G'_1) \cap C_1| \neq |V(G'_1) \cap C_2|$ and $|V(G'_2) \cap C_1| + |S| \neq |V(G'_2) \cap C_2| + |W|$.
- (ii) $G'[C_1 \cup C_2]$ is connected and $G'[S \cup W]$ has two connected components G'_1 and G'_2 such that $|V(G'_i) \cap S| \neq |V(G'_i) \cap W|$, $i = 1, 2$.

On particular cases

A direct consequence of Theorem 4.4.1 is that the cut and uniform cut polytopes of complete graphs have diameter one.

Corollary 4.4.8. *For all integers $n \geq 1$ and $k \leq \lfloor \frac{n}{2} \rfloor$, the uniform cut polytopes CUT_k^n have diameter 1.*

For a general graph G , the cut polytope $\text{CUT}(G)$ and uniform cut polytopes may have a diameter strictly larger than one. In fact, with respect to the cut polytope, it follows by Theorem 4.4.1 that $\text{CUT}(G)$ has a diameter strictly larger than one for any connected graph $G = (V, E)$ that is not complete and having at least three nodes. This contrasts with the case of uniform cut polytopes and is illustrated by the following proposition.

Proposition 4.4.9. [185] *If the graph G is a star, then $\text{CUT}_1(G)$ has diameter 1.*

Recall that for a star having n nodes the cut polytope $\text{CUT}(G)$ has diameter $n - 1$ (see Remark 2 above). So this illustrates the fact that in some cases the gap between the diameters of the cut polytope and uniform cut polytopes can be linear in the number of nodes of the graph.

A graph $G = (V, E)$ is said to be k -node connected if, for any node subset $Q \subseteq V$ with cardinality $|Q| < k$, the induced subgraph $G[V \setminus Q]$ is connected. The next result provides a simple upper bound on the diameter of uniform cut polytopes for 3-node connected graphs.

Proposition 4.4.10. [185] *If the graph G is 3-node connected, then*

$$\text{diameter}(\text{CUT}_k(G)) \leq k.$$

In addition to the results mentioned before, some connections between the 1-skeletons of uniform cut polytopes and bipartition polytopes [77, 145] could be established, see [185] for details.

4.5 Some perspectives

In this chapter we tried to provide an overview of the main results we could establish w.r.t. particular variants of the cut polytope defined as the convex hull of the incidence vectors of cuts having some fixed size. We could introduce many families of facet defining inequalities, some of which generalize facets of the equipartition polytope. Also we provided some of our contributions w.r.t. the diameter of the classical cut polytope and uniform ones.

We could see that uniform cut polytopes have a very rich structure as is illustrated above by many families of facet-defining inequalities. It would be interesting to look for further facets, and in particular for families generalizing some of the ones presented. Complete formulations could be obtained for the uniform cut polytopes CUT_k^n for $k = 1, 2$ [184]. A natural research direction is to look for complete formulations for larger values of k .

Recently, there has been some interest of the community in combinatorial optimization to investigate the extension complexity of some polytopes (the extension complexity of a polytope P is the smallest number of facets of any polyhedron whose affine image is P), see [102] and the references therein for an introduction. In particular, Kaibel and Weltge [137] gave an elegant proof of an exponential lower bound (in the number of nodes of the graph) on the extension complexity of the cut polytope. (In fact, their result is w.r.t. the correlation polytope, but the latter is linearly isomorphic to a cut polytope [79]). It seems likely that the extension complexity of uniform cut polytopes CUT_k^n is worth investigating, as this would provide us with another comparative view (i.e. different from the consideration of known families of facet defining inequalities) of the structure complexity of uniform versus (non-uniform) cut polytopes.

Another research direction is the study of the quality of approximations obtained depending on the families of inequalities considered in the relaxations, together with the analysis of the complexity status for the separation problem associated with the newly introduced families of inequalities and the development of efficient separation procedures (possibly heuristics for the case when the separation problem is difficult).

Chapter 5

Generalized domination concepts

This chapter presents results from our investigations on polyhedral structures that are related to generalized concepts of domination in graphs. For the three considered concepts, complete formulations could be obtained for stars, a complete formulation for trees (in the original space of variables which is indexed on the node set) for two of them, and several families of facet defining inequalities could be identified (for all of the three concepts).

The results related to f -dominating sets and total f -dominating sets stem from joint works with Prof. Dell'Amico [83, 84]. More information on the presented elements, together with some additional results can be obtained in [83, 84, 186, 187].

5.1 Introduction

Let $G = (V, E)$ denote a simple undirected graph having node set $V = \{1, 2, \dots, n\}$ and edge set E . For our purpose we may assume that G is connected and $n \geq 2$, unless otherwise stated. For each $v \in V$, let d_v denote its degree in G and let f_v be a given nonnegative integer value. Let \mathcal{F}_G stand for the set of vectors $\{f \in \mathbb{Z}_+^V : 0 \leq f_v \leq d_v, \forall v \in V\}$. Given a node subset $S \subseteq V$, its *open neighborhood* is the set $N(S) = \{v \in V \setminus S : \exists u \in S \text{ such that } \{u, v\} \in E\}$, and its *closed neighborhood* is the set $N[S] = N(S) \cup S$. In case S consists of a single node $v \in V$ we may write $N(v)$ (resp. $N[v]$) instead of $N(\{v\})$ (resp. $N[\{v\}]$).

A node subset $D \subseteq V$ is called an *f -dominating set* if each node $v \in V \setminus D$ (resp. $v \in V$) has at least f_v neighbor(s) in D . If, in addition, each node v in D also has at least f_v neighbors in D , then D is called a *total f -dominating set*. A node subset $D \subseteq V$ is called an *f -tuple dominating set* if the closed neighborhood of each node v in G intersects D in at least f_v nodes.

In the special case when $f_v = 1$, for all $v \in V$, an f -dominating set D is more simply called a *dominating set*. Similarly, in the same case, a total f -dominating set is called a *total dominating set* and an f -tuple dominating set is nothing but a dominating set, see [127, 128, 134]. The notion of k -tuple dominating set, for some given positive integer k , corresponds to the particular case of f -tuple dominating set, when $f_v = k$, for all $v \in V$ [110, 126], [128, p. 189].

Consider then optimization problems which consist in determining a minimum weight $\{f$ -tuple, total f - or f -} dominating set. The *minimum weight total f -dominating set problem*, denoted by $[MWT_f]$ can be formulated as follows: Given positive node weights $(c_v)_{v \in V}$, find a minimum weight total f -dominating set of G , i.e. find a node subset $D \subseteq V$ such that D is a total f -dominating set and the weight of D : $\sum_{v \in D} c_v$, is minimum. The problems w.r.t. f -dominating sets and f -tuple dominating sets can be defined similarly. They will be denoted by $[MW_f]$ and $[MWU_f]$, respectively. The problems $[MWT_f]$, $[MW_f]$ and $[MWU_f]$ may be formulated as the integer programs:

$$(\text{IP} - MWT_f) \begin{cases} \min & \sum_{v \in V} c_v x_v \\ & \sum_{u \in N(v)} x_u \geq f_v, v \in V, \\ & x \in \{0, 1\}^V, \end{cases}$$

$$\begin{aligned}
 (\text{IP-MW}_f) \left\{ \begin{array}{l} \min \sum_{v \in V} c_v x_v \\ f_v x_v + \sum_{u \in N(v)} x_u \geq f_v, v \in V, \\ x \in \{0, 1\}^V, \end{array} \right. \\
 (\text{IP-MWU}_f) \left\{ \begin{array}{l} \min \sum_{v \in V} c_v x_v \\ \sum_{u \in N[v]} x_u \geq f_v, v \in V, \\ x \in \{0, 1\}^V. \end{array} \right.
 \end{aligned}$$

The linear relaxations of these formulations (obtained replacing the constraints $x \in \{0, 1\}^V$ by $x \in [0, 1]^V$) will be denoted by (LP-MWT_f) , (LP-MW_f) and (LP-MWU_f) , respectively. Given a node subset $S \subseteq V$, let $\chi^S \in \{0, 1\}^V$ denote its incidence vector: $\chi_v^S = 1$ if $v \in S$, and $\chi_v^S = 0$ otherwise. Let \mathcal{T}_G^f denote the total f -dominating set polytope, i.e. the convex hull of all the incidence vectors of the total f -dominating sets in G . Then, problem $[\text{MWT}_f]$ can be reformulated as the linear program: $\min \{c^T x : x \in \mathcal{T}_G^f\}$. Similarly, we define the polytopes \mathcal{D}_G^f and \mathcal{U}_G^f w.r.t. f -dominating sets and f -tuple dominating sets; which can be used to reformulate $[\text{MW}_f]$ and $[\text{MWU}_f]$ as linear programs.

One can easily check the following inclusions hold:

$$\mathcal{T}_G^f \subseteq \mathcal{U}_G^f \subseteq \mathcal{T}_G^{f-\bar{1}} \subseteq \mathcal{D}_G^{f-\bar{1}}, \text{ and } \mathcal{T}_G^f \subseteq \mathcal{U}_G^f \subseteq \mathcal{D}_G^f,$$

where $\bar{1}$ stands for the n -dimensional all-ones vector.

Optimization problems involving dominating sets and some of their many variants arise in several important applications, in particular for the strategic placement of resources in network infrastructures (see, e.g., [127, 128]). The f -domination concept seems to appear first in the papers by [129, 210]. For further results related to the domination, total domination or multiple domination concepts and variants the reader may consult, e.g., [127, 128, 132, 134, 207, 222].

Complexity results

Many works on domination focus on minimum cardinality problems involving non-generalized concepts, i.e., with $f_v = c_v = 1$, for all $v \in V$. Let [DT] (resp. [DD]) stand for the decision problem associated with the minimum cardinality total dominating set problem (resp. the minimum cardinality dominating set problem). [DD] was shown to be NP-complete [111] for undirected path graphs in [48] using a reduction from the 3-dimensional matching problem. A variation of this reduction was used in [151] to prove the same result holds for [DT]. Further graph classes for which [DT] is known to be NP-complete include, e.g., split (and thus also chordal) graphs [150], line graphs of bipartite graphs [168] and circle graphs [140]. Connections between [DD] and [DT] are investigated in [147] which presents a linear time many-one reduction from [DT] to [DD]. This transformation allows the derivation of complexity results for one of the two decision problems from complexity results on the other for some particular graph families (closed for the graph transformation that is introduced there), among which the fact that the minimum cardinality total dominating set problem can be solved in polynomial time in permutation graphs, dually chordal graphs and k -polygon graphs. The decision problem corresponding to the minimum cardinality k -tuple dominating set is known to be NP-complete, even when considering restrictions on the input graph (split or bipartite) [91, 156].

Laskar et al. [151] gave the first linear time algorithm to find a minimum cardinality total dominating set in a tree. Their greedy algorithm uses a particular node labeling and iteratively processes a leaf and removes it from the current tree, which is initialized with the input graph. We extended their result in [84] by showing $[\text{MWT}_f]$ can be solved in linear time for trees (see Proposition 11 in the last mentioned reference). Other graph classes for which the minimum cardinality total dominating set problem can be solved in polynomial time include strongly chordal graphs [57] and cocomparability graphs [146]. In [40] a $\mathcal{O}(n \log n)$ algorithm is presented for solving the minimum weight total dominating set problem in interval graphs. A notable graph family for which the complexity status of the problems [DD] and [DT] differs is that of chordal bipartite graphs:

when restricted to this graph family [DD] is NP-complete whereas [DT] can be solved in polynomial time [75]. Let $\gamma_t(G)$ (resp. $\gamma_{t,f}(G)$) denote the minimum cardinality of a total dominating set (resp. total f -dominating set) in a graph $G = (V, E)$. Given the complexity of the problem for computing $\gamma_t(G)$, some works focused on getting bounds. Lower and upper bounds on $\gamma_t(G)$ appear in [68, 134]. To our present knowledge, a lower bound on $\gamma_{t,f}(G)$ only appears in [221], while the upper bound $\frac{6n}{7}$ is reported in [133] for the particular case $f_v = 2$, for all $v \in V$.

The minimum cardinality dominating set problem can be solved in linear time if the graph G is series-parallel [143] or a cactus [130]. It is known to be NP-hard in planar graphs with maximum node degree 3, regular planar graphs with nodes of degree 4 [111], chordal graphs [47] and undirected path graphs [48]. Also, for this problem, there exists a polynomial time $(H_{\Delta+1} - \frac{1}{2})$ -approximation algorithm, where Δ stands for the maximum degree in G , and, for some given positive integer q , $H_q = \sum_{i=1}^q \frac{1}{i}$ denotes the q th harmonic number [60, 92]. But it is NP-hard to approximate within a factor of $\ln(n) - c \ln \Delta$ for general graphs with $\Delta \geq \Delta_0$, where $c > 0$ and $\Delta_0 \geq 3$ are absolute constants [60]. For the minimum cardinality f -dominating set in block graphs, Hwang and Chang [135] presented a linear time algorithm. We extended their results to the weighted case for trees [83]. For the case when $f_v = k$, $v \in V$, for some positive integer k , Foerster [104] showed that the minimum cardinality f -dominating set problem is NP-hard to approximate with a factor better than $0.2267/k \ln(n/k)$, and he also provided a greedy-type algorithm with an approximation ratio of $\ln(\Delta + k) + 1 < \ln(\Delta) + 1.7$. For the case of unit disk graphs and $f_v = k$, $v \in V$, Couture et al. [74] introduced an incremental algorithm having a constant deterministic performance ratio of six (see also the references therein).

The minimum cardinality k -tuple dominating set problem can be solved in linear time when the graph is a tree [90, 155]. For any $f \in \mathcal{F}_G = \{f \in \mathbb{Z}_+^V : 0 \leq f_v \leq d_v + 1, \forall v \in V\}$, a minimum cardinality f -tuple dominating set can be found in polynomial time in strongly chordal graphs, and in linear time if a strong ordering of the nodes is given [156]. Dobson et al. [91] have shown that the minimum cardinality k -tuple dominating set problem can be solved in polynomial time in P_4 -tidy graphs. Several bounds on the k -tuple domination number, some of which involve other domination parameters, are reported in [126]; other upper bounds obtained with probabilistic approaches are reported, e.g., in [109, 200, 218]. The hardness of approximation for finding a minimum cardinality k -tuple dominating set is investigated in [144], which presents a $(\ln(n) + 1)$ -approximation algorithm. It is also shown that this problem cannot be approximated within a ratio of $(1 - \epsilon)\ln(n)$ for any $\epsilon > 0$ unless $\text{NP} \subseteq \text{DTIME}(n^{\log \log n})$.

Polyhedral approaches

The polyhedral structures of polytopes related to domination problems seem to have received rather little attention. With respect to the classical domination concept relevant works on such aspects are namely [52, 97]. Farber [97] proved that a complete description of $\mathcal{D}_G^{\bar{1}} = \mathcal{U}_G^{\bar{1}}$, i.e. the dominating set polytope, is given by the linear relaxation (LP-MW $_f$) for strongly chordal graphs. This follows from the fact that the closed neighborhood matrix of strongly chordal graphs is totally balanced, see [8, 97] and Theorem 6.13 in [73]. Bouchakour and Mahjoub's paper [52] provides properties and characterizations of facet defining inequalities, and also presents a peculiar decomposition result which may be formulated as follows. If $G = (V, E)$ is the 1-sum of the graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ (i.e. $V = V_1 \cup V_2$, $E = E_1 \cup E_2$ and $|V_1 \cap V_2| = 1$), then a complete formulation of \mathcal{D}_G can be deduced from the ones of \mathcal{D}_{G_1} and \mathcal{D}_{G_2} . We proceeded to similar investigations w.r.t. \mathcal{T}_f^G [84].

A complete description of $\mathcal{D}_G^{\bar{1}}$ for the case of domishold graphs appears in [162]. A graph is said to be *domishold* [39] if there exist real positive weights associated to the nodes so that a node subset is a dominating set if and only if the sum of the corresponding weights exceeds some threshold value. A complete formulation for cycles firstly appears in [51]. This work was extended in [42], leading to complete description for particular webs. Also recently, an exact extended formulation for cacti graphs was introduced in [18], together with a polynomial-time algorithm to solve [MW $_f$]

for cacti. Let us mention another particular, but interesting case: when $f_v = d_v^G$, for all $v \in V$, a node subset $S \subseteq V$ is an f -dominating set if and only if its complementary set $V \setminus S$ is a stable set. This implies that for this case it is equivalent to look for a maximum weight stable set and a minimum weight f -dominating set. From results on the stable set problem, it follows that $[MW_f]$ can be solved in polynomial time in perfect graphs for the particular case when $f_v = d_v$, for all $v \in V$, and that also a complete description of \mathcal{D}_G^f is known for that case [65, 193].

In the rest of the chapter we summarize part of the results from our investigations on the structures of the polytopes \mathcal{F}_G^f (Section 5.2), \mathcal{D}_G^f (Section 5.3) and \mathcal{U}_G^f (Section 5.4).

5.2 On the polyhedral structure of \mathcal{F}_G^f

5.2.1 Basic polyhedral results on \mathcal{F}_G^f

Let $G = (V, E)$ denote an undirected graph, and let $f \in \mathcal{F}_G$ such that $f_v < d_v$, for all $v \in V$. In this section we give basic polyhedral properties like dimension and facet-defining inequalities of \mathcal{F}_G^f .

Proposition 5.2.1. [84] *The following statements hold.*

- (i) *The polytope \mathcal{F}_G^f has dimension n , i.e. it is full dimensional.*
- (ii) *The trivial inequality $x_v \geq 0$ is facet defining for \mathcal{F}_G^f if and only if $f_w < d_w - 1$, for all $w \in N(v)$ such that $d_w \geq 2$.*
- (iii) *The trivial inequality $x_v \leq 1$ is facet defining for \mathcal{F}_G^f , for all $v \in V$.*

5.2.2 Decomposition results

In this section we provide results that allow us, in some cases, to decompose the search for a formulation of \mathcal{F}_G^f into several such searches but on graphs having a smaller order.

We start by introducing some simple auxiliary properties. Firstly, we show that when there exists a node for which the domination requirement equals its degree, in order to get a complete formulation of \mathcal{F}_G^f , we can easily reduce the situation to the case when $f_v < d_v$, for all $v \in V$.

Proposition 5.2.2. [84] *Let $f \in \mathcal{F}_G$ and assume there exists some node $v \in V$ such that $f_v = d_v$. Define $f' \in \mathcal{F}_G$ as follows: $f'_w = f_w$, for all $w \in V \setminus \{v\}$ and $f'_v = 0$. Then, a complete formulation of \mathcal{F}_G^f can be obtained by adding to a complete formulation of $\mathcal{F}_G^{f'}$ the set of equations $\{x_u = 1 : u \in N(v)\}$.*

Trivially, if the graph G is not connected, then a complete formulation of \mathcal{F}_G^f is obtained by aggregating the complete formulations corresponding to its connected components. So, in what follows, we can assume w.l.o.g. that G is connected. Given a graph $G = (V, E)$ and a node subset $S \subseteq V$, let $G[S]$ denote the subgraph of G that is induced by S , i.e. $G[S] = (S, E')$, where E' stands for the subset of edges in E having both endpoints in S .

Proposition 5.2.3. [84] *Let $G = (V, E)$ be an undirected graph, $f \in \mathcal{F}_G$, and assume that, for some node $u \in V$, the following holds: $f_u = d_u$ and $f_v \leq |N(v) \cap N(u)|$, for all $v \in N(u)$. Let $\tilde{G} = (\tilde{V}, \tilde{E}) = G[V \setminus N(u)]$ and let $\tilde{f}_v = f_v$, for all $v \in \tilde{V} \setminus S$ and $\tilde{f}_v = f_v - |N(v) \cap N(u)|$, for all $v \in S$ with $S = N(N(u))$. Then, a complete description of \mathcal{F}_G^f is obtained by adding to a complete description of $\mathcal{F}_{\tilde{G}}^{\tilde{f}}$ the equations $x_v = 1$, for all $v \in N(u)$ and the trivial inequalities $0 \leq x_u \leq 1$.*

The next result holds for the case when the node u used for decomposition is such that the number of connected components in $G[V \setminus \{u\}]$ equals d_u .

Proposition 5.2.4. [84] *Let $G = (V, E)$ be an undirected graph, $f \in \mathcal{F}_G$, such that $f_v < d_v$, for all $v \in V$. Consider some node $u \in V$, let C_1, C_2, \dots, C_p denote all the different connected components of*

$G[V \setminus \{u\}]$ and assume that $|C_i \cap N(u)| = 1$, for all $i \in \{1, 2, \dots, p\}$ (i.e. all the neighbors of u belong to different connected components of $G[V \setminus \{u\}]$). For each $i \in \{1, 2, \dots, p\}$, let $C_i = (V_i, E_i)$, define $\widehat{C}_i = G[V_i \cup \{u\}] = (\widehat{V}_i, \widehat{E}_i)$ and $\widehat{f}^i \in \mathcal{F}_{\widehat{C}_i}$ such that $(\widehat{f}^i)_w = f_w$, for all $w \in V_i$ and $(\widehat{f}^i)_u = 0$. Then, for any $\delta \in \{0, 1\}$, the polytope Q^δ defined by the aggregation of complete formulations of the polytopes $\mathcal{F}_{\widehat{C}_i}^{\widehat{f}^i}$ for $i = 1, 2, \dots, p$, together with the constraints $\sum_{v \in N(u)} x_v \geq f_u$ and $x_u = \delta$ is integral.

If the conditions for its application are satisfied, Proposition 5.2.4 together with Balas' result [19] allows the derivation of a simple extended formulation of \mathcal{F}_G^f from complete formulations related to the components $(\widehat{C}_i)_{i=1}^p$.

Proposition 5.2.5. [84] *In the setting of Proposition 5.2.4, \mathcal{F}_G^f is the projection onto the x -space of variables of the polytope defined by the following system.*

$$\begin{cases} A^i y^{k,i} \geq \lambda_k b^i, \forall i \in \{1, 2, \dots, p\}, \forall k \in \{1, 2\}, \\ \sum_{v \in N(u)} y_v^k \geq \lambda_k f_u, \forall k \in \{1, 2\}, \\ y_u^1 = 0, y_u^2 = \lambda_2, \lambda_1 + \lambda_2 = 1, \\ x = y^1 + y^2, \\ (x, y^1, y^2) \in (\mathbb{R}^V)^3, \lambda \in \mathbb{R}_+^2, \end{cases}$$

where $y^{k,i} \in \mathbb{R}^{\widehat{V}_i}$ denotes the restriction of y^k to the entries indexed on \widehat{V}_i , and such that $\mathcal{F}_{\widehat{C}_i}^{\widehat{f}^i} = \{z \in \mathbb{R}^{\widehat{V}_i} : A^i z \geq b^i\}$, for all $i \in \{1, 2, \dots, p\}$.

In the more particular case when, in addition to the setting of Proposition 5.2.4, we also have $f_v = 0$, for all $v \in N(u)$, then a complete formulation of \mathcal{F}_G^f in the original space of variables can be easily determined.

Proposition 5.2.6. [84] *In the setting of Proposition 5.2.4, assume in addition that $f_v = 0$, for all $v \in N(u)$. For each $i \in \{1, 2, \dots, p\}$, let f^i stand for the restriction of f to its entries indexed by V_i . Then, a complete formulation of \mathcal{F}_G^f is given by the aggregation of complete formulations of the polytopes $\mathcal{F}_{C_i}^{f^i}$ for $i = 1, 2, \dots, p$, together with the constraints $\sum_{v \in N(u)} x_v \geq f_u$, $0 \leq x_u \leq 1$.*

5.2.3 Complete formulations of \mathcal{F}_G^f for some special graph families

In this section we mention the graphs for which the trivial and neighborhood inequalities give a complete formulation of \mathcal{F}_G^f for all $f \in \mathcal{F}_G$, and then we provide complete formulations for cycles.

Given a graph $G = (V, E)$, its adjacency matrix is the matrix $A \in \{0, 1\}^{V \times V}$ satisfying $A_{uv} = 1$ if and only if $\{u, v\} \in E$. A graph is said to be *totally unimodular* [2] if its adjacency matrix is totally unimodular. Given $f \in \mathcal{F}_G$, let $P(f)$ denote the polytope corresponding to the feasible region of (LP-MWT_f) , i.e. $P(f) = \{x \in \mathbb{R}^V : Ax \geq f \text{ and } x \in [0, 1]^V\}$ (where $A \in \{0, 1\}^{V \times V}$ stands for the adjacency matrix of G). The next proposition characterizes the graphs for which $P(f)$ is integral for all $f \in \mathcal{F}_G$.

Proposition 5.2.7. [84] *The polytope $P(f)$ is integral for all $f \in \mathcal{F}_G$ if and only if G is totally unimodular.*

We now provide a complete description of \mathcal{F}_G^f , when G is a cycle, for any $f \in \{0, 1\}^V$.

Proposition 5.2.8. [84] *Let $G = (V, E)$ denote a cycle with node set $V = \{0, 1, \dots, n-1\}$, edge set $E = \{\{i, i+1 \bmod n\} : i \in V\}$, and let $f \in \{0, 1\}^V$. Then, a complete description of \mathcal{F}_G^f is given by the system (S_{odd}) (resp. (S_{even})) if n is odd (resp. even) with:*

$$(S_{\text{odd}}) \begin{cases} \sum_{v \in V} x_v \geq \lceil \frac{n}{2} \rceil \min_{v \in V} f_v, \\ \sum_{u \in N(v)} x_u \geq f_v, \forall v \in V, \\ x \in [0, 1]^V, \end{cases} \quad \text{and} \quad (S_{\text{even}}) \begin{cases} \sum_{k=0}^{\frac{n}{2}-1} x_{2k} \geq \lceil \frac{n}{4} \rceil \min_{k \in \llbracket 0, \frac{n}{2}-1 \rrbracket} f_{2k+1}, \\ \sum_{k=0}^{\frac{n}{2}-1} x_{2k+1} \geq \lceil \frac{n}{4} \rceil \min_{k \in \llbracket 0, \frac{n}{2}-1 \rrbracket} f_{2k}, \\ \sum_{u \in N(v)} x_u \geq f_v, \forall v \in V, \\ x \in [0, 1]^V. \end{cases}$$

5.3 On the polyhedral structure of \mathcal{D}_G^f

The results we give hereafter w.r.t. the polytope \mathcal{D}_G^f include different families of facet defining inequalities and a complete description for trees. In general, this formulation may contain an exponential number of constraints (whereas it is linear in the number of nodes for $\mathcal{D}_G^{\bar{1}}$), but we can show that the corresponding separation problem can be solved in polynomial time, thus implying the polynomial time solvability of $[MW_f]$ for trees, by the well-known results on the polynomial equivalence between optimization and separation established by Grötschel et al. [120]. In fact, for this case (of trees), we could also dedign a linear time combinatorial algorithm (via a dynamic programming approach) we do not discuss here (see [83] for details).

After some basic polyhedral properties are introduced, we present some nontrivial facet defining inequalities of \mathcal{D}_G^f and a complete formulation for trees.

5.3.1 Basic polyhedral properties of \mathcal{D}_G^f

Proposition 5.3.1. [83] *Let $f \in \mathcal{F}_G$. Then the following holds.*

- (i) *The polytope \mathcal{D}_G^f has dimension n , i.e., it is full dimensional.*
- (ii) *The trivial inequality $x_v \geq 0$ is facet defining for \mathcal{D}_G^f iff $d_w > f_w$, for all $w \in N[v]$, $v \in V$.*
- (iii) *The trivial inequality $x_v \leq 1$ is facet defining for \mathcal{D}_G^f , for all $v \in V$.*
- (iv) *Every facet defining inequality of \mathcal{D}_G^f which is not trivial is of the form $\sum_{v \in V} a_v x_v \geq b$, with $a_v \geq 0$, for all $v \in V$. Moreover $b > 0$, $|V_a = \{v \in V : a_v > 0\}| \geq 2$.*

Let $a^T x \geq b$ denote a non trivial facet-defining inequality of \mathcal{D}_G^f , and let $G_a = G[V_a]$ be the subgraph of G that is induced by the node subset V_a . The following property was shown to hold for the dominating set polytope [52], and its extension to f -dominating set polytopes is straightforward.

Proposition 5.3.2. [52] *The graph G_a is connected.*

5.3.2 Non trivial facet-defining inequalities

We start with a definition that will be useful for presenting different families of inequalities that are valid for \mathcal{D}_G^f .

Definition. Given an undirected simple graph $G = (V, E)$ and $f \in \mathcal{F}_G$, an f -clique is a node subset $Q \subseteq V$, such that $|Q| \geq 2$ and satisfying the following two conditions:

- (i) the nodes in Q are pairwise adjacent (i.e. Q is a clique in G), and
- (ii) $|\{v \in Q : f_v = d_v\}| \geq |Q| - 1$.

The next three propositions deal with cases when some nodes of the graph satisfy $f_v = d_v$. Notice that, if for each edge $\{u, v\} \in E$ we have $f_u = d_u$ or (not exclusively) $f_v = d_v$, then the following holds: $D \subseteq V$ is an f -dominating set iff D is a vertex cover. So in that case, the vertex cover and f -dominating set polytopes coincide (and are both affinely equivalent to the stable set polytope). The next proposition directly follows from this correspondence, see, e.g., [180].

Proposition 5.3.3. [83] *Let the graph $\mathcal{C} = (V(\mathcal{C}), E(\mathcal{C}))$ be an odd cycle such that each edge $\{u, w\} \in E(\mathcal{C})$ is an f -clique. Then, the inequality*

$$\sum_{v \in V(\mathcal{C})} x_v \geq \left\lceil \frac{|V(\mathcal{C})|}{2} \right\rceil \quad (5.1)$$

is facet defining for $\mathcal{D}_{\mathcal{C}}^f$ iff \mathcal{C} has no chord. (A chord is an edge joining two nonconsecutive nodes of the cycle).

The next two propositions illustrate the fact that, even though the property that “ $f_u = d_u$ or $f_v = d_v$ for each edge $\{u, v\} \in E$ ” may hold only on some parts of the graphs, some well-known families of inequalities valid for the vertex cover polytope, may, under some conditions, define facets of f -dominating set polytopes. We now formulate a simple sufficient condition for an inequality of type (5.1) corresponding to a node-induced subgraph of an arbitrary graph to define a facet of \mathcal{D}_G^f .

Proposition 5.3.4. [83] *Let $f \in \mathcal{F}_G$, and let $\mathcal{C} = (V(\mathcal{C}), E(\mathcal{C}))$ denote a node induced odd cycle in $G = (V, E)$ with no chord and such that each edge $\{u, w\} \in E(\mathcal{C})$ corresponds to an f -clique. If $|N(w) \cap V(\mathcal{C})| \leq 2$, for all $w \in V \setminus V(\mathcal{C})$, then the inequality (5.1) is facet-defining for \mathcal{D}_G^f .*

Proposition 5.3.5. [83] *Let $Q \subseteq V$ denote an f -clique. Then, the following inequality is valid for \mathcal{D}_G^f ,*

$$\sum_{v \in Q} x_v \geq |Q| - 1, \quad (5.2)$$

and it is facet-defining if and only if Q is maximal (w.r.t. inclusion).

We now introduce another family of valid inequalities for \mathcal{D}_G^f that we will call *partial neighborhood inequalities*, a denomination that is suggested by their support which is a subset of a closed neighborhood of a node having a positive domination requirement. These inequalities may be seen as a generalization of the classical neighborhood inequalities (obtained by setting $q = f_v = 1$ in (5.3) given hereafter) used in linear formulations of the dominating set problem. Recall that these inequalities, together with the trivial inequalities, are sufficient to completely describe the dominating set polytope ($f_v = 1$, for all $v \in V$) in the case of strongly chordal graphs [97]. Their relevance will be further stressed later, when considering complete formulations of the f -dominating set polytope for trees.

Proposition 5.3.6. [83] *Let $G = (V, E)$ denote an undirected simple graph, let $f \in \mathcal{F}_G$ and $u \in V$ such that $1 \leq f_u < d_u$. Then, the following partial neighborhood inequality is valid for \mathcal{D}_G^f :*

$$qx_u + \sum_{v \in N(u) \setminus Z_q} x_v \geq q, \quad (5.3)$$

with $q \in \{1, 2, \dots, f_u\}$, $Z_q \subseteq N(u)$ such that $|Z_q| = f_u - q$.

We now formulate a simple sufficient condition for an inequality of type (5.3) to be facet-defining for \mathcal{D}_G^f .

Proposition 5.3.7. [83] *Let $G = (V, E)$ denote an undirected simple graph and $f \in \mathcal{F}_G$. Let $u \in V$ such that $1 \leq f_u < d_u$, and assume that $|N(v) \setminus N[u]| \geq f_v$, for all $v \in V \setminus \{u\}$. Then the inequality (5.3) is facet-defining for \mathcal{D}_G^f , for all $q \in \{1, 2, \dots, f_u\}$ and $Z_q \subseteq N(u)$ such that $|Z_q| = f_u - q$.*

5.3.3 Complete descriptions of \mathcal{D}_G^f for trees

Theorem 5.3.8. [83] *If the graph $G = (V, E)$ is a tree and $f \in \mathcal{F}_G$, then a complete formulation of \mathcal{D}_G^f is given by the following set of inequalities.*

$$\left\{ \begin{array}{ll} 0 \leq x_v \leq 1, & v \in V, \\ x_u + x_v \geq 1, & f\text{-clique } \{u, v\}, \\ qx_u + \sum_{v \in N(u) \setminus Z_q} x_v \geq q, & u \in V \text{ such that } 1 \leq f_u < d_u, q \in \{1, 2, \dots, f_u\}, \\ & Z_q \subseteq N(u) \text{ with } |Z_q| = f_u - q. \end{array} \right.$$

5.4 On the polyhedral structure of \mathcal{U}_G^f

In this section we present some general properties on the polyhedral structure of f -tuple dominating set polytope. Then, we provide a complete formulation for the case of stars before we introduce a new family of valid inequalities, which led us to disprove a conjecture on a complete formulation for the 2-tuple dominating set polytope of trees. Finally, we conclude with some perspectives.

5.4.1 Basic polyhedral properties

Let \mathcal{R} denote the set of the nodes whose closed neighborhood belongs to all feasible solutions of (IP-MWU $_f$): $\mathcal{R} = \{v \in V: f_v = d_v + 1\}$.

Proposition 5.4.1. [186] Let $G = (V, E)$ denote an undirected simple graph and $f \in \widehat{\mathcal{F}}_G$. Then \mathcal{U}_G^f has dimension $|V| - |\mathcal{N}[\mathcal{R}]|$.

Proposition 5.4.2. [186] Let $f \in \mathcal{F}_G$ and $u \in V$. Then, the inequality $x_u \geq 0$ is facet defining for \mathcal{U}_G^f iff (if and only if) $f_v \leq d_v - 1$, for all $v \in \mathcal{N}[u]$.

Proposition 5.4.3. [186] Let $f \in \mathcal{F}_G$ and $u \in V$. Then, the inequality $x_u \leq 1$ is facet defining for \mathcal{U}_G^f .

Definition. Let $f \in \mathcal{F}_G$. A node $v \in V$ is said to be *critical* if $f_v = d_v$.

Proposition 5.4.4. [186] Let $f \in \mathcal{F}_G$ and let u denote a critical node such that $f_z \leq d_z - 1$, for all $z \in \mathcal{N}(u)$ such that $d_z \geq 2$. Then, the neighborhood inequality

$$\sum_{v \in \mathcal{N}[u]} x_v \geq f_u (= d_u), \quad (5.4)$$

is facet defining for \mathcal{U}_G^f .

Proposition 5.4.5. [186] Let $a^\top x \geq b$ denote a non trivial facet-defining inequality of \mathcal{U}_G^f , with $f \in \mathcal{F}_G$, and let its support be denoted by $S = \{v \in V: a_v \neq 0\}$. Then, the following holds:

(i) $|S| \geq 2$,

(ii) $a_v \geq 0$, for all $v \in V$, and $b > 0$.

5.4.2 Complete formulation for stars

For the case when the graph $G = (V, E)$ is a star having for center the node 1 and $f_1 \in \{0, 1\}$, the trivial inequalities together with the neighborhood inequalities provide a complete formulation of the polytope \mathcal{U}_G^f . This follows from the total balancedness of the closed neighborhood matrix (see [8, 69]). So, in the rest of this section we assume that the center 1 satisfies $f_1 \geq 2$ and that $n \geq 3$.

Proposition 5.4.6. Let $G = (V, E)$ denote a star having node 1 for center. Let $f \in \mathcal{F}_G$ such that $f_1 \geq 2$ and let (L_0, L_1) denote a partition of set of the leaves $\{2, 3, \dots, n\}$ with $L_0 = \{v \in V: f_v = 0\}$, (and $L_1 = V \setminus (\{1\} \cup L_0)$) (possibly $L_0 = \emptyset$ or $L_1 = \emptyset$). Then, a complete description of \mathcal{U}_G^f is given by the set of the trivial inequalities, the neighborhood inequalities of the nodes in L_1 , the following inequality:

$$[\max(|L_1|, f_1) - f_1 + 1]x_1 + \sum_{v \in V \setminus \{1\}} x_v \geq \max(|L_1|, f_1), \quad (5.5)$$

and the inequalities

$$(|L_1| - f_1 + |Z| + 1)x_1 + \sum_{v \in L_1 \cup (L_0 \setminus Z)} x_v \geq |L_1|, \quad (5.6)$$

for all node subsets $Z \subset L_0$ satisfying $\max(0, f_1 - |L_1|) < |Z| < f_1 - 1$ (in case such a set exists).

As an example of type (5.6) inequality, consider the star on eleven nodes illustrated by Figure 5.1, together with the domination requirements $(f_v)_{v \in V}$ reported close to each node. For the represented subsets L_0, L_1 and Z of the leaves, the corresponding facet defining inequality is $4x_1 + \sum_{v \in L_1 \cup (L_0 \setminus Z)} x_v \geq 5$. We can show all the inequalities of type (5.6) are facet defining for \mathcal{U}_G^f .

Proposition 5.4.7. [186] Under the setting of Proposition 5.4.6, each inequality of the type (5.6) is facet defining for \mathcal{U}_G^f .

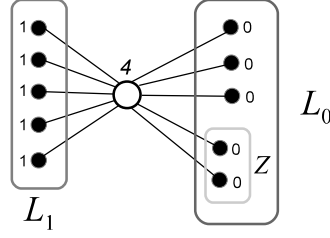


Figure 5.1 – Example to illustrate type (5.6) inequalities

Also, note that since \mathcal{U}_G^f is full dimensional (assuming $f \in \mathcal{F}_G$), the facets defined by the inequalities of type (5.6) are all distinct. Given this, a relevant consequence of Proposition 5.4.7 is the fact that there exist families of dominating set polytopes for which the number of facets grows exponentially with the number of nodes of the corresponding graph. This contrasts with the *classical* dominating set polytope $\mathcal{U}_G^{\bar{1}}$ for which the number of facets is $\mathcal{O}(n)$.

In the next proposition, we address the relevance of the inequalities (5.5)-(5.6) from an *optimization point of view* by considering the integrality gap of the formulation (LP1) for the case when G is a star and the weight function corresponds to the left-hand-side of an inequality of the type (5.5). The same result can be shown to hold if we consider a weight function corresponding to an inequality of type (5.6).

Proposition 5.4.8. [186] *Let $G = (V, E)$ denote a star having for center node 1, $n \geq 3$. Let $f \in \mathcal{F}_G$, $L_1 = \{v \in V \setminus \{1\} : f_v = 1\}$, $L_0 = V \setminus (\{1\} \cup L_1)$, and let $\hat{w} \in \mathbb{R}^n$ be defined as follows: $\hat{w}_1 = \max(|L_1|, f_1) - f_1 + 1$, and $\hat{w}_v = 1$, for all $v \in V \setminus \{1\}$. Then, the integrality gap of (LP1) with an objective function corresponding to \hat{w} is upper bounded by $\frac{4}{3}$ and this bound is asymptotically tight.*

5.4.3 Further valid inequalities

In this section, we introduce valid inequalities which extend the families (5.5) and (5.6) for stars. We also investigate on their facet-defining properties, disproving a conjecture on a complete formulation of the 2-tuple dominating set polytope for trees. We start with a simple extension of the inequalities (5.5).

Proposition 5.4.9. [186] *Let $G = (V, E)$ denote an undirected simple graph and $f \in \mathcal{F}_G$. Given any node u , let $L_0^u = \{v \in N(u) : f_v \leq d_v - 1\}$, and $L_1^u = N(u) \setminus L_0^u$. Then, the following inequalities are valid for \mathcal{U}_G^f .*

$$[\max(|L_1^u|, f_u) - f_u + 1] x_u + \sum_{v \in N(u)} x_v \geq \max(|L_1^u|, f_u), \text{ for all } u \in V. \quad (5.7)$$

Proposition 5.4.10. [186] *Let $G = (V, E)$ be an undirected simple graph and let $f \in \mathcal{F}_G$. Let $u \in V$ denote an articulation point such that $f_u \geq 2$ and all the neighbors of u in G belong to different connected components in $G[V \setminus \{u\}]$. Then the inequality (5.7) is facet defining w.r.t. \mathcal{U}_G^f .*

We now introduce a new family of inequalities generalizing (5.6).

Proposition 5.4.11. [186] *Let $G = (V, E)$ denote an undirected simple graph and let $f \in \mathcal{F}_G$. Let $u \in V$ denote an articulation point such that $f_u \geq 2$ and all the neighbors of u in G belong to different connected components in $G[V \setminus \{u\}]$. Let (Z, L_0, L_1) denote a tripartition of $N(u)$ such that*

- (i) $f_u - |L_1| < |Z| < f_u - 1$,
- (ii) $f_v \leq d_v - 1$, for all $v \in Z \cup L_0$,
- (iii) for each node $w \in L_1$, $f_w \geq 1$ and $|Q_w| \geq \bar{d}_w - f_w$, with $Q_w = \{v \in N(w) \setminus \{u\} : f_v \leq d_v - 1\}$.

For each node $w \in L_1$, let $\bar{N}(u, w)$ denote a subset of $Q_w \cup \{w\}$ such that $w \in \bar{N}(u, w)$ and $|\bar{N}(u, w)| = d_w - f_w + 1$. Then, the following inequality is valid for \mathcal{U}_G^f .

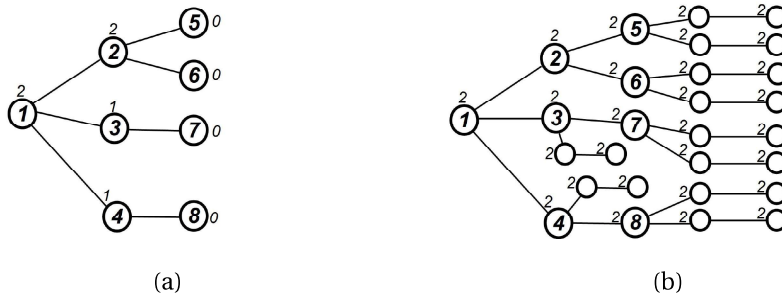
$$(|L_1| - f_u + |Z| + 1) x_u + \sum_{w \in L_1} \sum_{v \in \bar{N}(u, w)} x_v + \sum_{v \in L_0} x_v \geq |L_1|. \quad (5.8)$$

Proposition 5.4.12. [186] *If the graph G is a tree, then the inequality (5.8) is facet defining w.r.t. \mathcal{U}_G^f .*

For an example of a facet defining inequality of type (5.8), consider the graph G' of Figure 5.2-(a) where the node domination requirements $(f'_v)_{v \in V}$ correspond to the values close to the nodes. Then the inequality $2x_1 + x_2 + x_3 + x_4 + x_5 + x_7 + x_8 \geq 3$ is facet-defining for $\mathcal{U}_{G'}^{f'}$.

A conjecture formulated by Argiroffo (see Section 3 in [9]) on the formulation of the 2-dominating set polytope of trees (i.e. the f -tuple dominating set polytope of trees for the particular case when $f_v = 2$, for all $v \in V$) stated that a complete formulation of the 2-tuple dominating set polytope was given by a set of inequalities, each one having a support included in the closed neighborhood of a single node. From Proposition 5.4.12, it follows that this conjecture does not hold: the instance illustrated in Figure 5.2-(b) provides a counterexample since a complete formulation of \mathcal{U}_G^f can be obtained by adding to the one of the instance from Figure 5.2-(a), the set of equations $x_v = 1$, for all $v \in N[\mathcal{R}]$.

Figure 5.2 – Illustrations for a facetdefining inequality of type (5.8)



5.5 Some perspectives

In this chapter, we presented part of our results on polytopes related two three different kinds of domination in graphs. In particular, complete formulations could be obtained for some particular cases. On the algorithmic side, for conciseness, we did not presented it above but we can design linear time (dynamic programming type) algorithms to solve the weighted optimization problems related to the three concepts of domination for the case of trees.

Further research work may be directed towards the polyhedral structure of the polytopes related to the three concepts (and potentially others) for other graph families such as cacti and (strongly) chordal graphs. One may also look for (extended) compact formulations for such graph families and others.

On the algorithmic side, we should look for polynomial time algorithms to deal with the weighted optimization problems related to all of the generalized concepts for other graph families than trees.

Another aspect we started to study w.r.t. the f -tuple domination concept is the determination of the integrality gap of different formulations of the corresponding (weighted or not) optimization problem. This has to be pursued further and does no seem to have been tackled yet w.r.t. the other generalized concepts.

In addition to the results reported above, we also started to study the structure of the f -tuple dominating set polytope from another perspective, looking for characterizations of adjacency relations in its 1-skeleton graph. Such aspects seem to be almost uninvestigated yet in the literature for all of the generalized concepts.

Chapter 6

The k -separator problem

This chapter deals with a graph optimization problem that is called the k -separator problem, where k represents a positive integer value. It consists in identifying a node subset S in a node weighted undirected graph, such that the sum of the weights of the nodes in S is minimum and the removal of S induces a subgraph having connected components of size at most k only. We investigated this problem from different perspectives. In particular connections with the stable set problem could be established leading to an interesting LP relaxation and to the identification of many graph families for which the problem can be solved in polynomial time. Many efforts have also been directed towards the determination of new families of facet-defining inequalities, and complete formulations for some special cases. Exact combinatorial algorithms for some specific instances but also approximate algorithms have been developed. We summarize our contributions hereafter, mainly focusing on LP based or related approaches.

The content of this chapter stems from joint works with Prof. Walid Ben-Ameur and Mohamed-Ahmed Mohamed-Sidi. Further results and details are contained in our publications [30, 31].

6.1 Introduction

Basic definitions

Let $G = (V, E, w)$ denote a node-weighted undirected graph, where $w \in \mathbb{R}^V$, and let k be a positive integer. A node subset S whose removal leads to a graph (i.e. the subgraph of G that is induced by $V \setminus S$), such that the size of each connected component is less than or equal to k is called a k -separator. The *weight* of a k -separator is the quantity $w(S) = \sum_{v \in S} w_v$. We consider the following natural generalization of the well-known vertex cover problem, that we name the k -separator problem: compute a minimum weight k -separator in G . (The vertex cover problem corresponds to the case $k = 1$.)

Motivations

The k -separator problem has many applications. If the vertex weights are all equal to 1, the size of a minimum k -separator can be used to evaluate the robustness of a graph or a network. Intuitively, a graph for which the size of the minimum k -separator is large, is more robust. Unlike the classical robustness measure given by the connectivity, the new one seems to avoid to underestimate robustness when there are only some local weaknesses in the graph.

In the context of social networks, many approaches are proposed to detect communities. By solving a minimum k -separator problem, we get different connected components that may represent communities. The k -separator vertices represent persons making connections between communities. The k -separator problem can then be seen as a special partitioning/clustering graph problem.

Computing a k -separator can also be useful to build algorithms based on divide-and-conquer approaches. In some cases, a problem defined on a graph can be decomposed into many sub-

problems on smaller subgraphs obtained by the deletion of a k -separator (see, e.g., [209]). Other applications related to matrix decompositions and other fields are mentioned in [190].

Related works

To our knowledge, prior to our work, the only paper where the k -separator problem is considered in its general form seems to be [190]. The following extended formulation is proposed in [190]. In addition to binary variables $(x_i)_{i \in V}$ expressing whether a vertex i belongs to the k -separator, a binary variable x_{ij} is considered for each pair of vertices $\{i, j\}$ to indicate whether i and j belong to the same component ($x_{ij} = 0$) or not ($x_{ij} = 1$). Then, triangle inequalities (the first ones in the LP formulation given hereafter) are clearly valid. Moreover, to express the fact that a connected component does not contain more than k vertices, they added the constraints $\sum_{j \in V \setminus \{i\}} x_{ij} \geq n - k$, for all $i \in V$. Finally, other constraints are added to impose that if two adjacent vertices are not in the k -separator, then they belong to the same component: $x_i + x_j - x_{ij} \geq 0$, for all $\{i, j\} \in E$. The formulation is given below.

$$\left\{ \begin{array}{ll} \min & \sum_{v \in V} w_v x_v \\ & x_{ij} \leq x_{ik} + x_{jk}, \quad i, j, k \in V, \\ & \sum_{j \in V \setminus \{i\}} x_{ij} \geq n - k, \quad i \in V, \\ & x_i + x_j - x_{ij} \geq 0, \quad \{i, j\} \in E, \\ & x_{ij} \in [0, 1], \quad i, j \in V, \\ & x_i \in \{0, 1\}, \quad i \in V. \end{array} \right.$$

Observe that the x_{ij} variables are not required to be integer. In fact, as noticed in [190], relaxing the integrality constraint of x_{ij} variables does not modify the solution of the integer program. The polytope related to the formulation above is studied in [190] and many valid inequalities and facets are presented there.

The case $k = 1$ corresponds to the vertex cover problem (or the stable set problem) that received a lot of attention in the literature. When weights are unitary and $k = 2$, the k -separator problem is equivalent to compute the dissociation number of the graph [219]. A subset of vertices in a graph is called a dissociation set if it induces a subgraph so that each vertex has degree at most 1, and the dissociation number is the size of a largest dissociation set. In what follows, if the graph G does not contain an induced subgraph isomorphic to some given graph H , then G is said to be H -free. Computing the dissociation number is NP-hard if the graph is bipartite [219]. The NP-hardness still holds for $K_{1,4}$ -free bipartite graphs [46], C_4 -free bipartite graphs with a maximum vertex degree of 3 [46], planar graphs with a maximum vertex degree of 4 [195], and line graphs [192]. Several cases where the dissociation problem can be solved in polynomial time have been shown in the literature: chordal and weakly chordal graphs, asteroidal triple-free graphs [55], $(P_k, K_{1,n})$ -free graphs (for any positive numbers k and n) [157] and (G_1, G_2, G_3) -free graphs [192]. We recall the graphs G_1, G_2, G_3 and P_k for $k = 6$ (an elementary path of length $k - 1 = 5$) in Figure 6.1. Some inapproximability results related to the problem of computing the dissociation number

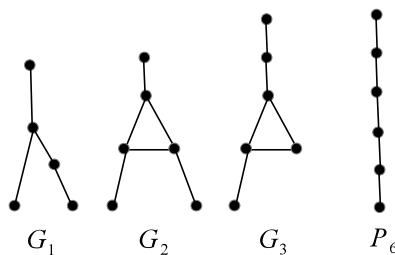


Figure 6.1 – The graphs G_1, G_2, G_3 and P_6

are also provided in [192].

The k -separator problem is closely related to the vertex-separator problem where we aim to remove a minimum-weight set of vertices such that each connected component in the remaining graph has a size less than $\alpha|V|$ (for a fixed $\alpha < 1$). A polyhedral study of this problem is proposed in [20] (see also the references therein). When the vertex-separator problem is considered, the graph is generally partitioned into 3 subgraphs: the separator, and two subgraphs each of size less than $\alpha|V|$. The philosophy is different in the case of the k -separator where the graph is partitioned into many components each of size at most k .

Additional notation

Given a vertex subset $U \subset V$, the complement of U in G , i.e. the vertex subset $V \setminus U$ is denoted by \overline{U} . The set of vertices (resp. edges) of the graph G may also be denoted by $V(G)$ (resp. $E(G)$). An edge $e \in E$ with endpoints u and v is denoted by $\{u, v\}$.

Given a vertex subset $S \subset V$, the set of vertices in \overline{S} that are adjacent to at least one vertex in S is denoted by $N(S)$. Given two subsets of vertices A and B , they are called *adjacent* if either $A \cap B \neq \emptyset$ or $N(A) \cap B \neq \emptyset$.

Given a subset of vertices $S \subset V$, $\chi^S \in \{0, 1\}^V$ denotes the incidence vector of S . The convex hull of all the incidence vectors of k -separators in the graph G is denoted by $\mathcal{S}_k(G)$. We also use $G[S]$ to denote the subgraph of G that is induced by a subset of vertices $S \subset V$.

The size of a graph denotes its number of vertices. Given some integer m , K_m denotes a complete graph with size m , mK_2 denotes a matching with m edges, P_m denotes a simple path containing m vertices, C_m denotes an elementary cycle of size m and $K_{1,m}$ denotes a complete bipartite graph with partition classes of cardinalities 1 and m (a star).

If G is a simple path with vertex set $\{v_1, \dots, v_n\}$ and edge set $\{\{v_i, v_{i+1}\} : i = 1, \dots, n-1\}$, then the notation $[v_i, v_j]$ (resp. $]v_i, v_j[$, $[v_i, v_j[$, $]v_i, v_j]$) with $i < j$, $i, j \in \{1, \dots, n\}$ stands for the vertex set $\{v_i, v_{i+1}, \dots, v_j\}$ (resp. $\{v_{i+1}, \dots, v_{j-1}\}$, $\{v_i, v_{i+1}, \dots, v_{j-1}\}$, $\{v_{i+1}, \dots, v_j\}$). The set of all the simple paths joining i and j will be denoted P_{ij} . Given a simple path p joining i and j , $x(p)$ stands for the sum of the x_v values over all vertices belonging to p (including i and j).

Organization

The rest of this chapter is organized as follows. We start introducing several results exhibiting some polynomially solvable cases of the k -separator problem in Section 6.2. Then, we delve in Section 6.3 into a polyhedral study of the convex hull of the incidence vectors of all the k -separators. Approximability results are presented in Section 6.4, before we conclude in Section 6.5.

6.2 Polynomially solvable cases

6.2.1 Graphs with bounded treewidth

A tree-decomposition of a graph $G = (V, E)$ is defined by a pair (\mathcal{X}, T) where $\mathcal{X} = (X_t)_{t \in V_T}$ is a set of vertex subsets of G indexed by vertices of a tree $T = (V_T, E_T)$ satisfying the following:

- (i) for each vertex $v \in V$, there is some $t \in V_T$ such that $v \in X_t$;
- (ii) for each edge $(u, v) \in E$, there is some $t \in V_T$ such that $u \in X_t$ and $v \in X_t$;
- (iii) for each vertex $v \in V$, if $v \in X_{t_1}$ and $v \in X_{t_2}$ then v belongs to X_t for each $t \in V_T$ on the path between t_1 and t_2 .

Property (iii) implies that the subgraph of T induced by the vertices t such that X_t contains v is a subtree. The width of the decomposition is given by $\max_{t \in V_T} \{|X_t| - 1\}$. The treewidth of G is the minimum width over all tree-decompositions of G .

We assume here that G has a treewidth bounded by a constant l . It is well-known that computing the treewidth of a graph and a corresponding minimum-width tree-decomposition can be done in linear time (assuming that l is constant) [45]. Many NP-hard optimization problems can be solved in polynomial time for bounded-treewidth graphs. The algorithms are generally based on dynamic programming and a tree-decomposition of the graph (see, e.g., [11, 49, 211]). Though the approaches in [11, 211] cannot be used to obtain a polynomial time algorithm for the k -separator problem, we could devise a polynomial time dynamic programming algorithm making use of a particular tree decomposition of G , leading to the next result (see Section 4.1 in [30] for details).

Proposition 6.2.1. [30] *The k -separator problem can be solved in polynomial time for bounded-treewidth graphs. This holds even if k is part of the input.*

6.2.2 Connections with the stable set problem and consequences

In this subsection, we firstly present a construction to reduce the k -separator problem to a maximum weight stable set problem. Then, we compare the feasible regions of two LP relaxations of the k -separator problem, one of them being derived from our reduction to the stable set problem. Finally we report complexity results that could be obtained, making use of this reduction.

Reducing the k -separator problem to a stable set problem

Given a vertex-weighted graph G , we build a vertex-weighted extended graph $G^* = (V^*, E^*, w)$ as follows. Each subset of vertices $S \subset V$ such that $1 \leq |S| \leq k$ and $G[S]$ is connected, is represented by a vertex in G^* . In others words, $V^* = \{S \subset V, |S| \leq k, G[S] \text{ is connected}\}$. The edge set is defined as follows: $E^* = \{(S, T), S \in V^*, T \in V^*, S \neq T, \text{ such that either } S \cap T \neq \emptyset, \text{ or } uv \in E \text{ for some } u \in S \text{ and } v \in T\}$. Said another way, $S \in V^*$ and $T \in V^*$ are connected by an edge if they are adjacent in G . The weight of a vertex $S \in V^*$ is defined by $w_S = \sum_{v \in S} w_v$.

Let R be a maximum weight stable set of G^* . If two nodes $S \in V^*$ and $T \in V^*$ belong to this stable set R , then $S \cap T = \emptyset$, and there are no edges in G with one endpoint in S and another in T . In other words, if we consider $\cup_{S \in R} S$, we get a node set in V inducing a subgraph, where each connected component has a size less than or equal to k . The complementary set of $\cup_{S \in R} S$ in V is a k -separator for the graph G . This graph construction can be seen as a generalization of a construction proposed by [157] for the dissociation problem ($k = 2$).

On the feasible region of an LP relaxation derived from a reduction to the stable set problem

Let us now illustrate the fact that the graph transformation we just described may be used to derive interesting formulations for the k -separator problem. Hereafter is an initial formulation which straightforwardly follows from the original problem statement: the objective is to find a set of nodes with minimum cost such that it intersects each set of nodes inducing a connected component having size $k + 1$ in the original graph. The Boolean variable x_v represents the fact that the node v belongs ($x_v = 1$) or not ($x_v = 0$) to the k -separator.

$$(IP1) \begin{cases} \min & \sum_{v \in V} w_v x_v \\ & \sum_{v \in S} x_v \geq 1, \quad S \subset V, |S| = k + 1, G[S] \text{ connected}, \\ & x_v \in \{0, 1\}, \quad v \in V. \end{cases}$$

From the graph transformation described above, we can easily derive another formulation that we may express as follows: find a stable set in G^* (this corresponds to the set of variables $(y_S)_{S \in V^*}$ and the constraints $y_S + y_T \leq 1, \forall (S, T) \in E^*$), so that the set of nodes in V that are not represented in this stable set (i.e. a k -separator in G , from our discussion above) has minimum total cost. The Boolean variable $y_S, S \in V^*$, represents the fact that the node $S \in V^*$ in G^* either belongs to the

stable set ($y_S = 1$) or not ($y_S = 0$).

$$(IP2) \left\{ \begin{array}{l} \min \quad \sum_{v \in V} w_v x_v \\ x_v = 1 - \sum_{S \in Q_v} y_S, \quad v \in V, \\ y_S + y_T \leq 1, \quad \{S, T\} \in E^*, \\ y_S \in \{0, 1\}, \quad S \in V^*, \end{array} \right.$$

with $Q_v = \{T \in V^* : v \in T\}$. Let F1 (resp. F2) stand for the set of feasible solutions of the linear relaxation of (IP1) (resp. (IP2)) with respect to variables $(x_v)_{v \in V}$.

Proposition 6.2.2. [30] *The following inclusion holds: $F2 \subseteq F1$.*

Proof. Let (x, y) stand for a feasible solution of the linear relaxation of (IP2). Let C denote a connected component of size $k + 1$ in the original graph $G = (V, E)$. So we have: $\sum_{v \in C} x_v = k + 1 - \sum_{v \in C} \sum_{S \in Q_v} y_S$. Notice that in the last expression each variable y_S such that S has a nonempty intersection with C occurs exactly $|S \cap C|$ times.

Let T denote a spanning tree of C (in the original graph) and consider the following quantity: $\sum_{(v,w) \in T} \sum_{S \in Q_v \cup Q_w} y_S$. Notice that in the last expression, the number of times a variable y_S occurs is equal to the number of edges of T that intersect with S , and thus is larger than or equal to $|S \cap C|$. From this we deduce that $\sum_{v \in C} \sum_{S \in Q_v} y_S \leq \sum_{(v,w) \in T} \sum_{S \in Q_v \cup Q_w} y_S$. Moreover, using the feasibility of (x, y) , we can write that $\sum_{(v,w) \in T} \sum_{S \in Q_v \cup Q_w} y_S \leq k$.

Combining the two previous inequalities leads to $\sum_{v \in C} \sum_{S \in Q_v} y_S \leq k$. Consequently, inequality $\sum_{v \in C} x_v \geq 1$ holds. In other words, x is a feasible solution of the linear relaxation of (IP1). \square

Both linear relaxations of (IP1) and (IP2) may be easily strengthened with many families of inequalities. In Section 6.3 a polyhedral study of the convex hull of the feasible region to (IP1) is reported, with several families of facet-defining inequalities and complete formulations for some particular cases.

Complexity results from connections with the maximum weight stable set problem

The reduction of the k -separator problem to the stable set problem presented in Section 6.2.2 can be used to derive complexity results on the former from known results on the latter. The starting point to get the results mentioned in the next proposition, is a graph family for which it is known that the maximum weight stable set problem can be solved in polynomial time. Assuming k is fixed (so that G^* has a polynomial size), it then suffices to show that, under some suitable assumptions on the original graph G , the transformed graph G^* belongs to this family.

Proposition 6.2.3. [30] *Assume that $k \geq 2$ is fixed. Then, the k -separator problem can be solved in polynomial time in the following cases:*

- mK_2 -free graphs, assuming m is a fixed positive integer.
- (G_1, G_2, G_3, P_6) -free graphs, assuming $k \geq 3$.
- interval filament, asteroidal triple-free, and weakly chordal graphs.

Removing the assumption that k is fixed, the transformed graph G^* may no longer have polynomial size. However, for the case of circular-arc graphs, we could reduce the k -separator problem to a stable set problem in an auxiliary graph of polynomial size, leading to the next result.

Proposition 6.2.4. [30] *The k -separator problem can be solved in polynomial time in circular-arc graphs.*

6.3 Polyhedral results

In this section, we provide results from our investigations on the polyhedral structure of the polytope $\mathcal{S}_k(G)$. After we present some general properties, we focus on the path and cycle cases. Finally, further families of valid inequalities are described.

6.3.1 Some general properties

We start our polyhedral study of $\mathcal{S}_k(G)$ giving its dimension, some elementary facets and general features that are satisfied by any facet defining inequality for $\mathcal{S}_k(G)$.

Proposition 6.3.1. [30] *The k -separator polytope $\mathcal{S}_k(G)$ is full-dimensional.*

Proposition 6.3.2. [30] *If $k \geq 2$, the trivial inequalities $x_v \leq 1$ and $x_v \geq 0$, for all $v \in V$ are facet defining for $\mathcal{S}_k(G)$.*

Proposition 6.3.3. [30] *If $a^\top x \geq \alpha$ denotes a facet defining inequality for $\mathcal{S}_k(G)$ different from the trivial inequalities (i.e. $0 \leq x_v \leq 1$, for all $v \in V$) then necessarily $a_v \geq 0$, for all $v \in V$ and $\alpha > 0$.*

Recall that the support of a valid inequality $a^\top x \geq \alpha$ is the set of vertices $v \in V$ such that $a_v \neq 0$.

Proposition 6.3.4. [30] *The support of any non-trivial facet of $\mathcal{S}_k(G)$ necessarily corresponds to a connected component of G having at least $k + 1$ vertices.*

The following proposition characterizes when a facet defining inequality for $\mathcal{S}_k(G)$ is also facet defining for $\mathcal{S}_k(G')$, where G' is obtained from G by adding a vertex (and a set of edges between this additional vertex and vertices in G).

Proposition 6.3.5. [30] *Let $a^\top x \geq b$ define a facet of the k -separator polytope $\mathcal{S}_k(G)$ with $G = (V, E)$. Let $G' = (V' = V \cup \{v\}, E')$ denote a graph obtained from G by adding a vertex v and some edges of the form $\{v, w\}$, $w \in V$. Then the inequality $a^\top x \geq b$ defines a facet of $\mathcal{S}_k(G')$ iff there exists a k -separator $S \subseteq V$ in G' whose incidence vector $\chi(S) \in \mathbb{R}^V$ satisfies $a^\top \chi(S) = b$.*

Proposition 6.3.6. [30] *Let the inequality $a^\top x \geq b$ be facet inducing for $\mathcal{S}_k(G)$, $G = (V, E)$. Then it is also facet defining for $\mathcal{S}_k(G')$ where $G' = (V', E')$ denotes the subgraph of G that is induced by the vertex set $V' \subseteq V$ and such that $a_v = 0$, for all $v \in V \setminus V'$.*

6.3.2 The path and cycle cases

The linear relaxation of (IP1) does not provide a complete formulation of $\mathcal{S}_k(G)$, even when restricting G to belong to the class of trees, and determining a complete formulation even in this very particular case is still an open problem. However, we were able to provide a complete formulation for some special cases and the purpose of this section is to illustrate such results. They are related to paths and cycles.

Let us assume first that $G = (V, E)$ is a path where $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{v_1 v_2, \dots, v_{n-1} v_n\}$. The connected components of size $k + 1$ considered in LP1 (the linear relaxation of the integer program (IP1)) are denoted by $S_1, \dots, S_i, \dots, S_{n-k}$, where $S_i = \{v_i, v_{i+1}, \dots, v_{i+k}\}$. The constraints of LP1 related to connected components of size $k + 1$ can be written in the matrix form $Ax \geq \vec{1}$, where the i -th row of A corresponds to the incidence vector of S_i . By showing that the constraint matrix is totally unimodular we obtained the next result.

Proposition 6.3.7. [30] *If G is a path, then formulation LP1 is exact (i.e. any optimal solution of (IP1) is also an optimal solution of (LP1)).*

Let us now assume that $G = (V, E)$ is a cycle where $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{v_1 v_2, \dots, v_i v_{i+1}, \dots, v_{n-1} v_n, v_n v_1\}$. It is clear that, if we know that $x_{v_i} = 1$ for some vertex v_i (i.e. v_i belongs to the k -separator), then any k -separator of the cycle should contain a k -separator of the path $V \setminus v_i$.

Using Proposition 6.3.7, we deduce that a minimum weight k -separator containing vertex v_i can be computed by solving the following linear program.

$$\left\{ \begin{array}{l} \min \quad \sum_{v \in V} w_v x_v \\ \text{s.t.} \\ \sum_{v \in S} x_v \geq 1, \quad \forall S, |S| = k+1, G[S] \text{ connected and } v_i \notin S, \\ x_{v_i} = 1, \\ x_v \in [0, 1], \quad \forall v \in V. \end{array} \right.$$

Let P_i be the polytope corresponding to the feasible region of the formulation above. Then, $P_i = \text{conv}(\{\chi^{(S)} \in \{0, 1\}^n, S \text{ is a } k\text{-separator containing } v_i\})$, where $\text{conv}(X)$ stands for the convex hull of X , for any set $X \subseteq \mathbb{R}^V$. Let T be an arbitrary subset of vertices such that $|T| = k+1$ and $G[T]$ is connected. Since at least one vertex of T belongs to the k -separator, we can write that $\mathcal{S}_k(G) = \text{conv}\left(\bigcup_{v_i \in T} P_i\right)$.

Using the projection result of Balas [19], we get the following equivalent formulation for the k -vertex separator problem when the graph is a cycle:

$$\left\{ \begin{array}{l} \min \quad \sum_{v \in V} w_v x_v \\ \text{s.t.} \\ x = \sum_{i \in T} z^i, \\ 0 \leq z_j^i \leq z_i^i, \quad \forall v_i \in T, v_j \in V, \\ \sum_{v_j \in S} z_j^i \geq z_i^i, \quad \forall S, |S| = k+1, G[S] \text{ connected, and } v_i \in T \setminus S, \\ \sum_{v_i \in T} z_i^i = 1. \end{array} \right.$$

In the formulation above, z^i is a vector of dimension n whose components are given by z_j^i for $v_j \in V$. Let us now focus on some special cases. Let C_k denote a cycle with length k .

Proposition 6.3.8. [30] $\mathcal{S}_k(C_{k+1}) = \{x \in [0, 1]^{k+1} : \sum_i x_i \geq 1\}$.

The formulations for $\mathcal{S}_k(C_{k+1})$ and $\mathcal{S}_k(P_{k+1})$ are the same (where P_{k+1} stands for an elementary path obtained from C_{k+1} by removing an edge).

Proposition 6.3.9. [30] $\mathcal{S}_k(C_{k+2}) = \{x \in [0, 1]^{k+2} : \sum_i x_i \geq 2\}$.

Note that differently from the case of C_{k+1} , the formulations of $\mathcal{S}_k(C_{k+2})$ and $\mathcal{S}_k(P_{k+2})$ do not coincide (since the constraint $\sum_i x_i \geq 2$ is not valid for $\mathcal{S}_k(P_{k+2})$).

For the particular case of C_5 and $k=2$ the following proposition shows that the addition of the constraints on all simple paths containing $k+1$ vertices provides an exact formulation.

Proposition 6.3.10. [30] $\mathcal{S}_2(C_5) = \{x \in [0, 1]^{k+1} : \sum_i x_i \geq 2, \sum_{i \in p} x_i \geq 1, \forall p \in \mathcal{P}_3\}$, where \mathcal{P}_3 stands for the set of all simple paths on C_5 containing 3 vertices.

6.3.3 Some families of valid inequalities for $\mathcal{S}_k(G)$

Hitting set inequalities

Hitting set inequalities are the basic inequalities, already mentioned above in the formulation (IP1):

$$\sum_{v \in S} x_v \geq 1, \quad \forall S \subset V, |S| = k+1, G[S] \text{ connected.} \quad (6.1)$$

Proposition 6.3.11. [30] Let S be a subset of vertices such that $G[S]$ is connected and $|S| = k+1$. Then the inequality $\sum_{v \in S} x_v \geq 1$ defines a facet of $\mathcal{S}_k(G)$ if each vertex $w \in V \setminus S$ is adjacent to at most one vertex in S .

If G is a tree and S is a subtree of G , then each vertex $v \in \bar{S}$ has at most one neighbor in G . This leads to the following corollary.

Corollary 6.3.12. [30] If G is a tree, then each inequality (6.1) induces a facet of $\mathcal{S}_k(G)$.

Connectivity inequalities

Proposition 6.3.13. [30] *Let $S \subset V$ be a subset of vertices such that $|S| \geq k + q$, $q \geq 1$ and $G[S]$ is q -vertex-connected. Then the following inequality is valid for $\mathcal{S}_k(G)$:*

$$\sum_{v \in S} x_v \geq q. \quad (6.2)$$

Notice that inequalities (6.2) were also considered in [190] but in a more restricted form (it is required in [190] that $|S| = k + q$). Let us focus on the special case where $|S| = k + q$. The next result can be seen as a generalization of Proposition 6.3.11 where q was equal to 1.

Proposition 6.3.14. [30] *Let $S \subset V$ be a subset of vertices such that $|S| = k + q$, $q \geq 1$ and $G[S]$ is q -vertex-connected. Then inequality (6.2) induces a facet of $\mathcal{S}_k(G)$ if each vertex $w \in V \setminus S$ is adjacent to at most q vertices in S .*

If we consider the more restrictive assumption: $G[S]$ is $q + 1$ -vertex-connected, then the condition related to the number of neighbors in S becomes necessary and sufficient to obtain a facet.

Proposition 6.3.15. [30] *Let $S \subset V$ be a subset of vertices such that $|S| = k + q$, $q \geq 1$ and $G[S]$ is $q + 1$ -vertex-connected. Then inequality (6.2) induces a facet of $\mathcal{S}_k(G)$ if and only if each vertex $w \in V \setminus S$ is adjacent to at most q vertices in S .*

The case where $G[S]$ is a clique was considered in [190]. The previous proposition can be seen as a generalization of the clique case.

Cycle inequalities

Proposition 6.3.16. [30] *Let S be a subset of vertices such that $|S| \geq k + 1$ and $G[S]$ is an elementary cycle, then the following inequality is valid for $\mathcal{S}_k(G)$:*

$$\sum_{v \in S} x_v \geq \left\lceil \frac{|S|}{k + 1} \right\rceil. \quad (6.3)$$

Notice that when $|S| = k + 1$, Proposition 6.3.15 can be applied with $q = 1$ to know under which conditions inequality (6.3) induces a facet. Let us then focus on the case where $|S| > k + 1$. It is clear that if $|S| \equiv 0 \pmod{k + 1}$, then (6.3) is just the sum of inequalities of type (6.1).

Proposition 6.3.17. [30] *Let S be a subset of vertices such that $|S| > k + 1$, $|S|$ is not a multiple of $k + 1$, and $G[S]$ is an elementary cycle, then inequality (6.3) induces a facet of $\mathcal{S}_k(G)$ if and only if for each vertex $w \in \bar{S}$, there is a k -separator of size $\lceil \frac{|S|}{k + 1} \rceil$ in $G[S \cup \{w\}]$.*

Further inequalities

Let S be a set of vertices with $|S| \geq k$, $G[S]$ connected and let $i \in \bar{S}$. For each $j \in S$, let $p_{ij} \in P_{ij}$ be a path joining i and j . Notice that $p_{ij} \setminus \{i\}$ is a path joining j and the neighbor of i in p_{ij} . Consider the following inequality

$$(|S| + 1 - k)(1 - x_i) \leq \sum_{j \in S} x(p_{ij} \setminus \{i\}). \quad (6.4)$$

Lemma 6.3.18. [30] *Inequalities (6.4) are valid for $\mathcal{S}_k(G)$.*

Observe that inequalities (6.4) can be separated in polynomial time as follows. The subset S is initially empty. For each vertex $i \in V$, we first compute the shortest path p_{ij} for each $j \neq i$. Then, we add to S the k closest vertices to i . We also add to S all the vertices $j \ni S$ for which $x(p_{ij} \setminus \{i\}) < (1 - x_i)$. If $(|S| + 1 - k)(1 - x_i) > \sum_{j \in S} x(p_{ij} \setminus \{i\})$ then we add the corresponding violated inequality and repeat this procedure for each vertex i . The complexity of the algorithm is obviously polynomial.

Inequalities (6.4) can be generalized as follows. Vertex i is replaced by a subset of vertices $A \subset V$ such that $G[A]$ is connected and $|A| \leq k$. Let S be a subset of vertices $S \subset V$ such that $S \cap A = \emptyset$ and $|S| + |A| \geq k + 1$. For each vertex $j \in S$, let p_{Aj} be a path connecting j to one vertex from A . The internal vertices of p_{Aj} can be assumed to be in \bar{A} . We use $p_{Aj} \setminus A$ to denote the path connecting j to the last vertex of p_{Aj} not belonging to A . It is then easy to see that the following inequalities are valid:

$$(|S| + |A| - k)(1 - x(A)) \leq \sum_{j \in S} x(p_{Aj} \setminus A). \quad (6.5)$$

Inequalities (6.5) can be written in a different way by making two observations. First, the paths p_{Aj} for $j \in S$ should be shortest paths from j to A (with respect to vertex weights $(x_v)_{v \in V}$). Second, we can assume that each path $p_{Aj} \setminus A$ is included in S since otherwise inequality (6.5) can be strengthened by deleting j from S , adding to S a vertex l from $p_{Aj} \setminus A$ and replacing p_{Aj} by the subpath of p_{Aj} connecting l to A . These two observations imply that we can assume that $\bigcup_{j \in S} p_{Aj}$ is in fact the disjoint union of some trees rooted at vertices in A . All vertices of each rooted tree (except the root) belong to S . Figure 6.2 illustrates the situation.

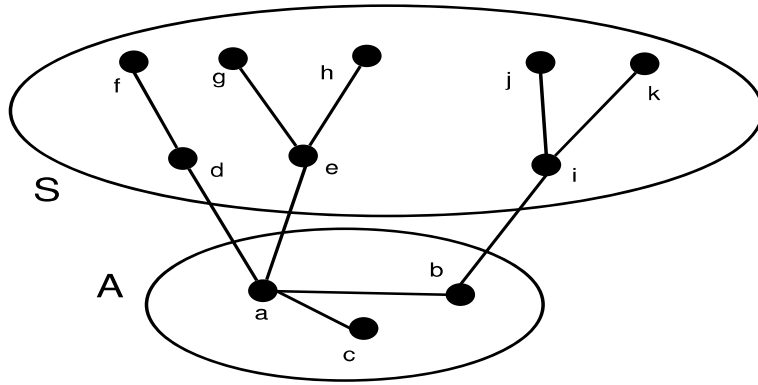


Figure 6.2 – Illustration of inequality (6.6)

Observe that in the sum $\sum_{j \in S} x(p_{Aj} \setminus A)$, the variable x_v related to vertex $v \in S$ appears as many times as the number of vertices in the subtree rooted at v . Let us use d'_v to denote this number. Then inequality (6.5) can be written as follows:

$$(|S| + |A| - k)(1 - x(A)) \leq \sum_{v \in S} d'_v x_v \quad (6.6)$$

where $A \subset V$ and $S \subset \bar{A}$ such that: $G[A]$ is connected, $|A| \leq k$, and $|S| + |A| \geq k + 1$. In the situation depicted by Figure 6.2, inequality (6.6) can be written as follows: $(|S| + |A| - k)(1 - x_a - x_b - x_c) \leq (x_f + x_g + x_h + x_j + x_k) + 2x_d + 3(x_e + x_i)$.

If we consider the special case where $S \subset N(A)$, then inequality (6.6) becomes

$$(|S| + |A| - k)(1 - x(A)) \leq x(S). \quad (6.7)$$

Since the exhibition of all cases where inequality (6.6) induces a facet requires some tedious proofs, we will only focus on a special case of inequality (6.7).

Proposition 6.3.19. [30] *Let $A = \{i\}$ and $S \subset N(A)$ be such that $|S| \geq k + 1$, and $G[S \cup \{j\}]$ does not contain a connected component of size greater than or equal to $k + 1$ for any $j \in S \cup A$. Then inequality (6.7) induces a facet of $\mathcal{S}_k(G)$.*

6.4 Approximations

In this section we present two approximation algorithms for the k -separator problem, one of them relying on an LP relaxation, another one which is of greedy type, with the same approximation guarantee. We also mention an inapproximability result.

The first approximation algorithm we give relies on the linear relaxation (LP1) of the integer program (IP1) introduced in Section 6.2.2. Notice that the separation of the inequalities (6.1) present in (IP1) is NP-hard even if all vertex-weights belong to $\{0,1\}$ when k is part of the input [116]. If k is constant, the separation is obviously easy. It is also known that a maximum-weight connected subgraph of size $k+1$ can be computed easily if the graph is a tree [116].

An LP-based approximation algorithm (Algorithm 1) is obtained by generalizing the basic approximation algorithm for the vertex cover problem. Hereafter, a connected subgraph $G[S]$ is said to be *large* if $|S| \geq k+1$.

Algorithm 1 LP-based Approximation Algorithm

- 1: **Input:** A vertex-weighted undirected graph $G = (V, E, w)$ and an integer k .
 - 2: **Output:** A k -separator S .
 - 3: Solve (LP1) and let x be an optimal solution of (LP1).
 - 4: Set $S := \emptyset$.
 - 5: **while** $G[V \setminus S]$ contains large connected components **do**
 - 6: Select $R \subset V \setminus S$ such that $|R| = k+1$ and $G[R]$ is connected.
 - 7: Select $v \in R$ such that x_v is maximum and set $S := S \cup \{v\}$.
 - 8: **end while**
-

Proposition 6.4.1. [30] *The LP-based approximation algorithm (Algorithm 1) is a $(k+1)$ -approximation algorithm.*

Observe that the presented Algorithm 1 is a polynomial time algorithm if we assume that k is bounded by a constant. This is necessary to guarantee that the size of (LP1) is polynomial. The primal-dual approach (see, e.g., [216]) leads also to a $(k+1)$ -approximation. In fact, the k -separator problem is a special-case of the hitting set problem where we want to hit large connected components.

If all vertex weights are equal to 1, then there is another simple $(k+1)$ -approximation algorithm (Algorithm 2).

Algorithm 2 Greedy Approximation Algorithm

- 1: **Input:** A graph $G = (V, E)$ and an integer k .
 - 2: **Output:** A k -separator S .
 - 3: Set $S := \emptyset$.
 - 4: **while** $G[V \setminus S]$ contains large connected components **do**
 - 5: Select $R \subset V \setminus S$ such that $|R| = k+1$ and $G[R]$ is connected.
 - 6: $S := S \cup R$.
 - 7: **end while**
-

Proposition 6.4.2. [30] *For the case when all the vertex weights are equal to 1, the greedy algorithm (Algorithm 2) is a $(k+1)$ -approximation algorithm for the k -separator problem.*

The greedy algorithm obviously has a polynomial time complexity even if k is part of the input.

Notice that we should not expect much better approximation algorithms since it is shown that the vertex cover (corresponding with $k=1$) cannot be approximated within a factor of 1.3606 [89] unless $P = NP$.

Finally, since computing a minimum weight k -separator is equivalent to maximizing the weight of the vertices that are not in the k -separator, one can also study the approximability of the maximization problem. Let us call this problem the maximum k -coseparator problem. For $k=2$, it is shown in [192] that this problem cannot be approximated within a factor of $|V|^{1/2-\epsilon}$ for any constant $\epsilon > 0$. We could extend their results for any k using the same reduction technique.

Proposition 6.4.3. [30] *Assuming that $P \neq NP$, the maximum k -coseparator problem cannot be approximated in polynomial time within a factor of $\left(\frac{|V|}{k}\right)^{1-\epsilon}$ for any constant $\epsilon > 0$.*

6.5 Some perspectives

In this chapter we introduced and investigated different aspects of the k -separator problem. Connections with the stable set problem led us to a linear relaxation generally stronger than the one stemming from the natural IP formulation. These connections also permitted us to identify families of graphs for which the problem can be solved efficiently. With respect to the polytope $\mathcal{S}_k(G)$ we established general properties that are satisfied by its facet defining inequalities, and we introduced many families of valid inequalities which may be facet defining (under some conditions). These can be used to strengthen relaxations of the problem. Preliminary computational results in this research direction are reported in Mohamed Sidi's PhD thesis [175].

A matter for future research work is the exhibition of some new classes of graphs for which the k -separator problem can be solved efficiently. It would be interesting to complete the polyhedral description when the problem can be solved in polynomial time (such as for trees). Also, the computational experiments carried out so far remain at a very preliminary status. Providing further insights on the quality of the approximations and the computation times, depending on the families of inequalities used, is also a substantial matter for future work. Another research direction is to look for approximation algorithms with better approximation guarantees.

Chapter 7

Orientation problems and connections to *max-cut*

In this chapter we present some of our results stemming from investigations on two graph orientation problems. Both problems involve the notion of *imbalance* of a node in a directed graph that is defined as the absolute difference between its outdegree and indegree. In one case, we wish to determine an orientation maximizing the minimum imbalance (taken over all the nodes). In the other, we wish to find an orientation maximizing the sum of the imbalances (over all the nodes). We investigate the complexity of the first problem, propose several MIP formulations and analyze some of their features. The second problem was shown to be equivalent to the maximum cardinality cut problem and this connection led us to study several new formulations for the latter.

The elements gathered in this chapter originate from joint works with Prof. Ben-Ameur and Antoine Glorieux; further details can be obtained in the publications [27–29, 113].

7.1 Introduction and notation

Let $G = (V, E)$ be an undirected simple graph with node set V and edge set E . We denote by δ_G the minimum degree of the vertices of G . An orientation Λ of G is an assignment of a direction to each undirected edge $uv \in E$, i.e. any function on E of the form $\Lambda(uv) \in \{\overrightarrow{uv}, \overrightarrow{vu}\}$, for all $\{uv\} \in E$, where \overrightarrow{uv} denotes the edge uv oriented from u to v . For each vertex v of G , we denote by $d_G(v)$ or $d(v)$ the degree of v in G and by $d_\Lambda^+(v)$ or $d^+(v)$ (resp. $d_\Lambda^-(v)$ or $d^-(v)$) the outdegree (resp. indegree) of v in G w.r.t. Λ . Given two integers a, b with $a < b$, let $\llbracket a, b \rrbracket$ stand for the set of integers $\{a, a + 1, \dots, b\}$.

Graph orientation is a well studied area in graph theory and combinatorial optimization. A large variety of constrained orientations as well as objective functions have been considered so far. Frank & Gyárfás [108] gave a simple characterization of the existence of an orientation such that the outdegree of every vertex is between a lower and an upper bound given for each vertex. Asahiro et al. in [12–14] proved the NP-hardness of the weighted version of the problem where the maximum outdegree is minimized, gave some inapproximability results, and studied similar problems for different classes of graphs. Chrobak & Eppstein proved that for every planar graph a 3-bounded outdegree orientation and a 5-bounded outdegree acyclic orientation can be constructed in linear time [62]. Other problems involving other criteria on the orientation have been studied such as acyclicity, diameter or connectivity [63, 67, 105, 179, 202]

For an orientation Λ of $G = (V, E)$ and a vertex v we call $|d_\Lambda^+(v) - d_\Lambda^-(v)|$ the *imbalance* of v in G w.r.t Λ and we call $\min_{v \in V} |d_\Lambda^+(v) - d_\Lambda^-(v)|$ the *imbalance* of Λ . Biedl et al. [43] studied the problem of finding an acyclic orientation of unweighted graphs minimizing the sum of the imbalances of all the vertices: they proved that it is solvable in polynomial time for graphs with maximum degree at most three but is NP-complete in general. Then Kára et al. closed the gap proving the NP-completeness for graphs with maximum degree four. Furthermore, they proved that the problem remains NP-complete for planar graphs with maximum degree four and for 5-regular graphs [138].

Landau's theorem [148] gives a condition for a sequence of non-negative integers to be the score sequence or outdegree sequence of some tournament (i.e. oriented complete graph) and later, Harary & Moser characterized score sequences of strongly connected tournaments [125]. Analogous results for the "imbalance sequences" of directed graphs are given by Mubayi et al. [176]. In 1962, Ford & Fulkerson characterized the mixed graphs (i.e. partially oriented graphs) whose orientation can be completed in an eulerian orientation, that is to say, an orientation for which the imbalance of each vertex equals zero [106]. Many other results related to orientation have been proposed. Some of them are reviewed in [21].

Let us denote by $\overline{\mathcal{O}}(G)$ the set of all the orientations of G , we consider the problem of finding an orientation with maximized imbalance:

$$(\text{MAXIM}) \quad \text{MAXIM}(G) = \max_{\Lambda \in \overline{\mathcal{O}}(G)} \min_{v \in V} |d_{\Lambda}^{+}(v) - d_{\Lambda}^{-}(v)|,$$

and we call $\text{MAXIM}(G)$ the value of MAXIM for G . The minimum degree δ_G of a graph G is a trivial upper bound for $\text{MAXIM}(G)$.

We tackled the problem MAXIM from different angles, characterizing the graphs for which this problem has an optimal objective value equal to zero, establishing complexity and inapproximability results, providing a polynomial time algorithm for the graphs that are cacti, proposing and evaluating different formulations of the problem [29]. After we summarize some of our results providing new insights on the complexity of MAXIM in Section 7.2, we present contributions related to linear programming formulations of the problem in Section 7.3. Then, we discuss connections with the maximum cardinality cut problem in Section 7.4, before we conclude with some perspectives in Section 7.5.

7.2 A sketch of new complexity results on MAXIM

Since the value of MAXIM for a graph is the minimum of the values of MAXIM on its connected components, from here on out, all the graphs we consider are assumed to be connected. For a graph G and a subgraph H of G , we will use the notations $V(H)$ and $E(H)$ to refer to the set of vertices of H and the set of edges of H , respectively.

7.2.1 Characterizing the graphs for which $\text{MAXIM}(G) = 0$

By investigating properties of orientations with zero imbalance, we could obtain characterizations of the graphs for which $\text{MAXIM}(G) = 0$. One of them involves a particular class of graphs: \mathcal{C}^{odd} , defined as follows.

Definition. A simple graph G is in \mathcal{C}^{odd} if there exists n odd cycles C_1, \dots, C_n (for some positive integer n), such that

- $\cup_{i=1}^n C_i = G$, and
 - $\left| V(\cup_{k=1}^{i-1} C_k) \cap V(C_i) \right| = 1$, for all $i \in \llbracket 2, n \rrbracket$.
- (7.1)

A simple characterization of the graphs for which $\text{MAXIM}(G) = 0$ is given by the next theorem.

Theorem 7.2.1. [29] *For any simple graph G , $\text{MAXIM}(G) = 0$ iff $G \in \mathcal{C}^{odd}$ iff G is eulerian with no even cycle.*

7.2.2 Complexity and (in)approximability

Concerning the complexity of MAXIM , we showed that the problem is NP-complete. More precisely, that answering if $\text{MAXIM}(G)$ equals 2 for a graph G such that $\delta_G = 2$ is NP-complete. The proof relies on a reduction of a particular variant of the satisfiability problem. This reduction was also used to obtain the inapproximability result stated in the next theorem.

Theorem 7.2.2. [29] *MAXIM is NP-complete and inapproximable within $\frac{1}{2} + \varepsilon$ where $\varepsilon \in \mathbb{R}_+ \setminus \{0\}$, unless $P = NP$.*

Given a node subset $A \subseteq V$, let $\delta(A)$ stand for the cut defined by A . In case the graph G is bipartite with node bipartition (V_1, V_2) , then the orientation that consists in assigning to each edge in E the orientation from its endpoint in V_1 to its endpoint in V_2 has an imbalance equal to δ_G , i.e. optimal. This simple case enabled us to obtain the following lower bound.

Theorem 7.2.3. [29] *For every graph G ,*

$$\text{MAXIM}(G) \geq \left\lceil \frac{\delta_G}{2} \right\rceil - 1.$$

Proof. Let (V_1, V_2) be a partition of V corresponding to a cut $C = \delta(V_1)$ such that $|\delta(\{v\}) \cap C| \geq \left\lceil \frac{d(v)}{2} \right\rceil$, for all $v \in V$. Such a cut can be easily found in polynomial time (for example, start with an arbitrary cut and iteratively switch from V_1 to V_2 or from V_2 to V_1 vertices not satisfying this condition). Then for any $i \in \{1, 2\}$, we orient the edges of the subgraph $G[V_i]$ of G that is induced by V_i . A new vertex v_0 is added and an edge between v_0 and each vertex with an odd degree in $G[V_i]$ if it is not Eulerian and we consider a decomposition of its edges into edge-disjoint cycles. Each of these cycles is oriented as directed cycle. Removing v_0 if necessary, the imbalance of each vertex in $G[V_i]$ is now in $\{-1, 0, 1\}$ which implies that, for all $v \in V$, we have $|d_\Lambda^+(v) - d_\Lambda^-(v)| \geq \left\lceil \frac{d(v)}{2} \right\rceil - 1$, hence, $\text{MAXIM}(G) \geq \left\lceil \frac{\delta_G}{2} \right\rceil - 1$. \square

The proof of the last result provides us with a polynomial time algorithm in order to find an orientation satisfying the inequality in the statement of Theorem 7.2.3. Note that, since $\text{MAXIM} \leq \delta_G$, this algorithm has approximation ratio at least $\frac{1}{2} - \frac{1}{8}$.

A special graph family for which we were able to design a polynomial time algorithm to compute $\text{MAXIM}(G)$ is that of cacti, which properly includes the class \mathcal{C}^{odd} introduced above. (Recall that a cactus is a connected simple graph for which every edge belongs to at most one elementary cycle.)

Theorem 7.2.4. [29] *Let G be a cactus graph, then $\text{MAXIM}(G)$ can be computed in polynomial time.*

The proof of Theorem 7.2.4 is algorithmic. It uses the fact that for cacti graphs $\text{MAXIM}(G) \leq 2$, thus implying $\text{MAXIM}(G) \in \{0, 1, 2\}$. Then, the result follows from a characterization of the cacti graphs for which $\text{MAXIM}(G) = 2$, together with Theorem 7.2.1.

7.3 MIP formulations for MAXIM

In this section, we introduce two MIP (Mixed-Integer Programming) formulations for the MAXIM problem. For our purposes, we shall consider the original graph G is directed (considering any arbitrary orientation), and let $B \in \{-1, 0, 1\}^{V \times E}$ stand for its incidence matrix, i.e. the column corresponding to the arc uv (or, equivalently, to the edge uv directed from node u to node v) has only nonzero entries in the rows corresponding to the nodes u and v : $B_{u,uv} = 1$ and $B_{v,uv} = -1$, respectively.

In order to describe an orientation of the graph G , we use orientation variables $x \in \{-1, 1\}^E$ interpreted as follows. For each edge $uv \in E$ which is originally directed from node u to node v : if $x_{uv} = 1$ then uv is directed from u to v (i.e. the orientation is the same as the original one) and is directed from v to u otherwise (i.e. the edge is “reversed” with respect to the original orientation). Then, taking the product of B with an orientation vector $x \in \{-1, 1\}^E$, we obtain $B_v x = d_x^+(v) - d_x^-(v)$, for all $v \in V$, where $d_x^+(v)$ (resp. $d_x^-(v)$) is the outdegree (resp. indegree) of $v \in V$ w.r.t. the

orientation described by x , and B_v denotes the row of the matrix B which corresponds to the node v . Hence the MAXIM problem can be expressed as the following nonlinear program.

$$\begin{cases} \max h \\ \text{s.t.} \\ h \leq |B_v x|, \quad \forall v \in V, \\ x \in \{-1, 1\}^E. \end{cases}$$

An equivalent linear formulation can be obtained as follows. Remove the absolute value of the inequality constraints by squaring: $h^2 \leq (B_v x)^2$, for all $v \in V$. Then, replace the x variables in $\{-1, 1\}$ by their binary ‘‘counterparts’’ and introduce variables representing the products of two variables. Developing, we obtain

$$\begin{aligned} (B_v(2x - \mathbf{1}))^2 &= d(v) + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} B_{v,uv} B_{v,wv} (4x_{uv}x_{wv} - 2x_{uv} - 2x_{wv} + 1), \\ &= d(v) + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} B_{v,uv} B_{v,wv} (4z_{uv,wv} - 2x_{uv} - 2x_{wv} + 1), \end{aligned}$$

where the variable $z_{uv,wv}$ represents the product $x_{uv}x_{wv}$. Furthermore, maximizing h is equivalent to maximizing its square root. Then substituting h^2 by h leads us to the next formulation where we added the constraints $z_{uv,wv} \leq x_{uv}$, $z_{uv,wv} \leq x_{wv}$ and $z_{uv,wv} \geq x_{uv} + x_{wv} - 1$ in order to ensure $z_{uv,wv} = x_{uv}x_{wv}$.

$$(MIP1) \begin{cases} \max h \\ \text{s.t.} \\ h \leq d(v) + 2 \sum_{\substack{uv, wv \in E \\ uv \neq wv}} B_{v,uv} B_{v,wv} (4z_{uv,wv} - 2x_{uv} - 2x_{wv} + 1), \quad \forall v \in V, \\ z_{uv,wv} \leq x_{uv}, \quad \forall uv, wv \in E, uv \neq wv, \\ z_{uv,wv} \leq x_{wv}, \quad \forall uv, wv \in E, uv \neq wv, \\ z_{uv,wv} \geq x_{uv} + x_{wv} - 1, \quad \forall uv, wv \in E, uv \neq wv, \\ x \in \{0, 1\}^E, z_{uv,wv} \geq 0, \forall uv, wv \in E, uv \neq wv, h \in \mathbb{R}. \end{cases}$$

Theorem 7.3.1. [29] For any graph G ,

$$MIP1(G) = \text{MAXIM}(G),$$

where $MIP1(G)$ is the square root optimal of the objective value of (MIP1) for G .

Unfortunately and in contrast with Theorem 7.3.1, the linear relaxation of (MIP1) is generally weak. Indeed, let the triplet (x^{lp}, z^{lp}, h^{lp}) be defined as follows: $x_{uv}^{lp} = \frac{1}{2}$, for all $uv \in E$; $z_{uv,wv}^{lp} = 0$ for all pairs of edges $uv, wv \in E$ that sharing no endpoint; $z_{uv,wv}^{lp} = \frac{1+B_{v,uv}B_{v,wv}}{4}$ for all pairs of edges $uv, wv \in E$ (i.e. all the pairs of edges sharing an endpoint), and $h^{lp} = \delta_G^2$. Then, the following holds.

Proposition 7.3.2. [29] (x^{lp}, z^{lp}, h^{lp}) is a feasible solution of the linear relaxation of (MIP1) with objective value δ_G^2 .

The last proposition shows that the optimal objective value of the linear relaxation of (MIP1) is at least the minimum degree of the input graph squared, which corresponds to the trivial upper bound of MAX2(G).

We now present a second formulation with a reduced number of variables and constraints. It involves orientation variables $x \in \{-1, 1\}^E$ and binary indicator variables y_k^v with $v \in V$, and $k \in \llbracket -d(v), d(v) \rrbracket$. They have the following interpretation: $y_k^v = 1$ if and only if $B_v x = d_x^+(v) - d_x^-(v) = k$, so that the following equations trivially hold.

$$\sum_{k \in \llbracket -d(v), d(v) \rrbracket} k y_k^v = B_v x, \quad \forall v \in V. \quad (7.2)$$

Given the interpretation for the variables y , among the ones of the form y_k^v , for some fixed node $v \in V$, exactly one of them has value 1. Thus,

$$\sum_{k \in \llbracket -d(v), d(v) \rrbracket} y_k^v = 1, \quad \forall v \in V. \quad (7.3)$$

Notice that for a vertex $v \in V$, the difference between its outdegree and indegree w.r.t. any orientation and its degree have the same parity. Thus instead of running k through $\llbracket -d(v), d(v) \rrbracket$, we can limit to $k \in \llbracket -d(v), d(v) \rrbracket$, s.t. $k \equiv d(v) \pmod{2}$, i.e. the only possible values of $d_x^+(v) - d_x^-(v)$ when $x \in \{-1, 1\}^E$. Then we can show the MAXIM problem may be formulated as the mixed integer program

$$(MIP2) \left\{ \begin{array}{l} \max h \\ \text{s.t.} \\ h \leq \sum_{\substack{k \in \llbracket -d(v), d(v) \rrbracket \\ k \equiv d(v) \pmod{2}}} \min(|k|, \delta_G) y_k^v, \quad \forall v \in V \\ \sum_{\substack{k \in \llbracket -d(v), d(v) \rrbracket \\ k \equiv d(v) \pmod{2}}} y_k^v = 1, \quad \forall v \in V \\ \sum_{\substack{k \in \llbracket -d(v), d(v) \rrbracket \\ k \equiv d(v) \pmod{2}}} k y_k^v = B_v x, \quad \forall v \in V \\ x \in \{-1, 1\}^E, y_k^v \in \{0, 1\}, (v, k) \in V \times \llbracket -d(v), d(v) \rrbracket, \text{ s.t. } k \equiv d(v) \pmod{2}, h \in \mathbb{R}. \end{array} \right.$$

Theorem 7.3.3. [29] For any graph G ,

$$MIP2(G) = \text{MAXIM}(G),$$

where $MIP2(G)$ is the optimal objective value of (MIP2) for G .

Let us consider the triplet (x^{lp}, y^{lp}, h^{lp}) where $x_{uv}^{lp} = 0$, for all $uv \in E$; $y_k^{v,lp} = 0$, for all $(v, k) \in V \times \llbracket -d(v) + 1, d(v) - 1 \rrbracket$, s.t. $k \equiv d(v) \pmod{2}$; and $y_{-d(v)}^{v,lp} = y_{d(v)}^{v,lp} = \frac{1}{2}$, for all $v \in V$. Observe that (x^{lp}, y^{lp}, h^{lp}) is a feasible solution of the linear relaxation of (MIP2) with objective value δ_G . So, the linear relaxation is generally weak, which led us to look for families of inequalities strengthening it. We restrict our presentation hereafter to two such families, see [29] for some others.

If l is a known lower bound for $\text{MAXIM}(G)$, then all variables y_k^v for which $|k| < l$ can be fixed to 0. To find such a lower bound, we may use a greedy algorithm to find a locally maximum cut and orient edges as described in the proof of Theorem 7.2.3. Let u be an upper bound for $\text{MAXIM}(G)$. We already know that $\text{MAXIM}(G) \leq \delta_G$, so we can assume that $u \leq \delta_G$. Consider the following inequality.

$$h \leq u - \sum_{v=1}^n \sum_{\substack{k \in \llbracket 0, u-1 \rrbracket \\ k \equiv d(v) \pmod{2}}} \lambda_k^v (y_k^v + y_{-k}^v), \quad \forall \lambda \in \Lambda_u, \quad (7.4)$$

where the vertices of G are numbered from 1 to $|V| = n$, and

$$\Lambda_u = \left\{ \lambda = (\lambda_k^v)_{(v,k) \in \llbracket 1, n \rrbracket \times \llbracket 0, u-1 \rrbracket} \in \mathbb{N}^{nu} \left| \begin{array}{l} \lambda_{k+1}^v \leq \lambda_k^v, \quad \forall (v, k) \in \llbracket 1, n \rrbracket \times \llbracket 0, u-2 \rrbracket \\ \sum_{v=1}^n \lambda_k^v = u - k, \quad \forall k \in \llbracket 0, u-1 \rrbracket \end{array} \right. \right\}.$$

Observe that the coefficients λ_k^v are nonnegative integer numbers and are non increasing in k . For each k , there exists only one v such that $\lambda_{k+1}^v = \lambda_k^v - 1$ while $\lambda_{k+1}^w = \lambda_k^w$ for any $w \neq v$. We can show that inequalities (7.4) are valid w.r.t. the convex hull of the feasible solutions of (MIP2) denoted by \mathcal{P}_2 in what follows.

Proposition 7.3.4. [29] Inequalities (7.4) are valid for \mathcal{P}_2 and can be separated in polynomial time.

In fact, the search for linear inequalities strengthening (MIP2) let us to investigate more closely the convex hull of polytopes related to extensions of assignment matrices. We were able to provide complete formulations for them; and it appeared that these descriptions essentially consist of the

inequalities (7.4) (and very few further constraints) and that they are all facet defining for these polytopes, see [28].

Let us now introduce another family of inequalities for (MIP2). Given any clique \mathcal{K} and any number $p \in \llbracket 1, |\mathcal{K}| \rrbracket$, we consider the two following inequalities.

$$\sum_{v \in \mathcal{K}} \sum_{k=0}^{\min(p-1, d(v))} (p-k) y_{d(v)-2k}^v \leq \frac{p(p+1)}{2}, \quad \forall p \in \llbracket 1, |\mathcal{K}| \rrbracket, \quad (7.5)$$

$$\sum_{v \in \mathcal{K}} \sum_{k=0}^{\min(p-1, d(v))} (p-k) y_{2k-d(v)}^v \leq \frac{p(p+1)}{2}, \quad \forall p \in \llbracket 1, |\mathcal{K}| \rrbracket. \quad (7.6)$$

Inequalities (7.5) can be seen as a generalization of the obvious inequalities $\sum_{v \in \mathcal{K}} y_{d(v)}^v \leq 1$, obtained when $p = 1$.

Proposition 7.3.5. [29] *Inequalities (7.5) and (7.6) are valid for \mathcal{P}_2 .*

Inequalities such as the ones introduced above have been tested within the framework of a cutting-plane algorithm followed by a branch-and-bound. Roughly summarizing results from our preliminary computational experiments, it appears that the formulation (MIP2) together with some inequalities added (including the two families presented above) leads to the best results, both in terms of the quality of the obtained bound for the linear relaxation, and the total computation times for its exact resolution. Also, due to important memory requirements for some instances (the graphs tested had between 10 and 3750 nodes and between 33 and 22500 edges), the formulation (MIP1) given as input to a MIP solver (CPLEX) did not permit to solve many of the largest ones, whereas they could be solved within 15 minutes using (MIP2) (strengthened with some inequalities). See [29] for details.

7.4 New formulations for unweighted *max-cut*

We now introduce new formulations for the maximum cardinality cut problem, i.e. the maximum cut problem (also called “max-cut”) for the case when all the edge weights are equal to 1: $w_e = 1$, for all $e \in E$. The optimal objective value is denoted by w^* . For our purposes, the original graph $G = (V, E)$ is assumed to be directed and we shall resume the notation used in the Section 7.3.

Consider the following MIP involving the set of variables y_k^v and x having the same interpretation as for (MIP2).

$$(MC1) \begin{cases} MC1(G) = \frac{1}{2} \max \sum_{v \in V} \sum_{k=-d_v}^{d_v} |k| y_k^v \\ \text{s.t. (7.2), (7.3),} \\ x \in [-1; 1]^E, \\ y_k^v \in \{0, 1\}, \quad v \in V, \quad k \in \llbracket -d_v, d_v \rrbracket. \end{cases}$$

The next result evidences the fact that looking for the maximum cardinality of a cut is equivalent to looking for an orientation maximizing the sum of the imbalances taken over all the nodes of the graph.

Proposition 7.4.1. [27] *The optimal objective value of (MC1) equals the maximum cardinality of a cut in the graph G : $MC1(G) = w^*$.*

Proof. First, note that it is equivalent to take $x \in [-1; 1]^E$ or $x \in \{-1; 0; 1\}^E$, given that B is totally unimodular and y is integral. implies the integrity of x . Observe that introducing sign variables $z \in \{-1, 1\}^V$ with the following interpretation: z_v is the sign of the only k for which $y_k^v \neq 0$, (MC1) is equivalent to the following

$$\begin{cases} \max \frac{1}{2} \sum_{v \in V} z_v \sum_{k=-d_v}^{d_v} k y_k^v \\ \text{s.t. (7.2), (7.3),} \\ y_k^v \in \{0, 1\}, \quad v \in V, \quad k \in \llbracket -d_v, d_v \rrbracket, \\ z \in \{-1, 1\}^V, \quad x \in [-1; 1]^E. \end{cases} \Leftrightarrow \begin{cases} \max \frac{1}{2} \sum_{v \in V} z_v B_v x = \frac{1}{2} z^T B x \\ \text{s.t.} \\ z \in \{-1, 1\}^V, \quad x \in \{-1, 0, 1\}^E. \end{cases}$$

$$\Leftrightarrow \begin{cases} \max \frac{1}{2} \sum_{uv \in E} x_{uv} (z_v - z_u) = \frac{1}{2} x^T B^T z \\ \text{s.t.} \\ z \in \{-1, 1\}^V, x \in \{-1, 1\}^E. \end{cases} \Leftrightarrow \begin{cases} \max \frac{1}{2} \sum_{uv \in E} |z_v - z_u| \\ \text{s.t.} \\ z \in \{-1, 1\}^V. \end{cases}$$

Noting that, for any $z \in \{-1, 1\}^V$, the quantity $\sum_{uv \in E} |z_v - z_u|$ equals twice the number of edges in $\delta(S)$, with $S = \{v \in V : z_v = 1\}$, the proposition follows. \square

Given a cut $\delta(S)$ with maximum cardinality, we can associate to it the following feasible solution of (MC1): the vector x corresponds to an orientation of all edges in the cut from S to $V \setminus S$, all other edges are not oriented ($x_{uv} = 0, \forall uv \in E \setminus \delta(S)$), and $y_k^v = 1$ if and only if k is equal to the outdegree minus the indegree w.r.t. the orientation given by x (ignoring edges which are not oriented). Then, for any edge $uv \in E$ which has the original orientation from u to v (i.e. the one given by the matrix B), the following equation holds: $x_{uv} = \sum_{k=1}^{d_u} y_k^u - \sum_{k=1}^{d_v} y_k^v$. From the latter we deduce (developing the expression $B_v x$):

$$\sum_{k=-d_v}^{d_v} k y_k^v = B_v x = \sum_{uv \in E} \left(\sum_{k=1}^{d_v} y_k^v - \sum_{k=1}^{d_u} y_k^u \right). \quad (7.7)$$

Note that, since each vertex v is incident to at least $\lceil \frac{d_v}{2} \rceil$ edges in a maximum cut, the variables y_k^v with $k \in \{1 - \lceil \frac{d_v}{2} \rceil, \dots, \lceil \frac{d_v}{2} \rceil - 1\}$ may be removed from the formulation (MC1), while Proposition 7.4.1 remains valid. It follows that, in place of (MIP1), we may consider a formulation involving variables of the form y_k^v only: replace equations (7.2) by (7.7). For each vertex v , we only consider y_k^v variables for $k \in I_v = I_v^- \cup I_v^+$ with $I_v^- = \left[-d_v, -\lceil \frac{d_v}{2} \rceil \right]$, $I_v^+ = \left[\lceil \frac{d_v}{2} \rceil, d_v \right]$.

$$(MC2) \begin{cases} MC2(G) = \frac{1}{2} \max \sum_{v \in V} \sum_{k \in I_v} |k| y_k^v \\ \text{s.t.} \\ \sum_{k \in I_v} y_k^v = 1, v \in V, \\ \sum_{k \in I_v} k y_k^v = \sum_{uv \in E} \left(\sum_{k \in I_v^+} y_k^v - \sum_{k \in I_u^+} y_k^u \right), v \in V, \\ y_k^v \in \{0, 1\}, v \in V, k \in I_v. \end{cases}$$

Formulation (MC2) may be strengthened using reformulation-linearization techniques. Let Y_{kl}^{uv} , with $(u, v) \in V^2$, $(k, l) \in I_u \times I_v$, denote a binary variable representing the product $y_k^u y_l^v$. Then, given that the variables y_k^v are binary and satisfy (7.3) we have $Y_{kl}^{uu} = 0$, for all $u \in V$, for all $k \neq l$. Considering then the product of the left side of (7.3) with itself we deduce $\sum_{k \in I_v} Y_{kk}^{vv} = 1$. Using other equations obtained by multiplying variables of the form y_k^v with equations (7.3) and others obtained from (7.7), we can deduce from (MC2) the following exact formulation for the maximum cardinality cut problem.

$$(MC3) \begin{cases} MC3(G) = \frac{1}{2} \max \sum_{v \in V} \sum_{k \in I_v} |k| Y_{kk}^{vv} \\ \text{s.t.} \\ \sum_{k \in I_v} Y_{kk}^{vv} = 1, v \in V, & (7.8a) \\ \sum_{k \in I_v} k Y_{kk}^{vv} = \sum_{uv \in E} \left(\sum_{k \in I_v^+} Y_{kk}^{vv} - \sum_{k \in I_u^+} Y_{kk}^{uu} \right), v \in V, & (7.8b) \\ Y_{kk}^{vv} = \sum_{l \in I_u} Y_{kl}^{vu}, u, v \in V, k \in I_v, & (7.8c) \\ (d_v - k) Y_{kk}^{vv} = \sum_{uv \in E} \sum_{l \in I_u^+} Y_{kl}^{vu}, v \in V, k \in I_v^+, & (7.8d) \\ -k Y_{kk}^{vv} = \sum_{uv \in E} \sum_{l \in I_u^+} Y_{kl}^{vu}, v \in V, k \in I_v^-, & (7.8e) \\ \sum_{l \in I_u} l Y_{kl}^{vu} = \sum_{uw \in E} \left(\sum_{l \in I_w^+} Y_{kl}^{vu} - \sum_{l \in I_u^+} Y_{kl}^{uu} \right), v \neq u \in V, k \in I_v, & (7.8f) \\ Y_{kl}^{vu} = Y_{lk}^{uv}, u, v \in V, (k, l) \in I_v \times I_u, & (7.8g) \\ Y_{kl}^{vu} \in \{0, 1\}, u, v \in V, (k, l) \in I_v \times I_u, & (7.8h) \end{cases}$$

Proposition 7.4.2. [27] *The optimal objective value of (MC3) equals the maximum cardinality of a cut in the graph G : $MC3(G) = w^*$.*

(MC3) was derived from (MC2) using lifting. Other formulations may be obtained using some other well-known lifting techniques such as the one of Lasserre, the Sherali-Adams technique or the lifting of Lovász-Schrijver [152, 204, 215]. An interesting feature w.r.t. the formulation (MC3) that we proved, is that a semidefinite relaxation derived from it provides an upper bound on w^* that is lower than or equal to the one given by Goemans and Williamson's relaxation [114]. Further formulations of the maximum cardinality cut problem, together with some numerical experiments are presented in [27].

7.5 Some perspectives

While computing the most imbalanced orientation of a graph is generally difficult, the problem turns out to be easy for cacti. It may be the same for other graph classes: It would be nice to get a characterization of them.

Two mixed integer linear programming formulations of MAXIM have been presented. It may be worth to pursue further investigations to determine efficient families of constraints to be added to their linear relaxations (different from the ones presented in [29]) in the framework of cutting plane algorithms. It would also be interesting to study the weighted version of the most imbalanced orientation problem.

Then, w.r.t. the maximum cardinality cut problem, starting from graph orientations, we have seen that it could be modeled in several new ways. In particular, one may get bounds that at least as strong as the ones given by the standard semidefinite bound of Goemans and Williamson [114]. We could identify cases when bounds stemming from semidefinite relaxations of our new formulation were strictly better than the latter. (This notably occurs for complete graphs with odd order, see [27].) It would be nice if one could characterize the graphs for which such a strict improvement holds.

Also, further efforts could be dedicated to determine if the results obtained for the maximum cardinality cut problem can be extended to the weighted case.

Bibliography

- [1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993. [9](#)
- [2] S. Akbari and S.J. Kirkland. On unimodular graphs. *Linear Algebra and its Applications*, 421:3–15, 2007. [65](#)
- [3] D. Alevras. Small min cut polyhedra. *Mathematics of Operations Research*, 24:35–49, 1999. [50](#)
- [4] K. Allemand, K. Fukuda, T.M. Liebling, and E. Steiner. A polynomial case of unconstrained zero-one quadratic optimization. *Mathematical Programming*, 91(1):49–52, 2001. [23](#)
- [5] R. Armstrong, A. Charnes, and F. Phillips. Page cuts for integer interval linear programming. *Discrete Applied Mathematics*, 1:1–14, 1979. [18](#)
- [6] M.F. Anjos and J. Neto. A class of spectral bounds for max k -cut. Technical report, Télécom SudParis, 2018. [33](#), [42](#), [43](#), [44](#)
- [7] M.F. Anjos and H. Wolkowicz. Strengthened semidefinite relaxations via a second lifting for the maximum cut problem. *Discrete Applied Mathematics*, 119:79–106, 2002. [39](#)
- [8] R. Anstee and M. Farber. Characterizations of totally balanced matrices. *J. Algorithms*, 5:215–230, 1984. [63](#), [68](#)
- [9] G. Argiroffo. New facets of the 2-dominating set polytope of trees. In *Proceedings of the 11th SIO, Argentina*, pages 30–34, 2013. [70](#)
- [10] M. Armbruster, C. Helmberg, M. Fügenschuh, and A. Martin. On the graph bisection cut polytope. *SIAM Journal on Discrete Mathematics*, 22(3):1073–1098, 2008. [50](#)
- [11] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340. [76](#)
- [12] Y. Asahiro, J. Jansson, E. Miyano, and H. Ono. Degree constrained graph orientation: maximum satisfaction and minimum violation. In *WAOA 2013*, volume 8447 of *LNCS*, pages 24–36, 2014. [85](#)
- [13] Y. Asahiro, E. Miyano, and H. Ono. Graph classes and the complexity of the graph orientation minimizing the maximum weighted outdegree. In *Proceedings of the fourteenth Computing: the Australasian Theory Symposium(CATS2008), Wollongong, NSW, Australia*, 2008. [85](#)
- [14] Y. Asahiro, E. Miyano, H. Ono, and K. Zenmyo. Approximation algorithms for the graph orientation minimizing the maximum weighted outdegree. In *Proc. of the 3rd Int. Conf. on Algorithmic Aspects in Information and Management (AAIM2007)*, volume 4508 of *LNCS*, pages 167–177, 2007. [85](#)
- [15] A. Astorino and M. Gaudioso. Polyhedral separability through successive lp. *Journal of Optimization Theory and Applications*, 112(2):265–293, 2002. [10](#)

-
- [16] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65:21–46, 1996. [25](#)
- [17] Mohar B. and S. Poljak. Eigenvalues and the max-cut problem. *Czechoslovak Mathematical Journal*, 40:343–352, 1990. [39](#), [41](#)
- [18] M. Baïou and F. Barahona. The dominating set polytope via facility location. In *Proceedings of ISCO 2014*, volume 8596 of LNCS, pages 38–49, 2014. [63](#)
- [19] E. Balas. Disjunctive programming: properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89:1–44, 1998. [65](#), [79](#)
- [20] E. Balas and C. de Souza. The vertex separator problem: a polyhedral investigation. *Mathematical Programming*, 103:563–608, 2005. [75](#)
- [21] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and applications*, chapter Orientations of graphs and digraphs, pages 417–472. Springer, 2nd edition edition, 2009. [86](#)
- [22] F. Barahona. On the computational complexity of ising spin glass models. *J. Phys. A: Math. Gen.*, 15:3241–3253, 1982. [49](#)
- [23] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36:493–513, 1988. [23](#)
- [24] F. Barahona, M. Grötschel, and A.R. Mahjoub. Facets of the bipartite subgraph polytope. *Mathematics of Operations Research*, 10:340–358, 1985. [50](#)
- [25] F. Barahona and A.R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36(2):157–173, 1986. [50](#), [57](#)
- [26] D.E. Bell. A cutting plane algorithm for integer programs with an easy proof of convergence. Technical Report WP-73-15, International Institute for Applied Systems Analysis, Laxenburg, 1973. [18](#)
- [27] W. Ben-Ameur, A. Glorieux, and J. Neto. From graph orientation to the unweighted maximum cut problem. In *Proceedings of COCOON 2016*, volume 9797 of LNCS, pages 370–384, 2016. [85](#), [90](#), [91](#), [92](#)
- [28] W. Ben-Ameur, A. Glorieux, and J. Neto. Complete formulations of polytopes related to extensions of assignment matrices. *Discrete Optimization*, 29:133–147, 2018. [85](#), [90](#)
- [29] W. Ben-Ameur, A. Glorieux, and J. Neto. On the most imbalanced orientation of a graph. *Journal of Combinatorial Optimization*, 36(2):637–669, 2018. [85](#), [86](#), [87](#), [88](#), [89](#), [90](#), [92](#)
- [30] W. Ben-Ameur, M.-A. Mohamed-Sid, and J. Neto. The k -separator problem: polyhedra, complexity and approximation results. *Journal of Combinatorial Optimization*, 29:276–307, 2015. [73](#), [76](#), [77](#), [78](#), [79](#), [80](#), [81](#), [82](#)
- [31] W. Ben-Ameur, M.-A. Mohamed-Sidi, and J. Neto. The k -separator problem. In *Proceedings of COCOON 2013*, volume 7936 of LNCS, pages 337–348. Springer, 2013. [73](#)
- [32] W. Ben-Ameur and J. Neto. A constraint generation algorithm for large scale linear programs using multiple-points separation. *Mathematical Programming*, 107(3):517–537, 2006. [11](#), [12](#), [13](#), [16](#), [17](#)
- [33] W. Ben-Ameur and J. Neto. Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks*, 49:3–17, 2007. [5](#), [6](#), [7](#), [8](#)

- [34] W. Ben-Ameur and J. Neto. Spectral bounds for the maximum cut problem. *Networks*, 52(1):8–13, 2008. [33](#), [35](#), [39](#), [43](#)
- [35] W. Ben-Ameur and J. Neto. Spectral bounds for unconstrained $(-1, 1)$ -quadratic optimization problems. *European Journal of Operational Research*, 207:15–24, 2010. [33](#), [34](#), [35](#), [36](#), [38](#), [40](#), [41](#), [44](#)
- [36] W. Ben-Ameur and J. Neto. A polynomial-time recursive algorithm for some unconstrained quadratic optimization problems. *Discrete Applied Mathematics*, 159:1689–1698, 2011. [23](#), [24](#), [30](#), [31](#), [33](#)
- [37] J.F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4:238–252, 1962. [4](#)
- [38] K.P. Bennett and O.L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992. [10](#)
- [39] C. Benzaken and P.L. Hammer. Linear separation of domination sets in graphs. *Annals of Discrete Math.*, 3:1–10, 1978. [63](#)
- [40] A.A. Bertossi and A. Gori. Total domination and irredundance in weighted interval graphs. *SIAM Journal on Discrete Mathematics*, 1(3):317–327, 1988. [62](#)
- [41] B. Betrò. An accelerated central cutting plane algorithm for linear semi-infinite programming. *Mathematical Programming*, 101(3):479–495, 2004. [5](#)
- [42] S. Bianchi, G. Nasini, and P. Tolomei. The set covering problem on circulant matrices: polynomial instances and the relation with the dominating set problem on webs. *Electronic Notes in Discrete Mathematics*, 36:1185–1192, 2010. [63](#)
- [43] T. Biedl, T. Chan, Y. Ganjali, M. Hajiaghayi, and D.R. Wood. Balanced vertex-orderings of graphs. *Discrete Applied Mathematics*, 48(1):27–48, 2005. [85](#)
- [44] A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109(1):55–68, 2007. [33](#)
- [45] H.L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. In *Proceedings of STOC'93*, pages 226–234, 1993. [76](#)
- [46] R. Boliac, K. Cameron, and V. Lozin. On computing the dissociation number and the induced matching number of bipartite graphs. *Ars Combin.*, 72:241–253, 2004. [74](#)
- [47] K.S. Booth. Dominating sets in chordal graphs. Technical Report CS-80-34, Univ. Waterloo, Waterloo, Ontario, Canada, 1980. [63](#)
- [48] K.S. Booth and J.H. Johnson. Dominating sets in chordal graphs. *SIAM J. Comput.*, 11:191–199, 1982. [62](#), [63](#)
- [49] R.B. Borie. Generation of polynomial-time algorithms for some optimization problems on tree-decomposable graphs. *Algorithmica*, 14:123–137, 1995. [76](#)
- [50] R.A. Bosch and J.A. Smith. Separating hyperplanes and the authorship of the disputed federalist papers. *American Mathematical Monthly*, 105(7):601–608, 1998. [10](#)
- [51] M. Bouchakour, T.M. Contenza, C.W. Lee, and A.R. Mahjoub. On the dominating set polytope. *European Journal of Combinatorics*, 29:652–661, 2008. [63](#)
- [52] M. Bouchakour and A.R. Mahjoub. One-node cutsets and the dominating set polytope. *Discrete Mathematics*, 165/166:101–123, 1997. [63](#), [66](#)

-
- [53] V.J. Bowman and G.L. Nemhauser. A finiteness proof for modified dantzig cuts in integer programming. *Naval Research Logistics Quarterly*, 17:309–313, 1970. 18
- [54] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 10, 13
- [55] K. Cameron and P. Hell. Independent packings in structured graphs. *Mathematical programming, Ser. B*, 105:201–213, 2006. 74
- [56] E. Çela, B. Klinz, and C. Meyer. Polynomially solvable cases of the constant rank unconstrained quadratic 0-1 programming problem. *Journal of Combinatorial Optimization*, 12:187–215, 2006. 23, 31
- [57] C.J. Chang. Labeling algorithms for domination problems in sun-free chordal graphs. *Discrete Applied Mathematics*, 22:21–34, 1988. 62
- [58] W. Cheney and A.A. Goldstein. Newton’s method for convex programming and chebyshev approximation. *Numerische Mathematik*, 1(5):253–268, 1959. 4
- [59] G.A. Cheston, G. Fricke, S.T. Hedetniemi, and D.P. Jacobs. On the computational complexity of upper fractional domination. *Discrete Applied Mathematics*, 27:195–207, 1990. 17
- [60] M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Inform. Comput.*, 206:1264–1275, 2008. 63
- [61] T. Christof and A. Loeb. Porta (polyhedron representation transformation algorithm). Software package, available for download at http://typo.zib.de/opt-long_projects/Software/Porta/50
- [62] M. Chrobak and D. Eppstein. Planar orientations with low out-degree and compaction of adjacency matrices. *Theoretical Computer Sciences*, 86:243–266, 1991. 85
- [63] F. Chung, M. Garey, and R. Tarjan. Strongly connected orientations of mixed multigraphs. *Networks*, 15:477–484, 1985. 85
- [64] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973. 4
- [65] V. Chvátal. On certain polytopes associated with graphs. *J. Combin. Theory (B)*, 18:138–154, 1975. 64
- [66] V. Chvatal. *Linear Programming*. W.H. Freeman, New York, 1983. 3
- [67] V. Chvátal and C. Thomassen. Distances in orientation of graphs. *Journal of Combinatorial Theory Series B*, 24:61–75, 1978. 85
- [68] E.J. Cockayne, R.M. Dowes, and S.T. Hedetniemi. Total domination in graphs. *Networks*, 10:211–219, 1980. 63
- [69] M. Conforti, G. Cornuéjols, and K. Truemper. From totally unimodular to balanced $0, \pm 1$ matrices: a family of integer polytopes. *Math. Oper. Res.*, 19:21–23, 1994. 68
- [70] M. Conforti, M.R. Rao, and A. Sassano. The equipartition polytope. i: Formulations, dimension and basic facets. *Mathematical Programming*, 49:49–70, 1990. 50, 51, 52
- [71] M. Conforti, M.R. Rao, and A. Sassano. The equipartition polytope. ii: Valid inequalities and facets. *Mathematical Programming*, 49:71–90, 1990. 50, 53, 55
- [72] M. Conforti, G. Rinaldi, and L.A. Wolsey. On the cut polyhedron. *Discrete Mathematics*, 277:279–285, 2004. 50

- [73] G. Cornuéjols. *Combinatorial Optimization: Packing and Covering*. CBMS-NSF Regional Conference Series in Applied Mathematics CBMS 74, SIAM, Philadelphia, PA, 2001. [63](#)
- [74] M. Couture, M. Barbeau, P. Bose, and E. Kranakis. Incremental construction of k -dominating sets in wireless sensor networks. In *Proceedings of the 10th International Conference On Principles Of Distributed Systems*, pages 202–214, 2006. [63](#)
- [75] P. Damaschke, H. Müller, and D. Kratsch. Domination in convex and chordal bipartite graphs. *Information Processing Letters*, 36:231–236, 1990. [63](#)
- [76] G.B. Dantzig, R. Fulkerson, and S.M. Johnson. Solution of a large-scale traveling salesman problem. *Operations Research*, 2:393–410, 1954. [4](#)
- [77] J.A. De Loera and E.D. Kim. Combinatorics and geometry on transportation polytopes: An update. *Contemporary Mathematics*, 625:37–76, 2014. [59](#)
- [78] C. De Simone. The cut polytope and the boolean quadric polytope. *Discrete Mathematics*, 79:71–75, 1990. [23](#)
- [79] C. De Simone. Lifting facets of the cut polytope. *Operations Research Letters*, 9:341–344, 1990. [50](#), [59](#)
- [80] C. De Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi. Exact ground states of ising spin glasses: New experimental results with a branch and cut algorithm. *Journal of Statistical Physics*, 80:487–496, 1995. [49](#)
- [81] C.C. de Souza. *The Graph Equipartition Problem: Optimal Solutions, Extensions and Applications*. PhD thesis, Faculté des Sciences Appliquées, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1993. [50](#)
- [82] C.C. de Souza and M. Laurent. Some new classes of facets for the equicut polytope. *Discrete and Applied Mathematics*, 62:167–191, 1995. [50](#)
- [83] M. Dell’Amico and J. Neto. On f -domination: polyhedral and algorithmic results. Technical report, University of Modena and Reggio Emilia, 2017. (Accepted for publication in *Mathematical Methods of Operations Research*). [61](#), [63](#), [66](#), [67](#)
- [84] M. Dell’Amico and J. Neto. On total f -domination: polyhedral and algorithmic results. Technical report, University of Modena and Reggio Emilia, 2017. (Accepted for publication in *Discrete Applied Mathematics*). [61](#), [62](#), [63](#), [64](#), [65](#)
- [85] C. Delorme and S. Poljak. Combinatorial properties and the complexity of a max-cut approximation. *European Journal of Combinatorics*, 14:313–333, 1993. [33](#), [39](#), [40](#)
- [86] C. Delorme and S. Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62:557–574, 1993. [33](#), [39](#), [40](#)
- [87] S.S. Dey and M. Molinaro. Theoretical challenges towards cutting-plane selection. *Math. Program., Ser. B*, 170(1):237–266, 2018. [18](#)
- [88] M.M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer, Berlin, 1997. [49](#), [50](#)
- [89] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162:439–485, 2005. [82](#)
- [90] M.P. Dobson, V. Leoni, and G. Nasini. Arbitrarily limited packings in trees. In *II MACI*, 2009. [63](#)

-
- [91] M.P. Dobson, V. Leoni, and G. Nasini. The multiple domination and limited packing problems in graphs. *Information Processing Letters*, 111:1108–1113, 2011. [62](#), [63](#)
- [92] R. Duh and M. Fürer. Approximation of k -set cover by semi-local optimization. In *Proceedings of the 29th ACM Symposium on Theory of Computing, STOC*, pages 256–264, 1997. [63](#)
- [93] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer, 1987. [25](#)
- [94] H. Edelsbrunner, J. O’Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM Journal on Computing*, 15:341–363, 1986. [25](#)
- [95] J. Edmonds. Systems of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards (B)*, 71:241–245, 1967. [31](#)
- [96] J. Elzinga and T.G. Moore. A central cutting plane algorithm for the convex programming problem. *Mathematical Programming*, 8:134–145, 1975. [5](#)
- [97] M. Farber. Domination, independent domination, and duality in strongly chordal graphs. *Discrete Applied Mathematics*, 7:115–130, 1984. [63](#), [67](#)
- [98] U. Feige and G. Schechtman. On the integrality ratio of semidefinite relaxations of MAX CUT. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 433–442, 2001. [39](#), [40](#)
- [99] C.E. Ferreira, A. Martin, C.C. de Souza, R. Weismantel, and L.A. Wolsey. Formulations and valid inequalities for the node capacitated graph partitioning problem. *Mathematical Programming*, 74:247–266, 1996. [50](#)
- [100] C.E. Ferreira, A. Martin, C.C. de Souza, R. Weismantel, and L.A. Wolsey. The node capacitated graph partitioning problem: a computational study. *Mathematical Programming*, 81:229–256, 1998. [50](#)
- [101] J.A. Ferrez, K. Fukuda, and T.M. Liebling. Solving the fixed rank convex quadratic maximization in binary variables by a parallel zonotope construction algorithm. *European Journal of Operational Research*, 166(1):35–50, 2005. [25](#), [26](#)
- [102] S. Fiorini, S. Massar, S. Pokutta, H.R. Tiwary, and R. de Wolf. Linear vs. semidefinite extended formulations: exponential separation and strong lower bound. In *STOC*, pages 95–106, 2012. [59](#)
- [103] I. Fisher, G. Gruber, F. Rendl, and R. Sotirov. Computational experience with a bundle approach for semidefinite cutting plane relaxations of max-cut and equipartition. *Math. Program.*, 105:451–469, 2006. [39](#)
- [104] K.T. Foerster. Approximating fault-tolerant domination in general graphs. In *Proceedings of the Tenth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 25–32, 2013. [63](#)
- [105] F. Fomin, M. Matamala, and I. Rapaport. Complexity of approximating the oriented diameter of chordal graphs. *Journal of Graph Theory*, 45(4):255–269, 2004. [85](#)
- [106] L.R. Ford and D.R. Fulkerson. *Flows in networks*. Princeton University Press, Princeton, NJ, 1962. [86](#)
- [107] A. Frangioni. About lagrangian methods in integer optimization. *Annals of Operations Research*, 139(1):163–193, 2005. [4](#)

- [108] A. Frank and A. Gyárfás. How to orient the edges of a graph ? *Colloquia Mathematica Societatis János Bolyai*, 18:353–364, 1976. [85](#)
- [109] A. Gagarin. Improved upper bounds for the k -tuple domination number. *Australasian journal of combinatorics*, 41:257–261, 2008. [63](#)
- [110] R. Gallant, G. Gunther, B. Hartnell, and D. Rall. Limited packings in graphs. *Electronic Notes in Discrete Mathematics*, 30:15–20, 2008. [61](#)
- [111] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, San Francisco, 1979. [4](#), [16](#), [17](#), [50](#), [62](#), [63](#)
- [112] A.M. Geoffrion. Generalized benders decomposition. *J. Optim. Theory and Appl.*, 10:237–260, 1972. [4](#)
- [113] A. Glorieux. *Optimiser les déséquilibres dans un graphe*. PhD thesis, Télécom SudParis, Evry, France, 2017. Numéro National de Thèse (NNT) : 2017TELE0011. [85](#)
- [114] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995. [33](#), [39](#), [92](#)
- [115] J.L. Goffin and J.P. Vial. Convex nondifferentiable optimization: a survey focussed on the analytic center cutting plane method. *Optimization Methods and Software*, 17:805–867, 2002. [5](#)
- [116] O. Goldschmidt and D. Hochbaum. K -edge subgraphs problems. *Discrete Applied Mathematics*, 74:159–169, 1997. [82](#)
- [117] R. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, The Rand Corporation (Santa Monica, CA), 1960. [18](#), [21](#)
- [118] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958. [4](#), [18](#)
- [119] M. Grötschel, M. Jünger, and G. Reinelt. Via minimization with pin preassignment and layer preference. *Zeitschrift für Angewandte Mathematik und Mechanik*, 69:393–399, 1989. [23](#), [49](#)
- [120] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. [66](#)
- [121] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, 1988. [3](#), [5](#), [11](#), [12](#), [13](#)
- [122] B. Grünbaum. *Convex Polytopes*. Springer, New York, 2 edition, 2003. [49](#)
- [123] P. Hammer. Some network flow problems solved with pseudo-boolean programming. *Operations Research*, 32:388–399, 1965. [23](#)
- [124] P. Hammer and A.A. Rubin. Some remarks on quadratic programming with 0-1 variables. *Revue Française d'Automatique, Informatique et de Recherche Opérationnelle*, 4(3):67–79, 1970. [33](#)
- [125] F. Harary, J. Krarup, and A. Schwenk. Graphs suppressible to an edge. *Canadian Mathematical Bulletin*, 15:201–204, 1971. [86](#)
- [126] Y. Harary and T.W. Haynes. Double domination in graphs. *Ars Combin.*, 55:201–213, 2000. [61](#), [63](#)

-
- [127] T.W. Haynes, S.T. Hedetniemi, and J.B. Slater. *Domination in Graphs: Advanced topics*. Marcel Dekker, 1998. [61](#), [62](#)
- [128] T.W. Haynes, S.T. Hedetniemi, and J.B. Slater. *Fundamentals of Domination in Graphs*. Marcel Dekker, 1998. [61](#), [62](#)
- [129] S. Hedetniemi, S. Hedetniemi, and R. Laskar. *Graph Theory with Applications to Algorithms and Computer Science*, chapter Domination in trees: models and algorithms, pages 423–442. John Wiley & Sons, New York, 2nd edition edition, 2009. [62](#)
- [130] S.T. Hedetniemi, R. Laskar, and J. Pfaff. A linear algorithm for finding a minimum dominating set in a cactus. *Discrete Applied Mathematics*, 13:287–292, 1986. [63](#)
- [131] C. Helmberg. A cutting plane algorithm for large scale semidefinite relaxations. Technical Report Report 01-26, Konrad-Zuse-Zentrum, Berlin, 2001. [39](#)
- [132] M.A. Henning. A survey of selected recent results on total domination in graphs. *Discrete Mathematics*, 309:32–63, 2009. [62](#)
- [133] M.A. Henning and A. Yeo. Strong transversals in hypergraphs and double total domination in graphs. *SIAM Journal on Discrete Mathematics*, 24(4):1336–1355, 2010. [63](#)
- [134] M.A. Henning and A. Yeo. *Total Domination in Graphs*. Springer Monographs in Mathematics, Springer, 2013. [61](#), [62](#), [63](#)
- [135] S.F. Hwang and G.J. Chang. The k -neighbor domination problem. *European Journal of Operational Research*, 52:373–377, 1991. [63](#)
- [136] M. Iri. On an extension of the maximum-flow minimum-cut theorem to multicommodity flows. *Journal of the Operations Research Society of Japan*, 13(3):129–135, 1971. [9](#)
- [137] V. Kaibel and S. Weltge. A short proof that the extension complexity of the correlation polytope grows exponentially. *Discrete and Computational Geometry*, 53(2):396–401, 2015. [59](#)
- [138] J. Kára, J. Kratochvíl, and D.R. Wood. On the complexity of the balanced vertex ordering problem. In *Proceedings of COCOON 2005*, volume 3595 of LNCS, pages 849–858, 2005. [85](#)
- [139] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum, New York, 1972. [23](#)
- [140] J.M. Keil. The complexity of domination problems in circle graphs. *Discrete Applied Mathematics*, 42:51–63, 1993. [62](#)
- [141] J.E. Kelley. The cutting-plane method for solving convex programs. *Journal of the SIAM*, 8:703–712, 1960. [4](#), [8](#)
- [142] L.G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979. [5](#)
- [143] T. Kikuno, N. Yoshida, and Y. Kakuda. A linear algorithm for the domination number of a series-parallel graph. *Discrete Applied Mathematics*, 5:299–311, 1983. [63](#)
- [144] R. Klasing and C. Laforest. Hardness results and approximation algorithms of k -tuple domination in graphs. *Information Processing Letters*, 89:75–83, 2004. [63](#)
- [145] V. Klee and C. Witzgall. Facets and vertices of transportation polytopes. In G.B. Dantzig and A.F. (eds) Veinott, editors, *Mathematics of the Decision Sciences, vol.1*, pages 257–282. Mathematics Research Laboratory, Boeing Scientific Research Laboratories, 1968. [59](#)

- [146] D. Kratsch and L. Stewart. Domination on cocomparability graphs. *SIAM Journal on Discrete Mathematics*, 6:400–417, 1993. [62](#)
- [147] D. Kratsch and L. Stewart. Total domination and transformation. *Information Processing Letters*, 63:167–170, 1997. [62](#)
- [148] H.G. Landau. On dominance relations and the structure of animal societies iii. the condition for a score structure. *The Bulletin of Mathematical Biophysics*, 15:143–148, 1953. [86](#)
- [149] T. Larsson and D. Yuan. An augmented lagrangian algorithm for large scale multicommodity routing. *Computational Optimization and Applications*, 27(2):187–215, 2004. [9](#)
- [150] R. Laskar and J. Pfaff. Domination and irredundance in split graphs. Technical Report 430, Dept. Mathematical Sciences, Clemson Univ., 1983. [62](#)
- [151] R. Laskar, J. Pfaff, S.M. Hedetniemi, and S.T. Hedetniemi. On the algorithmic complexity of total domination. *SIAM Journal Alg. Discr. Methods*, 5(3):420–425, 1984. [62](#)
- [152] M. Laurent. A comparison of the sherali-adams, lovász-schrijver and lasserre relaxations for 0-1 programming. *Mathematics of Operations Research*, 28(3):470–496, 2003. [92](#)
- [153] C. Lemaréchal. An extension of davidon methods to nondifferentiable problems. *Mathematical programming*, 3:95–109, 1975. [4](#)
- [154] A.Y. Levin. On an algorithm for the minimization of convex functions over convex functions. *Soviet Mathematical Doklady*, 6:245–282, 1965. [5](#)
- [155] C.S. Liao and G.J. Chang. Algorithmic aspect of k -tuple domination in graphs. *Taiwanese Journal of Mathematics*, 6(3):415–420, 2002. [63](#)
- [156] C.S. Liao and G.J. Chang. k -tuple domination in graphs. *Information Processing Letters*, 87:45–50, 2003. [62](#), [63](#)
- [157] V. Lozin and D. Rautenbach. Some results on graphs without long induced paths. *Inform. Process. Letters*, 88:167–171, 2003. [74](#), [76](#)
- [158] M.E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Oper. Res.*, 53:1007–1023, 2005. [37](#)
- [159] D. Luenberger. *Linear and nonlinear programming*. Addison Wesley, 1984. [8](#)
- [160] I.J. Lustig, R.E. Marsten, and D.F. Shanno. Interior point methods for linear programming: Computational state of the art. *ORSA Journal on Computing*, 6:1–14, 1994. [4](#)
- [161] T. Magnanti and R.T. Wong. Accelerating Benders decomposition: algorithmic enhancement and model selection criteria. *Operations Research*, 29:464–484, 1981. [4](#)
- [162] A.R. Mahjoub. Le polytope des absorbants dans une classe de graphe à seuil. *Annals of Discrete Mathematics*, 17:443–452, 1983. [63](#)
- [163] U. Malik, I.M. Jaimoukha, G.D. Halikias, and S.K. Gungah. On the gap between the quadratic integer programming problem and its semidefinite relaxation. *Mathematical Programming*, 107:505–515, 2006. [33](#), [39](#)
- [164] O.L. Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13(3):444–452, 1965. [10](#)
- [165] O.L. Mangasarian, R. Setiono, and W.W. Wolberg. Pattern recognition via linear programming: Theory and applications to medical diagnosis. in *Large-Scale Numerical Optimization*, Thomas F Coleman and Yuying Li, (Eds.), SIAM, Philadelphia, pages 22–30, 1990. [10](#)

-
- [166] O.L. Mangasarian, W.N. Street, and W.W. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43:570–577, 1995. [10](#)
- [167] T. Matsui and S. Tamura. Adjacency on combinatorial polyhedra. *Discrete Applied Mathematics*, 56:311–321, 1995. [57](#)
- [168] A.A. McRae. *Generalizing NP-Completeness Proofs for Bipartite and Chordal Graphs*. PhD thesis, Clemson Univ., 1994. [62](#)
- [169] M. Minoux. Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, 19:313–360, 1989. [9](#)
- [170] J.E. Mitchell. Cutting plane and column generation methods using interior point methods. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, 1997. [5](#)
- [171] J.E. Mitchell. Fixing variables and generating classical cutting planes when using an interior point branch and cut method to solve integer programming problems. *European Journal of Operational Research*, 97:139–148, 1997. [5](#)
- [172] J.E. Mitchell. Computational experience with an interior point cutting plane algorithm. *SIAM J. Optim.*, 10(4):1212–1227, 2000. [5](#)
- [173] J.E. Mitchell, K. Farwell, and D. Ramsden. Interior point methods for large-scale linear programming. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY, 2004. [5](#)
- [174] J.E. Mitchell, P. Pardalos, and M.G.C. Resende. Interior point methods for combinatorial optimization. In D.-Z. Du and P. (ed.) Pardalos, editors, *Handbook of combinatorial optimization*. Kluwer Academic Publishers, 1998. [5](#)
- [175] M.A. Mohamed Sidi. *Problème de k -séparateur*. PhD thesis, Télécom SudParis, Evry, France, 2014. Numéro National de Thèse (NNT) : 2014TELE0032. [83](#)
- [176] D. Mubayi, T.G. Will, and D.B. West. Realizing degree imbalances in directed graphs. *Discrete Mathematics*, 239(1-3):147–153, 2001. [86](#)
- [177] D. Naddef. The hirsch conjecture is true for (0,1)-polytopes. *Mathematical Programming*, 45:109–110, 1989. [57](#)
- [178] D. Naddef and W.R. Pulleyblank. Hamiltonicity and combinatorial polyhedra. *Journal of Combinatorial Theory, Series B*, 31:297–312, 1981. [57](#)
- [179] C. Nash-Williams. On orientations, connectivity and odd vertex pairings in finite graphs. *Canadian Journal of Mathematics*, 12:555–567, 1960. [85](#)
- [180] G.L. Nemhauser and L.E. Trotter. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6:48–61, 1974. [66](#)
- [181] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, 1988. [3](#), [18](#), [21](#)
- [182] J. Neto. From equipartition to uniform cut polytopes: Extended polyhedral results. *Discrete Mathematics*, 311:705–714, 2011. [49](#), [51](#), [52](#), [53](#), [54](#), [55](#)
- [183] J. Neto. A simple finite cutting plane algorithm for integer programs. *Operations Research Letters*, 40:578–580, 2012. [18](#)
- [184] J. Neto. On the polyhedral structure of uniform cut polytopes. *Discrete Applied Mathematics*, 175:62–70, 2014. [49](#), [51](#), [55](#), [56](#), [59](#)

- [185] J. Neto. On the diameter of cut polytopes. *Discrete Mathematics*, 339:1605–1612, 2016. [49](#), [51](#), [57](#), [58](#), [59](#)
- [186] J. Neto. A polyhedral view to generalized multiple domination and limited packing. In *Proceedings of ISCO 2018*, volume 10856 of *LNCS*, pages 352–363, 2018. [61](#), [68](#), [69](#), [70](#)
- [187] J. Neto. A polyhedral view to generalized multiple domination and limited packing. Technical report, Télécom SudParis, Evry, France, 2018. [61](#)
- [188] V. Nikiforov. Max k -cut and the smallest eigenvalue. *Linear Algebra and its Applications*, 504:462–467, 2016. [41](#), [42](#), [43](#)
- [189] K. Onaga and O. Kakusho. On feasibility conditions of multicommodity flows in networks. *IEEE Transactions on Circuit Theory*, 18(4):425–429, 1971. [9](#)
- [190] M. Oosten, J. Rutten, and F. Spikma. Disconnecting graphs by removing vertices: a polyhedral approach. *Statistica Neerlandica*, 61:35–60, 2007. [74](#), [80](#)
- [191] J.B. Orlin. A finitely converging cutting plane technique. *Operations Research Letters*, 4(1):1–3, 1985. [18](#)
- [192] Y. Orlovich, A. Dolgui, G. Finke, V. Gordon, and F. Werner. The complexity of dissociation set problems in graphs. *Discrete Applied Mathematics*, 159:1352–1366, 2011. [74](#), [75](#), [82](#)
- [193] M. Padberg. Perfect zero-one matrices. *Mathematical Programming*, 6:180–196, 1974. [64](#)
- [194] M. Padberg. *Linear Optimization and Extensions*. Springer Verlag, Heidelberg, 1995. [3](#)
- [195] C.H. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *J. Assoc. Comput. Mach.*, 29:285–309, 1982. [74](#)
- [196] J.C. Picard and H.D. Ratliff. Minimum cuts and related problems. *Networks*, 5:357–370, 1974. [23](#)
- [197] R.Y. Pinter. Optimal layer assignment for interconnect. *VLSI Comput. Syst.*, 1:123–137, 1984. [23](#)
- [198] S. Poljak and F. Rendl. Nonpolyhedral relaxations of graph bisection problems. *SIAM Journal on Optimization*, 5:467–487, 1995. [33](#), [39](#), [40](#)
- [199] S. Poljak and F. Rendl. Nonpolyhedral relaxations of graph bisection problems. *SIAM Journal on Optimization*, 5:467–487, 1995. [39](#)
- [200] D. Rautenbach and L. Volkmann. New bounds on the k -domination number and the k -tuple domination number. *Appl. Math. Lett.*, 20:98–102, 2007. [63](#)
- [201] F. Rendl, Rinaldi G., and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming (A)*, 121:307–335, 2010. [39](#), [45](#)
- [202] H. Robbins. A theorem on graphs with an application to a problem of traffic control. *American Mathematical Monthly*, 46:281–283, 1939. [85](#)
- [203] J.B. Rosen. Pattern separation by convex programming. *Journal of Mathematical Analysis and Applications*, 10:123–134, 1965. [10](#)
- [204] T. Rothvoß. The lasserre hierarchy in approximation algorithms. Lecture notes for the MAPSP Tutorial, 2013. preliminary version. [92](#)
- [205] A. Schrijver. *Linear and Integer Programming*. John Wiley, New York, 1986. [3](#)

-
- [206] A. Schrijver. *Combinatorial optimization: Polyhedra and Efficiency*. Springer, 2003. 49
- [207] W. Shang, P. Wan, F. Yao, and X. Hu. Algorithms for minimum m -connected k -tuple dominating set problem. *Theoretical Computer Science*, 381:241–247, 2007. 62
- [208] A. Shapiro and K. Scheinberg. Duality and optimality conditions. In H. Wolkowicz, R. Saigal, and Editors Vandenberghe, L., editors, *Handbook of Semidefinite Programming: Theory, Algorithms and Applications*, pages 67–110. Kluwer Academic Publishers, Boston, 2000. 38
- [209] D. Shmoys. Cut problems and their application to divide-and-conquer. In editor Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pages 192–235. PWS Publishing, 1997. 74
- [210] C. Stracke and L. Volkmann. A new domination conception. *J. Graph Theory*, 17:315–323, 1993. 62
- [211] J.A. Telle and A. Proskurowski. Algorithms for vertex partitioning problems on partial k -trees. *SIAM J. Discret. Math.*, 10:529–550, 1997. 76
- [212] P. Turán. On an extremal problem in graph theory. *Matematikai és Fizikai Lapok*, 48:436–452, 1941. 44
- [213] van Dam E.R. and R. Sotirov. New bounds for the max- k -cut and chromatic number of a graph. *Linear Algebra and its Applications*, 488:216–234, 2016. 41
- [214] A.F. Veinott. The supporting hyperplane method for unimodal programming. *Operations research*, 1:147–152, 1967. 5, 6
- [215] A. Wiegele. *Nonlinear Optimization Techniques Applied to Combinatorial Optimization Problems*. PhD thesis, Alpen-Adria-Universität, Klagenfurt, 2006. 92
- [216] D. Williamson. The primal-dual method for approximation algorithms. *Mathematical Programming, Ser B*, 91:447–478, 2002. 82
- [217] P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable convex functions. *Mathematical programming*, 3:145–173, 1975. 4
- [218] G. Xu, L. Kang, E. Shan, and H. Yan. Proof of a conjecture on k -tuple domination in graphs. *Appl. Math. Lett.*, 21:287–290, 2008. 63
- [219] M. Yannakakis. Node-deletion problems on bipartite graphs. *SIAM Journal on Computing*, 10:310–327, 1981. 74
- [220] Y. Ye. *Interior point algorithms. Theory and analysis*. John Wiley & Sons Inc., New-York, 1997. 5
- [221] S.M. Zhou. Invariants concerning f -domination in graphs. *Bulletin of the Malaysian Mathematical Sciences Society*, 37:1047–1056, 2014. 63
- [222] S.M. Zhou. On f -domination number of a graph. *Czechoslovak Mathematical Journal*, 46(3):489–499, 2014. 62
- [223] G.M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, New-York, 1995. 25, 49