Charles University in Prague
Faculty of Mathematics and Physics
Department of Applied Mathematics

Abstract Models of Optimization Problems

Doctoral Thesis

Petr Škovroň

Supervised by Prof. RNDr. Jiří Matoušek, DrSc.
I4 Discrete models and algorithms
Prague, 2007

Acknowledgements

During the time of preparation of this thesis, numerous people helped and influenced me. I would like to thank all of them. Those mentioned below deserve extra thanks.

First of all, I would like to express my gratitude to Jiří Matoušek for his supervision, questions, advice, ideas, and encouragement. You were much better advisor than I was a student.

Robert Babilon, Tomáš Gavenčiak, and Milan Hladík helped me with proofreading of the thesis and have suggested many improvements to the text.

I am indebted to my foreign coworkers Bernd Gärtner and Leo Rüst from ETH Zürich for exciting collaboration and for the joint publications.

All the professors and fellow students in our department have created an environment worth working in. Special thanks go to Petr Kolman, Jan Kratochvíl, Jaroslav Nešetřil, and Pavel Valtr for their enthusiasm and for interesting lectures, and to Martin Bálek, Honza Foniok, Martin Pergel, Diana Piguet, and Aleš Přívětivý for sharing the office. The secretaries Hana Čásenská, Nana Giorgadze, and Hana Polišenská helped me with all the official forms and documents.

The most stimulating environment I have experienced during the work on this thesis has been Emo Welzl's group at ETH Zürich. I thank all members of the group for making me feel at home during the two month stays spent with them.

Student's life is not only science. I thank Ilona Veselovská and Lucie Čistecká for teaching me to play flute. While I worked in industry, Pavel Kaňkovský, Karel Miko, and Honza Pavelka impressed me with their practical skills and competence.

Last but not least I would like to thank my family and my girlfriend Katka for their care and love.

Charles University in Prague
Faculty of Mathematics and Physics
Department of Applied Mathematics

Abstract Models of Optimization Problems

Doctoral Thesis

Petr Škovroň (xofon@kam.mff.cuni.cz)
Supervised by Prof. RNDr. Jiří Matoušek, DrSc.
Prague, 2007

Electronic version available at http://kam.mff.cuni.cz/~xofon/thesis/.

Hereby I declare that I have written all of the thesis on my own with the exceptions explicitly mentioned, and that I cited all used sources of information. I agree with public availability and lending of the thesis.

Petr Škovroň
2. 7. 2007, Prague

Table of Contents

Notation

$\mathbb{R}$	The set of real numbers
$\mathbb{N}$	The set of positive integers; i.e., $\{1, 2, 3, \ldots\}$
$a := b$	Define a symbol a to mean b
$\boldsymbol{v}$	A column vector
$\boldsymbol{u} \le \boldsymbol{v}$	Every component of a vector $\boldsymbol{u}$ is less or equal to the corresponding component of a vector $\boldsymbol{v}$
$\boldsymbol{u} \le_{\mathrm{L}} \boldsymbol{v}$	A vector $\boldsymbol{u}$ is lexicographically less or equal to a vector $\boldsymbol{v}$
$A^{\mathrm{T}}, \boldsymbol{v}^{\mathrm{T}}$	Transpose of a matrix A or a vector $\boldsymbol{v}$
$\binom{n}{k}$	The binomial coefficient for integers $0 \le k \le n$; otherwise 0
$\exp x$	The x-th power of the number $\mathrm{e} \approx 2.71828$
$\ln x$	Natural logarithm of a number x (with base e)
$[\varphi]$	Indicator for a formula φ; i.e., 1 if φ holds, 0 otherwise
$\lvert X \rvert$	Size of a set X
$X \mathbin{\dot\cup} Y$	Disjoint union of X and Y; i.e., $X \cup Y$ assuming $X \cap Y = \emptyset$
$X \setminus Y$	The set difference of sets X and Y; i.e., $\{x \in X : x \notin Y\}$
$\bigcap \mathcal{G}$	Intersection of sets in a family $\mathcal{G}$; i.e., $\bigcap_{G \in \mathcal{G}} G$
$\binom{H}{k}$	Family of all k-element subsets of a set H
$f\vert_M$	Restriction of a mapping f to a set M

Preface

In this thesis we study abstract models of optimization problems. Such research has both theoretical and algorithmic results. On the theoretical side it may reveal hidden analogies in different problems; on the algorithmic side it allows to employ known algorithms to new tasks.

The main model on which this thesis is based is the class called LP-type problems. This is an axiomatic framework invented by Sharir and Welzl [SW92] designed to generalize linear programming. Since its invention it became a well-established tool in the field of geometric optimization. As applications of LP-type problems we may name algorithms for minimum enclosing ball for a given set of points [SW92], parity games [BSV03], and determining the Tukey depth [Cha04]. Theoretical results concerning LP-type problems include proving subexponential running time bounds for certain randomized variants of simplex algorithm for linear programming [MSW96], and relating LP-type problems to Helly-type theorems [Ame94], [Ame96] and to lexicographic Helly-type theorems [Hal04].

Whenever we prove that an optimization task is an LP-type problem and we implement certain algorithmic primitives, we can immediately use several efficient algorithms: Sharir-Welzl algorithm [SW92], Clarkson's algorithms [Cla95], [GW96], its deterministic version [CM96], and an algorithm for finding an optimum solution satisfying all but at most k of the given constraints [Mat95].

An LP-type problem is given by a finite set H and a value $w(G)$ for every subset G of H. We interpret the elements of H as constraints, and $w(G)$ as the cost of a minimum solution that satisfies all constraints in G.

Other two abstract models explored in this thesis are concrete LP-type problems and violator spaces. The model of concrete LP-type problems is based on the following property of LP-type problems: in practice, we can often represent constraints by sets of solutions satisfying the particular constraint. A concrete LP-type problem is essentially an LP-type problem in which the constraints themselves are subsets of a linearly ordered set X (whose elements are solutions), and where $w(G)$ is the value of the minimum point in the intersection of the respective constraints.

Violator spaces were developed as a theoretical tool for a proof that every LP-type problem has a concrete representation. However, with discovering that a basis in a violator space can be found using Clarkson's algorithm [GMRŠ06], violator spaces turned out to be an algorithmically useful generalization of LP-type problems.

Another abstract model closely related to LP-type problems are unique sink orientations (USO) of cubes and grids. In this context, by a cube we mean a graph whose vertices correspond to vertices of an n-dimensional hypercube, and whose edges connect adjacent vertices. A unique sink orientation is such an orientation of edges that every subgraph corresponding to a face of the cube (including the whole graph) has a unique sink. In applications we wish to find the sink of the cube.

Unique sink orientations of cubes date back to the seventies when they appeared in connection to linear complementarity problems [SW78]. Current research includes among others [SW01], [SS04], [Sch04], and [GMR]. Algorithmic applications of the model include linear programming [GS06] and minimum enclosing balls of balls [Fis05]. An relation between unique sink orientations and violator spaces is described in [GMRŠ06]. In this thesis we do not study unique sink orientations in more depth.

Overview of the thesis. In Chapter 1 we motivate and define the models of optimization problems studied in this thesis: abstract and concrete LP-type problems and violator spaces. We introduce important concepts like bases and combinatorial dimension. We add a few notes concerning structure, usage, and relation between the models.

In Chapter 2 we suggest a modification of frameworks of concrete LP-type problems and violator spaces that allow to model sets with value $-\infty$ (i.e., unbounded sets). We prove a theorem on equivalence of the models, extending the analogous result achieved in the author's master thesis [Ško02].

The message of Chapter 3 is that removing degeneracy from optimization problems is hard. We prove the following theorem.

Theorem 3.2 (abridged). *For any positive integer Δ there exists a degenerate LP-type problem $\mathcal{P}$ such that to get a nondegenerate version of $\mathcal{P}$, the combinatorial dimension needs to increase at least by Δ.*

We show a representation of the constructed problem by a linear program with nonnegative variables. We conclude with an open question of how large increase of dimension we need to remove degeneracy from fixed-dimensional problems.

In Chapter 4 we exhibit some examples of cyclic violator spaces of small combinatorial dimension. These serve as counterexamples to several conjectures concerning cyclicity. The result of the chapter is the following proposition.

Proposition 4.2. *There exists a 2-dimensional cyclic violator space with 4 constraints. There exists a 2-dimensional basis-regular cyclic violator space. There exists a 3-dimensional nondegenerate basis-regular cyclic violator space.*

In Chapter 5 we present several bounds on the number of violator spaces of prescribed parameters:

Theorem 5.1 (abridged). *The number of violator spaces with n constraints is at most* $\exp(n2^{n-1}\ln 2)$.

The number of violator spaces with n constraints and with combinatorial dimension at most d is at most $\exp((\mathrm{e}/d)^d\, n^{d+1}\ln 2)$.

The number of basis-regular nondegenerate violator spaces with n constraints and with combinatorial dimension exactly d is between $\exp(\Omega(d^{-1/2}(\mathrm{e}/d)^d\,(n-d)^d))$ *and* $\exp(O(d(\mathrm{e}/d)^d\,n^d \ln n))$.

In Chapter 6 we prove that Clarkson's algorithm originally developed for linear programming in small dimension works in the context of violator spaces with minus infinity. We derive the following result on the running time of the algorithm.

Theorem 6.13 (abridged). *A basis of a violator space with* $-\infty$ *with* n *constraints and with combinatorial dimension* d *can be found in time* $O\big(t(dn + d^{O(d)})\big)$ *if certain elementary operations in the violator space are implemented and run in time* t.

In Chapter 7 we present another, generally known abstract model of optimization problems: oriented matroid programming. We apply the machinery developed in previous chapters to obtain Clarkson's algorithm for nondegenerate OM programming. To the best of our knowledge, this is the first algorithm for solving nondegenerate OM programs of fixed rank running in expected linear time.

Original results. The following results included in this thesis are published or accepted for publication in papers authored or coauthored by the author of this thesis. The text of the sections mentioned here originates in the corresponding papers. The results based on a joint work are used with a kind permission of the coauthors.

- The construction and the general proof of the result on increase of the dimension when removing degeneracy, and the representation of the example by a linear program; i.e., Sections 3.1–3.3 and 3.5 [MŠ07].
- The special case of the previous result for $\Delta = 2$; i.e., Section 3.4 [Ško06].
- Most of the bounds on the number of violator spaces; i.e., Chapter 5 [Ško05].
- Clarkson's algorithm for violator spaces without $-\infty$; i.e., a special case of Sections 6.1–6.6 [GMRŠ06].

Moreover, the following original results contained in this thesis are previously unpublished.

- The ideas of representing minus infinity in concrete LP-type problems and violator spaces; i.e., Sections 2.1 and 2.2.
- Small modifications of the theorem on equivalence of the models that allow for minus infinity; i.e., Section 2.3.
- Role of $-\infty$ when removing degeneracy and the results concerning degeneracy in 2-dimensional problems; i.e., Sections 3.6 and 3.7.
- Particular examples of interesting small-dimensional violator spaces; i.e., Chapter 4.
- Small modifications of Clarkson's algorithm for violator spaces that allow for minus infinity; i.e., Sections 6.1–6.7.
- The proposition that violator spaces are exactly the problems effectively solvable by Clarkson's algorithm; i.e., Section 6.8. This result has been achieved in collaboration with Bernd Gärtner.
- Relation between violator spaces and oriented matroid programs; i.e., Section 7.3.

Part I

Structure

Chapter 1

Motivation and basic definitions

In this chapter we introduce several models of optimization problems. We present some related terminology and provide several examples.

1.1. Linear programming

Since this thesis concerns abstract models of linear programming, we start with a brief description of linear programming itself. We can interpret linear programming algebraically or geometrically. We present both of these approaches.

To a reader interested in a more detailed introduction to linear programming, solving linear programming problems by simplex method, and a review of applications, we recommend Chvátal's textbook [Chv83].

The algebraic setting. Algebraically, the linear programming problem consists of finding a minimum of a linear function of n variables on a set of all solutions to a system of m linear inequalities. To be more specific, we are presented with a matrix A of size $m \times n$, a vector $\boldsymbol{b}$ of length m and a vector $\boldsymbol{c}$ of length n. The matrix A and the vector $\boldsymbol{b}$ represent the system of the inequalities and the vector $\boldsymbol{c}$ represents the function to be minimized. To avoid trivial cases, we assume that $\boldsymbol{c}$ is nonzero and that every row $\boldsymbol{a}_i$ of A is nonzero. The linear program requires us to find a vector $\boldsymbol{x}_0$ of length n satisfying $A\boldsymbol{x}_0 \leq \boldsymbol{b}$ (with the inequality holding in each component) for which the value of the linear function $\boldsymbol{x} \mapsto \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ is minimal.

The vector $\boldsymbol{c}$ is called the *cost vector* of the problem and the function $\boldsymbol{x} \mapsto \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ is called the *objective function.* The value of the objective function in a point $\boldsymbol{x} \in \mathbb{R}^n$ is called *cost* of $\boldsymbol{x}$. The rows of the system $A\boldsymbol{x} \leq \boldsymbol{b}$, that is, the inequalities

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \qquad \text{for } i = 1, \ldots, m,$$

are called the *constraints.* If a vector $\boldsymbol{x}$ satisfies $A\boldsymbol{x} \leq \boldsymbol{b}$, it is called a *feasible solution* to the linear program, otherwise is it called *infeasible.*

A problem of *maximizing* a linear function $\boldsymbol{x} \mapsto \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ may be expressed as a problem of minimizing the function $\boldsymbol{x} \mapsto (-\boldsymbol{c})^{\mathrm{T}}\boldsymbol{x}$; therefore we consider it to be a problem of linear programming as well. Generally, the linear programming problem should contain the information whether the objective function has to be minimized or maximized. We use the term *optimizing*, if we want to avoid explicitly stating whether minimizing or maximizing is demanded. Since this thesis is centered around LP-type problems, which are usually defined in such a way that the minimization is more natural to express, we prefer the minimizing formulation of problems.

In some applications of linear programming we may encounter constraints expressed as *equalities* instead of inequalities. We can describe this situation as a linear programming problem in the sense defined above if we replace every equality $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i$ by two inequalities $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i$ and $-\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq -b_i$.

The geometric setting. Consider the n-dimensional Euclidean space $\mathbb{R}^n$. A linear programming problem in the geometric setting is given by a nonzero vector $\boldsymbol{c} \in \mathbb{R}^n$ and a finite set $\mathcal{H}$ of closed halfspaces. We define the set $P \subseteq \mathbb{R}^n$ to be the polyhedron given as the intersection of the halfspaces in $\mathcal{H}$. The task is to find a point $\boldsymbol{x}_0 \in P$ for which the value of the function $\boldsymbol{x} \mapsto \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ is minimal.

We call the polyhedron P the *feasible region* of the problem and we call the points $\boldsymbol{x} \in P$ *feasible*. The halfspaces in $\mathcal{H}$ are called the *constraints*.

Geometrically we interpret the vector $\boldsymbol{c}$ as a direction; the value of $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ increases as the point $\boldsymbol{x}$ progresses in the direction of $\boldsymbol{c}$. Thus minimizing $\boldsymbol{x} \mapsto \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ corresponds to looking for a point $\boldsymbol{x}_0$ furthermost in the direction of $\boldsymbol{c}$.

We easily see that the algebraic and the geometric description of linear programming are equivalent: A closed halfspace can be given analytically by a linear inequality $\sum_j a_j x_j \leq b$ with suitable coefficients a_j and b, hence the polyhedron P can be given by a system of linear inequalities. We leave the cost vector unchanged. This expresses the geometric problem in the algebraic way.

We often give examples of 2-dimensional linear programs geometrically. We draw a picture that for each constraint displays the boundary line of the constraint halfplane and we mark on which side of the line the constraint is satisfied. As we mentioned above, we prefer minimization problems, therefore we interpret such a picture so that the y coordinate has to be minimized. However, the common practice is to draw an arrow denoting the maximizing direction. We keep this tradition by drawing an arrow pointing downwards.

Existence and uniqueness of the optimum solution. In some cases the optimum solution does not exist. There are several different reasons for such an outcome, and whenever it occurs, the output of a good linear program solver should contain an information which of the reasons applies.

The first, most obvious reason is that for the given linear program no feasible solution $\boldsymbol{x}$ exists at all. In this case we say that the linear program is *infeasible*.

Even if some feasible solutions do exist, it may happen that none of them is optimum. This happens when there are feasible solutions $\boldsymbol{x}$ with arbitrarily small values of $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ (or arbitrarily large if we are maximizing). In this case we say that the linear program is *unbounded*.

Furthermore, we often desire that the optimum solution is unique. However, even if an optimum solution exists, there may be many of them. We can see an example of a linear program where this happens in Figure 1.1. The set of all optimum

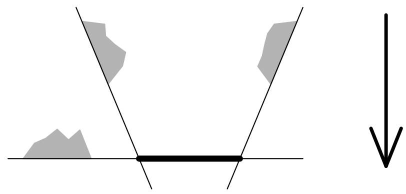

Figure 1.1. A linear program where optimum solution is not unique

solutions is marked by the bold line. The lack of the uniqueness is particularly unwelcome when expressing linear programming as an abstract LP-type problem. We therefore introduce a lexicographic variant of linear programming, in which the optimum solution is unique whenever it exists.

Put $\boldsymbol{c}^1 := \boldsymbol{c}$ (the cost vector) and fix vectors $\boldsymbol{c}^2, \dots, \boldsymbol{c}^n$ so that the set $B = \{\boldsymbol{c}^1, \dots, \boldsymbol{c}^n\}$ forms an orthogonal basis of $\mathbb{R}^n$. For a point $\boldsymbol{x} \in \mathbb{R}^n$, let $(x_1, \dots, x_n)$ denote the coordinates of $\boldsymbol{x}$ with respect to B. We define the *lexicographic ordering* $\leq_{\mathrm{L}}$ on $\mathbb{R}^n$ as follows. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we say that $\boldsymbol{x} <_{\mathrm{L}} \boldsymbol{y}$ if there exists some k such that $x_k < y_k$, and $x_i = y_i$ for all $i < k$. Furthermore, we say that $\boldsymbol{x} \leq_{\mathrm{L}} \boldsymbol{y}$ if $\boldsymbol{x} <_{\mathrm{L}} \boldsymbol{y}$ or $\boldsymbol{x} = \boldsymbol{y}$.

In the lexicographic variant of linear programming we look for the feasible point for which the coordinate vector is lexicographically minimal, instead of minimizing the first coordinate only. Now if the minimum is attained, it is unique, since every point has a distinct coordinate vector.

Note that the existence of an optimum solution in standard linear programming does not imply that the optimum solution exists in the lexicographic variant. A simple example in $\mathbb{R}^2$ is the problem of minimizing x_1, i.e., $\boldsymbol{c} = (1, 0)^{\mathrm{T}}$, with a single constraint $x_1 \geq 0$. The optimum value 0 of the objective function is attained for instance in $\boldsymbol{x}_0 = (0, 0)^{\mathrm{T}}$. However, the lexicographic minimum does not exist.

On the other hand, if we consider the linear programs including special constraints $x_i \geq 0$ for all $i = 1, \dots, n$, it is possible to prove that the unique lexicographic minimum exists in every feasible problem. We refer to the problems of this type as *lexicographic linear programming in the positive orthant.*

1.2. Abstract LP-type problems

In this section we introduce abstract LP-type problems and some basic terminology. Although we try to give some motivation, we proceed slightly more formally that in the previous section, notably we state precise definitions of the presented concepts. However, we do not demonstrate in detail how to represent a general linear programming problem in the form of an abstract LP-type problem.

Abstract LP-type problems were introduced by Sharir and Welzl [SW92] as a framework for a certain randomized algorithm for solving linear programs. We refer to this algorithm as the Sharir-Welzl algorithm.

Originally, the name LP-type problems was used. We add the prefix 'abstract' to emphasize the distinction from a different model called concrete LP-type problems (see Section 1.3).

An abstract LP-type problem is a structure capturing some combinatorial properties of a linear programming problem, or more generally, of an instance of some classes of problems of looking for a minimum solution satisfying some constraints. An instance of abstract LP-type problem consists of two basic ingredients:

- an abstract finite set H representing the constraints of the problem;
- a function w that for a given set of constraints $G \subseteq H$ gives the cost of the optimum solution satisfying the constraints in G.

We call the function w the *weight function* and for a set $G \subseteq H$ we call $w(G)$ the *value* of G. If $h_1, \dots, h_k$ are some elements of H, we write $w(h_1, \dots, h_k)$ instead of $w(\{h_1, \dots, h_k\})$. We assume that the values $w(G)$ are elements of a set W linearly ordered by a relation $\leq$. If the structure of W is not important, we often omit its specification.

To cope with infeasible and unbounded problems, the set W may contain a special minimum element $-\infty$ and a maximum element $+\infty$. If the problem represented by G is unbounded, we set $w(G) = -\infty$, and for an infeasible problem we set $w(G) = +\infty$.

To complete the formal definition of LP-type problems we postulate two properties of the function w.

Consider the optimum solution with respect to constraints G with the cost $w(G)$. Now remove some of the constraints, so that only the constraints of F remain. The original solution still remains a solution to the new problem, and maybe even some new better solution is possible. Thus we have $w(F) \leq w(G)$ whenever $F \subseteq G$. This property of w is called *monotonicity*.

Let us assume that for every feasible subproblem the optimum solution exists and is unique. Let $F \subseteq G \subseteq H$ be sets of constraints satisfying $w(F) = w(G)$; then the optimum solution $\boldsymbol{x}$ with respect to F is actually the same as the optimum solution with respect to G. Now consider an additional constraint $h \in H$ such that $w(F) = w(F \cup \{h\})$. The vector $\boldsymbol{x}$ is optimum for $F \cup \{h\}$ as well, hence $\boldsymbol{x}$ satisfies the constraint h. Therefore $\boldsymbol{x}$ is optimum for $G \cup \{h\}$ too. In other words, $w(F) = w(F \cup \{h\}) = w(G)$ implies $w(G) = w(G \cup \{h\})$. This property is called *locality*.

Actually, if $w(F) = w(G) = -\infty$, i.e., when the problems are unbounded, we cannot claim that the optimum solution with respect to F is the same as the one with respect to G, since they do not exist at all. In this case the argument does not work, and the locality can fail actually.

To summarize, here is the formal definition.

Definition 1.1. *Let W be a fixed set with a linear ordering $\leq$. An* (abstract) LP-type problem *is a pair (H, w), where H is a finite set and $w : 2^H \to W$ is a mapping satisfying the following conditions:*

- *$w(F) \leq w(G)$ for all $F \subseteq G \subseteq H$* (monotonicity)*;*
- *if $w(F) = w(F \cup \{h\}) = w(G) \neq -\infty$ for some $F \subseteq G \subseteq H$ and $h \in H$, then also $w(G) = w(G \cup \{h\})$* (locality).

For example, consider the linear program in Figure 1.2 with the set of constraints $H = \{a, b, c, d\}$. As agreed, the y coordinate has to be minimized. The values of

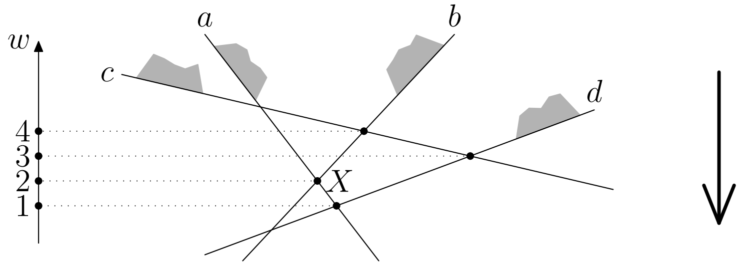

Figure 1.2. A linear program interpreted as an abstract LP-type problem

the objective function are marked on the scale on the left side. To determine for example the value of $w(a,b)$, we note that the optimum solution satisfying the constraints a, b is the point X, whose cost is 2. Hence we have $w(a,b) = 2$.

The values of all subsets of H (i.e., the main part of the specification of the LP-type problem) follow:

$$\begin{array}{lllll} w(\emptyset) = -\infty, & w(a) = -\infty, & w(b) = -\infty, & w(c) = -\infty, & w(d) = -\infty, \\ w(a,b) = 2, & w(a,c) = -\infty, & w(a,d) = 1, & w(b,c) = 4, & w(b,d) = -\infty, \\ w(c,d) = 3, & w(a,b,c) = 4, & w(a,b,d) = 2, & w(a,c,d) = 3, & w(b,c,d) = 4, \\ & & w(a,b,c,d) = 4. & & \end{array}$$

Bases, combinatorial dimension, and nondegeneracy. The optimum solution with respect to $G \subseteq H$ is often actually determined by much fewer constraints than by the whole G. It is useful to introduce a special terminology for this.

Definition 1.2. *Consider an abstract LP-type problem (H, w). A set $B \subseteq H$ is called a* basis *in (H, w) if for all proper subsets $F \subset B$ we have $w(F) < w(B)$.*

For $G \subseteq H$, we say that a basis B is a basis of G *in (H, w) if $B \subseteq G$ and $w(B) = w(G)$.*

In other words, B is a basis if it contains no elements that are redundant for determining the optimum solution. For instance, $G = \{a, b, d\}$ is not a basis in the example problem, since $F = \{a, b\} \subset G$ has the same value, namely $w(F) = w(G) = 2$.

The terminology 'a basis of G' suggests that every set has a basis. This is indeed the case, since we can take any inclusion-minimal subset B of G with $w(B) = w(G)$.

In the example problem, the bases are $\emptyset$, $\{a, d\}$, $\{a, b\}$, $\{c, d\}$ and $\{b, c\}$ (ordered by the value).

To measure the complexity of the structure of the problem, we introduce the combinatorial dimension.

Definition 1.3. *The* combinatorial dimension *of an abstract LP-type problem (H, w), denoted by* $\dim(H, w)$, *is the maximum cardinality of a basis.*

To illustrate the relevance of the definition of combinatorial dimension we remark that LP-type problems representing linear programs in $\mathbb{R}^d$ have combinatorial dimension at most d. For instance, in the example LP-type problem above we have $d = 2$.

The basis of a set G does not need to be unique. However, for some algorithms the uniqueness of a basis is desirable. This leads to the notion of nondegeneracy. In a sense, it is analogous to the concept of general position in combinatorial geometry.

Definition 1.4. *We say that an abstract LP-type problem* (H, w) *is* nondegenerate *if we have* $w(B) \neq w(B')$ *for every two distinct bases* B, B'*. If the problem is not nondegenerate, we say that it is* degenerate.

Consequently, in a nondegenerate abstract LP-type problem every set $G \subseteq H$ has exactly one basis.

We study the problem of removing degeneracy in Chapter 3.

Optimization aspect of the LP-type problem. We interpret the abstract LP-type problem (H, w) as a problem of finding a basis B of H. Actually, we are interested in $w(H)$; this seems to be an easy task, since the function w is included in the specification of the LP-type problem. However, if the function $G \mapsto w(G)$ is given as a subroutine, its running time likely depends on $|G|$ or it even requires that $|G| \leq \dim(H, w)$. Since we often have $\dim(H, w) \ll |H|$, we expect that $w(B)$ is much simpler to evaluate than $w(H)$.

Violation. In fact, most of the known algorithms related to LP-type problems do not need to evaluate the exact values of w. A knowledge of a combinatorial structure of the problem is often sufficient. This knowledge is usually provided by the following two subroutines:

- For a basis B and a constraint $h \in H$ determine whether $w(B) < w(B \cup \{h\})$ *(violation test)*.
- For a basis B and $h \in H$ with $w(B) < w(B \cup \{h\})$ find a basis of $B \cup \{h\}$ *(basis computation)*.

The first subroutine is closely related to the following concept. For a set of constraints $G \subseteq H$ and an additional constraint $h \in H$, we are interested in whether the value of G increases by adding h; in other words whether $w(G) < w(G \cup \{h\})$ or $w(G) = w(G \cup \{h\})$ (note that $w(G) \leq w(G \cup \{h\})$ by monotonicity). In the former case the optimum solution with respect to G does not satisfy the constraint h; therefore we say that the constraint h *violates* the set G. For a set of constraints $G \subseteq H$, we define

$$\mathsf{V}(G) := \big\{h \in H : w(G) < w(G \cup \{h\})\big\}$$

to be the set of all constraints violating G. We regard V as a mapping $2^H \to 2^H$ and we call it the *violator mapping* of the LP-type problem.

To demonstrate the power of the language of V, we prove a proposition that allows us to remove the reference to the weight function in some statements about LP-type problems.

Proposition 1.5. *Let* (H, w) *be an LP-type problem with a violator mapping* V*. Let* F, G *be sets of constraints with* $F \subseteq G \subseteq H$ *and* $w(F) \neq -\infty$*. Then* $w(F) = w(G)$ *if and only if* $G \cap \mathsf{V}(F) = \emptyset$.

Proof. If $w(F) = w(G)$ and $g \in G$, we have $w(F) \le w(F \cup \{g\}) \le w(G) = w(F)$, thus $w(F) = w(F \cup \{g\})$, which means that $g \notin \mathsf{V}(F)$. Therefore $G \cap \mathsf{V}(F) = \emptyset$.

Conversely, if $G \cap \mathsf{V}(F) = \emptyset$, we have $w(F) = w(F \cup \{g\})$ for every $g \in G$. Let the elements of $G \setminus F$ be $g_1, \ldots, g_k$. By locality successively applied to sets F, $F \cup \{g_1\}$, $F \cup \{g_1, g_2\}$, $\ldots$, $F \cup \{g_1, \ldots, g_k\} = G$, we obtain $w(F) = w(G)$. ∎

Basis-regularity. The analysis of the Sharir-Welzl algorithm for solving LP-type problems [MSW96] identifies a class of LP-type problems for which the running time is subexponential. The class is characterized by the following property.

Definition 1.6. *We say that an LP-type problem (H, w) of combinatorial dimension d is* basis-regular *if for every $G \subseteq H$ with d or more elements, each basis B of G has exactly d elements.*

To see how we can profit from basis-regularity, note how we can implement the basis computation. Consider a basis-regular LP-type problem of combinatorial dimension d. If G is a set with $d+1$ elements, we know that a basis B of G has d elements; that is, exactly one element of G is missing in B. Therefore, if a subroutine for the violation test is provided, we can implement the basis computation for a set G with $|G| = d+1$, using at most $d+1$ calls to the violation test.

On the other hand, if the basis-regularity of the problem is not guaranteed, virtually any proper subset of G can be the basis, so in an extreme case we may need to check all $2^{|G|} - 1 = 2^{d+1} - 1$ proper subsets of G. A real-life example of an LP-type problem of combinatorial dimension d that is not basis-regular is the problem of finding a minimum ball enclosing a given set of points in $\mathbb{R}^{d-1}$; an analysis of the running time of LP-type based algorithms for this problem has been provided by Gärtner [Gär95].

Need for the special treatment of $-\infty$. Note that for linear programs interpreted as LP-type problems, the condition of locality indeed may fail if we omit the assumption $w(G) \neq -\infty$. In the example problem in Figure 1.2 putting $F := \emptyset$, $G := \{a\}$, $h := b$, we have $w(F) = w(\emptyset) = -\infty$, $w(G) = w(a) = -\infty$, and $w(F \cup \{h\}) = w(b) = -\infty$, but $w(G \cup \{h\}) = w(a, b) = 2$.

If the function w attains values $w(G) \neq -\infty$ for all $G \subseteq H$, we call (H, w) an *abstract LP-type problem without* $-\infty$. In this setting we have the axiom of locality in a simplified form without the assumption $w(G) \neq -\infty$.

In applications, some important classes of optimization problems lead to abstract LP-type problems without $-\infty$. These include problem of finding the minimum ball enclosing a given set of points, and lexicographic linear programming in the positive orthant.

The role of the structure of W. The set H is finite, hence we have only a finite number of distinct values of $w(G)$. Since all linear orderings of a finite set are isomorphic, the structure of the set W is not important. Without loss of generality we can even assume that $W = \mathbb{R} \cup \{-\infty, +\infty\}$.

On the other hand, in many applications some choice of W based on the setting of the problem is more appropriate than $\mathbb{R}$. Therefore we stick to the more general definition. It is even possible to relax the assumption of linearity of the ordering $\leq$; for details see the definition of LP-type problems as given by Fischer [Fis05].

1.3. Concrete LP-type problems

In this section we continue the presentation of basic models of optimization problems by introducing concrete LP-type problems.

In geometric examples of abstract LP-type problems, we often have some set X representing the feasible solutions, and the constraints actually are subsets of X. Moreover, the elements of X are ordered according to their cost, and the value $w(G)$ represents the optimum solution in the intersection of the constraints in G.

This view of LP-type problems is formalized in the following definition.

Definition 1.7. *Let X be a set linearly ordered by a relation $\preceq$ and let $\mathcal{H}$ be a finite multiset whose elements (called* constraints*) are subsets of X. We say that a triple $(X, \preceq, \mathcal{H})$ is a* concrete LP-type problem (without $-\infty$)*, if whenever the intersection $\bigcap \mathcal{G} = \bigcap_{G \in \mathcal{G}} G$ is nonempty for some $\mathcal{G} \subseteq \mathcal{H}$, then it has a minimum element with respect to $\preceq$. In the case that $\mathcal{G} = \emptyset$, by $\bigcap \mathcal{G}$ we mean X.*

The definition allows $\mathcal{H}$ to be a multiset, so we may include a single constraint $A \subseteq X$ several times. For example, in an instance of linear programming in the algebraic setting, some inequalities can describe identical halfspaces, and setting $\mathcal{H}$ to be a multiset represents this in a natural way.

In this section we assume that the problem and all of its subproblems are bounded; i.e., the presented definition introduces the problems without $-\infty$. We study the unbounded case in Chapter 2.

On the other hand, representing the infeasible problems makes no special difficulty. We allow the intersection of constraints to be empty. For the subsequent discussion, we set $\min(\emptyset) := +\infty$ for this case, and we put $x < +\infty$ for every $x \in X$.

We define bases and combinatorial dimension analogously as for abstract LP-type problems.

Definition 1.8. *Consider a concrete LP-type problem $(X, \preceq, \mathcal{H})$. We say that a multiset $\mathcal{B} \subseteq \mathcal{H}$ is a* basis *in $(X, \preceq, \mathcal{H})$ if for all proper submultisets $\mathcal{F} \subset \mathcal{B}$ we have $\min(\bigcap \mathcal{F}) < \min(\bigcap \mathcal{B})$.*

For $\mathcal{G} \subseteq \mathcal{H}$, we say that a basis $\mathcal{B}$ is a basis of $\mathcal{G}$ *in $(X, \preceq, \mathcal{H})$ if $\mathcal{B} \subseteq \mathcal{G}$ and $\min(\bigcap \mathcal{B}) = \min(\bigcap \mathcal{G})$.*

The combinatorial dimension *of $(X, \preceq, \mathcal{H})$ denoted by* $\dim(X, \preceq, \mathcal{H})$ *is the maximum cardinality of a basis.*

A related model. The framework of concrete LP-type problems is similar to the model presented by Amenta [Ame94] as a *mathematical programming problem.* The small technical differences are that in a mathematical programming problem, more points can have the same value; on the other hand, in an LP-type problem we allow identical constraints.

1.4. Violator spaces

In this section we present yet another abstract framework for optimization problems called violator spaces, which is a proper generalization of LP-type problems.

Violator spaces were introduced in [Ško02] where they were used as a tool for studying structural properties of LP-type problems. Later, several algorithms related to LP-type problems were noted to work with violator spaces. The added generality turned out to be useful in some applications, for example unique sink orientations of grids [GMRŠ06].

For LP-type problems we have defined $\mathsf{V}(G)$ to be the set of all constraints violating G, that is,

$$\mathsf{V}(G) = \{h \in H : w(G) < w(G \cup \{h\})\},$$

and we regard V as a mapping $2^H \to 2^H$ called the *violator mapping*. In a violator space we specify the structure of the problem by the mapping V instead of the weight function.

The definition of violator spaces postulates some properties of the violator mapping. In abstract LP-type problems without $-\infty$ with the mapping V defined as above, the properties are satisfied.

Definition 1.9. *A* violator space *is a pair* (H, V), *where* H *is a finite set and* $\mathsf{V} : 2^H \to 2^H$ *is a mapping satisfying the following properties:*

- $G \cap \mathsf{V}(G) = \emptyset$ *for every* $G \subseteq H$ (consistency)*;*
- *if* $F \subseteq G$ *and* $G \cap \mathsf{V}(F) = \emptyset$ *then* $\mathsf{V}(G) = \mathsf{V}(F)$, *for every* $F, G \subseteq H$ (locality).

The presented definition encompasses only LP-type problems without $-\infty$; indeed, we require the locality axiom to hold for all sets F, G without exceptions. We return to the problem of unboundedness in Chapter 2.

To exhibit an example of violator space we start with the linear programming problem in Figure 1.3. To ensure the existence and uniqueness of optima of subproblems, we consider the lexicographic variant in the positive orthant. The constraints representing the positive orthant are marked in light grey; we do not include them into H, but they are implicitly present in all subproblems. We thus set $H := \{a, b, c, d\}$. To determine $\mathsf{V}(G)$ for instance for $G = \{b, d\}$, we note that the optimum solution with respect to the constraints b and d is the point X. Now we have $a \in \mathsf{V}(G)$ since the optimum solution with respect to $G \cup \{a\} = \{a, b, d\}$ is the point $Y > X$, or in other words, the point X does not satisfy the constraint a. On the other hand $c \notin \mathsf{V}(G)$ since the point X satisfies the constraint c, so the optimum solution with respect to $\{b, c, d\}$ is X. To summarize, we have $\mathsf{V}(b, d) = \{a\}$. All values of V are recorded in the following list:

$\mathsf{V}(\emptyset) = \{a, b\}$, $\mathsf{V}(a) = \{c, d\}$, $\mathsf{V}(b) = \{a, d\}$, $\mathsf{V}(c) = \{a, b\}$, $\mathsf{V}(d) = \{a, b\}$,
$\mathsf{V}(a, b) = \{c, d\}$, $\mathsf{V}(a, c) = \{d\}$, $\mathsf{V}(a, d) = \emptyset$, $\mathsf{V}(b, c) = \{a, d\}$, $\mathsf{V}(b, d) = \{a\}$,
$\mathsf{V}(c, d) = \{a, b\}$, $\mathsf{V}(a, b, c) = \{d\}$, $\mathsf{V}(a, b, d) = \emptyset$, $\mathsf{V}(a, c, d) = \emptyset$, $\mathsf{V}(b, c, d) = \{a\}$,
$\mathsf{V}(a, b, c, d) = \emptyset$.

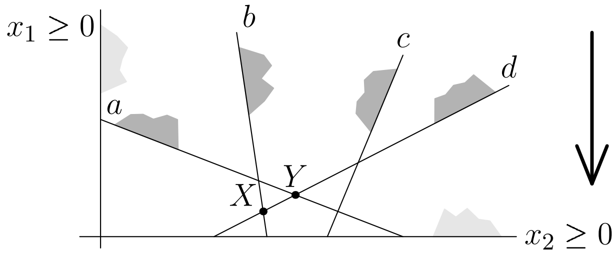

Figure 1.3. A linear program interpreted as a violator space

Bases and combinatorial dimension. The bases in LP-type problems are defined using the weight function. However, Proposition 1.5 suggests the following way of defining bases in the setting of violator spaces:

Definition 1.10. *Consider a violator space* (H, V)*. We call a set* $B \subseteq H$ *a* basis *in* (H, V) *if for all proper subsets* $F \subset B$ *we have* $B \cap \mathsf{V}(F) \neq \emptyset$*.*

For a set $G \subseteq H$*, we say that a basis* B *is a* basis of G *in* (H, V) *if* $B \subseteq G$ *and* $G \cap \mathsf{V}(B) = \emptyset$*.*

The combinatorial dimension *of* (H, V) *denoted by* $\dim(H, \mathsf{V})$ *is the maximum cardinality of a basis.*

From the axiom of locality we immediately get that for a basis B of G we have $\mathsf{V}(B) = \mathsf{V}(G)$. Furthermore we remark that every set $G \subseteq H$ has a basis B (not necessarily unique). To see this, we can take any inclusion-minimal subset $B \subseteq G$ with $\mathsf{V}(B) = \mathsf{V}(G)$.

Basis-regularity and nondegeneracy. We defined basis-regularity of LP-type problems using only the concept of basis in the problem. Now when we have defined basis in violator spaces too, defining basis-regularity of violator spaces is simple.

Definition 1.11. *We say that a violator space* (H, V) *of combinatorial dimension* d *is* basis-regular *if for every* $G \subseteq H$ *with* d *or more elements, each basis* B *of* G *has exactly* d *elements.*

Since the definition of nondegenerate LP-type problems refers to values of the weight function, we cannot get the definition of nondegenerate violator spaces by copying the LP-type version. Instead we proceed in accord with the observation following Definition 1.4.

Definition 1.12. *A violator space* (H, V) *is* nondegenerate *if every* $G \subseteq H$ *has exactly one basis. If the violator space is not nondegenerate, we say that it is* degenerate.

Acyclicity. Influenced by Proposition 1.5 valid for LP-type problems, it is tempting to interpret an information that $G \cap \mathsf{V}(F) \neq \emptyset$ for some sets of constraints $F \subset G$ as that $w(F) < w(G)$, where $w(G)$ is some kind of cost of the optimum solution with respect to G. However, in some violator spaces we have a sequence of sets of constraints $G_1, \ldots, G_k$ with bases $B_1, \ldots, B_k$ for which the interpretation above together with a natural assumption that $w(B_i) = w(G_i)$ asserts that

$$w(G_1) = w(B_1) < w(G_2) = w(B_2) < \cdots < w(G_n) = w(B_n) < w(G_1),$$

which is at least suspicious. We therefore introduce the following definition, characterizing the class of violator spaces where this happens.

Definition 1.13. *We say that a violator space* (H, V) *is* cyclic *if there exists a sequence of sets* $G_1, \ldots, G_k \subseteq H$ *with* $k \geq 2$, $\mathsf{V}(G_1) \neq \mathsf{V}(G_2)$, *and*

$$G_1 \cap \mathsf{V}(G_2) = G_2 \cap \mathsf{V}(G_3) = \cdots = G_k \cap \mathsf{V}(G_1) = \emptyset.$$

Such a sequence $G_1, \ldots, G_k$ *is called* a cycle. *If* (H, V) *is not cyclic, we call it* acyclic.

Examples of cyclic violator spaces are given in [GMRŠ06] and in Chapter 4.

1.5. Relations between the models

In this section we summarize results on relation between the models presented in the previous sections.

A concrete LP-type problem is an abstract LP-type problem. Consider a concrete LP-type problem $(X, \preceq, \mathcal{H})$. Define the function $w : 2^{\mathcal{H}} \to X \cup \{+\infty\}$ by setting

$$w(\mathcal{G}) := \begin{cases} \min \bigcap \mathcal{G} & \text{if } \bigcap \mathcal{G} \text{ is nonempty,} \\ +\infty & \text{if } \bigcap G = \emptyset, \end{cases}$$

for $\mathcal{G} \subseteq \mathcal{H}$. Then $(\mathcal{H}, w)$ is an abstract LP-type problem without $-\infty$. The proof consists of checking the axioms of monotonicity and locality.

Strictly speaking, if the multiset $\mathcal{H}$ has elements with multiplicity greater than 1 then we are cheating: since $\mathcal{H}$ is not a set, it cannot be used as the set of constraints for an abstract LP-type problem. However, this is only a formal obstacle. We can bijectively map $\mathcal{H}$ to a set, i.e., we take any set H with $|H| = |\mathcal{H}|$ and a mapping $f : H \to \mathcal{H}$ so that for every $\bar{h} \in \mathcal{H}$, the number of elements $h \in H$ that map to $\bar{h}$ is equal to the multiplicity of $\bar{h}$. For $G \subseteq H$ we then define $w(G) := \min(\bigcap_{g \in G} f(g))$, and this gives us a honest abstract LP-type problem $P = (H, w, X, \preceq)$ with H being a set.

An abstract LP-type problem is a violator space. Consider an abstract LP-type problem (H, w) without $-\infty$. Let $\mathsf{V} : 2^H \to 2^H$ be its violator mapping, defined as $\mathsf{V}(G) = \{h \in H : w(G) < w(G \cup \{h\})\}$. Then (H, V) is an acyclic violator space. The proof of consistency and locality consists of checking the axioms; acyclicity is obtained from transitivity and antisymmetricity of the ordering of values of the function w.

Basis equivalence. We intuitively feel that the structure of an optimization problem stays the same when we transform between the abstract models. Let us formalize this feeling. We need to compare some feature of the problem that is defined in all models. We use an approach based on comparing the bases in the problem.

Consider an abstract LP-type problem $P = (H, w)$ without $-\infty$ with violator mapping V, and the violator space $P' = (H, \mathsf{V})$. For $B \subseteq G \subseteq H$, we observe that B is a basis of G in P if and only if B is a basis of G in P'. We say that P' is *basis-equivalent* to P.

Similarly, consider a concrete LP-type problem $\mathcal{P} = (X, \preceq, \mathcal{H})$ with the weight mapping w, and the abstract LP-type problem $P = (\mathcal{H}, w)$. For $\mathcal{B} \subseteq \mathcal{G} \subseteq \mathcal{H}$, we see that $\mathcal{B}$ is a basis of $\mathcal{G}$ in P if and only if $\mathcal{B}$ is a basis of $\mathcal{G}$ in $\mathcal{P}$. Again, we say that P is *basis-equivalent* to $\mathcal{P}$. If $\mathcal{H}$ has elements of multiplicity greater than 1, we have to be a bit careful on the formal side. In such a case, by basis-equivalence we mean the existence of a set H and a bijection $f : H \to \mathcal{H}$ such that $B \subseteq G$ is a basis of G in P if and only if the multiset $\{f(b) : b \in B\}$ is a basis of $\{f(g) : g \in G\}$ in $\mathcal{P}$.

Basis-equivalence of a concrete LP-type problem and a violator space is defined in the obvious way.

Equivalence of the models. The main result on the relation of the models is given by the following theorem.

Theorem 1.14. *The axioms of abstract LP-type problems without $-\infty$, of concrete LP-type problems, and of acyclic violator spaces are equivalent. More precisely, every problem in one of the three classes has a basis-equivalent counterpart in each of the other two classes.*

The surprising part of the theorem is that every acyclic violator space (H, V) can be obtained from a suitable concrete LP-type problem $(X, \preceq, \mathcal{H})$. The idea of the proof is to construct the set X as the set of bases of the violator space (H, V). The ordering imposed on X comes out from the requirement that for $F \subseteq G$ we have $w(F) < w(G)$ if and only if $G \cap \mathsf{V}(F) \neq \emptyset$, as Proposition 1.5 asserts. Some technicalities arise in the proof, such as the formal definition of the ordering.

A complete proof of a slightly more general version of this theorem is contained in Section 2.3.

The theorem has a noteworthy algorithmic corollary. Every algorithm for LP-type problems can be used for acyclic violator spaces, whenever it does not inquire about the values of w and instead uses only queries about violation.

Chapter 2

The role of minus infinity

The abstract LP-type framework allows us to represent optimization problems that have some unbounded subproblems. This is achieved by allowing the unbounded sets to break the locality axiom. In this chapter we closely examine the role of unboundedness and of $-\infty$. We present modifications of the models of concrete LP-type problems and violator spaces that allow representing unbounded problems. We state and prove a $-\infty$ analogue of Theorem 1.14 concerning equivalence of LP-type problems, concrete LP-type problems, and acyclic violator spaces.

Expressive power of minus infinity. Every problem with $-\infty$ can be represented by a problem without $-\infty$ exhibiting the same relations on the bounded sets; more precisely, a constraint h violates a bounded set G in the original problem if and only if it violates G in the new problem. The details of such a construction depend on which model we are using. For instance, for abstract LP-type problem we can set $w'(G) := w(G)$ for G bounded and $w'(G) := |G| - M$ for G unbounded, where M is a sufficiently big real number. However, sometimes we need to increase the combinatorial dimension of the problem substantially; see Proposition 3.12. Therefore in applications we may be able to prove better running time bounds etc., if we use the models with minus infinity.

2.1. Concrete LP-type problems with minus infinity

This section's goal is to modify the framework of concrete LP-type problems to be able to represent $-\infty$. We provide two equivalent modifications of the framework.

Concrete LP-type problems as defined in Definition 1.7 cannot encompass abstract LP-type problems where some sets with minus infinite weight break locality. The reason is that the weight mapping of a concrete LP-type problem *always* satisfies the locality axiom. To see this, recall that concrete LP-type problem is a linearly ordered set X of points together with a family $\mathcal{H}$ of subsets of X interpreted as constraints. We defined a weight mapping w of a concrete LP-type problem as $w(\mathcal{G}) := \min(\bigcap \mathcal{G})$. Now it is straightforward to check that for $\mathcal{F} \subseteq \mathcal{G} \subseteq \mathcal{H}$ and $Y \in \mathcal{H}$ with $w(\mathcal{F}) = w(\mathcal{G}) = w(\mathcal{F} \cup \{Y\})$, we have $m := \min(\bigcap \mathcal{F}) \in G$ for every $G \in \mathcal{G}$, and $m \in Y$, hence $w(\mathcal{G} \cup \{Y\}) = w(G)$.

It may not be obvious how to modify the framework of concrete LP-type problems to allow for $-\infty$. However, note that linear programming has a concrete nature, although it is an exemplary class of problems that do need the minus infinity. By inspecting linear programming, we can come up with the following modification of the definition of concrete LP-type problems: we do not require the minimum $\min(\bigcap \mathcal{G})$ to exist if the set $\bigcap \mathcal{G}$ is unbounded from below, and in this case we set $w(\mathcal{G}) := -\infty$. This leads to the following definition.

Definition 2.1. *Let X be a set linearly ordered by a relation $\leq$ and let $\mathcal{H}$ be a finite multiset whose elements are subsets of X. We say that the system $(X, \leq, \mathcal{H})$ is a* concrete LP-type problem with $-\infty$ represented by unbounded sets *if for every $\mathcal{G} \subseteq \mathcal{H}$ with $\bigcap \mathcal{G}$ nonempty and bounded from below, the minimum $\min \bigcap \mathcal{G}$ is attained.*

We define the weight function $w : 2^{\mathcal{H}} \to X \cup \{-\infty, +\infty\}$ of the problem as

$$w(\mathcal{G}) := \begin{cases} +\infty & \text{if } \bigcap \mathcal{G} \text{ is empty,} \\ \min \bigcap \mathcal{G} & \text{if } \bigcap \mathcal{G} \text{ is nonempty and bounded from below,} \\ -\infty & \text{if } \bigcap \mathcal{G} \text{ is not bounded from below.} \end{cases}$$

This definition has a small drawback. From the proof of Theorem 1.14 follows that every abstract LP-type problem without $-\infty$ is basis-equivalent to a suitable concrete LP-type problem $(X, \leq, \mathcal{H})$ with the set X finite. In other words, by requiring the set X to be finite we do not lose any generality. On the other hand, to model a problem with $-\infty$ using the definition above we need some nonempty subsets of X that do not attain the minima. However, if X is finite, this is not possible.

An alternative definition allows to represent any abstract LP-type problem by a concrete one in which the set X is finite. The idea is to introduce several distinct values for minus infinity that do satisfy locality. We keep a separate list Y containing the minus infinities. We keep the ability to tell the minus infinities apart, but to the observer having access only to the values of w they seem the same. Formally we have the following definition.

Definition 2.2. *Let X be a set linearly ordered by a relation $\leq$, let $\mathcal{H}$ be a finite multiset whose elements are subsets of X, and let Y be a subset of X. We say that the system $(X, \leq, \mathcal{H}, Y)$ is a* concrete LP-type problem with $-\infty$ represented by a list *if for every $y \in Y$ and $x \in X \setminus Y$ we have $y \leq x$, and for every $\mathcal{G} \subseteq \mathcal{H}$ with $\bigcap \mathcal{G} \neq \emptyset$, the minimum $\min \bigcap \mathcal{G}$ is attained.*

We define the weight function $w : 2^{\mathcal{H}} \to (X \setminus Y) \cup \{-\infty, +\infty\}$ as

$$w(G) := \begin{cases} +\infty & \text{if } \bigcap \mathcal{G} \text{ is empty,} \\ \min \bigcap \mathcal{G} & \text{if } \bigcap \mathcal{G} \text{ is nonempty and } (\bigcap \mathcal{G}) \cap Y \text{ is empty,} \\ -\infty & \text{if } \bigcap \mathcal{G} \cap Y \text{ is nonempty.} \end{cases}$$

The spirit of Definition 2.1 with unbounded sets is more closely related to actual geometric optimization problems. On the other hand, the theory is more elegant if we adopt Definition 2.2 with the lists. Luckily, the definitions are equivalent in the following sense.

Proposition 2.3. *For every concrete LP-type problem with $-\infty$ with weight function w there exists a concrete LP-type problem with $-\infty$ in the other representation with the weight function w' identical to w up to an isomorphism of the set of constraints; i.e., $w(\mathcal{G}) = w'(\{H' : H \in \mathcal{G}\})$.*

Proof. First assume that we have a problem $(X, \leq, \mathcal{H}, Y)$ with $-\infty$ represented by a list. We replace every element $y \in Y$ by an unbounded infinite chain $Y_y := \{(-1, y), (-2, y), ...\}$. We define

$$X' := (X \setminus Y) \cup \bigcup_{y \in Y} Y_y$$

and we define the ordering $\leq'$ on X' by saying that $a \leq' b$ if

- $a, b \in X \setminus Y$ and $a \leq b$, or
- $a \in Y$ and $b \in X \setminus Y$, or
- $a, b \in Y$, $a = (i, y)$, $b = (j, z)$, and $i < j$, or
- $a, b \in Y$, $a = (i, y)$, $b = (i, z)$, and $y \leq z$.

For every $H \in \mathcal{H}$ we define

$$H' := (H \setminus Y) \cup \bigcup_{y \in H \cap Y} Y_y$$

and we set $\mathcal{H}' := \{H' : H \in \mathcal{H}\}$. Now $(X', \leq', \mathcal{H}')$ is the desired concrete LP-type problem with $-\infty$ represented by a list, equivalent to $(X, \leq, \mathcal{H}, Y)$. We omit the straightforward check of this fact.

In the other direction assume that in the problem $(X, \leq, \mathcal{H})$ we have $-\infty$ represented by unbounded sets. We define Y as the list of unbounded subsets of $\mathcal{H}$:

$$Y := \{y_{\mathcal{G}} : \mathcal{G} \subseteq \mathcal{H} \text{ with } w(\mathcal{G}) = -\infty\},$$

where $y_{\mathcal{G}} \notin X$ is a formal symbol. We set $X' := X \cup Y$. We define a relation $\leq'$ on X' as a linear ordering extending $\leq$ so that for every $a \in Y, b \in X$ we have $a \leq b$, and for every $y_F, y_G \in Y$ with $F \subseteq G$ we have $y_F \leq y_G$. Now, for $H \in \mathcal{H}$ we define

$$H' := H \cup \{y_{\mathcal{G}} : \mathcal{G} \subseteq \mathcal{H} \text{ with } w(\mathcal{G}) = -\infty \text{ and } H \in \mathcal{G}\}$$

and we set $\mathcal{H}' := \{H' : H \in \mathcal{H}\}$. Now $(X', \leq', \mathcal{H}', Y)$ is a concrete LP-type problem with $-\infty$ represented by unbounded sets, equivalent to $(X, \leq, \mathcal{H})$, which is again routine to check. ∎

Linear programming. With these definitions it is tempting to think that representing linear programs by concrete LP-type problems with $-\infty$ is simple. Having a linear program $\mathcal{P}$ of minimizing x_1 subject to $A\boldsymbol{x} \leq \boldsymbol{b}$, it seems reasonable to take $H_i := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i\}$ and expect that $(\mathbb{R}^n, \leq_{\mathrm{L}}, \{H_i : i = 1, \ldots, m\})$ is a concrete LP-type problem with $-\infty$ represented by unbounded sets. But generally this does not work. Consider a linear program with variables x_1, x_2 and a single constraint $x_1 \geq 0$. We have H_1 nonempty, since $(1, 0) \in H_1$, and bounded from below for instance by $(-1, 0)$, but H_1 has no lexicographic minimum; therefore the conditions of Definition 2.1 are not satisfied.

However, every linear program does have a concrete representation conforming to our definitions. It can be obtained from an abstract LP-type representation by processing it as described in the following sections.

2.2. Violator spaces with minus infinity

The goal of this section is to introduce violator spaces with $-\infty$. To get an inspiration for the axioms in the formal definition, we start with proving some results concerning violation in abstract LP-type problems.

Some of the propositions below strongly resemble the non-$-\infty$ versions, differing only by omitting the assumption $w(F) \neq -\infty$. This kind of extension may seem tautological. However, the final result of the following section, i.e., the $-\infty$ version of equivalence of LP-type problems and violator spaces, is not trivial.

Lemma 2.4. *Let (H, w) be an LP-type problem with violator mapping V. If for some $F \subseteq G \subseteq H$ we have $w(F) = w(G) \neq -\infty$ then $\mathsf{V}(F) = \mathsf{V}(G)$.*

Proof. If $h \in \mathsf{V}(F)$, then $w(G) = w(F) < w(F \cup \{h\}) \leq w(G \cup \{h\})$, hence $h \in \mathsf{V}(G)$.

Conversely assume that $h \notin \mathsf{V}(F)$, that is, $w(F) = w(F \cup \{h\})$. Since $w(F) = w(G) \neq -\infty$, locality applies and gives $w(G) = w(G \cup \{h\})$, that is, $h \notin \mathsf{V}(G)$. ∎

Proposition 2.5. *Let (H, w) be an LP-type problem with violator mapping V. Then the following holds.*

(i) For every $G \subseteq H$ we have $G \cap \mathsf{V}(G) = \emptyset$.
(ii) For every $F \subseteq G \subseteq H$ with $w(F) \neq -\infty$ and $G \cap \mathsf{V}(F) = \emptyset$, we have $\mathsf{V}(F) = \mathsf{V}(G)$.
(iii) For every $F \subseteq G \subseteq H$ with $w(G) = -\infty$ we have $w(F) = -\infty$.
(iv) For every $G \subseteq H$ with $w(G) = -\infty$ we have $\mathsf{V}(G) = \{h \in H : G \cup \{h\} \neq -\infty\}$.

Proof. The statements (i) and (iv) are obvious from the definition of V. The statement (ii) follows by combining Proposition 1.5 (which gives $w(F) = w(G)$) and Lemma 2.4. The statement (iii) is immediate from monotonicity. ∎

Now we are ready to introduce the $-\infty$ into the world of violator spaces. We proceed as in Definition 2.2: to an ordinary violator space we add a list $\mathcal{U}$ representing unbounded sets. The axioms in the definition match the conclusions of Proposition 2.5.

Definition 2.6. *Let H be a finite set, V a mapping $2^H \to 2^H$, and $\mathcal{U}$ a subset of 2^H. We say that $(H, \mathsf{V}, \mathcal{U})$ is a* violator space with minus infinity *if the following four conditions hold:*

- *for every $G \subseteq H$, we have $G \cap \mathsf{V}(G) = \emptyset$* (consistency)*;*
- *for every $F \subseteq G \subseteq H$ with $F \notin \mathcal{U}$ and $G \cap \mathsf{V}(F) = \emptyset$, we have $\mathsf{V}(F) = \mathsf{V}(G)$* (locality)*;*

- *for every* $F \subseteq G \subseteq H$ *with* $G \in \mathcal{U}$ *we have* $F \in \mathcal{U}$ (monotonicity)*;*
- *for every* $G \in \mathcal{U}$*, we have* $\mathsf{V}(G) = \{h \in H : G \cup \{h\} \notin \mathcal{U}\}$ (matching of V and $\mathcal{U}$)*.*

We say that a subset $G \subseteq H$ *is* unbounded *if* $G \in \mathcal{U}$*, otherwise it is* bounded.

From every LP-type problem we can get a basis-equivalent violator space with $-\infty$. This follows from Proposition 2.5. In the next section we prove that the obtained violator space is acyclic.

We continue by few more definitions.

Definition 2.7. *Let* $(H, \mathsf{V}, \mathcal{U})$ *be a violator space with* $-\infty$*. We say that* $B \subseteq H$ *is a* basis *if*

- *either* B *is bounded and all bounded proper subsets* $F \subset B$ *satisfy* $B \cap \mathsf{V}(F) \neq \emptyset$*,*
- *or* B *is unbounded and empty.*

By $\mathcal{B}$ *we mean the set of all bases.*

For a set $G \subseteq H$ *we say that a basis* $B \subseteq G$ *is a* basis of G *if either* B *is bounded and* $G \cap \mathsf{V}(B) = \emptyset$*, or* G *is unbounded.*

By a combinatorial dimension *of the problem denoted by* $\dim(H, \mathsf{V}, \mathcal{U})$ *we mean the maximum cardinality of a basis.*

Observe that every basis is a basis of itself, and that every set has a basis. Moreover, if B is a basis of $G \subseteq H$ then B and G are either both bounded or both unbounded. Furthermore, if B is a basis of a bounded set G, from locality follows that $\mathsf{V}(B) = \mathsf{V}(G)$.

2.3. Equivalence of the models with minus infinity

In this section we prove the $-\infty$ version of Theorem 1.14. The proof goes along the lines of the proof of non-$-\infty$ version [GMRŠ06]. We just have to take care of the unbounded sets.

First we define the basis-equivalence as existence of a bijection preserving bases and boundedness.

Definition 2.8. *Let* $\mathcal{P}$ *be an abstract LP-type problem or violator space with* $-\infty$*; let* $\mathcal{P}'$ *be an abstract or concrete LP-type problem or violator space with* $-\infty$*. Let* H, H' *denote the (multi)sets of constraints of* $\mathcal{P}, \mathcal{P}'$*. We say that* $\mathcal{P}$ *is* basis-equivalent *to* $\mathcal{P}'$ *if there exists a bijection* $\psi : H \to H'$ *such that*

- *for every* $G \subseteq H$*, the (multi)set* $\psi(G) := \{\psi(g) : g \in G\}$ *is bounded in* $\mathcal{P}'$ *if and only if* G *is bounded in* $\mathcal{P}$*,*
- *for every* $B \subseteq G \subseteq H$*, the (multi)set* $\psi(B)$ *is a basis of* $\psi(G)$ *in* $\mathcal{P}'$ *if and only if* B *is a basis of* G *in* $\mathcal{P}$*.*

Hereat, if H' *is a multiset, by a bijection* $H \to H'$ *we mean a mapping* ψ *for which the number of elements* $h \in H$ *that map to a particular* $h' \in \mathcal{H}'$ *is equal to the multiplicity of* h' *in* H'*.*

We successively prove several results on existence of basis-equivalent representations; we sum them up in Theorem 2.17. We start with exhibiting abstract LP-type representations of concrete LP-type problems.

Proposition 2.9. *Every concrete LP-type problem with $-\infty$ can be represented by a basis-equivalent abstract LP-type problem.*

Proof. Let $(X, \leq, \mathcal{H}, Y)$ be a concrete LP-type problem with $-\infty$ represented by a list. Let w be its weight function. Let H be a set with $|H| = |\mathcal{H}|$ and let ψ be a bijection $H \to \mathcal{H}$. Let the mapping $w' : 2^H \to (X \setminus Y) \cup \{-\infty, +\infty\}$ be defined as $w'(G) := w(\{\psi(g) : g \in G\})$. It is easy to check that (H, w') is an abstract LP-type problem basis-equivalent to $(X, \leq, \mathcal{H}, Y)$. ∎

The other two results on representations concern *acyclic* violator spaces. In order to define acyclicity we need some auxiliary notions.

Definition 2.10. *Let $(H, \mathsf{V}, \mathcal{U})$ be a violator space with $-\infty$. We say that bases B, C are* equivalent*, written as $B \sim C$, if either both B, C are bounded and $\mathsf{V}(B) = \mathsf{V}(C)$, or both B, C are unbounded.*

Clearly, the relation $\sim$ defined on the set $\mathcal{B}$ of all bases is an equivalence relation. If B is a basis, by $[B]$ we mean the equivalence class containing B. We write $\mathcal{B}/\!\sim$ for the set of all equivalence classes.

In LP-type problems, equivalence of bases preserves the weight function:

Observation 2.11. *Let (H, w) be an abstract LP-type problem. Let V be its violator mapping and let $\mathcal{U} \subseteq 2^H$ contain the unbounded sets. Then for any two equivalent bases B, B' we have $w(B) = w(B')$.*

Proof. If B, B' are both bounded, we have

$$(B \cup B') \cap \mathsf{V}(B) = (B \cap \mathsf{V}(B)) \cup (B' \cap \mathsf{V}(B')) = \emptyset,$$

hence Proposition 1.5 implies that $w(B) = w(B \cup B')$; symmetrically we get that $w(B \cup B') = w(B')$. If B, B' are both unbounded, we have $w(B) = -\infty = w(B')$. ∎

Now we define an ordering of the bases, from this we derive an ordering of the equivalence classes, and finally we define the acyclicity of violator spaces.

Definition 2.12. *Let $(H, \mathsf{V}, \mathcal{U})$ be a violator space with $-\infty$. Let us define a relation $\leq_0$ between subsets of H. If F, G are both bounded, we say that $F \leq_0 G$ if $F \cap \mathsf{V}(G) = \emptyset$. For unbounded $F \subseteq H$ we say that $F \leq_0 G$ for all $G \subseteq H$.*

For equivalence classes $[B], [C] \in \mathcal{B}/\!\sim$ we say that $[B] \leq_0 [C]$ if there exist bases $B' \in [B]$ and $C' \in [C]$ such that $B' \leq_0 C'$. We write $[B] <_0 [C]$ if we have $[B] \leq_0 [C]$ and $[B] \neq [C]$.

We define a relation $\leq_1$ on $\mathcal{B}/\!\sim$ as the transitive closure of $\leq_0$. The relation $\leq_1$ is clearly reflexive and transitive. If it is moreover antisymmetric, we say that the violator space $(H, \mathsf{V}, \mathcal{U})$ is acyclic.

Now we are ready to prove that abstract LP-type problems can be represented as acyclic violator spaces, and those in turn as concrete LP-type problems.

Proposition 2.13. *Let (H, w) be an abstract LP-type problem with violator mapping V. Set $\mathcal{U} := \{G \subseteq H : w(G) = -\infty\}$. Then $(H, \mathsf{V}, \mathcal{U})$ is an acyclic violator space with $-\infty$ basis-equivalent to (H, w).*

Proof. Proposition 2.5 implies that $(H, \mathsf{V}, \mathcal{U})$ is a violator space with $-\infty$.

We proceed with acyclicity. First we prove that we have $w(B) < w(C)$ whenever $[B] <_0 [C]$. For this, let B, C be bases such that $[B] \neq [C]$ and $[B] \leq_0 [C]$. If B is unbounded then C is bounded, therefore $w(B) = -\infty < w(C)$. If B is bounded, we have $B' \cap \mathsf{V}(C') = \emptyset$ for some $B' \in [B]$ and $C' \in [C]$. Proposition 1.5 implies that $w(C') = w(B' \cup C')$. We claim that we have $w(B') < w(C')$. To prove this, assume for contradiction that $w(B') \geq w(C')$. We have $w(B') \geq w(C') = w(B' \cup C') \geq w(B')$, hence $w(B') = w(B' \cup C') = w(C')$. Since $w(B') \neq -\infty$, Lemma 2.4 applies and gives $\mathsf{V}(B') = \mathsf{V}(B' \cup C') = \mathsf{V}(C')$, which contradicts $[B] \neq [C]$. This finishes the proof that $w(B') < w(C')$. By Observation 2.11 we have $w(B) < w(C)$.

If we have $[B] <_1 [C]$, we can chain several $<_0$ to get $w(B) < w(C)$. Since the relation $\leq$ on the range of the weight function is an ordering, antisymmetricity of $\leq_1$ follows. This concludes the proof of acyclicity.

Finally we prove the basis-equivalence. We set $\psi : H \to H$ to be the identity mapping. By Lemma 2.4 and Observation 2.11, a set B is an inclusion-minimal subset of G with $w(B) = w(G)$ if and only if B is an inclusion-minimal subset of G with $\mathsf{V}(B) = \mathsf{V}(G)$. Hence B is a basis of G in the LP-type problem if and only if B is a basis of G in the violator space. Furthermore we get that $w(G) = -\infty$ if and only if $G \in \mathcal{U}$ from the definition of $\mathcal{U}$. Therefore $(H, \mathsf{V}, \mathcal{U})$ is basis-equivalent to (H, w). ■

Proposition 2.14. *Every acyclic violator space with $-\infty$ can be represented by a basis-equivalent concrete LP-type problem with $-\infty$.*

Proof. Let $(H, \mathsf{V}, \mathcal{U})$ be an acyclic violator space with $-\infty$. In the concrete representation we take the following two kinds of points: the equivalence classes of bases, and the unbounded sets; i.e., we set $X := (\mathcal{B}/\!\sim) \cup \mathcal{U}$.

We define the mapping $S : H \to 2^X$ that will act as a concretization of the constraints in H:

$$S(h) := \big\{[B] : B \text{ is a basis satisfying } h \notin \mathsf{V}(B)\big\} \cup \big\{U \in \mathcal{U} : h \in U\big\}.$$

Let $\mathcal{H}$ be the image of the mapping S taken as a multiset:

$$\mathcal{H} := \big\{S(h) : h \in H\big\}.$$

Thus, S is a bijection between H and $\mathcal{H}$. Let σ be the induced bijection of 2^H and $2^{\mathcal{H}}$ defined as $\sigma(G) := \{S(h) : h \in G\}$ for $G \subseteq H$.

Let $\leq$ be an arbitrary linear ordering on X such that all elements of $\mathcal{U}$ are less than all elements of $\mathcal{B}/\!\sim$, on $\mathcal{U}$ the ordering $\leq$ respects inclusion, and on $\mathcal{B}/\!\sim$ it respects $\leq_1$ (which is an ordering since $(H, \mathsf{V}, \mathcal{U})$ is acyclic).

We claim that $(X, \leq, \mathcal{H}, \mathcal{U})$ is a concrete LP-type problem with $-\infty$ represented by a list. The existence of a minimum element in every nonempty intersection $\bigcap \mathcal{G}$ for $\mathcal{G} \subseteq \mathcal{H}$ is guaranteed by finiteness of X and the linearity of $\leq$; the condition $y \leq x$ for $y \in Y$ and $x \in X \setminus Y$ is guaranteed by the construction of $\leq$.

It remains to prove the basis-equivalence.

First we prove that $G \subseteq H$ is unbounded in the violator space if and only if $\sigma(G)$ is unbounded in the concrete LP-type problem. First assume that $G \in \mathcal{U}$.

Now for every $g \in G$ we have $G \in S(g)$, hence $G \in \bigcap_{g\in G} S(g) \cap \mathcal{U}$. Therefore $w(\sigma(G)) = -\infty$. In the other direction, let $w(\sigma(G)) = -\infty$; then there exists $F \in \mathcal{U}$ with $F \in \bigcap_{g\in G} S(g)$. Therefore for every $g \in G$ we have $F \in S(g)$, i.e., $F \in \{U \in \mathcal{U} : g \in U\}$, therefore $g \in F$. We infer that $G \subseteq F$, and since $F \in \mathcal{U}$, we have $G \in \mathcal{U}$.

To prove that B is a basis of G in the violator space if and only if $\sigma(B)$ is a basis of $\sigma(G)$ in the concrete LP-type problem, we use the following two lemmas.

Lemma 2.15. *Let B be a bounded basis of $G \subseteq H$ in the violator space. Then $w(\sigma(B)) = w(\sigma(G))$ in the concrete LP-type problem.*

Proof. We certainly have $[B] \in \bigcap \sigma(G)$. We claim that $[B]$ is actually the minimum in $\bigcap \sigma(G)$. To prove this, consider a basis C with $[C] \leq_0 [B]$; i.e., $[C] \in \bigcap \sigma(G)$, and $C \cap \mathsf{V}(B) = \emptyset$. Since $[C] \in \bigcap \sigma(G)$, we have $G \cap \mathsf{V}(C) = \emptyset$, which is equivalent to $(G \cup C) \cap \mathsf{V}(C) = \emptyset$, and $C \cap \mathsf{V}(B) = \emptyset$ is equivalent to $(G \cup C) \cap \mathsf{V}(B) = \emptyset$. By applying locality in $(H, \mathsf{V}, \mathcal{U})$ to these two equations, we get $\mathsf{V}(C) = \mathsf{V}(G \cup C) = \mathsf{V}(B)$, i.e., $[C] = [B]$. Therefore $[B] = \min \bigcap \sigma(G)$ as claimed. Since B is a basis of itself, we can use the just proved result to obtain $[B] = \min \bigcap \sigma(B)$. The equality $w(\sigma(G)) = [B] = w(\sigma(B))$ follows. ∎

Lemma 2.16. *Let B and G be subsets of H such that $\sigma(B)$ is a bounded basis of $\sigma(G)$ in the concrete LP-type problem. Then $\mathsf{V}(B) = \mathsf{V}(G)$ in the violator space.*

Proof. Since $\sigma(B)$ is bounded, B is bounded too. Let A be a basis of B, so $\mathsf{V}(A) = \mathsf{V}(B)$. Note that $[A] \in \bigcap \sigma(B)$. Let $[C] := \min \bigcap \sigma(B)$. Then $B \cap \mathsf{V}(C) = \emptyset$, thus $A \cap \mathsf{V}(C) = \emptyset$. This means that $[A] \leq_0 [C]$, hence $[A] = [C]$. From $\min \bigcap \sigma(G) = [C]$ we get $G \cap \mathsf{V}(C) = \emptyset$, therefore $G \cap \mathsf{V}(B) = \emptyset$. Since $\sigma(B) \subseteq \sigma(G)$ if and only if $B \subseteq G$, we can apply locality, which gives $\mathsf{V}(B) = \mathsf{V}(G)$. ∎

Now we are ready to finish the proof of the basis-equivalence of the violator space and the concrete LP-type problem. Let B be a basis of G in the violator space. First assume that B, G are bounded. By the first lemma we have $w(\sigma(B)) = w(\sigma(G))$. Then for every proper subset $A \subset B$ we have $w(\sigma(A)) < w(\sigma(G))$; otherwise the second lemma yields a contradiction to the minimality of B. Hence $\sigma(B)$ is indeed a basis of $\sigma(G)$. If B, G are unbounded then $\sigma(G)$ is unbounded, $B = \emptyset$, and $\sigma(B) = \emptyset$ is obviously a basis of $\sigma(G)$. The reasoning that a basis in the concrete LP-type problem gives a basis in the violator space is analogous. This concludes the proof of Proposition 2.14. ∎

We conclude with summing up the equivalence of the models.

Theorem 2.17. *The axioms of abstract LP-type problems, of concrete LP-type problems with minus infinity, and of acyclic violator spaces with minus infinity are equivalent. More precisely, every problem in one of the three classes has a basis-equivalent problem in each of the other two classes.*

Proof. This follows by combination of Propositions 2.9, 2.13, and 2.14. ∎

Chapter 3

Removing degeneracy may need to increase dimension

Many descriptions of algorithms in computational geometry and in geometric optimization, as well as numerous proofs in discrete geometry, start with a sentence similar to "Let us assume that the given points are in general position." General position may mean that no three among the points are collinear, or we may also require than no four are cocircular, etc., depending on the considered problem. Violations of general positions, such as three points on a line, are referred to as *degeneracies*.

Assuming the input to be nondegenerate (i.e., in general position) usually simplifies the description, analysis, and implementation of a geometric algorithm significantly. For many algorithms, this assumption can be avoided with some extra work and careful attention to detail. However, for some algorithms, the nondegeneracy assumption is not only a convenient simplification, but rather an essential condition for correctness and/or for running time analysis, that seems difficult to circumvent.

General methods have been developed for removing degeneracies in geometric algorithms, based on *infinitesimal perturbations* of the input. Roughly speaking, the coordinates of each input object are changed by a suitable function of a real parameter $\varepsilon > 0$, and the considered algorithm is executed with these new input objects, treating ε as a formal quantity smaller than any concrete nonzero number occurring in the algorithm. These methods can actually be implemented, but they have several drawbacks: They slow down the computations significantly (typically by a large constant factor, but sometimes even much more), they increase space requirements, and sometimes it may be difficult or impossible to reconstruct the correct result for the original input from the result for the perturbed input.

Removing degeneracies means "breaking ties" in some sense. Of course, the ties cannot be broken arbitrarily, since geometric algorithms almost always depend on some kind of global consistency of the input. Still, one might hope for some simpler, perhaps combinatorial, way of removing degeneracies. The present chapter was motivated by this question, and its results can be regarded as an indication that a simple, general, and efficient combinatorial method is unlikely to exist.

Degeneracy in LP-type problems. In the setting of LP-type problems, degeneracy can be an issue too. In particular, Matoušek [Mat95] described an algorithm for

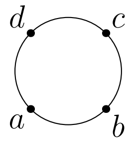

Figure 3.1. A degenerate LP-type problem where removing degeneracy increases dimension

finding the optimum solution satisfying all but at most k of the given constraints. However, for degenerate LP-type problems the algorithm can give a wrong answer.

Recall that an LP-type problem (H, w) is denegerate if we have $w(B_1) = w(B_2)$ for some distinct bases B_1, B_2 (see Definition 1.4). Furthermore we remind that in a nondegenerate LP-type problem, every $G \subseteq H$ has exactly one basis.

To remove degeneracies from an LP-type problem, we want to break the ties $w(B_1) = w(B_2)$ by slightly modifying the values of w, while retaining all strict inequalities between the original values.

Definition 3.1. *An LP-type problem (H, w') is a* refinement *of an LP-type problem (H, w) on the same set of constraints if for all $F, G \subseteq H$ with $w(F) < w(G)$ we have $w'(F) < w'(G)$.*

We thus formalize removing degeneracies from an LP-type problem (H, w) as the question of finding a nondegenerate refinement of (H, w).

Observe that a basis in (H, w) is a basis in any refinement (H, w') as well. Indeed, if B is a basis in (H, w), then for every proper subset $F \subset B$ we have $w(F) < w(B)$, hence $w'(F) < w'(B)$. Note that this implies that refining the problem does not decrease the dimension. More formally, for a problem (H, w) and its refinement (H, w') we have $\dim(H, w) \leq \dim(H, w')$.

At first sight it might seem that in order to produce a nondegenerate refinement, it should suffice to impose some suitable linear ordering on every group of bases sharing the same value of w—perhaps one could even take an arbitrary ordering. However, some thought reveals that things are not that simple. As was observed by Matoušek [Mat95], sometimes we also have to create new bases, and even larger ones than those present in (H, w). Namely, consider the problem of the smallest enclosing ball for points $H = \{a, b, c, d\}$ forming the vertices of a square; see Figure 3.1. The set H has two bases $B_1 = \{a, c\}$ and $B_2 = \{b, d\}$, and the combinatorial dimension of the problem is 2. We will refer to this particular 2-dimensional LP-type problem as the *square example* denoted by (H_0, w_0). Any nondegenerate refinement has dimension at least 3, as we check in Section 3.1 below.

The main result of this chapter is that we exhibit LP-type problems where removing degeneracies requires arbitrarily large increase of the dimension; see Sections 3.1–3.3.

Theorem 3.2. *There exists a positive constant $\varepsilon > 0$ such that for infinitely many values of D, there is a D-dimensional LP-type problem without $-\infty$ for which every nondegenerate refinement has combinatorial dimension at least $(1 + \varepsilon)D$.*

The example of an LP-type problem as in the theorem is obtained by an "iterated join" of the square example. We also show that an essentially equivalent example can be represented as a (highly degenerate) linear program in the usual sense; see Section 3.5.

The result can also be understood as telling us that for degenerate LP-type problems, the combinatorial dimension doesn't convey a full "dimensionality" information about the problem. An alternative dimension parameter might be the smallest possible dimension of a nondegenerate refinement; however, this appears quite hard to determine. In actual geometric problems, we can resort to the dimension of the ambient space, which doesn't grow when degeneracies are removed by perturbation methods. But this brings us back to our initial point—geometric perturbations are not easy to deal with.

The main open question is, can the smallest possible dimension of a nondegenerate refinement be bounded by some function of the dimension of the original degenerate LP-type problem? In particular, does every 2-dimensional LP-type problem have a nondegenerate refinement of dimension bounded by a universal constant? We present some observations about the structure of 2-dimensional degenerate LP-type problems in Section 3.7, but it seems that our methods are not sufficient to give a final answer.

Actually, constructing a D-dimensional LP-type problem with $-\infty$ that does not have a nondegenerate refinement of dimension $2D-1$ is not hard; see Section 3.6. But we feel that here the growth of dimension is caused primarily by the presence of $-\infty$. We therefore focus on LP-type problems without $-\infty$ in the remainder of this chapter.

3.1. Structure of nondegenerate LP-type problems

Let (H, w) be an LP-type problem with $w : 2^H \to \mathbb{R}$. We consider the partially ordered set (poset) $(2^H, \subseteq)$ of all subsets of H ordered by inclusion. For every $x \in \mathbb{R}$, the system $\mathcal{P}_x := \{G \subseteq H : w(G) = x\}$ is a subposet of $(2^H, \subseteq)$, and these subposets for all $x \in \mathbb{R}$ form a disjoint cover of 2^H. Monotonicity implies that a poset $\mathcal{P}_x$ has no "holes": If $F \subseteq M \subseteq G$ and $w(F) = w(G) = x$, then $w(M) = x$ as well. The following lemma states that for *nondegenerate* LP-type problems, each $\mathcal{P}_x$ is actually a copy of a Boolean algebra.

Lemma 3.3 (Cube lemma). *Let (H, w) be a nondegenerate LP-type problem without $-\infty$. For every $x \in \mathbb{R}$ with $\mathcal{P}_x \neq \emptyset$ there exist two (uniquely determined) sets $B, C \subseteq H$ such that $\mathcal{P}_x = \{F \subseteq H : B \subseteq F \subseteq C\}$. The set B is the basis of all $F \in \mathcal{P}_x$.*

We call the set $[B, C] := \{F \subseteq H : B \subseteq F \subseteq C\}$ a *cube* with the *bottom vertex* B and the *top vertex* C. By *dimension* of the cube we mean $|C \setminus B|$.

Proof. Let G be an arbitrary set in $\mathcal{P}_x$. Let B be the basis of G and let C be the set of constraints that may be added to B without changing the value of w:

$$C := \big\{h \in H : w(B) = w(B \cup \{h\})\big\} = H \setminus \mathsf{V}(B).$$

We claim that this choice of B and C satisfies the desired conditions. First we note that Proposition 1.5 readily implies that $w(B) = w(C)$. Therefore $[B, C] \subseteq \mathcal{P}_x$.

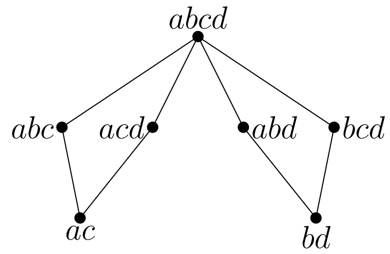

Figure 3.2. The poset $\mathcal{P}_{w_0(H_0)}$ for the square example

Now let us assume that $w(F) = w(B)$ for some $F \subseteq H$. Let B' be a basis of F; we have $w(B') = w(F) = w(B)$, and thus $B = B'$ by nondegeneracy. In particular, $B \subseteq F$. For every $f \in F$ we have $w(B) \leq w(B \cup \{f\}) \leq w(F) = w(B)$, so $w(B) = w(B \cup \{f\})$, and hence $f \in C$. Thus $\mathcal{P}_x \subseteq [B, C]$. The lemma is proved. ∎

To see how this lemma can be used, let us check that every nondegenerate refinement of the square example (H_0, w_0) has dimension at least 3. The poset $\mathcal{P}_{w_0(H_0)}$ of all subsets of H_0 with the same smallest enclosing circle as that of H_0 consists of all subsets of $\{a, b, c, d\}$ containing $\{a, c\}$ or $\{b, d\}$; see Figure 3.2.

In any nondegenerate refinement, $\mathcal{P}_{w_0(H_0)}$ has to be expressed as a disjoint union of cubes. If the dimension of the refinement was 2, all of these cubes would have to have a 2-element set as the bottom vertex. Such a covering is obviously impossible, since in order to cover $\{a, b, c, d\}$, we have to use a 2-dimensional cube, say $[\{a, c\}, \{a, b, c, d\}]$, and any covering of the remaining sets $\{b, d\}$, $\{a, b, d\}$, and $\{b, c, d\}$ by disjoint cubes must use at least one of the 0-dimensional (single-vertex) cubes $[\{a, b, d\}, \{a, b, d\}]$ and $[\{b, c, d\}, \{b, c, d\}]$.

3.2. The construction

We begin by defining a binary operation on LP-type problems.

Definition 3.4. *Let (H_1, w_1) and (H_2, w_2) be LP-type problems with weights in $\mathbb{R}$, and assume that $H_1 \cap H_2 = \emptyset$. We define a new LP-type problem denoted by $(H, w) = (H_1, w_1) * (H_2, w_2)$ and called the* join *of (H_1, w_1) and (H_2, w_2), by setting $H := H_1 \cup H_2$ and $w(G) := w_1(G \cap H_1) + w_2(G \cap H_2)$ for all $G \subseteq H$.*

Lemma 3.5. *The join $(H, w) = (H_1, w_1) * (H_2, w_2)$ is indeed an LP-type problem, and $\dim(H, w) = \dim(H_1, w_1) + \dim(H_2, w_2)$.*

Proof. First note that if $F \subseteq G$ and $w(F) = w(G)$, then $w_i(F \cap H_i) = w_i(G \cap H_i)$ for $i = 1, 2$. Indeed, since $F \cap H_i \subseteq G \cap H_i$, we have $w_i(F \cap H_i) \leq w_i(G \cap H_i)$, and to get equality of the sum, equality must hold in both summands.

Now we verify the axioms for (H, w). Monotonicity is obvious, and for locality, let $F \subseteq G \subseteq H$ and $h \in H$ satisfy $w(F) = w(G) = w(F \cup \{h\})$. Supposing $h \in H_1$, we have $w_1(F \cap H_1) = w_1(G \cap H_1) = w_1((F \cap H_1) \cup \{h\})$ by the observation above, and locality in (H_1, w_1) yields $w_1((G \cap H_1) \cup \{h\}) = w_1(G)$. Then $w(G \cup \{h\}) =$

$w_1((G \cap H_1) \cup \{h\}) + w_2(G \cap H_2) = w_1(G \cap H_1) + w_2(G \cap H_2) = w(G)$. Therefore (H, w) is indeed an LP-type problem.

Now we check that $\dim(H, w) \geq \dim(H_1, w_1) + \dim(H_2, w_2)$. Let B_i be a basis in (H_i, w_i) witnessing $\dim(H_i, w_i)$. It suffices to check that $B = B_1 \cup B_2$ is a basis in (H, w); that is, $w(A) < w(B)$ for every proper subset of B. Letting $A_i := A \cap H_i$, we have $A_i \subseteq B_i$ with at least one of the inclusions proper, say $A_1 \subset B_1$. Since B_1 is a basis, we have $w_1(A_1) < w_1(B_1)$, and $w(A) < w(B)$ follows.

To prove the opposite inequality $\dim(H, w) \leq \dim(H_1, w_1) + \dim(H_2, w_2)$, we choose a basis B in (H, w) with $|B| = \dim(H, w)$ and we set $B_i := B \cap H_i$. It suffices to check that B_i is a basis in (H_i, w_i). Let us consider a proper subset $A_1 \subset B_1$; then $w_1(B_1) + w_2(B_2) = w(B_1 \cup B_2) > w(A_1 \cup B_2) = w_1(A_1) + w_2(B_2)$, and we get $w_1(A_1) < w_1(B_1)$ as needed. The lemma is proved. ■

The example. For the proof of Theorem 3.2 we define, for a positive integer m, an LP-type problem $\mathcal{L}_m = (H, w)$ as the m-fold join of the square example (H_0, w_0). More formally, we choose distinct elements $a_1, a_2, \ldots, a_m$, $b_1, \ldots, b_m$, $c_1, \ldots, c_m$, $d_1, \ldots, d_m$, we let $H_i := \{a_i, b_i, c_i, d_i\}$, and we let $w_i : H_i \to \mathbb{R}$ be a copy of the value function w_0 from the square example, defined on H_i. We let $\mathcal{L}_m = (H, w) := (H_1, w_1) * (H_2, w_2) * \cdots * (H_m, w_m)$. We have $|H| = 4m$ and by the above lemma, $\mathcal{L}_m$ is an LP-type problem of combinatorial dimension $D = 2m$.

We want to bound from below the dimension of any nondegenerate refinement of $\mathcal{L}_m$. Similar to the warm-up argument for (H_0, w_0), any nondegenerate refinement $\mathcal{L}' = (H, w')$ of $\mathcal{L}_m$ of dimension D' yields a covering of the poset

$$\mathcal{P}_{w(H)} = \big\{G \subseteq H : w(G) = w(H)\big\}$$

by disjoint cubes $[B_j, C_j]$, where each bottom vertex B_j satisfies $|B_j| \leq D'$. We deal with this combinatorial problem in the next section.

The case $m = 2$. We analyze the 4-dimensional LP-type problem $\mathcal{L}_2$ in Section 3.4 below. We prove by a case analysis that every nondegenerate refinement has dimension at least 6. The corresponding poset $\mathcal{P}_{w(H)}$ is illustrated in Figure 3.3. Interestingly, this $\mathcal{P}_{w(H)}$ does admit a cover by disjoint cubes with bottom vertices of cardinality at most 5; see Figure 3.4. However, covers corresponding to nondegenerate LP-type problems have to satisfy an additional condition, analogous to acyclicity in violator spaces. The proof in Section 3.4 verifies that every acyclic cover has a bottom vertex of cardinality 6 or larger. On the other hand, arbitrary covers by disjoint cubes correspond to nondegenerate violator spaces. One can thus say that $\mathcal{L}_2$ has a 5-dimensional nondegenerate refinement in the realm of violator spaces, but not in the realm of LP-type problems. However, the subsequent proof of Theorem 3.2 doesn't use acyclicity in any way and thus it applies equally well to violator spaces.

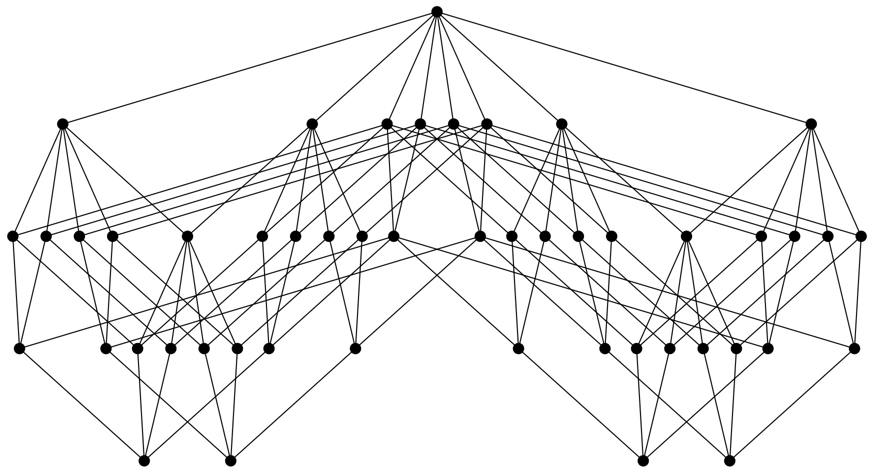

Figure 3.3. The poset $\mathcal{P}_{w(H)}$ for $m = 2$

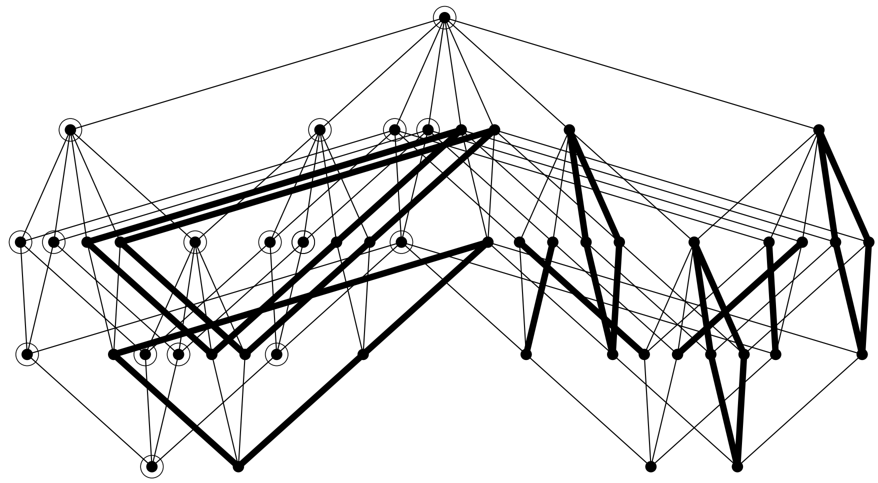

Figure 3.4. A covering of $\mathcal{P}_{w(H)}$ by disjoint cubes; a 4-dimensional cube is marked by circles around its vertices

3.3. Dimension may need to grow by any Δ

The basic strategy for the proof of Theorem 3.2 is simple. We suppose that the LP-type problem described above has a nondegenerate refinement of a small dimension and we want to arrive at a contradiction. The existence of the refinement implies that the poset $\mathcal{P}_{w(H)}$ can be covered by disjoint cubes. We count the number F_ℓ of vertices of the poset that have cardinality ℓ. We compare F_ℓ with the number of vertices of cardinality ℓ contained in the covering cubes. This gives a system of linear equations with variables corresponding to the numbers of cubes $[B, C]$ with given $|B|$ and $|C|$. Then we prove that this system of equations has no nonnegative real solution, therefore obtaining the contradiction.

Setting up the linear system. Let $\mathcal{L}_m = (H, w)$ be the join of m copies of the square example, as constructed above. Let us suppose that the poset $\mathcal{P} := \mathcal{P}_{w(H)} \subset 2^H$ can be covered by disjoint cubes $[B_j, C_j]$ with bottom vertices of size

$|B_j| \le D'$; here D' represents the combinatorial dimension of the refinement. Since $\dim(H,w) = D = 2m$, we have $2m \le |B_j| \le |C_j| \le |H| = 4m$ for all j. Let $x_{d,k}$ be the number of cubes with $|B_j| = 2m + d$ and $|C_j| = 2m + k$, where d, k satisfy $d \le \Delta := D' - 2m$ and $d \le k \le 2m$. A cube $[B_j, C_j]$ with $|B_j| = 2m + d$ and $|C_j| = 2m + k$ contains sets of cardinality $2m + \ell$ for $d \le \ell \le k$, and the number of sets of this cardinality in $[B_j, C_j]$ equals $\binom{k-d}{\ell-d}$ (this formula is actually valid for every ℓ if we adopt the convention that $\binom{a}{b} = 0$ for $b < 0$ or $b > a$). If we let

$$F(m,\ell) := \left|\left\{G \in \mathcal{P} : |G| = 2m + \ell\right\}\right|,$$

we get that the values $x_{k,d}$ have to satisfy the following system of linear equations:

$$\sum_{d=0}^{\Delta} \sum_{k=d}^{2m} \binom{k-d}{\ell-d} x_{d,k} = F(m,\ell), \qquad \ell = 0, 1, \ldots, 2m. \tag{3.1}$$

We are going to prove that with $\Delta = \lceil \varepsilon D \rceil$, where ε is a sufficiently small positive constant, this system of equations for variables $x_{k,d}$ has no *nonnegative real* solution, provided that m is sufficiently large.

Now we evaluate $F(m,\ell)$.

Lemma 3.6. *We have*

$$F(m,\ell) = \sum_s \binom{m}{s, \ell - 2s, m - \ell + s} 2^{m+\ell-3s},$$

with the sum being over all s with $0 \le 2s \le \ell$ and $s \ge \ell - m$ (here $\binom{n}{k_1,k_2,k_3} = \frac{n!}{k_1!k_2!k_3!}$ is the multinomial coefficient, $k_1 + k_2 + k_3 = n$).

Proof. We count the number of sets $G \in \mathcal{P}$ of cardinality $2m+\ell$. First we observe, reasoning as in the proof of Lemma 3.5, that a set $B \subseteq H$ is a basis of H in $\mathcal{L}_m$ if and only if each $B_i = B \cap H_i$ is a basis of H_i in (H_i, w_i). Hence the bases of H are the sets B with $B \cap H_i = \{a_i, c_i\}$ or $B \cap H_i = \{b_i, d_i\}$ for all $i = 1, 2, \ldots, m$. A set $G \subseteq H$ is in $\mathcal{P}$ if and only if it contains at least one of these bases; i.e., if it contains at least one of the pairs $\{a_i, c_i\}, \{b_i, d_i\}$ for every i.

For $G \in \mathcal{P}$ of cardinality $2m + \ell$ let

$$s_r := \left|\left\{i \in \{1, 2, \ldots, m\} : |G \cap H_i| = r\right\}\right|$$

for $r = 2, 3, 4$. We have $s_2 + s_3 + s_4 = m$ and $2s_2 + 3s_3 + 4s_4 = |G| = 2m + \ell$. Calculation shows that $s_2 = m - \ell + s_4$ and $s_3 = \ell - 2s_4$.

For counting the number of possible ways of choosing G, we first fix $s = s_4$. Then s_2 and s_3 are fixed as well, and there are $\binom{m}{s_2,s_3,s_4}$ ways to choose the indices i contributing to each s_r (in other words, to choose for which sets H_i does G take 2, 3, or 4 elements, respectively). Knowing that $|G \cap H_i| = 2$, there are two possibilities for $G \cap H_i$, for $|G \cap H_i| = 3$ we have 4 possibilities, and for $|G \cap H_i|$ there is just one possibility. Therefore, once $|G \cap H_i|$ has been fixed for all i, there are $2^{s_2} \cdot 4^{s_3} = 2^{m+\ell-3s_4}$ possibilities for G. Summation over $s = s_4$ yields the statement of the lemma; the conditions on the range of s in the summation correspond to the obvious restrictions $s_2, s_3, s_4 \geq 0$. ■

Unsolvability of the linear system. We recall that for finishing the proof of Theorem 3.2, it suffices to prove that for $\Delta := \lceil 2\varepsilon m \rceil$ and m sufficiently large, the linear system (3.1) has no nonnegative solution $x = (x_{d,k})_{d=0}^{\Delta}{}_{k=d}^{2m}$.

Before starting with the formal proof, which is a sequence of somewhat frightening calculations, we say a few words about how it was found. We started by testing the solvability for concrete values of parameters via linear programming. We used the function `LinearProgramming` in Mathematica, which uses arbitrary precision arithmetic and computes the solution exactly; this allowed us to deal with m up to about 1000 (other LP solvers we tried failed for large instances because of insufficient precision). By the Farkas lemma, the unsolvability is always witnessed by a linear combination of the equations that has nonnegative coefficients on the left-hand side and negative right-hand side. By minimizing the sum of absolute values of (suitably normalized) coefficients providing such a linear combination, we found that the unsolvability was witnessed, in all examples we tried, by a linear combination of only 3 of the equations. For simplifying the analytic approach, we then tried 3 *consecutive* equations, and found that such combinations work as well, provided that the index of the middle equation is chosen in a suitable range. These numerical results encouraged us to try finer and finer estimates, until we finally reached the following proof.

Proof of the unsolvability of (3.1). We set, somewhat arbitrarily, $t := \frac{1}{2}m$, assuming m even (we suspect that $t = \tau m$ for any fixed $\tau \in (0,1)$ would work, but we haven't checked). We prove that for sufficiently large m already the system of the three consecutive equations with $\ell = t-1$, t, and $t+1$ has no nonnegative solution. To this end, we find a linear combination of these three equations with suitable coefficients α, β, γ, such that the resulting equation has all coefficients on the left-hand side nonnegative, while the right-hand side is strictly negative. We normalize the coefficients so that $\beta = -1$. Then, explicitly, we need

$$\alpha\binom{k-d}{t-d-1} - \binom{k-d}{t-d} + \gamma\binom{k-d}{t-d+1} \geq 0 \qquad \text{for all } 0 \leq d \leq \Delta \tag{3.2}$$

and

$$\alpha F(m, t-1) - F(m,t) + \gamma F(m, t+1) < 0. \tag{3.3}$$

The choice of suitable α and γ turns out to be surprisingly subtle. Namely, we need to choose $\alpha = \alpha_0 + \alpha_1/t$ and $\gamma = \gamma_0 + \gamma_1/t$, where

$$\alpha_0 := \frac{\sqrt{10}+2}{4} \approx 1.29057, \quad \gamma_0 := \frac{\sqrt{10}-2}{6} \approx 0.193713$$

are uniquely determined real constants and α_1, γ_1 are constants in certain ranges. For concreteness we set $\alpha_1 := 1$ and $\gamma_1 := \frac{1}{8}$.

We get (3.2) from the following lemma:

Lemma 3.7. *There is a positive constant $\varepsilon > 0$ such that, with this choice of t, α, and γ, Equation (3.2) holds for all $d \leq \Delta = \lceil 2\varepsilon m \rceil$ and $k \geq d$, provided that m, and hence t, are sufficiently large.*

Proof. We use the substitution $x := t - d$ and $y := k - d$. We thus want to prove

$$\alpha\binom{y}{x-1} - \binom{y}{x} + \gamma\binom{y}{x+1} \geq 0.$$

For $y < x - 1$ all three terms are 0, therefore we may assume $y \geq x - 1$. We rewrite the left-hand side to

$$\frac{y!}{(x+1)!(y-x+1)!}\Big(\alpha x(x+1) - (x+1)(y-x+1) + \gamma(y-x+1)(y-x)\Big).$$

Let $f(\alpha, \gamma, x, y)$ be the expression in parentheses; we want to prove that it is non-negative.

Let us choose constants $\alpha_1' < \alpha_1$ and $\gamma_1' < \gamma_1$. Assuming that ε in the lemma is sufficiently small, we have d sufficiently small compared to x, and hence $\alpha = \alpha_0 + \alpha_1/(x + d) \geq \alpha_0 + \alpha_1'/x$ and $\gamma = \gamma_0 + \gamma_1/(x + d) \geq \gamma_0 + \gamma_1'/x$.

Since f is increasing in α and in γ (for the relevant x and y), it suffices to check that

$$f(\alpha_0 + \frac{\alpha_1'}{x}, \gamma_0 + \frac{\gamma_1'}{x}, x, y) \geq 0,$$

and we will verify this for all sufficiently large *real* x and all real y. One of the properties of α_0 and γ_0 needed here is $\alpha_0\gamma_0 = \frac{1}{4}$. Things can be simplified a little by the substitution $y = x(z + 1)$. Then $f(\alpha_0 + \alpha_1'/x, \gamma_0 + \gamma_1'/x, x, x(z + 1))$ is a polynomial in x and z. For x fixed it is a quadratic polynomial in z, and the coefficient of z^2 is $x^2/4\alpha_0 + \gamma_1' x > 0$ (this calculation and the subsequent ones were done using Mathematica). Therefore it has a unique minimum, which can be found by setting the first derivative according to z to 0. This minimum occurs at

$$z_0 = z_0(x) = \frac{x^2 + (1 - \gamma_0)x - \gamma_1'}{2x(\gamma_0 x + \gamma_1')}$$

(the expression was simplified using the property $\alpha_0\gamma_0 = \frac{1}{4}$). Substituting this into $f(\alpha_0 + \alpha_1'/x, \gamma_0 + \gamma_1'/x, x, z(x + 1))$ yields a function of x of the form

$$\frac{-\gamma_0 - 2\gamma_0^2 + 4\alpha_1'\gamma_0^2 + \gamma_1'}{4\gamma_0^2}x + O(1),$$

with the $O(\cdot)$ notation referring to $x \to \infty$. Calculation checks that the coefficient of x is a positive real number (for α_1' and γ_1' sufficiently close to α_1 and γ_1, respectively). Hence f is indeed positive for the considered values of the variables. ■

Remark. It is easy to check that if α, γ are positive *constants*, then the inequality $f(\alpha, \gamma, x, y) \geq 0$ holds for all y and all sufficiently large x if and only if $\alpha\gamma > \frac{1}{4}$. However, for such α and γ the equation (3.3) fails. We are thus forced to choose α and γ depending on x so that $\alpha\gamma \to \frac{1}{4}$ as $x \to \infty$.

We now proceed to establish (3.3). We set

$$Q(m,t,s) := \binom{m}{s, t-2s, m-t+s} 2^{m+t-3s},$$

so that $F(m,t) = \sum_s Q(m,t,s)$. First we look for the s maximizing $Q(m,t,s)$. Let

$$r(m,t,s) := \frac{Q(m,t,s)}{Q(m,t,s-1)} = \frac{(t-2s+1)(t-2s+2)}{8s(m-t+s)}$$

be the ratio of two consecutive terms. As a function of s it is decreasing, hence $Q(m,t,s)$ is maximum for the largest s with $r(m,t,s) \geq 1$.

We stick to our choice $t = \frac{1}{2}m$. It is more convenient to use t as a parameter instead of m. Let us write

$$\tilde{r}(t,s) := r(2t,t,s) \quad \text{and} \quad \tilde{Q}(t,s) := Q(2t,t,s),$$

and let us note that $m - t = t$. Let $\sigma := (\sqrt{10} - 3)/2 \approx 0.0811388$ be the positive root of the equation $(1-2\sigma)^2 = 8\sigma(1+\sigma)$, which is an asymptotic version of the equality $\tilde{r}(t,\sigma t) = 1$. Now for $s_0 := \lfloor \sigma t \rfloor$ the maximum of $\tilde{r}(t,s)$ is attained, and $\tilde{r}(t,s_0) = 1 + O(t^{-1})$. Next, we need an estimate on the rate of decrease of $\tilde{Q}(t, s_0 + a)$ as $|a|$ increases.

Lemma 3.8. *Let $c_0 := \frac{4}{1-2\sigma} + \frac{1}{\sigma} + \frac{1}{1+\sigma} \approx 18.0244$. Suppose that $a = o(t^{2/3})$. Then*

$$\frac{\tilde{Q}(t,s_0+a)}{\tilde{Q}(t,s_0)} = (1+o(1))e^{-c_0a^2/2t},$$

where $o(.)$ refers to $t \to \infty$ and the convergence is uniform in a.

Proof. We will be summing over $j = 1, 2, \ldots, a$ in the proof. Let us write $\xi = j/t$; thus $\xi = o(1)$. We have

$$\tilde{r}(t,s_0+j) = (1+O(t^{-1}))\frac{\tilde{r}(t,s_0+j)}{\tilde{r}(t,s_0)} = (1+O(t^{-1}))\frac{\left(1-\frac{2j}{t-2s_0+1}\right)\left(1-\frac{2j}{t-2s_0+2}\right)}{\left(1+\frac{j}{s_0}\right)\left(1+\frac{j}{t+s_0}\right)} =$$

$$= (1+O(t^{-1}))\frac{\left(1-\frac{2}{1-2\sigma}\xi\right)^2}{\left(1+\frac{1}{\sigma}\xi\right)\left(1+\frac{1}{1+\sigma}\xi\right)} = (1+O(t^{-1})+O(\xi^2))e^{-c_0\xi}.$$

Then

$$\ln\frac{\tilde{Q}(t,s_0+a)}{\tilde{Q}(t,s_0)} = \sum_{j=1}^{a}\ln\tilde{r}(t,s_0+j) = \left(\sum_{j=1}^{a} -\frac{c_0 j}{t}\right) + O\left(\frac{a}{t}\right) + O\left(\frac{a^3}{t^2}\right) =$$

$$= -\frac{c_0a^2}{2t} + O\left(\frac{a}{t} + \frac{a^3}{t^2}\right).$$

By exponentiation we get the statement of the lemma. ∎

Next, we consider the expression

$$\tilde{D}(t,s) := \alpha Q(m,t-1,s) - Q(m,t,s) + \gamma Q(m,t+1,s)$$

with $m = 2t$, $\alpha = \alpha_0 + \alpha_1/t$, and $\gamma = \gamma_0 + \gamma_1/t$ as above. The idea is to show that for s close to s_0 we have $\tilde{D}(t,s)$ negative, while for s further from s_0 it can be positive but it is sufficiently small compared to $|\tilde{D}(t,s_0)|$. Again, the calculation has to be done rather precisely in order to work.

Lemma 3.9. *Let us suppose that $a = o(t)$, and let $s_0 = \lfloor \sigma t \rfloor$ be as above. Then*

$$\tilde{D}(t,s_0+a) = \tilde{Q}(t,s_0+a)\cdot(1+o(1))\frac{C}{t}\left(\frac{c_1}{t}a^2 - 1 + o(1)\right),$$

where C is a certain positive constant whose value will not be important, $c_1 := (14584\sqrt{10} + 46192)/5877 \approx 15.7071$, the $o(\cdot)$ notation refers to $t \to \infty$, and the convergence is uniform in a.

Proof. Similar to the proof of Lemma 3.7 we rewrite

$$\tilde{D}(t,s) = \tilde{Q}(t,s)\cdot\frac{1}{2(t+1-2s)(t+s+1)}g(\alpha,\gamma,t,s),$$

with $g(\alpha,\gamma,t,s) := \alpha(t-2s+1)(t-2s) - 2(t-2s+1)(t+s+1) + 4\gamma(t+s)(t+s+1)$. With the constant σ as above, $\tilde{g}(t) := g(\alpha_0+\alpha_1/t, \gamma_0+\gamma_1/t, t, \sigma t)$ becomes a polynomial, which is a priori quadratic, but we chose α_0 and γ_0 so that the coefficient at t^2, which equals $14-5\sqrt{10}+(26-8\sqrt{10})\alpha_0+(11-2\sqrt{10})\gamma_0$, vanishes. (This, together with $\alpha_0\gamma_0 = \frac{1}{4}$, are the two conditions that uniquely determine α_0 and γ_0.) The coefficient of the linear term equals $-c_2 := (191 - 62\sqrt{10})/8 \approx -0.632652$, hence $g(\alpha,\gamma,t,s)$ is indeed negative and of order t for s sufficiently near to σt.

More quantitatively, expanding and simplifying gives

$$g\left(\alpha_0 + \frac{\alpha_1}{t}, \gamma_0 + \frac{\gamma_1}{t}, t, \sigma t + b\right) = -c_2 t + c_3 b^2 + O\left(\frac{b^2}{t} + b + 1\right)$$

with $c_3 := (14 + 5\sqrt{10})/3 \approx 9.93712$. For $a = b + \sigma t - s_0 = b + \sigma t - \lfloor \sigma t \rfloor \le b+1$ we then obtain

$$g\left(\alpha_0 + \frac{\alpha_1}{t}, \gamma_0 + \frac{\gamma_1}{t}, t, s_0 + a\right) = -c_2 t + c_3 a^2 + O\left(\frac{a^2}{t} + a + 1\right).$$

Therefore, using $a = o(t)$, we arrive at

$$\begin{aligned}\tilde{D}(t,s_0+a) &= \tilde{Q}(t,s_0+a)\cdot\frac{-c_2 t + c_3 a^2 + O(a^2/t + a + 1)}{2(t+1-2s)(t+s+1)}\\ &= \tilde{Q}(t,s_0+a)\cdot\frac{C}{t}\left(\frac{c_3 a^2}{c_2 t} - 1 + o(1)\right)\end{aligned}$$

as required. ∎

We are ready to prove (3.3). For our choice of α, γ, and t we have

$$\alpha F(m,t-1) - F(m,t) + \gamma F(m,t+1) = \sum_a \tilde{D}(t,s_0+a).$$

For concreteness let us set $a_0 := t^{3/5}$. We will show that

$$\sum_{|a|\le a_0} \tilde{D}(t,s_0+a) \le -\frac{\delta}{\sqrt{t}}\,\tilde{Q}(t,s_0)$$

for a constant $\delta > 0$. Now for $a > a_0$ we have $|\tilde{D}(t,s_0+a)| \le 3\tilde{Q}(t,s_0+a) \le 3\tilde{Q}(t,s_0+a_0)$, and the last expression is smaller that $\tilde{Q}(t,s_0)$ by a factor exponential in t. A similar argument applies for $a < -a_0$ and thus the sum over $|a| > a_0$ is negligible.

Combining Lemmas 3.8 and 3.9, we get that for $|a| \le a_0$ we have

$$\tilde{D}(t,s_0+a) = \tilde{Q}(t,s_0)\frac{C}{t}(1+\varphi_t(a))e^{-c_0a^2/2t}\left(\frac{c_1a^2}{t} - 1 + \psi_t(a)\right),$$

where $\varphi_t(a)$ and $\psi_t(a)$ are some functions converging to 0 as $t\to\infty$, uniformly in a.

We prove that

$$\sum_{|a|\le a_0}(1+\varphi_t(a))e^{-c_0a^2/2t}\left(1-\frac{c_1a^2}{t}-\psi_t(a)\right) = \Omega(\sqrt{t}\,). \tag{3.4}$$

Let us fix an arbitrarily small $\nu > 0$ and let us assume that t has been chosen so large that $|\varphi_t(a)| \le \nu$ and $|\psi_t(a)| \le \nu$ for all a. Then the left-hand side of (3.4) is bounded from below by

$$\sum_{|a|\le a_0} e^{-c_0a^2/2t}(1-c_1a^2/t) - \sum_{|a|\le a_0}(1+|\varphi_t(a)|)e^{-c_0a^2/2t}|\psi_t(a)| - \sum_{|a|\le a_0}|\varphi_t(a)|e^{-c_0a^2/2t}$$

$$\ge \sum_{|a|\le a_0} e^{-c_0a^2/2t}\left(1-\frac{c_1a^2}{t}\right) - 3\nu\sum_{|a|\le a_0} e^{-c_0a^2/2t}.$$

By basic properties of Riemann integration and uniform continuity arguments it is routine to check that both of these sums converge to the corresponding integrals as $t\to\infty$. So it suffices to bound from below

$$(1-3\nu)\int_{-a_0}^{a_0} e^{-c_0a^2/2t}\,\mathrm{d}a - \frac{c_1}{t}\int_{-a_0}^{a_0} a^2e^{-c_0a^2/2t}\,\mathrm{d}a.$$

Since $a_0^2/t = t^{1/5}\to\infty$ as $t\to\infty$ and the integrands decrease exponentially in a^2/t, we make only a negligible error by taking both integrals from $-\infty$ to ∞. We have

$$(1-3\nu)\int_{-\infty}^{\infty} e^{-c_0a^2/2t}\,\mathrm{d}a = (1-3\nu)\sqrt{2\pi t/c_0} \approx 0.590419\sqrt{t},$$

while

$$\frac{c_1}{t}\int_{-\infty}^{\infty} a^2 e^{-c_0a^2/2t}\,\mathrm{d}a = c_1\sqrt{2\pi}c_0^{-3/2}\sqrt{t} \approx 0.514513\sqrt{t}.$$

This finally proves (3.3) and concludes the proof of Theorem 3.2.

3.4. Dimension may need to grow by 2

In this section we examine $\mathcal{L}_2$, i.e., the join of $m = 2$ copies of the square example. We prove that already for this case, the growth of dimension when removing degeneracy must be at least 2. This result is quantitatively better than what is given by Theorem 3.2 for the case $\Delta = 2$, since Theorem 3.2 does not say anything about the value of m.

The proof consists of tedious case analysis.

Proposition 3.10. *Let $(H, w) := \mathcal{L}_2 = (H_1, w_1) * (H_2, w_2)$ be the join of two copies of the square example. Then (H, w) is an LP-type problem of dimension 4 and every its nondegenerate refinement (H, w') is of dimension at least 6.*

Proof. Proposition 3.5 readily confirms that (H, w) is an LP-type problem of dimension 4. To prove that there exists no nondegenerate refinement of (H, w) of dimension at most 5 we proceed by contradiction. Let us assume that such a nondegenerate refinement (H, w') does exist and let us see what happens.

We assume that $H_1 = \{a, b, c, d\}$ and $H_2 = \{t, x, y, z\}$.

We are going to analyze the possibilities how the poset $\mathcal{P}_{w(H)}$ can be covered by disjoint cubes. We will have to employ monotonicity in some places in the proof. The poset $\mathcal{P}_{w(H)}$ is displayed in Figure 3.3.

In the subsequent discussion we distinguish whether the basis B of H in the hypothesized refinement (H, w') has four or five elements. Note that since B is an element of $\mathcal{P}_{w(H)}$, it has at least four elements.

Case I: The basis of *H* has four elements.

There is quite a lot of symmetry in our poset $\mathcal{P}_{w(H)}$. Without loss of generality we can assume that the basis of H in (H, w') is the set $\{a, b, t, x\}$. The parts of $\mathcal{P}_{w(H)}$ with $w'(G) = w'(H)$ and $w'(G) \neq w'(H)$ are then displayed in Figure 3.5.

We focus on the sets G with $w'(G) \neq w'(H)$. Figure 3.6 depicts them after rearranging. For the sake of brevity let us omit the braces and commas when enumerating sets. From the sets $bcdyz$, $acdyz$, $cdxyz$, $cdtyz$ choose the one with the biggest value of w'; again there is enough symmetry to safely assume this is $bcdyz$.

Now we claim that $w'(bcdyz) > w'(cdyz)$. To prove it, assume for contradiction that $w'(bcdyz) = w'(cdyz)$. From maximality of $bcdyz$ and monotonicity of w' we get $w'(acdyz) \leq w'(bcdyz) = w'(cdyz) \leq w'(acdyz)$, that is, $w'(acdyz) = w'(bcdyz)$. In the same way we show that $w'(cdxyz) = w'(cdtyz) = w'(bcdyz)$. From the Cube lemma follows that $w'(cdyz) = w'(abcdtxyz)$, which contradicts the fact that the

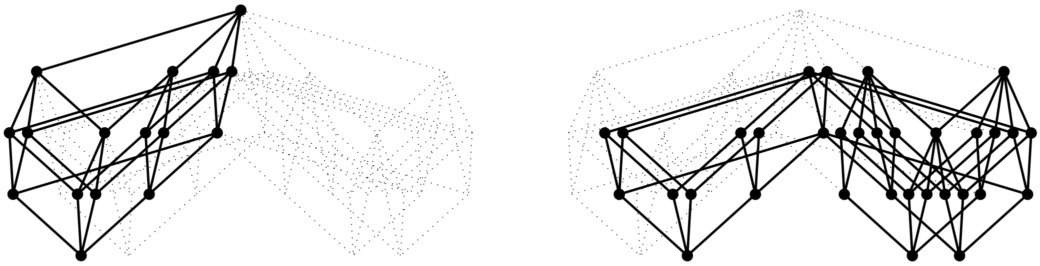

Figure 3.5. Sets G with $w'(G) = w'(H)$ (left) and $w'(G) \neq w'(H)$ (right)

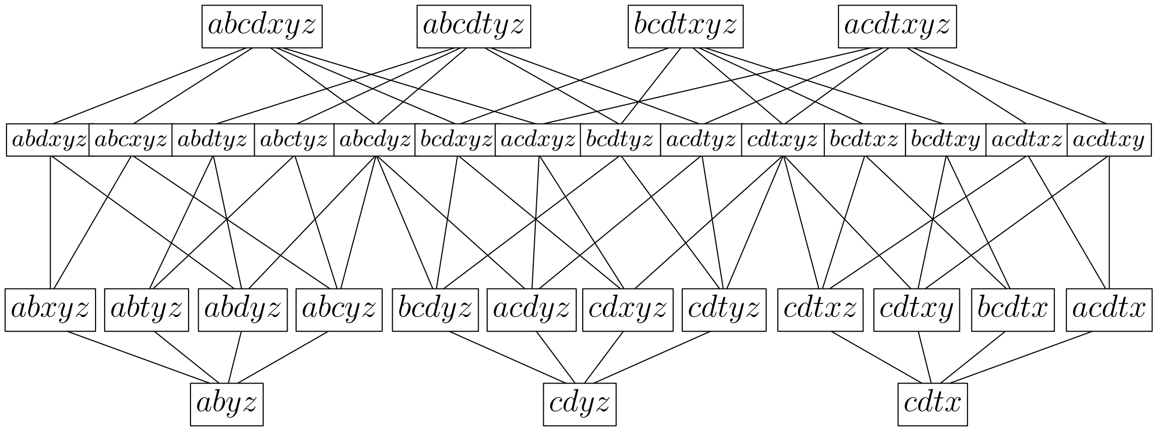

Figure 3.6. Sets G with $w'(G) \neq w'(H)$

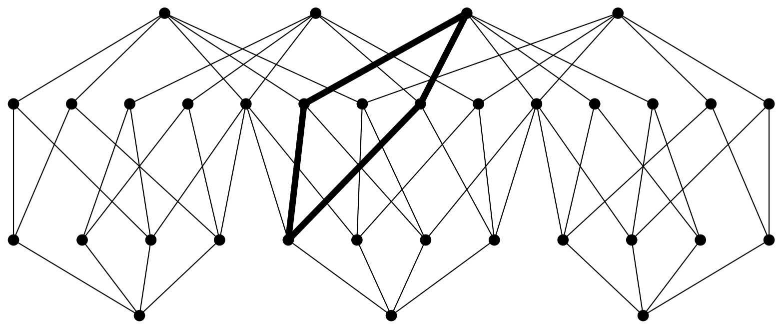

Figure 3.7. $w'(bcdyz) = w'(bcdtxyz)$

only basis of $H = abcdtxyz$ is $abtx$. Therefore $w'(bcdyz)$ is indeed strictly greater than $w'(cdyz)$.

Now, $bcdxyz$ is not a basis, since it has six elements; hence $w'(bcdxyz)$ is equal to the greatest $w'(G)$ of a proper subset G of $bcdxyz$. From the previous discussion follows that this maximum occurs for $bcdyz$; so $w'(bcdyz) = w'(bcdxyz)$. Similarly we get $w'(bcdyz) = w'(bcdtyz)$ and by the Cube lemma $w'(bcdyz) = w'(bcdtxyz)$. The present situation is demonstrated in Figure 3.7. The bold lines connect the sets with the same value of w.

Now from the sets $cdtxz$, $cdtxy$, $bcdtx$, $acdtx$ we choose Y to be the one with the greatest $w'(Y)$. We claim that $Y = acdtx$. Otherwise, if $Y = bcdtx$, from maximality we get $w'(bcdtx) = w'(bcdtxy) = w'(bcdtxz)$ and from the Cube lemma

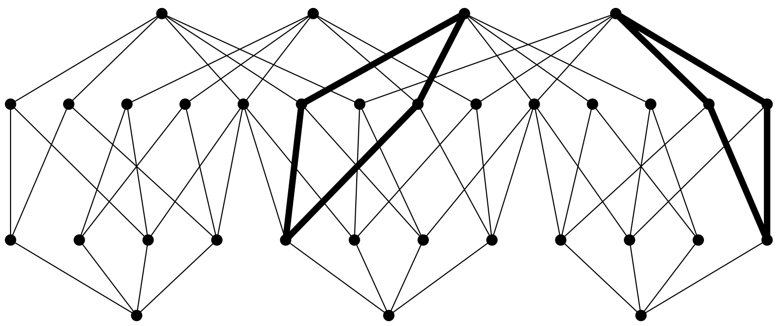

Figure 3.8. $w'(acdyz) = w'(acdtxyz)$

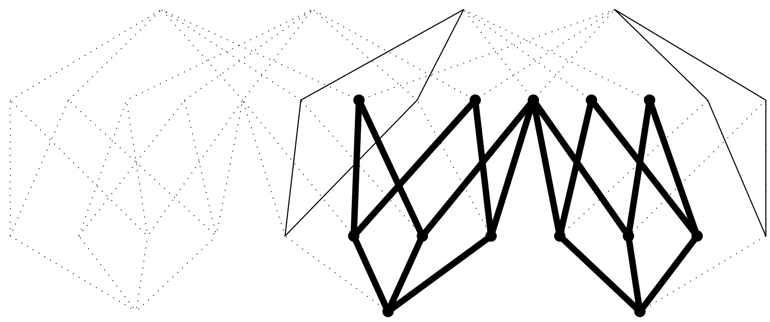

Figure 3.9. The part interesting for further steps

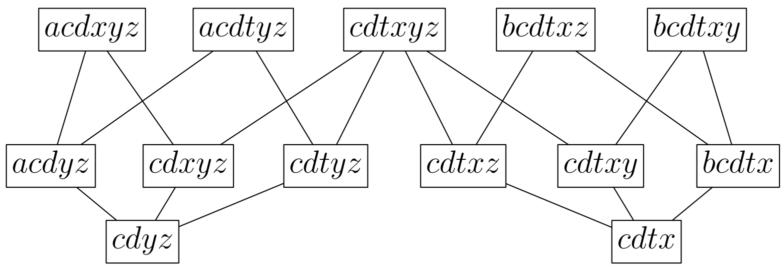

Figure 3.10. The part interesting for further steps, zoomed in

$w'(bcdtx) = w'(bcdtxyz)$, but we already have $w'(bcdtxyz) = w'(bcdyz)$, which is not possible. If on the other hand $Y = cdtxy$, we similarly obtain $w'(cdtxy) = w'(acdtxy) = w'(bcdtxy) = w'(abcdtxy) = w'(abtx)$, which is a contradiction. For $cdtxz$, we proceed analogously. Hence the greatest w' among the four sets is indeed attained by $acdtx$. By a similar reasoning as for $cdyz$ we prove that $w'(cdtx) < w'(acdtx)$, and from this we get that $w'(acdtx) = w'(acdtxyz)$. The current state of affairs is demonstrated in Figure 3.8.

Now we consider the poset $\mathcal{G}'$ marked in bold in Figure 3.9. Note that no two maximal vertices of the poset have a common dominating vertex not yet assigned to any cube. Thus we can restrict our attention to the marked part. For reference, we show the labels of the vertices in Figure 3.10.

Now we are again in a situation where we can enjoy symmetry. Consider the basis Z of $cdtxyz$. Without loss of generality assume that Z is some of the sets in

Figure 3.11. The basis of $cdtxyz$ has 4 elements (left), 5 elements (right)

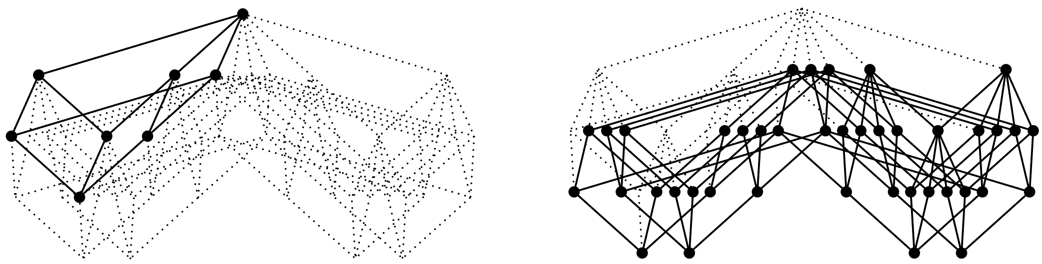

Figure 3.12. Sets G with $w'(G) = w'(H)$ (left) and $w'(G) \neq w'(H)$ (right)

the left part of the picture. We distinguish two cases depending on the number of elements of the basis.

If the basis is four-element (let us assume that it is $cdyz$), both the sets $acdxyz$ and $acdtyz$ have $acdyz$ as the basis, which is not possible. See the left part of Figure 3.11.

If the basis of $cdtxyz$ is five-element (without loss of generality $cdtyz$), we get that the basis of $acdtyz$ is $acdyz$ and now the basis of $acdxyz$ is $cdxyz$; see the right part of Figure 3.11. Now by repeated use of monotonicity we get

$$\begin{aligned} w(acdtyz) = w(acdyz) < w(acdxyz) = w(cdxyz) < \\ < w(cdtxyz) = w(cdtyz) < w(acdtyz), \end{aligned}$$

which is a contradiction.

So we checked that all the possibilities inevitably lead to a contradiction.

Case II: The basis of $\boldsymbol{H}$ has five elements.

There is enough symmetry that we do not lose generality if we assume that the basis of H in (H, w') is the set $abdtx$. The parts of $\mathcal{P}_{w(H)}$ with $w'(G) = w'(H)$ and $w'(G) \neq w'(H)$ are as in Figure 3.12.

Again we focus on the vertices G with $w'(G) \neq w'(H)$. We get Figure 3.13, which is the union of Figure 3.6 with the cube $[abtx, abctxyz]$.

Choose X to be the one of the sets $bcdyz$, $acdyz$, $cdxyz$, $cdtyz$ attaining the greatest value of w'. If $X = bcdyz$ or $X = acdyz$, we proceed exactly as we did in Case I. However, if $X = cdxyz$ or $X = cdtyz$, we have to analyze some new possibilities, since the new cube $[abtx, abctxyz]$ comes into play. Without loss of generality we assume that $X = cdtyz$.

For a while we argue as in Case I. We claim that $w'(cdyz) < w'(cdtyz)$; otherwise we get that $w'(cdyz) = w'(abcdtxyz)$, contradicting the assumption that $w'(abcdtxyz) = w'(abdtx)$. Furthermore $w'(cdtyz) = w'(bcdtyz)$ and $w'(cdtyz) = w'(acdtyz)$, and this implies that $w'(cdtyz) = w'(abcdtyz)$.

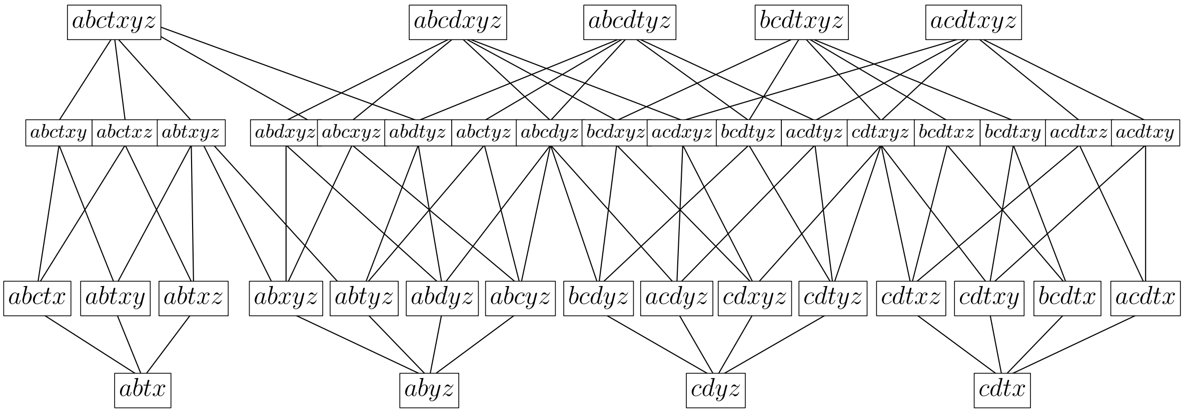

Figure 3.13. Sets G with $w'(G) \neq w'(H)$

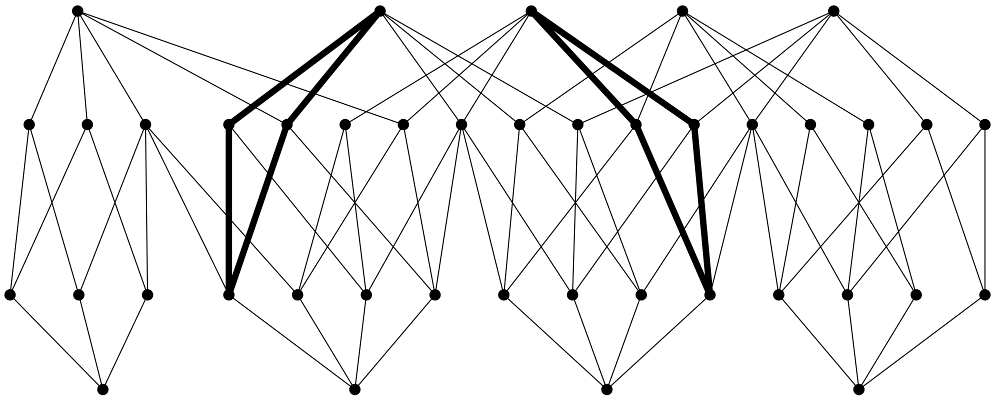

Figure 3.14. $w'(cdtyz) = w'(abcdtyz)$ and $w'(abxyz) = w'(abcdxyz)$

Now we consider the sets $abxyz$, $abtyz$, $abdyz$, $abcyz$. We conclude that $abxyz$ has the greatest w' of them, therefore $w'(abxyz) = w'(abcdxyz)$. The proof of this mimics the proof leading to Figure 3.8; now we yield Figure 3.14.

Here we leave the similarities to the Case I and we enter the Unknown. We distinguish what is the basis B of the set $abctxyz$. Employing the Cube lemma and recalling that the combinatorial dimension is 5, we see that only a few possibilities arise: $abtyz$, $abtx$, $abtxz$, $abtxy$ and $abctx$.

If $B = abtyz$ (see Figure 3.15), we get $w'(abctxyz) = w'(abtyz) < w'(abxyz) < w'(abctxyz)$ (the first inequality follows from maximality of $abxyz$), which is not possible. If $B = abtx$, $B = abtxy$, or $B = abtxz$ (see Figure 3.16), we get the same configuration as in Figure 3.10, which leads to a contradiction, as we already know. Therefore the basis of $abctxyz$ is $abctx$; see the left part of Figure 3.17.

Now we consider the basis C of $abtxyz$: if $C \neq abtyz$, we again refer to the configuration of Figure 3.10. If $C = abtyz$, we get $w'(abdtyz) = w'(abdyz)$ and $w'(abctyz) = w'(abcyz)$. Now we get the marked configuration in the right part of Figure 3.17; one can easily check that it is not possible to cover it by the cubes without breaking monotonicity.

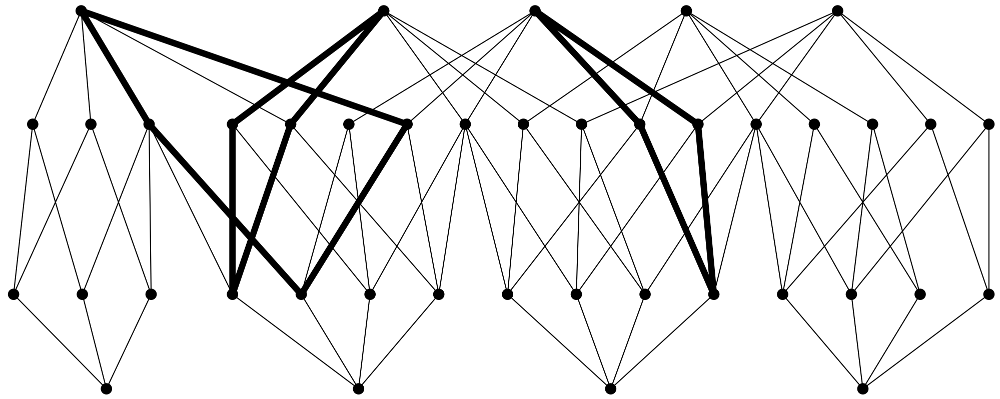

Figure 3.15. $w'(abctxyz) = w'(bcdtyz)$

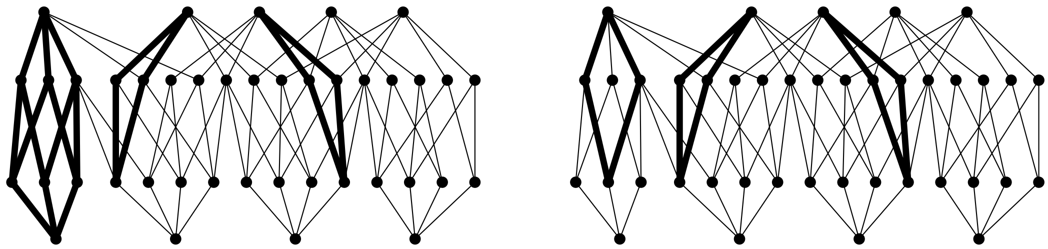

Figure 3.16. The basis of $abctxyz$ is $abtx$ or $abtxy$

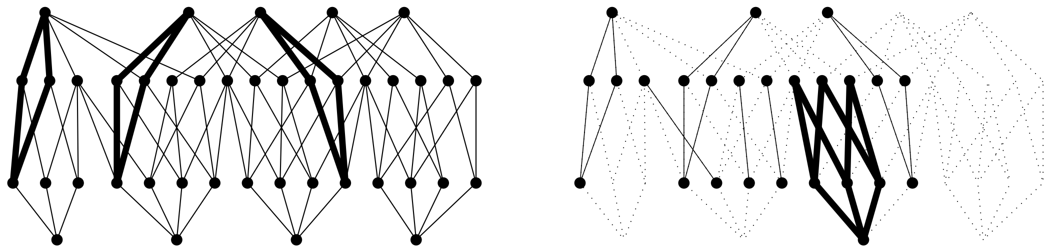

Figure 3.17. The last steps

So neither in Case II we managed to cover the poset $\mathcal{P}_{w(H)}$ by cubes with the bottom vertices of cardinality at most five in a way satisfying monotonicity. Thus we can conclude that no nondegenerate refinement of (H, w) of combinatorial dimension at most 5 exists. ∎

We remark that a 6-dimensional refinement can be constructed easily by covering the poset $\mathcal{P}_{w(H)}$ by cubes, or, more systematically, by joining two copies of 3-dimensional refinement of the square example.

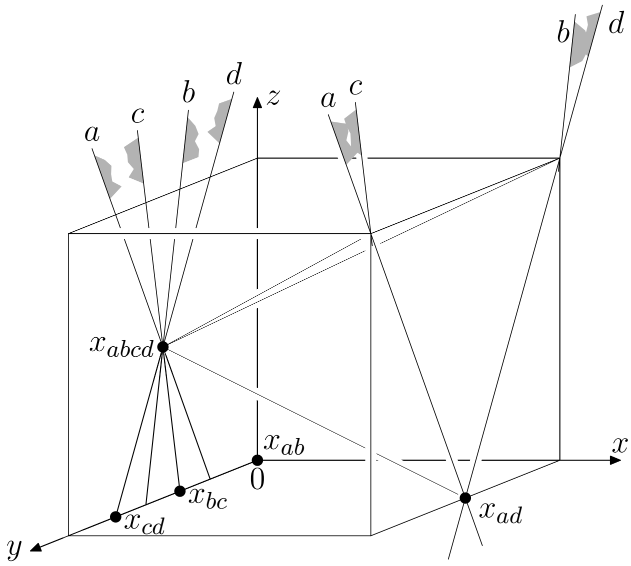

Figure 3.18. A linear program in $\mathbb{R}^3$ essentially representing the square example

3.5. A geometric representation by a linear program

It turns out that an LP-type problem $\hat{\mathcal{L}}_m = (H, \hat{w})$ similar to $\mathcal{L}_m$ that can also be used as an example establishing Theorem 3.2, can be represented as a linear program. To see that our proof of Theorem 3.2 works for $\hat{\mathcal{L}}_m$ as well, it suffices to verify that its poset $\mathcal{P}_{\hat{w}(H)}$ of maximum-weight sets is isomorphic to $\mathcal{P}_{w(H)}$ of $\mathcal{L}_m$. This follows from the discussion below.

We begin by setting up the following linear program with variables x, y, z ($\varepsilon > 0$ is a very small positive real number):

$$\begin{array}{rr} \multicolumn{2}{c}{\text{minimize } z + \varepsilon y + \varepsilon^2 x \text{ subject to}} \\ a: & x + 4y - 2z \leq 1 \\ b: & 3x + 8y + 2z \leq 5 \\ c: & 3x - 8y + 2z \leq -3 \\ d: & -x - 4y - 2z \leq -3 \\ & x, y, z \geq 0. \end{array}$$

The corresponding LP-type problem $(H_0, \hat{w}_0)$ has the set $H_0 = \{a, b, c, d\}$ of four constraints corresponding to the four inequalities of the linear program. The value $\hat{w}_0(G)$ of a subset $G \subseteq H_0$ is the minimum of the linear program where the constraints of $H_0 \setminus G$ have been deleted (we stress that the implicit nonnegativity constraints $x, y, z \geq 0$ are always present, even for $G = \emptyset$). In this way, $\hat{w}_0(G)$ is well defined for every G.

The linear program is illustrated in Figure 3.18. For better visualization, the picture shows the unit cube $[0, 1]^3$ and intersections of the bounding planes of the constraints with the planes $x = 0$ and $x = 1$. The minimum of the linear programs containing both the constraints a and c or both the constraints b and d is attained at the point $x_{abcd} = (0, \frac{1}{2}, \frac{1}{2})$; thus, $\hat{w}_0(H_0) = \frac{1}{2}$. It can be checked that for every subset G of constraints containing neither $\{a, c\}$ nor $\{b, d\}$, the minimum is attained at a point with $z = 0$, and thus with $\hat{w}_0 < \frac{1}{2}$ (the picture shows the minima for all G of cardinality 2). Thus $\hat{\mathcal{L}}$ is an LP-type problem of combinatorial dimension 2 with the poset $\mathcal{P}_{\hat{w}_0(H_0)}$ isomorphic to $\mathcal{P}_{w_0(H_0)}$ of the square example.

Next, we observe that if (H, w) is an LP-type problem corresponding to a linear program with variables $x_1, \ldots, x_n$ and with objective $\min \sum c_i x_i$, and (H', w') is an LP-type problem corresponding to a linear program with variables $x'_1, \ldots, x'_m$ and with objective $\min \sum c'_i x'_i$, then the join $(H, w) * (H', w')$ corresponds to the linear program obtained by putting the constraints of both linear programs aside and setting the objective $\min(\sum c_i x_i + \sum c'_i x'_i)$. Indeed, it suffices to check that the value function in $(H, w) * (H', w')$ coincides with the value function obtained from the combined linear program, and this is immediate. In particular, the m-fold join $\hat{\mathcal{L}}_m$ of m disjoint copies of $(H_0, \hat{w}_0)$ corresponds to the following linear program in $3m$ variables:

$$\text{minimize } \textstyle\sum_{i=1}^{m}(z_i + \varepsilon y_i + \varepsilon^2 x_i) \text{ subject to}$$

$$\left.\begin{aligned} x_i + 4y_i - 2z_i &\le 1 \\ 3x_i + 8y_i + 2z_i &\le 5 \\ 3x_i - 8y_i + 2z_i &\le -3 \\ -x_i - 4y_i - 2z_i &\le -3 \\ x_i, y_i, z_i &\ge 0 \end{aligned}\right\} i = 1, 2, \ldots, m.$$

We could have presented the example for Theorem 3.2 in this form, but we find the abstract construction of join more transparent.

3.6. Removing degeneracy in presence of minus infinity

In this section we examine the role of minus infinity in the context of removing degeneracy. We present a very simple proof of $-\infty$ version of Theorem 3.2. On the other hand we argue that allowing for $-\infty$ has a considerable influence on the dimension; more precisely, removing $-\infty$ from LP-type problems may require unbounded dimension growth.

Proposition 3.11. *For every $D \ge 2$ there exists a D-dimensional degenerate LP-type problem (H, w) that does not have any nondegenerate refinement of combinatorial dimension smaller than $2D - 1$.*

Proof. Let $D \ge 2$ be a fixed positive integer. To construct the LP-type problem (H, w), we put $H := \{a_1, \ldots, a_D,\ b_1, \ldots, b_D\}$ and we define

$$w(G) := \begin{cases} 0 & \text{if } \{a_1, \ldots, a_D\} \subseteq G \text{ or } \{b_1, \ldots, b_D\} \subseteq G, \\ -\infty & \text{otherwise.} \end{cases}$$

It is straightforward to check that (H, w) is an LP-type problem. Its bases are $\emptyset$, $\bar{A} := \{a_1, \ldots, a_D\}$, and $\bar{B} := \{b_1, \ldots, b_D\}$, hence $\dim(H, w) = D$.

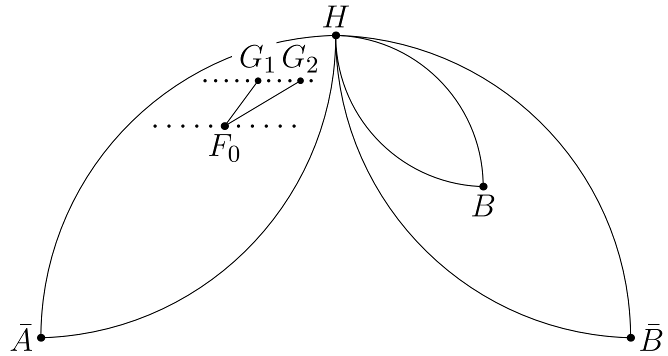

Figure 3.19. The proof of Proposition 3.11

Let (H, w') be some nondegenerate refinement of (H, w) and for contradiction let us assume that $\dim(H, w') \le 2D - 2$. Let B be the basis of H in the refined problem. We have $w(H) = w(B)$, therefore $\bar{A} \subseteq B$ or $\bar{B} \subseteq B$; without loss of generality we assume the latter. Let

$$M := \max\left\{w'(F) : F \subseteq H, \bar{A} \subseteq F, |F| = 2D - 2\right\}$$

and let F_0 be a set where this maximum is attained; see Figure 3.19. We have $F_0 = H \setminus \{b_k, b_\ell\}$ for some k, ℓ. Consider the sets $G_1 := F_0 \cup \{b_k\}$ and $G_2 := F_0 \cup \{b_\ell\}$; note that $G_1 \cup G_2 = H$. Since we assume $\dim(H, w') \le 2D - 2$, we have $w'(G_i) = \max\{w'(F) : F \subseteq G_i, |F| \le 2D - 2\} = M$. This gives $w'(F_0) = w'(G_1) = w'(G_2) = M$. Using locality we get $w'(H) = M$. On the other hand, from nondegeneracy we have $w'(H) > M$, which is a contradiction. ∎

It is not clear whether the construction above can be modified to work without $-\infty$, since removing $-\infty$ may be comparably hard to removing degeneracy. This claim is justified by the following proposition demonstrating that removing $-\infty$ even from 2-dimensional problems may need large dimension.

Proposition 3.12. *There are 2-dimensional LP-type problems with $-\infty$ that do not have (possibly degenerate) refinements without $-\infty$ in any fixed dimension.*

Proof. Let $D \ge 2$ be a fixed positive integer. We construct an LP-type problem (H, w) with $\dim(H, w) = 2$ such that any its refinement without $-\infty$ has dimension at least D.

Put $H := \{a_1, \ldots, a_D,\ b_1, \ldots, b_D\}$ and define

$$w(G) := \begin{cases} 0 & \text{if } a_i, b_i \in G \text{ for some } i \in \{1, \ldots, D\}, \\ -\infty & \text{otherwise.} \end{cases}$$

This (H, w) is an LP-type problem. Its bases are $\emptyset$ and $\{a_i, b_i\}$ for $i = 1, \ldots, D$, hence $\dim(H, w) = 2$.

Now let (H, w') be any refinement of (H, w) with $w'(\emptyset) \ne -\infty$. We prove that $\dim(H, w') \ge D$. We proceed by contradiction. Assume that $\dim(H, w') \le D - 1$. Let

$$M := \max\left\{w'(F) : F \subseteq H, |F| = D - 1, w(F) = -\infty\right\}$$

and let F_0 be a set where this maximum is attained. Since $|F_0| = D - 1$, there is some k such that both a_k, b_k are missing in F_0. Let $G_a := F_0 \cup \{a_k\}$ and $G_b := F_0 \cup \{b_k\}$. Since $|G_a| = |G_b| = D$, the sets G_a and G_b are not bases in (H, w'), hence $w'(G_a) = \max\{w'(F) : F \subseteq G_a, |F| \leq D - 1\} = M$, and the same holds for G_b. Now locality for F_0, G_a, and b_k gives $w'(F_0 \cup \{a_k, b_k\}) = w'(F_0)$. On the other hand, $w(F_0 \cup \{a_k, b_k\}) = 0 > -\infty = w(F_0)$, which contradicts the fact that w' refines w. ■

3.7. Degeneracy in 2-dimensional problems

As we mentioned above, we do not know whether every 2-dimensional LP-type problem has a nondegenerate refinement of dimension bounded by a universal constant. In this section we study degeneracy of 2-dimensional problems. The results presented here form a bunch of observations and examples rather than a compact theory.

We have seen that the problem of removing degeneracy is closely related to properties of certain posets. Here we investigate what do the posets look like for 2-dimensional LP-type problems. We describe a connection of these posets and graphs. We give some results stating what graphs we get.

Basis graphs. Let (H, w) be a degenerate 2-dimensional LP-type problem without $-\infty$. For simplicity assume that its degeneracy is caused by H having more than one basis. Furthermore assume that the empty set and all single-element sets have smaller values of w than every two-element set. Under these conditions, every basis of H has size exactly 2. We want to describe the poset $\mathcal{P}_{w(H)}$, i.e., to characterize sets $G \subseteq H$ with $w(G) = w(H)$. Since $\dim(H, w) = 2$, we have

$$w(G) = \max\left\{w(F) : F \subseteq G, |F| \leq 2\right\} \quad \text{for every } G \subseteq H.$$

This implies that $w(G) = w(H)$ if and only if G has some two-element subset F such that $w(F) = w(H)$. This leads us to constructing a graph $\mathcal{B} = (H, E)$ whose vertices are the constraints of the LP-type problem, and where the vertices $g, h \in H$ are connected by an edge if and only if $w(g, h) = w(H)$. Now we can distinguish the sets $G \subseteq H$ with $w(G) = w(H)$ using only the graph $\mathcal{B}$: we have $w(G) = w(H)$ if and only if the subgraph induced by G contains an edge. Note that $E \neq \emptyset$ (this is an easy consequence of the trivial equality $w(H) = w(H)$).

The graph $\mathcal{B}$ is meaningful for degenerate LP-type problems where H has more than one basis, but its construction does make sense for other 2-dimensional LP-type problems as well; $\mathcal{B}$ then contains exactly one edge.

We call $\mathcal{B}$ the *basis graph of* (H, w). We say that a graph (V, E) is a *basis graph*, if it is isomorphic to a basis graph of some 2-dimensional LP-type problem. If a graph (V, E) is not a basis graph, we call it *nonbasis*.

Observe that given the basis graph $\mathcal{B}$ of an LP-type problem (H, w), we can determine the number $F(k)$ of k-element sets $G \subseteq H$ with $w(G) = w(H)$. In total there are $\binom{|H|}{k}$ sets of size k, and the sets G with $w(G) = w(H)$ are exactly those containing an edge. This gives

$$F(k) = \binom{|H|}{k} - \operatorname{ind}(\mathcal{B}, k),$$

where $\operatorname{ind}(\mathcal{B}, k)$ stands for the number of independent sets of size k in the graph $\mathcal{B}$.

If it is the case that removing degeneracy in 2-dimensional LP-type problems may require unbounded dimension, we believe that the following method might help to prove it. We suggest an approach similar to the proof of Theorem 3.2 above. We try to find a degenerate LP-type problem (H, w) that does not admit any nondegenerate refinement of combinatorial dimension $2 + \Delta$. To prove that such a refinement does not exist, we would construct a system of equations and then prove that it has no solutions. We get the system as follows: according to the Cube lemma 3.3, the refinement gives a partitioning of the poset $\mathcal{P}_{w(H)}$ into disjoint cubes. We compare the number of sets of size $2 + \ell$ covered by the cubes with the total number $F(2 + \ell)$ of the sets of this size contained in $\mathcal{P}_{w(H)}$. This gives the system of equations

$$\sum_{d=0}^{\Delta} \sum_{k=d}^{|H|-2} \binom{k-d}{\ell-d} x_{d,k} = F(2+\ell), \qquad \ell = 0, 1, \ldots, \Delta,$$

with nonnegative variables $x_{d,k}$ corresponding to the number of cubes $[B, C]$ with $|B| = 2 + d$ and $|C| = 2 + k$.

To proceed in the described way, we would like to be able to construct examples of basis graphs, in particular such ones for which we can exactly determine the number of independent sets of each size.

Equivalent description of basis graphs. We present an alternate definition of basis graphs without the terminology of LP-type problems. It is based on a characterization of mappings $\bar{w} : \binom{H}{2} \to \mathbb{R} \cup \{\infty\}$ that can be obtained by restricting the weight mapping of some LP-type problem (H, w) to the family of 2-element subsets of H.

Let V be a finite set. Let a mapping $\bar{w} : \binom{V}{2} \to \mathbb{R} \cup \{\infty\}$ assign real or infinite weights to pairs of elements of V. For simplicity of notation we write $\bar{w}(a, b)$ for $\bar{w}(\{a, b\})$. Let us agree on putting $x < \infty$ for all $x \in \mathbb{R}$.

We say that the mapping $\bar{w}$ is *good* if both of the following conditions are satisfied:

(i) In any four-element subset of V, if there is a strictly maximal value of $\bar{w}$, the second greatest value is not attained on the pair consisting of the two remaining elements. More formally, for every $a, b, c, d \in V$ and $T \in \mathbb{R}$ with $\bar{w}(a, b) \leq T$, $\bar{w}(a, c) \leq T$, $\bar{w}(a, d) \leq T$, $\bar{w}(b, c) \leq T$, $\bar{w}(b, d) \leq T$, and $\bar{w}(c, d) > T$, we have a strict inequality $\bar{w}(a, b) < T$. In yet other words, the ordering *others* $\leq \bar{w}(a, b) < \bar{w}(c, d)$ is prohibited.

(ii) For some $a, b \in V$ we have $\bar{w}(a, b) = \infty$.

With this, we can give the following characterization of basis graphs:

Proposition 3.13. *A graph $G = (V, E)$ is a basis graph if and only if there exists a good mapping $\bar{w}$ such that the edges connect exactly the pairs of vertices with infinite value of $\bar{w}$:*

$$E = \big\{\{a, b\} : \bar{w}(a, b) = \infty\big\}.$$

Proof. Given a good mapping $\bar{w}$, let m, M be numbers satisfying $m < \bar{w}(f, g) < M$ for all $f, g \in G$ with a finite value of $w(f, g)$. For $G \subseteq H$ we define

$$w(G) := \max\big\{w(F) : F \subseteq G, |F| = 2\big\} \quad \text{if } |G| \geq 2 \text{ and the maximum is finite,}$$

$w(G) := M$ if the maximum is ∞, $w(\emptyset) := m-1$, and $w(\{g\}) := m$ for every single-element set. We claim that (V, w) is an LP-type problem; then it is straightforward to see that its basis graph is exactly G. Monotonicity follows from the definition of w. To get locality we proceed by contradiction: assume that for $F \subset G$ we have $w(F) = w(G) = w(F \cup \{h\}) \neq w(G \cup \{h\})$. First notice that $|F| \geq 2$, otherwise we cannot achieve $w(F) = w(G)$. By the definition of w, we have some $f, f' \in F$ with $w(f, f') = w(F)$, and some $g \in G$ with $w(g, h) = w(G \cup \{h\})$. These vertices f, f', g, h break the condition (i) in the definition of a good mapping.

In the other direction assume that (H, w) is a 2-dimensional LP-type problem. We claim that $\bar{w}$ defined as

$$\bar{w}(f, g) := \begin{cases} w(\{f, g\}) & \text{if } w(\{f, g\}) \neq w(H), \\ \infty & \text{if } w(\{f, g\}) = w(H) \end{cases}$$

is a good mapping. The condition (ii) is satisfied since $\dim(H, w) = 2$. To prove (i), let $F := \{a, b\}$, $G := \{a, b, c\}$, and $h := d$, and invoke locality. ∎

Independent sets. For a graph G, let $\alpha(G)$ be the size of the largest independent set. The following observation asserts that in our quest we are interested in graphs with large α.

Observation 3.14. *Let (H, w) be an LP-type problem with a basis graph $\mathcal{B} = (H, E)$. Then there exists a nondegenerate refinement (H, w') of (H, w) of combinatorial dimension at most $\alpha(\mathcal{B}) + 1$.*

Proof. First observe that every set $G \subseteq H$ of size larger than $\alpha(\mathcal{B})$ contains an edge, hence $w(G) = w(H)$. To each set of size at most $\alpha(\mathcal{B})$ we assign its own weight; then we need to break the ties for sets of size $\alpha(\mathcal{B})+1$. Essentially, we order these sets lexicographically.

Formally, assume that $H = \{h_1, \ldots, h_n\}$ and let ε represent a suitable positive real number. We define

$$w'(G) := \max\Big\{\sum_{h_i \in P} \varepsilon^i : P \subseteq G, |P| \leq \alpha(\mathcal{B}) + 1\Big\}.$$

Now for every proper subset F of some G, we have $w'(F) < w'(G)$ whenever $|F| \leq \alpha(\mathcal{B})$. On the other hand, if $|F| > \alpha(\mathcal{B})$, we have $w(F) = w(H) = w(G)$. This proves that w' refines w.

Figure 3.20. Nonbasis graphs with $|V| = 6$ and $|V| = 7$

Figure 3.21. Nonbasis graphs with $|V| = 8$

To prove nondegeneracy of (H, w'), assume that all the expressions $\sum_{h_i \in P} \varepsilon^i$ are distinct for distinct sets P of size at most $\alpha(\mathcal{B}) + 1$; we can achieve this by setting ε to be transcendent.

Every $G \subseteq H$ has a basis in (H, w') of size at most $\alpha(\mathcal{B}) + 1$: take a set where the maximum in the definition of w' is attained. Hence $\dim(H, w') \leq \alpha(\mathcal{B}) + 1$.

Monotonicity of w' is straightforward from its definition. To prove locality we again use that ε is transcendent. Now if $w(F) = w(F \cup \{h_k\}) = w(G)$ with $h_k \notin F$, the set P achieving the maximum in the definition of $w(F)$ achieves the maximum for $F \cup \{h_k\}$ and $w(G)$ as well, therefore $\varepsilon^k < \varepsilon^i$ and $\varepsilon^j < \varepsilon^i$ for every $i \in P$ and $j \in G \setminus P$. Therefore P achieves the maximum for $G \cup \{h_k\}$ too, hence $w(G \cup \{h_k\}) = w(G)$, which proves locality. ■

Examples of basis and nonbasis graphs. As we observed above, a graph with no edges is not a basis graph. However, we consider this to be a trivial case.

An induced subgraph $G' = (H', E')$ of a basis graph $\mathcal{B} = (H, E)$ with $E' \neq \emptyset$ is a basis graph. This follows easily from the definition of basis graph by restricting w to $2^{H'}$. This means that if a graph G contains a nontrivial nonbasis graph as an induced subgraph, G is not a basis graph.

The graphs in Figures 3.20 and 3.21 are nontrivial examples of nonbasis graphs. A computer check has shown that there are no more minimal nonbasis graphs with 8 or fewer vertices.

A nontrivial infinite class of basis graphs is formed by complete multipartite graphs. Let $H = H_1 \dot\cup \cdots \dot\cup H_k$ with $k \geq 2$. For $a, b \in H$ define $w(a, b) := \infty$ when $a \in H_i, b \in H_j$ for some $i \neq j$, and $w(a, b) := 0$ when $a, b \in H_i$ for some i. One can easily check that such a w is a good mapping. Unfortunately, this class of graphs does not help in our quest, since the corresponding LP-type problems have 2-dimensional nondegenerate refinements. We can obtain them by a lexicographic perturbation: we define

$$w'(G) := \begin{cases} \max\left\{ \sum_{h_j \in P} \varepsilon^j : P \subseteq G, |P| \leq 2 \right\} & \text{for } G \subseteq H_i \text{ for some } i, \\ \max\left\{ \sum_{h_j \in P} \varepsilon^j : P \subseteq G, |P| \leq 2 \right\} + 100 & \text{otherwise.} \end{cases}$$

It is routine to check that under suitable restrictions on ε, the problem (H, w') is indeed a 2-dimensional nondegenerate refinement of (H, w). We proceed along the lines of the proof of Observation 3.14, and when checking locality we distinguish whether $F \subseteq H_i$ for some i. We omit the details.

Adding edges to a basis graph. If we take a basis graph arising from a good mapping w and we add edges connecting pairs of vertices with high values of w, we get again a basis graph. More precisely, if we have a good mapping w and $l \in \mathbb{R}$, the mapping $w_{(l)}$ given as follows is good:

$$w_{(l)}(G) := \begin{cases} w(G) & \text{if } w(G) < l, \\ \infty & \text{if } w(G) \geq l. \end{cases}$$

The condition (ii) in the definition of good mapping is clearly satisfied. To prove the condition (i) we assume for contradiction that $w_{(l)}(c, d) > w_{(l)}(a, b) \geq \mathit{others}$; this implies the same relations for w, which is not possible.

A necessary condition for basis graphs. Let w be a good mapping giving rise to a basis graph $\mathcal{B} = (V, E)$. Assume that $\mathcal{B}$ is not a complete graph. Let M be the maximum finite value of w:

$$M := \max\big\{w(a, b) : a, b \in V, w(a, b) < \infty\big\}.$$

Let a, b be any pair of vertices where M is attained. Consider any edge $\{x, y\} \in E$ with $x, y \in V \setminus \{a, b\}$. We have $w(x, y) = \infty$, which is certainly the maximum on the set $S = \{a, b, x, y\}$. If this maximum is unique, in other words if there is no other edge in S, we break the first rule in the definition of a good mapping w.

This proves that in every good graph $\mathcal{B} = (V, E)$ that is not complete, there exists a pair of vertices $\{a, b\}$ that does not form an edge, such that for every edge $\{x, y\} \in E$ with a, b, x, y distinct, at least one of $\{a, x\}$, $\{a, y\}$, $\{b, x\}$ or $\{b, y\}$ is an edge.

Let us say that a pair of vertices $\{a, b\}$ in a graph (V, E) forms a *nonedge*, if it does not form an edge. Furthermore we say that a nonedge $\{a, b\}$ is *close* to an edge $\{x, y\}$, if

- either $a = x$, $a = y$, $b = x$, or $b = y$,
- or at least one of $\{a, x\}$, $\{a, y\}$, $\{b, x\}$, or $\{b, y\}$ is an edge;

in other words, if the graph distance between the sets $\{a, b\}$ and $\{x, y\}$ is at most 1. With this terminology we can summarize the observation proved above as follows:

Proposition 3.15. *Let $\mathcal{B}$ be a basis graph distinct from a complete graph. Then $\mathcal{B}$ contains a nonedge $\{a, b\}$ that is close to every edge.*

In other words, let G be a noncomplete graph in which for every nonedge $\{a, b\}$ we have some edge that is not close to $\{a, b\}$; then G is nonbasis. All the examples of nonbasis graphs presented in Figures 3.20 and 3.21 are of this nature.

An algorithm for finding a mapping $\boldsymbol{w}$. Actually, the proof of Proposition 3.15 shows a stronger thing: the nonedges close to all edges are the only possible places where the maximum finite value of w can be attained. With this we get an algorithm that for a given basis graph G finds an associated good mapping w.

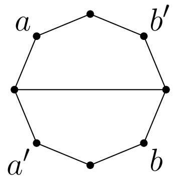

Figure 3.22. A basis graph where the rule (i) fails

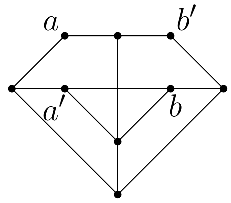

Figure 3.23. A basis graph where the rule (ii) fails

Algorithm 3.16 (nondeterministic).
Input: *Graph G.*
Output: *Good mapping w such that G is the basis graph arising from (H, w).*

For all edges $\{x, y\}$ put $w(x, y) := \infty$. Set $M := 1000$.
`REPEAT`
  Let S be the set of all nonedges $\{a, b\}$ that are close to all edges
  `IF` $S = \emptyset$ `THEN` *exit unsuccessfully*
  `IF` $|S| > 1$ `THEN` *wisely choose a nonempty subset $S' \subseteq S$*
  `FOR` $\{a, b\} \in S'$
    Set $w(a, b) := M$ and connect $\{a, b\}$ by an edge
  Set $M := M - 1$
`UNTIL` *no nonedges remain*
`RETURN` w

Unfortunately, the wisdom required to choose S' seems to be nontrivial. One may come up with the following reasonably looking rules:

(i) choose S' to be a suitable one-element subset of S,
(ii) choose $S' := S$.

However, both for (i) and for (ii) we can construct a basis graph for which the rule fails (that is, it gets stuck so that the algorithm cannot add any more edges and exits in the very next iteration of the cycle).

First consider the rule (i). Let G be the graph in Figure 3.22. The set S found in the first iteration of the cycle consists of nonedges ab and $a'b'$. By choosing any of these nonedges and converting it into an edge we get a nonbasis graph (see Figure 3.21), therefore the algorithm gets stuck. However, if we choose both of these nonedges, we can arrive to a suitable w.

Now consider the rule (ii). Let G be the graph in Figure 3.23. In the first iteration of the cycle the algorithm finds the nonedges ab and $a'b'$. After choosing both of then and converting them to edges, the algorithm gets stuck. However, if we choose any single one, we can construct a suitable w.

Chapter 4

Examples of cyclic violator spaces

For some abstract models of optimization problems there are theorems saying that problems of a very small dimension cannot entail a cyclic structure. Namely, unique sink orientations of grids of dimension 2 are acyclic [GMR] and oriented matroid programs of rank 3 are Euclidean [BLVS$^+$99]. Gärtner (personal communication, June 2004) suggested that a similar result might hold for violator spaces. In this chapter we present some examples of cyclic violator spaces of combinatorial dimension 2. However, we conjecture that such examples are always degenerate.

Conjecture 4.1. *Let (H, V) be a nondegenerate basis-regular violator space of combinatorial dimension $d \leq 2$. Then (H, V) is acyclic.*

However, in higher dimension we can get cyclic violator spaces which are nondegenerate and basis-regular. We show another example proving this.

The examples presented in this chapter prove the following proposition.

Proposition 4.2. *There exists a 2-dimensional cyclic violator space with 4 constraints. There exists a 2-dimensional basis-regular cyclic violator space. There exists a 3-dimensional nondegenerate basis-regular cyclic violator space.*

A notation for medium-size violator spaces. We are going to present particular examples of violator spaces. In doing this, we need to specify the mapping V. Since giving values of $\mathsf{V}(G)$ for all $2^{|H|}$ subsets G of H is cumbersome, we devise a more condensed notation.

We use a property of V analogous to the Cube lemma 3.3.

Observation 4.3. *Let (H, V) be a violator space and let B be any basis in (H, V). Then for all sets G with $B \subseteq G \subseteq H \setminus \mathsf{V}(B)$ we have $\mathsf{V}(G) = \mathsf{V}(B)$.*

Proof. We get the statement as an immediate consequence of locality. ∎

Now consider the following game. Suppose that an adversary has a hidden violator space (H, V) and we are given the list of the values of $\mathsf{V}(B)$ for all bases B in (H, V). The adversary gives us a set $G \subseteq H$ and our task is to guess the value of $\mathsf{V}(G)$. If we can find a basis B in the list such that $B \subseteq G \subseteq H \setminus \mathsf{V}(B)$, we win, since Observation 4.3 guarantees us that $\mathsf{V}(G) = \mathsf{V}(B)$. On the other hand, if no such basis exists, the adversary must have cheated: in the hidden violator space,

G has some basis B, and we have $G \cap \mathsf{V}(B) = \emptyset$, therefore $B \subseteq G \subseteq H \setminus \mathsf{V}(B)$, and we should have been able to find this basis B in the list.

To summarize, we have proved the following result.

Proposition 4.4. *To determine the mapping* V *in a violator space* (H, V)*, it is sufficient to specify only the values of* $\mathsf{V}(B)$ *for all bases.*

To make simpler to spot a suitable basis for a given set G in the list, we introduce the following notation. By writing $\mathsf{V}(B \dots C) = X$, where $B \subseteq C$, we mean that we have $\mathsf{V}(G) = X$ for every G with $B \subseteq G \subseteq C$. With this notation we give $\mathsf{V}(B \dots (H \setminus \mathsf{V}(B)))$ for every basis B in the violator space; from the previous paragraphs follows that this uniquely determines V. Finally, we omit the commas and braces in the notation for sets. For instance, if $H = \{a, b, c, d, e, f\}$ and for a basis $B = \{a, b\}$ we have $\mathsf{V}(B) = \{e, f\}$, we write $\mathsf{V}(ab \dots abcd) = ef$.

We continue with the promised examples of cyclic violator spaces.

A 2-dimensional cyclic violator space with four constraints. We choose the set of constraints to be $H := \{a, b, c, d\}$. Let the mapping V be given by the following list.

$\mathsf{V}(\emptyset) = abcd$, $\mathsf{V}(a \dots ad) = bc$, $\mathsf{V}(b \dots ab) = cd$, $\mathsf{V}(c \dots bc) = ad$,
$\mathsf{V}(d \dots cd) = ab$, $\mathsf{V}(ac \dots abcd) = \emptyset$, $\mathsf{V}(bd \dots abcd) = \emptyset$.

One can check that the axioms of violator spaces are satisfied. The cycle in the violator space is given by the sets $G_1 = \{a\}$, $G_2 = \{b\}$, $G_3 = \{c\}$, $G_4 = \{d\}$. We have $G_1 \cap \mathsf{V}(G_2) = \{a\} \cap \mathsf{V}(b) = \{a\} \cap \{c, d\} = \emptyset$, etc., as required by the definition of a cycle in Definition 1.13.

The bases in (H, V) are the sets $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{a, c\}$, $\{b, d\}$. We see that the maximum cardinality of a basis, i.e., the combinatorial dimension of (H, V), is indeed 2.

A 2-dimensional basis-regular cyclic violator space. The violator space in the previous example is not basis-regular: we have $\dim(H, \mathsf{V}) = 2$, but the two-element set $\{a, b\}$ has a single-element basis $\{a\}$. However, it turns out that a cyclic violator space of dimension 2 can be found even if we require it to be basis-regular.

We set $H := \{a, b, c, d, e, f\}$. We give the mapping V by the following list.

$\mathsf{V}(\emptyset) = abcdef$, $\mathsf{V}(a) = bcdef$, $\mathsf{V}(b) = acdef$, $\mathsf{V}(c) = abdef$,
$\mathsf{V}(d) = abcef$, $\mathsf{V}(e) = abcdf$, $\mathsf{V}(f) = abcde$,
$\mathsf{V}(ab \dots abf) = cde$, $\mathsf{V}(bc \dots abc) = def$, $\mathsf{V}(cd \dots bcd) = aef$, $\mathsf{V}(de \dots cde) = abf$,
$\mathsf{V}(ef \dots def) = abc$, $\mathsf{V}(af \dots aef) = bcd$,
$\mathsf{V}(ac \dots ace) = bdf$, $\mathsf{V}(ae) = bcdf$, $\mathsf{V}(bd \dots bdf) = ace$, $\mathsf{V}(bf) = acde$,
$\mathsf{V}(ce) = abdf$, $\mathsf{V}(df) = abce$,
$\mathsf{V}(ad \dots abcdef) = \emptyset, \mathsf{V}(be \dots abcdef) = \emptyset$, $\mathsf{V}(cf \dots abcdef) = \emptyset$.

The two first lines settles the sets of small cardinality; they ensure that the problem is basis-regular. Sets in the third and fourth line form the desired cycle: $G_1 = \{a, b\}$, $G_2 = \{b, c\}$, $G_3 = \{c, d\}$, $G_4 = \{d, e\}$, $G_5 = \{e, f\}$, $G_6 = \{a, f\}$.

The problem is degenerate, since the set $H = \{a, b, c, d, e, f\}$ has three bases: $\{a, b\}$, $\{c, d\}$ and $\{e, f\}$. Note that the sets that account for degeneracy do not appear in the cycle.

A 3-dimensional basis-regular nondegenerate cyclic violator space. It is not obvious how to find an example of a cyclic basis-regular nondegenerate violator space, regardless of the dimension. The following example can be constructed with help of noneuclidean oriented matroid programs (see Chapter 7) or cyclic unique sink orientations of cubes [GMRŠ06].

We set $H := \{a, b, c, d, e, f\}$. We give the mapping V by the following list.

$\mathsf{V}(cde \dots bcdef) = a$, $\mathsf{V}(ace \dots acde) = bf$, $\mathsf{V}(aef \dots acdef) = b$,
$\mathsf{V}(abf \dots abef) = cd$, $\mathsf{V}(bdf \dots abdef) = c$, $\mathsf{V}(bcd \dots bcdf) = ae$,
$\mathsf{V}(abd \dots abde) = cf$, $\mathsf{V}(cdf \dots acdf) = be$, $\mathsf{V}(bce \dots bcef) = ad$,
$\mathsf{V}(abc \dots abcdef) = \emptyset$.

For the sets F not present on the list we define $\mathsf{V}(F) := H \setminus F$. The cycle is formed by the sets in the first two lines: $G_1 = \{c, d, e\}$, $G_2 = \{a, c, e\}$, $G_3 = \{a, e, f\}$, $G_4 = \{a, b, f\}$, $G_5 = \{b, d, f\}$, $G_6 = \{b, c, d\}$.

Chapter 5

The number of violator spaces

In this chapter we prove several bounds on the number of violator spaces of given parameters.

Let $V(n,d)$ be the number of violator spaces on a fixed set of constraints H with $|H| = n$ and with the combinatorial dimension at most d. Let $V(n)$ be the number of all violator spaces without the restriction on the dimension. Since the dimension of a violator space (H, V) is at most $|H|$, we have $V(n) = V(n,n)$. Let $V^*_{\mathrm{RN}}(n,d)$ be the number of *basis-regular nondegenerate* violator spaces with $|H| = n$ and combinatorial dimension *exactly* d, and let the number of acyclic ones among them be $V^*_{\mathrm{RNA}}(n,d)$. Obviously $V^*_{\mathrm{RNA}}(n,d) \le V^*_{\mathrm{RN}}(n,d) \le V(n,d)$.

The bounds proved in this chapter are summarized in the following theorem.

Theorem 5.1. *For the functions $V(n)$, $V(n,d)$, $V^*_{\mathrm{RN}}(n,d)$, and $V^*_{\mathrm{RNA}}(n,d)$ the following bounds hold.*

- *Equation (5.1):* $V(n) \le \exp\big(n2^{n-1}\ln 2\big)$
- *Equation (5.2):* $V(n,d) \le \exp\big((\mathrm{e}/d)^d\, n^{d+1} \ln 2\big)$
- *Equation (5.4):* $V^*_{\mathrm{RN}}(n,d) \le \exp\big(O\big(d(\mathrm{e}/d)^d\, n^d \ln n\big)\big)$
- *Equation (5.5):* $V^*_{\mathrm{RNA}}(n,d) \ge \exp\big(\Omega\big(d^{-1/2}(\mathrm{e}/d)^d\,(n-d)^d\big)\big)$

The exact values of V and V^*_{RN} for small n and d are given in Table 5.1. The values have been determined by a computer search.

$d\backslash n$	0	1	2	3	4	5
0	1	1	1	1	1	1
1		2	6	26	150	1082
2			9	183	28732	?
3				246	265214	?
4					336852	?

$d\backslash n$	0	1	2	3	4	5
0	1	1	1	1	1	1
1		1	2	6	24	120
2			3	12	240	?
3				51	1844	?
4					27451	?

Table 5.1. Exact values of $V(n,d)$ (left) and $V^*_{\mathrm{RN}}(n,d)$ (right)

5.1. Upper bounds

In this section the upper bounds on the number of violator spaces are established.

A simple bound. We establish a simple upper bound on the number of violator spaces by relating violator spaces to the orientations of edges of a hypercube.

Consider a finite set H with $|H| = n$. By an *n-dimensional hypercube* we mean a graph whose vertices are subsets of H and whose edges connect sets differing by presence of exactly one element.

For each violator space (H, V) with the set of constraints H we construct an orientation $\mathcal{E}$ of the edges of the hypercube. We proceed in the following way: for every $G \subseteq H$ and $h \in H \setminus G$, the vertices G and $G \cup \{h\}$ are adjacent in the hypercube. If $h \notin \mathsf{V}(G)$, we add into $\mathcal{E}$ the oriented edge $(G \cup \{h\}, G)$, otherwise we add the oriented edge $(G, G \cup \{h\})$.

From the orientation $\mathcal{E}$ we can reconstruct the original violator space: given a set G of constraints, the set $\mathsf{V}(G)$ contains exactly the elements $h \in H \setminus G$ satisfying $(G, G \cup \{h\}) \in \mathcal{E}$. Using consistency of violator spaces one can see that for distinct violator spaces we obtain distinct orientations.

The n-dimensional hypercube has $n2^{n-1}$ edges, hence the number of its orientations is $2^{n2^{n-1}} = \exp(n2^{n-1} \ln 2)$, and from the discussion above we know that the number of violator spaces on n constraints is bounded from above by this number.

To summarize, we have proved that

$$V(n) \leq \exp(n2^{n-1} \ln 2). \tag{5.1}$$

An estimate for violator spaces of a bounded dimension. We proceed with establishing a bound employing the combinatorial dimension. We use Proposition 4.4 asserting that for determining a violator space it is sufficient to specify the values of $\mathsf{V}(B)$ for all bases B. In a violator space of combinatorial dimension bounded by d, the possible bases are only sets of cardinality at most d. The number of violator spaces is bounded by the number of possible mappings $\mathsf{V}' : \mathcal{P} \to 2^H$, where $\mathcal{P} = \{G \subseteq H : |G| \leq d\}$ is the set of prospective bases, since every such mapping V' can be extended to a full violator mapping $\mathsf{V} : 2^H \to 2^H$ in at most one way.

For a n-element set H, the number of possible bases is

$$|\mathcal{P}| = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{d} \leq \left(\frac{\mathrm{e}n}{d}\right)^d$$

(for a proof of the inequality see, e.g., [MN98]). This gives

$$V(n, d) \leq \left|2^H\right|^{|\mathcal{P}|} \leq (2^n)^{\left(\frac{\mathrm{e}n}{d}\right)^d} = \exp\left((\mathrm{e}/d)^d n^{d+1} \ln 2\right). \tag{5.2}$$

It might seem that the knowledge of the set of bases carries quite a lot of information already, under lucky conditions possibly even enough as to completely determine the violator space. If this was true, this might give a better bound on

$V(n,d)$. Unfortunately, this is not the case. In particular, in basis-regular violator spaces of combinatorial dimension d, *every* set of d elements is a basis, and no set with more than d elements is a basis. Yet, basis-regular violator spaces of fixed dimension abound.

A bound for nondegenerate basis-regular violator spaces. In the next bound we employ nondegeneracy and basis-regularity.

First we prove a useful lemma concerning the structure of nondegenerate basis-regular violator spaces. The lemma was originally proved by Clarkson [Cla93] for linear programming problems. An LP-type version was formulated by Gärtner and Welzl [GW01]. The proof presented here is a straightforward generalization of the previous proofs.

Lemma 5.2. *A basis-regular nondegenerate violator space (H, V) of combinatorial dimension d has exactly $\binom{d+k-1}{d-1}$ bases B with exactly k violators (i.e., with $|\mathsf{V}(B)| = k$), for $k = 0, \dots, n-d$.*

Proof. Let b_k be the number of bases with exactly k violators.

Since a basis B is a basis of G if and only if $B \subseteq G \subseteq H \setminus \mathsf{V}(B)$, a d-element basis with k violators is a (unique) basis of exactly $\binom{n-d-k}{r-d}$ sets of size r. By considering all sets of size r for $d \le r \le n$ we get

$$\sum_{k=0}^{n-d} b_k \binom{n-d-k}{r-d} = \binom{n}{r}. \tag{5.3}$$

Regarding the values b_k as unknowns, this gives a system of $n-d+1$ equations with $n-d+1$ unknowns. The last $r-d$ coefficients in the equation (5.3), i.e., those with $k \ge n-r+1$, are 0, and the preceding one, i.e., the one with $k = n-r$, is equal to 1. Therefore the matrix of the system is triangular and all elements on the diagonal are 1, hence the system has a unique solution. One can check that the claimed solution $b_k = \binom{d+k-1}{d-1}$ works by substituting it into the equation (5.3), getting

$$\sum_{k=0}^{n-d} \binom{d-1+k}{d-1} \binom{n-d-k}{r-d} = \binom{n}{r}.$$

This identity can be proved by standard manipulations of binomial coefficients; see Equation (5.26) in [GKP89].

This completes the proof of the lemma. ∎

Now we describe how to encode a violator space into a series of choices. For each set we determine its basis, beginning with large sets. We count the number of choices in each step; then by multiplying we get an upper bound on the number of all basis-regular nondegenerate violator spaces.

We start with the whole of H. We choose any d-element subset B_H of H to become the basis of H. We have $\binom{n}{d}$ possible choices. Note that besides H, this also fixes the basis for all sets G with $B_H \subseteq G \subseteq H$, which are exactly the sets with $\mathsf{V}(G) = \emptyset$.

Now we proceed with sets of size $n-1$. For each of these sets whose basis remains unknown after the previous step, we want to pick the basis now. For each of them, we have at most $\binom{n-1}{d} < \binom{n}{d}$ choices. The sets in question are exactly those that will have exactly one violator in the resulting violator space; to see this, note that a $(n-1)$-element set has at most one violator, and the sets F with $\mathsf{V}(F) = \emptyset$ have their bases determined in the previous step. Therefore Lemma 5.2 asserts that there are exactly $\binom{d+1-1}{d-1}$ sets to process in this step. We stick to any suitable choice of bases for these sets and we proceed to sets of smaller size.

In the s-th step, we consider all sets of size $r := n - s + 1$ that do not have a basis determined in the previous steps. We claim that the sets in question are those with exactly $k := n - r$ violators in the resulting violator space. To see this, note that a set with more than k violators must have fewer than r elements to satisfy consistency. On the other hand, choose an r-element set F with the number of violators $|\mathsf{V}(F)| < k$; we claim that its basis is already known from previous steps. Let B_F be the basis of F and put $G := H \setminus \mathsf{V}(F) \supset F$. From locality we get $\mathsf{V}(G) = \mathsf{V}(F) = \mathsf{V}(B_F)$ and from nondegeneracy $B_F = B_G$, therefore the basis of F is known since we have picked B_G to be the basis of G.

Now, from Lemma 5.2 follows that there are exactly $\binom{d+k-1}{d-1}$ sets for which the basis has to be determined in this step. For each of them we choose its basis to be some of its $\binom{r}{d} \leq \binom{n}{d}$ subsets.

The number N of choices we were allowed to take up until now is

$$N \leq \prod_{k=0}^{n-d} \binom{n}{d}^{\binom{d+k-1}{d-1}} = \binom{n}{d}^{\sum_{k=0}^{n-d} \binom{d+k-1}{d-1}} = \binom{n}{d}^{\binom{n}{d}}.$$

When all sets G of size $|G| \geq d$ have been processed, the description of the violator space is nearly complete. It remains only to decide about the values of $\mathsf{V}(F)$ for sets F of size $|F| < d$; for a while, call these sets *small*. The number of small sets is

$$S = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{d-1} \leq \left(\frac{\mathrm{e}n}{d}\right)^d.$$

For each small set F we need $H \setminus \mathsf{V}(F)$ to be a small set, for otherwise we lose basis-regularity. Therefore the number of ways to define the mapping V on small sets is bounded by S^S.

At the end the violator space is completely determined, and we get that the number of violator spaces is

$$V^*_{\mathrm{RN}}(n,d) \leq N \cdot S^S \leq \binom{n}{d}^{\binom{n}{d}} \left(\frac{\mathrm{e}n}{d}\right)^{d\left(\frac{\mathrm{e}n}{d}\right)^d} \leq \left[\left(\frac{\mathrm{e}n}{d}\right)^{d\left(\frac{\mathrm{e}n}{d}\right)^d}\right]^2 =$$

$$= \exp\left((\ln \mathrm{e} + \ln n - \ln d)\, 2d \left(\frac{\mathrm{e}n}{d}\right)^d\right) = \exp\left(O\left(d\left(\frac{\mathrm{e}}{d}\right)^d n^d \ln n\right)\right). \tag{5.4}$$

5.2. Lower bound

To get a lower bound on V^*_{RNA}, we present a recursive construction of nondegenerate basis-regular acyclic violator spaces.

We start with small cases. For any n, there is exactly one violator space of dimension 0, namely $\mathsf{V}(G) = \emptyset$ for all G, and it is basis-regular, nondegenerate and acyclic. Therefore $V^*_{\mathrm{RNA}}(n, 0) = 1$.

Now we consider $d = 1$. We choose any linear ordering $<$ of H and we put

$$\mathsf{V}(G) := \{h \in H : h > g \text{ for all } g \in G\}.$$

Here (H, V) is a violator space, and since every nonempty set G has its largest element as a basis, (H, V) is a basis-regular nondegenerate acyclic violator space of dimension 1. For various choices of the ordering $<$ we obtain distinct violator spaces, so $V^*_{\mathrm{RNA}}(n, 1) \geq n!$.

For a given $d \geq 0$, the following construction gives a basis-regular nondegenerate acyclic violator space on $n = d$ constraints. Let H be a d-element set, and set $\mathsf{V}(G) := H \setminus G$ for every $G \subseteq H$. Then H is a basis of itself, therefore the dimension of the violator space is exactly d. This proves that $V^*_{\mathrm{RNA}}(d, d) \geq 1$ for every $d \geq 0$.

We continue with a construction that allows us to glue two violator spaces together.

Proposition 5.3. *Consider two violator spaces (H, V_1), (H, V_2) on the same set of constraints. Let d_i be the combinatorial dimension of (H, V_i) for $i = 1, 2$. Let $h \notin H$ be a new constraint; set $H' := H \cup \{h\}$. We define a mapping $\mathsf{V} : 2^{H'} \to 2^{H'}$ as*

$$\mathsf{V}(G) := \begin{cases} \mathsf{V}_1(G) \cup \{h\} & \text{if } h \notin G, \\ \mathsf{V}_2(G \setminus \{h\}) & \text{if } h \in G. \end{cases}$$

Then (H', V) is a violator space. Its combinatorial dimension d is $\max(d_1, d_2 + 1)$. If both of the spaces (H, V_i) are basis-regular then (H', V) is basis-regular. If both of the spaces (H, V_i) are nondegenerate and $d_1 = d_2 + 1$ then (H', V) is nondegenerate. If both of the spaces (H, V_i) are acyclic then (H', V) is acyclic.

Proof. The proof consist of a technical reduction of the claimed statements to the corresponding statements in the original violator spaces, branching to cases depending on whether $h \in F, G$.

Consistency. If $h \notin G$, we have $\mathsf{V}(G) \cap G = (\mathsf{V}_1(G) \cup \{h\}) \cap G = (\mathsf{V}_1(G) \cap G) \cup (\{h\} \cap G) = \emptyset$. If $h \in G$, we have $\mathsf{V}(G) \cap G = \mathsf{V}_2(G \setminus \{h\}) \cap G = \mathsf{V}_2(G \setminus \{h\}) \cap \big((G \setminus \{h\}) \cup \{h\}\big) = \big(\mathsf{V}_2(G \setminus \{h\}) \cap (G \setminus \{h\})\big) \cup \big(\mathsf{V}_2(G \setminus \{h\}) \cap \{h\}\big) = \emptyset$.

Locality. Assume that $F \subseteq G$ and $G \cap \mathsf{V}(F) = \emptyset$; we want to check that $\mathsf{V}(G) = \mathsf{V}(F)$. In the following paragraphs we distinguish three cases.

If $h \notin F$ and $h \notin G$ then $\mathsf{V}(G) = \mathsf{V}_1(G) \cup \{h\}$ and $\mathsf{V}(F) = \mathsf{V}_1(F) \cup \{h\}$. Thus $G \cap \mathsf{V}_1(F) \subseteq G \cap (\mathsf{V}_1(F) \cup \{h\}) = G \cap \mathsf{V}(F) = \emptyset$, so locality for V_1 applies. We get $\mathsf{V}_1(F) = \mathsf{V}_1(G)$, and $\mathsf{V}(F) = \mathsf{V}(G)$ follows.

If $h \notin F$ and $h \in G$ then $h \in \mathsf{V}(F) = \mathsf{V}_1(F) \cup \{h\}$, so the intersection $G \cap \mathsf{V}(F)$ contains h, therefore it is nonempty. So this case does not occur.

If $h \in F$ and $h \in G$, we have $\mathsf{V}(G) = \mathsf{V}_2(G \setminus \{h\})$ and $\mathsf{V}(F) = \mathsf{V}_2(F \setminus \{h\})$. Thus $(G \setminus \{h\}) \cap \mathsf{V}_2(F \setminus \{h\}) \subseteq G \cap \mathsf{V}_2(F \setminus \{h\}) = F \cap \mathsf{V}(F) = \emptyset$, so locality for V_2 applies. We get $\mathsf{V}_2(F) = \mathsf{V}_2(G)$, and $\mathsf{V}(F) = \mathsf{V}(G)$ follows.

Dimension. For a basis of one of (H, V_i) we construct a basis of (H', V) and vice versa. The constructions are straightforward but tedious. This will however proof the desired equality for dimensions.

Consider a basis B of (H, V_1). We claim that B is a basis of (H', V). For a proper subset $F \subset B$ we have $B \cap \mathsf{V}(F) = B \cap (\mathsf{V}_1(F) \cup \{h\}) \supseteq B \cap \mathsf{V}_1(F)$ which is nonempty since B is a basis for V_1. So the claim holds. This proves that $d_1 \leq d$.

Consider a basis B of (H, V_2). We claim that $B \cup \{h\}$ is a basis of (H', V). For a proper subset $F \subset B \cup \{h\}$ we distinguish two cases. If $h \notin F$, we have $h \in (B \cup \{h\}) \cap (\mathsf{V}_1(F) \cup \{h\}) = (B \cup \{h\}) \cap \mathsf{V}(F)$. If $h \in F$, we have $(B \cup \{h\}) \cap \mathsf{V}(F) \supseteq B \cap \mathsf{V}_2(F \setminus \{h\})$, which is nonempty since B is a basis for V_2. In both cases we proved that $(B \cup \{h\}) \cap \mathsf{V}(F)$ is nonempty, so the claim holds. Therefore $d_2 + 1 \leq d$.

Finally, consider a basis B of (H', V). We claim that if $h \notin B$ then B is a basis in (H, V_1), and if $h \in B$ then $B \setminus \{h\}$ is a basis in (H, V_2). In the first case, for $F \subset B$ we have $B \cap \mathsf{V}_1(F) = B \cap \mathsf{V}(F) \neq \emptyset$; in the second case, for $F \subset B \setminus h$ (i.e., $F \cup \{h\} \subset B$), we have $B \cap \mathsf{V}_2(F) = B \cap \mathsf{V}(F \cup \{h\}) \neq \emptyset$. So the claim holds and this proves that $d \leq \max(d_1, d_2 + 1)$.

Basis-regularity. Assume that the violator spaces (H, V_1) and (H, V_2) are basis-regular. Consider a set $G \subseteq H'$; we claim that its basis B in (H', V) is unique. Again, the proof depends on whether $h \in G$.

If $h \notin G$, we have $\mathsf{V}(G) = \mathsf{V}_1(G) \cup \{h\}$. Assume that $B \subseteq G$ is a basis of G in (H', V); we have $h \notin G$, therefore $\mathsf{V}(B) = \mathsf{V}_1(B) \cup \{h\}$, and from the discussion regarding the dimension, B is a basis in (H, V_1). Since $\mathsf{V}(B) = \mathsf{V}(G)$, we have $\mathsf{V}_1(B) = \mathsf{V}_1(G)$, therefore B is a basis of G in the basis-regular violator space (H, V_1), which determines B uniquely.

If $h \in G$, we have $\mathsf{V}(B) = \mathsf{V}_2(G \setminus \{h\})$. Assume that $B \subseteq G$ is a basis of G in (H', V). Note that $h \in B$, since otherwise $h \in \mathsf{V}(B) \setminus \mathsf{V}(G)$, i.e., $\mathsf{V}(B) \neq \mathsf{V}(G)$. Now we have $\mathsf{V}(B) = \mathsf{V}_2(B \setminus \{h\})$ and from the discussion of the dimension, $B \setminus \{h\}$ is a basis in (H, V_2). Since $\mathsf{V}(B) = \mathsf{V}(G)$, we have $\mathsf{V}_2(B \setminus \{h\}) = \mathsf{V}_2(G \setminus \{h\})$, therefore $B \setminus \{h\}$ is a basis of $G \setminus \{h\}$ in the basis-regular violator space (H, V_2), which determines B uniquely.

Nondegeneracy. Assume that the violator spaces (H, V_1) and (H, V_2) are nondegenerate and $d_1 = d_2 + 1 = d$. Consider a set $G \subseteq H'$ with $|G| \geq d$; we claim that for its basis B in (H', V) we have $|B| = d$. Again, we distinguish the cases $h \in G$ and $h \notin G$.

If $h \notin G$, from the previous parts of the proof we know that B is a basis in (H, V_1). Since (H, V_1) is nondegenerate, and $|G| \geq d = d_1$, we have $|B| = d_1 = d$.

If $h \in G$, we have $h \in B$ and we know that $B \setminus \{h\}$ is a basis of $G \setminus \{h\}$ in (H, V_2). Since $|G \setminus \{h\}| \geq d - 1 = d_2$, we have $|B \setminus \{h\}| = d_2$, i.e., $|B| = d_2 + 1 = d$.

Acyclicity. Assume that (H', V) is cyclic. By definition, this means that for some $G_1, \ldots, G_k \subseteq H'$ we have $G_i \cap \mathsf{V}(G_{i+1}) = \emptyset$ for all $i = 1, \ldots, k$ (for simplicity we set $G_{k+1} := G_1$), and $\mathsf{V}(G_1) \neq \mathsf{V}(G_2)$. We distinguish three cases depending on how many of the sets G_i contain h.

If $h \notin G_i$ for all i, we have a cycle $G_1, \ldots, G_k$ in H_1. If $h \in G_i$ for all i, we have a cycle $G_1 \setminus \{h\}, \ldots, G_k \setminus \{h\}$ in H_2. If h is contained in some of the sets G_i but not in all of them, there is certainly some ℓ such that $h \in G_\ell$ but $h \notin G_{\ell+1}$. However, now $h \in \mathsf{V}(G_{\ell+1})$ by definition of V. Hence $h \in G_\ell \cap \mathsf{V}(G_{\ell+1})$, therefore the intersection is nonempty, and the witness $G_1, \ldots, G_k$ for cyclicity is bogus.

This completes the proof of Proposition 5.3. ∎

From the construction one can see that for distinct choices of V_1 and V_2 we get distinct resulting mappings V. Therefore the proposition gives the recursive bound

$$V^*_{\mathrm{RNA}}(n,d) \geq V^*_{\mathrm{RNA}}(n-1,d) \cdot V^*_{\mathrm{RNA}}(n-1,d-1) \qquad \text{for } n,d \geq 1.$$

For logarithms $L(n,d) := \ln V^*_{\mathrm{RNA}}(n,d)$ we get the recurrence

$$L(n,d) \geq L(n-1,d) + L(n-1,d-1).$$

The examples of violator spaces preceding the statement of Proposition 5.3 prove that $L(n,1) \geq \ln n! \geq n-2$ and $L(d,d) \geq \ln 1 = 0$ for every $n,d \geq 1$. Using Lemma 5.4 below, the recurrence implies that $L(n,d)$ is bounded from below by $\binom{n-2}{d}$. By Stirling's approximation of the factorial (see, e.g., Equation (4.23) in [GKP89]) we have

$$\binom{n-2}{d} \geq \frac{(n-2-d)^d}{d!} \geq \frac{(n-2-d)^d}{C(d/\mathrm{e})^d\sqrt{d}} = \Omega\big(d^{-1/2}(\mathrm{e}/d)^d\,(n-d)^d\big),$$

where C is a constant slightly greater than $\sqrt{2\pi}$. Therefore we conclude with the bound

$$V^*_{\mathrm{RNA}}(n,d) \geq \exp\Big(\Omega\big(d^{-1/2}(\mathrm{e}/d)^d\,(n-d)^d\big)\Big). \tag{5.5}$$

It remains to state and prove the lemma concerning the recurrence.

Lemma 5.4. *Let $L(n,d)$ be a function of two integer variables satisfying*

- $L(d,d) \geq 0$ *for all* $d \geq 1$,
- $L(n,1) \geq n-2$ *for all* $n \geq 1$,
- $L(n,d) \geq L(n-1,d) + L(n-1,d-1)$ *for all* $n,d \geq 2$.

Then we have

$$L(n,d) \geq \binom{n-2}{d} \tag{5.6}$$

for all $n \geq d \geq 1$.

Proof. We proceed by double induction. For $d = 1$ we have $L(n,1) \geq n-2 = \binom{n-2}{1}$, hence the inequality (5.6) holds.

For an induction step let d be fixed. To prove (5.6) for all n, let us assume that it holds for $d-1$ for every n. We have $L(d,d) \geq 0 = \binom{d-2}{d}$, and induction on n gives

$$L(n,d) \geq L(n-1,d) + L(n-1,d-1) \geq \binom{n-3}{d} + \binom{n-3}{d-1} = \binom{n-2}{d},$$

as claimed. ■

Conclusion. The described construction for the lower bound gives violator spaces of a very specific structure, so I believe that the actual value of V^*_{RNA} is closer to the upper bound from Equation (5.4).

Part II

Applications

Chapter 6

Clarkson's algorithm

Clarkson [Cla95] developed an elaborate randomized algorithm for solving linear programs with few variables and many constraints. The expected running time of the algorithm in fixed dimension is linear in the number of constraints.

Gärtner and Welzl [GW96] noted that the algorithm works for abstract LP-type problems. Chazelle and Matoušek [CM96] presented a derandomized version of the algorithm. We can use the equivalence of violator spaces and LP-type problems (Theorem 1.14) to conclude that these algorithms can be used for acyclic violator spaces.

In this chapter we prove that Clarkson's algorithm works for violator spaces with $-\infty$ even without assuming acyclicity. The analysis we give is almost identical on the abstract level to the analysis of the LP-type version of the algorithm. Furthermore we prove that the class of violator spaces in a certain sense coincides with the class of problems solvable by Clarkson's algorithm (Proposition 6.14).

6.1. Sampling lemma

Expected number of violators. The analysis of the running time of Clarkson's algorithm relies on a bound on the expected number of violators of a random set of constraints. Before presenting the algorithm, let us derive the bound.

We can talk about some kind of monotonicity in violator spaces, even if the order is absent here. The following is an easy consequence of Definition 2.6.

Lemma 6.1 (Monotonicity for violator spaces). *Let $(H, \mathsf{V}, \mathcal{U})$ be a violator space with $-\infty$. Let $F \subseteq E \subseteq G \subseteq H$ and let $F \notin \mathcal{U}$. Then*

$$\mathsf{V}(F) = \mathsf{V}(G) \text{ implies } \mathsf{V}(F) = \mathsf{V}(E) = \mathsf{V}(G).$$

Proof. We have $E \cap \mathsf{V}(F) \subseteq G \cap \mathsf{V}(G) = \emptyset$, so locality yields $\mathsf{V}(E) = \mathsf{V}(F)$. ∎

The definition of basis can be used to prove the following observation, well-known to hold for LP-type problems [GW96].

Observation 6.2. *Let (H, V) be a violator space. For $R \subseteq H$ with $R \notin \mathcal{U}$ and all $h \in H$, we have the following equivalences:*

(i) $\mathsf{V}(R) \neq \mathsf{V}(R \cup \{h\})$ *if and only if* $h \in \mathsf{V}(R)$*;*
(ii) $\mathsf{V}(R) \neq \mathsf{V}(R \setminus \{h\})$ *if and only if* h *is contained in every basis of* R*.*

We say that a constraint h *is* extreme *in* R *if the conditions of the equivalence (ii) hold.*

Proof. The equivalence (i). If $h \notin \mathsf{V}(R)$, we get $(R \cup \{h\}) \cap \mathsf{V}(R) = \emptyset$ and by locality $\mathsf{V}(R) = \mathsf{V}(R \cup \{h\})$. On the other hand, if $h \in \mathsf{V}(R)$ then we get $\mathsf{V}(R) \neq \mathsf{V}(R \cup \{h\})$ from consistency applied to $R \cup \{h\}$.

The equivalence (ii). First assume that $R \setminus \{h\} \notin \mathcal{U}$. If $\mathsf{V}(R) = \mathsf{V}(R \setminus \{h\})$, there exists a basis B of $R \setminus \{h\}$, and this B is also a basis of R that does not contain h. Conversely, if there is some basis B of R that does not contain h, then $\mathsf{V}(R) = \mathsf{V}(R \setminus \{h\})$ follows from monotonicity.

Now assume that $R \setminus \{h\} \in \mathcal{U}$. We claim that in this case h is extreme, i.e., that both sides of the equivalence to prove are true. Since $R \notin \mathcal{U}$, we have $h \in \mathsf{V}(R \setminus \{h\})$ from relation between V and $\mathcal{U}$ (Definition 2.6). Consistency gives $\mathsf{V}(R) \neq \mathsf{V}(R \setminus \{h\})$, which is the left-hand side of the equivalence. Now let B be any basis of R. If $h \notin B$, we have $B \subseteq R \setminus \{h\}$, hence $B \in \mathcal{U}$, which is not possible. Therefore h is contained in every basis of R, and this is the right-hand side of the equivalence. ∎

Observation 6.2 implies that in a violator space of combinatorial dimension d, every bounded set has at most d extreme elements. This in turn yields a bound for the *expected* number of violators of a random subset of constraints. To prove the bound we use a general lemma due to Gärtner and Welzl [GW01]. For the reader's convenience and for sake of completeness, we present the lemma including the original proof.

Lemma 6.3 (Sampling lemma). *Consider arbitrary sets* H *and* M *and a mapping* $\psi : 2^H \to M$*. Let* $n = |H|$*. For* $Q, R \subseteq H$ *we define*

$$\mathcal{V}(R) := \big\{h \in H \setminus R : \psi(R) \neq \psi(R \cup \{h\})\big\},$$
$$\mathcal{X}(Q) := \big\{h \in Q : \psi(Q) \neq \psi(Q \setminus \{h\})\big\}.$$

For $0 \leq r \leq |H|$*, let* v_r *be the expected value of* $|\mathcal{V}(R)|$ *for* R *chosen uniformly at random among all subsets of* H *with* r *elements. Let* x_r *be defined as the expected value of* $|\mathcal{X}(R)|$ *under the same conditions. Then for every* $r = 0, \ldots, n$ *we have*

$$v_r = \frac{n-r}{r+1} x_{r+1}.$$

Proof. By definitions of $\mathcal{V}$ and $\mathcal{X}$ we have for every $s \in H \setminus R$

$$s \in \mathcal{V}(R) \text{ if and only if } s \in \mathcal{X}(R \cup \{s\}).$$

Therefore we get

$$\binom{n}{r} v_r = \sum_{R \in \binom{H}{r}} \sum_{s \in H \setminus R} \big[s \in \mathcal{V}(R)\big] = \sum_{R \in \binom{H}{r}} \sum_{s \in H \setminus R} \big[s \in \mathcal{X}(R \cup \{s\})\big] =$$

$$= \sum_{Q \in \binom{H}{r+1}} \sum_{s \in Q} [s \in \mathcal{X}(Q)] = \binom{n}{r+1} x_{r+1} = \binom{n}{r} \frac{n-r}{r+1} x_{r+1}.$$

■

For our situation we have the following corollary.

Corollary 6.4. *Let $(H, \mathsf{V}, \mathcal{U})$ be a violator space with $-\infty$ of combinatorial dimension d with $|H| = n$. Fix a set $W \subseteq H$ with $W \notin \mathcal{U}$. For $r = 0, \ldots, n - |W|$, let v_r be the expected number of violators of the set $W \cup R$, where R is uniformly random subset of $H \setminus W$ of size r. Then*

$$v_r \leq d \frac{n-r}{r+1}.$$

Proof. We use the Sampling lemma for random subsets of $H \setminus W$. We define $\psi(R) := \mathsf{V}(W \cup R)$. Then the set $\mathcal{X}(R)$ contains exactly the extreme elements of $W \cup R$; hence $|X(R)| \leq d$ for all R. The bound on v_r follows. ■

6.2. A setting for the algorithm

Let $(H, \mathsf{V}, \mathcal{U})$ be a violator space with $-\infty$ of combinatorial dimension d. The violator space is given implicitly by the following primitive operations:

Primitive 6.5 (Violation test). *Given $F \subseteq H$ with $|F| \leq d$ and $h \in H \setminus F$, decide whether $h \in \mathsf{V}(F)$.*

Primitive 6.6 (Boundedness test). *Given $F \subseteq H$ with $|F| \leq d$, decide whether $F \in \mathcal{U}$.*

We assume that an initial bounded basis $B_0 \subseteq H$, $B_0 \notin \mathcal{U}$ is provided. If we have a violator space without $-\infty$, i.e., $\mathcal{U} = \emptyset$, we can safely set $B_0 := \emptyset$.

Our goal is to find a basis of H. We build the algorithms so that they can find a basis of $G_0 \cup B_0$ for any given $G_0 \subseteq H$. Let the size of $G := G_0 \setminus B_0$ be denoted by n.

6.3. The trivial algorithm

With Primitives 6.5 and 6.6, the problem can be solved in a brute-force manner by going through all sets of size at most d, testing each of them for being a basis of $G' := G \cup B_0$. A set $B \subseteq G'$ is a basis of G' if and only if $B \notin \mathcal{U}$ and

$$\begin{array}{ll} h \in \mathsf{V}(B \setminus \{h\}) & \text{for every } h \in B, \\ h \notin \mathsf{V}(B) & \text{for every } h \in G' \setminus B. \end{array}$$

Consequently, the number of invocations of the primitive tests carried out in order to find a basis of G' is at most

$$(1 + n + d) \sum_{i=0}^{d} \binom{n}{i} = O(n^{d+1}),$$

where in the parentheses the 1 accounts for the boundedness test and the $n + d$ for the violator tests. In the next three sections we show that this can be substantially improved.

6.4. Clarkson's first algorithm

Clarkson's first algorithm calls Clarkson's second algorithm (`Basis2`) as a subroutine. Given an initial bounded basis B_0 in H and a set $G \subseteq H$ with $G \cap B_0 = \emptyset$, both algorithms compute a basis of $G \cup B_0$.

Algorithm 6.7 (`Basis1`).
Input: *a set $G \subseteq H$, a basis B_0 in H with $B_0 \cap G = \emptyset$ and $B_0 \notin \mathcal{U}$.*
Output: *a basis of $G \cup B_0$.*

```
n := |G|
IF n ≤ 9d² THEN RETURN Basis2(G, B₀)
ELSE
    r := ⌊d√n⌋
    W := ∅
    REPEAT
        choose uniformly random R ⊆ G \ W with |R| = r
        C := Basis2(W ∪ R, B₀)
        V := G ∩ V(C)
        IF |V| ≤ 2√n THEN
            W := W ∪ V
    UNTIL V = ∅
    RETURN C
```

Assuming `Basis2` is correct, this algorithm is correct as well: if B is a basis of $F \subseteq G \cup B_0$ with $F \notin \mathcal{U}$ and in addition B has no violators in $G \cup B_0$, then B is a basis of $G \cup B_0$.

The algorithm augments the working set W at most d times, which is guaranteed by the following observation.

Observation 6.8. *Let $(H, \mathsf{V}, \mathcal{U})$ be a violator space with $-\infty$. Let $G' \subseteq H$. Let $F \subseteq G'$ with $F \notin \mathcal{U}$ and $G' \cap \mathsf{V}(F) \neq \emptyset$. Then every basis B of G' contains at least one element from $G' \cap \mathsf{V}(F)$.*

Proof. Let B be a basis of G' such that $B \cap G' \cap \mathsf{V}(F) = \emptyset$; we want to prove that this leads to a contradiction.

Since $B \subseteq G'$, we have $B \cap \mathsf{V}(F) = \emptyset$. From consistency applied to the set F we get $(B \cup F) \cap \mathsf{V}(F) = \emptyset$. Now locality gives $\mathsf{V}(F) = \mathsf{V}(B \cup F)$. Monotonicity for sets $B \subseteq B \cup F \subseteq G'$ gives $\mathsf{V}(B \cup F) = \mathsf{V}(G')$, hence $\mathsf{V}(F) = \mathsf{V}(G')$. Therefore $G' \cap \mathsf{V}(F) = G' \cap \mathsf{V}(G') = \emptyset$, which contradicts the condition imposed on F. ■

Since the set W always grows at most by $2\sqrt{n}$ elements, we see that the size of W does not exceed $2d\sqrt{n}$. Therefore the first argument for every invocation of `Basis2` is of size at most $3d\sqrt{n}$.

The set V can be computed as $G \cap \mathsf{V}(C) = \{h \in G \setminus C : h \in \mathsf{V}(C)\}$. Hence in every iteration of the `REPEAT` loop we invoke the violation test at most n times.

Now we determine the expected number of iterations through the loop. Corollary 6.4 applied to $(G, \mathsf{V}|_G, \mathcal{U} \cap 2^G)$ bounds the expected number v of violators of $W \cup R \cup B_0$ with uniformly random set R of $r = \lfloor d\sqrt{n} \rfloor$ elements:

$$v \leq d\frac{n-r}{r+1} \leq \frac{dn}{\lfloor d\sqrt{n} \rfloor + 1} \leq \sqrt{n}.$$

The Markov inequality implies that the expected number of calls to `Basis2` before we next augment W is at most 2. Therefore the expected number of iterations of the loop is bounded by $2d$.

Lemma 6.9. *Algorithm* `Basis1` *computes a basis of G with $|G| = n$ using an expected number of at most $2dn$ calls to Primitive 6.5, and an expected number of at most $2d$ calls to* `Basis2` *with sets of size at most $3d\sqrt{n}$.*

6.5. Clarkson's second algorithm

This algorithm calls the trivial algorithm as a subroutine. Instead of adding violated constraints to a working set, it increases their probability of being selected in further iterations. Technically this is done by maintaining G as a multiset, where $\mu(h)$ denotes the multiplicity of h. For a set $F \subseteq G$ we define $\mu(F) := \sum_{h \in F} \mu(h)$ to be the compound multiplicity of all elements of F. Sampling from G is done as before, imagining that G contains $\mu(h)$ copies of every element h.

Algorithm 6.10 (`Basis2`).
Input: *a set $G \subseteq H$, a basis B_0 in H with $B_0 \cap G = \emptyset$ and $B_0 \notin \mathcal{U}$.*
Output: *a basis of $G \cup B_0$.*

```
n := |G|
IF n ≤ 6d² THEN
    RETURN Trivial(G ∪ B₀)
ELSE
    r := 6d²
    REPEAT
        choose μ-distributed random R ⊆ G with |R| = r
        replace repeated elements of R by a single instance
        C := Trivial(R ∪ B₀)
        V := G ∩ V(C)
        IF μ(V) ≤ μ(G)/3d THEN
            for every h ∈ V set μ(h) := 2μ(h)
    UNTIL V = ∅
    RETURN C
```

Again we see that the algorithm `Basis2` is correct, provided that `Trivial` is correct.

We say that an iteration of the loop is *successful* if we change the weights of elements. To estimate how many unsuccessful iterations pass between two successful ones we again use Corollary 6.4. To make its application formally correct for the case of multisets, we assume that $G = \{g_1, \ldots, g_n\}$, and we sample from the set

$$\hat{G} = \{g_1^{(1)}, \ldots, g_1^{(\mu(g_1))}, \ \ldots, \ g_n^{(1)}, \ldots, g_n^{(\mu(g_n))}\}.$$

For the expected value v of number $\mu(V)$ of elements of $\hat{G}$ that violate the random set R of $r = 6d^2$ elements, Corollary 6.4 gives

$$v \le d\,\frac{\mu(G) - r}{r+1} < \frac{d\mu(G)}{6d^2} = \frac{\mu(G)}{6d}.$$

From the Markov inequality we get that the expected number of calls to `Trivial` before the next successful iteration is at most 2.

It remains to bound the number of successful iterations.

Lemma 6.11. *Let k be a positive integer. After kd successful iterations, we have*

$$2^k \le \mu(B) \le \mu(G) \le ne^{k/3}$$

for every basis B of G. In particular, $k < 3 \ln n$.

Proof. Every successful iteration multiplies the total weight of elements in G by at most $(1 + 1/3d)$, which gives the upper bound. For the lower bound, we use Observation 6.8 to argue that each successful iteration doubles the weight of some element in B, meaning that after kd successful iterations, some element has been doubled at least k times. Because the left-hand side exceeds the right-hand side for $k \ge 3 \ln n$, the bound on k follows. ∎

Summarizing, we get the following lemma.

Lemma 6.12. *Algorithm* `Basis2` *computes a basis of G with an expected number of at most $6dn \ln n$ calls to Primitive 6.5, and an expected number of at most $6d \ln n$ calls to* `Trivial` *with sets of size at most $6d^2 + d$.*

6.6. Combining the Algorithms

Theorem 6.13. *Let $(H, \mathsf{V}, \mathcal{U})$ be a violator space with $-\infty$ of combinatorial dimension d. Let $n := |H|$. Using a combination of the above algorithms, a basis of H can be found using expected number of*

$$O\left(dn + d^{O(d)}\right)$$

calls to the primitive tests, provided that an initial basis $B_0 \notin \mathcal{U}$ is available.

Proof. Using the bound for the trivial algorithm, `Basis2`(G, B_0) can be implemented to require an expected number of at most

$$O\left(d \log |G| \left(|G| + d^{O(d)}\right)\right)$$

calls to the primitive tests. Applying this as a subroutine in $\texttt{Basis1}(H\setminus B_0, B_0)$, size of G is bounded by $3d\sqrt{n}$, and we get an overall number of calls to the primitives at most

$$O\left(dn + d^2(\log n(d\sqrt{n} + d^{O(d)}))\right)$$

The terms $d^2 \log n\, d\sqrt{n}$ and $d^2 \log n\, d^{O(d)}$ are asymptotically dominated either by dn or by $d^{O(d)}$, and we get the simplified bound of $O\left(dn + d^{O(d)}\right)$. ■

6.7. Need for the special care of unbounded sets

We feel that one thing concerning a behavior of unbounded sets deserves explicit mentioning.

Recall that we stop the algorithm when we find a set F with $\mathsf{V}(F) = \emptyset$. If $F \notin \mathcal{U}$ then locality implies that every basis of F is a basis of H. However, for $F \in \mathcal{U}$ this does not need to hold. In particular, we can construct LP-type problems of small fixed dimension with very large unbounded sets F with no violators.

For a positive integer n let

$$H := \{a_1, \ldots, a_n,\ b_1, \ldots, b_n,\ c_1, \ldots, c_n\}.$$

Define the weight mapping w as follows:

$$w(G) := \begin{cases} 0 & \text{if there exists } i \text{ with all } a_i, b_i, c_i \in G, \\ -\infty & \text{otherwise.} \end{cases}$$

One can easily verify that (H, w) is an LP-type problem by checking the conditions of Definition 1.1. Bases in this problem are the empty set and the sets $B_i := \{a_i, b_i, c_i\}$ for $i = 1, \ldots, n$, therefore $\dim(H, w) = 3$. However, the n-element set $\{a_1, \ldots, a_n\}$ has no violators.

6.8. If Clarkson's algorithm works, we have a violator space

One can ask whether one can invent yet another framework, possibly more general than violator spaces, for which the Clarkson's algorithm works and which admits an analysis of the running time similar to the one presented above. In this section we show that in a well-defined sense this is not possible. We identify a property that any abstract structure needs to satisfy if the analysis of the algorithm is applicable. Then we prove that this property already implies that we have a violator space.

For simplicity let us assume that all subproblems are bounded. Let us consider algorithm schemes that represent constraints by an abstract finite set and that use the violation test as a subroutine to access to the structure of the problem. We retain the notation H for the set of constraints and $\mathsf{V}(G)$ for the set of constraints violating the optimum solution with respect to G. Let us admit that the condition of consistency is justified by its own.

Now note that in the analysis of `Basis1` (Algorithm 6.7) we needed to bound the number of iterations of the loop, which we did using Observation 6.8. Gärtner (personal communication, June 2004) suggested to take the property resulting from Observation 6.8 as an axiom, and he conjectured that a pair (H, V) that satisfies consistency and this special axiom is necessarily a violator space. In this section we prove this conjecture.

Proposition 6.14. *Consider a set H and a mapping $\mathsf{V} : 2^H \to 2^H$ satisfying the following properties:*

- $\mathsf{V}(G) \cap G = \emptyset$ *for every $G \subseteq H$ (consistency);*
- *if a set $C \subseteq G$ satisfies $G \cap \mathsf{V}(C) \neq \emptyset$, then every $B \subseteq G$ with $\mathsf{V}(B) = \mathsf{V}(G)$ contains at least one element from $\mathsf{V}(C)$ (Gärtner's condition).*

Then (H, V) is a violator space.

We prove the proposition using a series of lemmas. First we give a slight reformulation of the Gärtner's condition that is more convenient to work with. Note that for the proof of Lemma 6.16 below we need both implications of the equivalence.

Lemma 6.15. *Let H be a finite set and let V be a mapping $2^H \to 2^H$. Assume that the pair (H, V) satisfies consistency. Then the Gärtner's condition is equivalent to the following statement:*

> *Let G be a subset of H and let F, C be subsets of G. Suppose that $\mathsf{V}(F) = \mathsf{V}(G)$ and $F \cap \mathsf{V}(C) = \emptyset$. Then $G \cap \mathsf{V}(C) = \emptyset$.*

Proof. Both implications of the equivalence follow easily by contradiction. ∎

From now on, whenever we refer to Gärtner's condition, we actually use the condition of Lemma 6.15. We continue with proving that a certain way of deriving subproblems preserves consistency and the Gärtner's condition.

Lemma 6.16. *Let a pair (H, V) satisfy consistency and the Gärtner's condition. For a fixed set $M \subseteq H$ we consider a problem in which the constraints of M are enforced. More precisely, we set $H' := H \setminus M$ and for $G \subseteq H'$ we define $\mathsf{V}'(G) := \mathsf{V}(G \cup M)$. Then the pair (H', V') satisfies consistency and the Gärtner's condition.*

Proof. Consistency is immediate from the definition of V' and consistency of V: we have $\mathsf{V}'(G) \cap G = \mathsf{V}(G \cup M) \cap G = \emptyset$.

We proceed by establishing the Gärtner's condition. Let C, F, G satisfy the hypotheses, i.e., $C, F \subseteq G \subseteq H$ with $\mathsf{V}'(F) = \mathsf{V}'(G)$ and $F \cap \mathsf{V}'(C) = \emptyset$; we want to deduce that $G \cap \mathsf{V}'(C) = \emptyset$. We have

$$(F \cup M) \cap \mathsf{V}(C \cup M) = (F \cap \mathsf{V}(C \cup M)) \cup (M \cap \mathsf{V}(C \cup M)) = F \cap \mathsf{V}'(C) = \emptyset;$$

$$\mathsf{V}(F \cup M) = \mathsf{V}'(F) = \mathsf{V}'(G) = \mathsf{V}(G \cup M).$$

By using the Gärtner's condition for V with the sets $C \cup M$, $F \cup M$, $G \cup M$ we get

$$(G \cup M) \cap \mathsf{V}(C \cup M) = \emptyset,$$

therefore $G \cap \mathsf{V}'(C) = \emptyset$. ∎

Lemma 6.17. *Let (H, V) be a pair satisfying consistency and the Gärtner's condition. Then for every $C \subseteq H$ we have*

$$\mathsf{V}(C) = \mathsf{V}(H \setminus \mathsf{V}(C)).$$

Consequently, if $\mathsf{V}(C) \neq H \setminus C$ then $\mathsf{V}(C) = \mathsf{V}(D)$ for some proper superset D of C.

Proof. We proceed by induction on $|H|$. If $H = \emptyset$, the statement of the lemma trivially holds.

Now let $|H| = n$ and assume that for sets smaller than n the statement holds. Let $C \subseteq H$ and put $F := H \setminus \mathsf{V}(C)$. We want to deduce that $\mathsf{V}(F) = \mathsf{V}(C)$.

From consistency we have $\mathsf{V}(F) \cap (H \setminus \mathsf{V}(C)) = \emptyset$, hence $\mathsf{V}(F) \subseteq \mathsf{V}(C)$. Let us assume that the inclusion is proper; i.e., $\mathsf{V}(F) \subset \mathsf{V}(C)$. We want to arrive at a contradiction.

We have $C \subseteq F$ from consistency, and since $\mathsf{V}(F) \neq \mathsf{V}(C)$, the inclusion is proper. In particular we get $|F| > 0$. We define V' by putting $\mathsf{V}'(X) := \mathsf{V}(X \cup F)$ for every $X \subseteq H \setminus F$. From Lemma 6.16 we infer that $(H \setminus F, \mathsf{V}')$ satisfies consistency and the Gärtner's condition. Therefore we can use the induction hypothesis to get

$$\mathsf{V}'(\emptyset) = \mathsf{V}'\big((H \setminus F) \setminus \mathsf{V}'(\emptyset)\big).$$

We set $D := (H \setminus F) \setminus \mathsf{V}'(\emptyset)$. We claim that $D \neq \emptyset$; otherwise we had $\mathsf{V}(F) = \mathsf{V}'(\emptyset) = H \setminus F = \mathsf{V}(C)$, which is not the case. Now we set $G := D \cup F$. We have $\mathsf{V}(F) = \mathsf{V}'(\emptyset) = \mathsf{V}'(D) = \mathsf{V}(G)$, moreover $C, F \subseteq G$, and finally $F \cap \mathsf{V}(C) = \emptyset$ by definition of F. Therefore the Gärtner's condition applies and gives $G \cap \mathsf{V}(C) = \emptyset$. On the other hand, we have $D \subseteq G$ by definition of G, and $D \subseteq H \setminus F = \mathsf{V}(C)$; therefore $D \subseteq G \cap \mathsf{V}(C) = \emptyset$. This is a contradiction with $D \neq \emptyset$. ∎

Now we are finally ready to prove that a pair (H, V) satisfying consistency and the Gärtner's condition is a violator space.

Proof of Proposition 6.14. It is sufficient to check that (H, V) satisfies locality. We assume that sets $P, Q \subseteq H$ satisfy $P \subseteq Q$ and $Q \cap \mathsf{V}(P) = \emptyset$, and we want to deduce that $\mathsf{V}(P) = \mathsf{V}(Q)$.

First we set $C := Q$, $F := P$, and $G := H \setminus \mathsf{V}(P)$. We are going to use the Gärtner's condition. We have $F \subseteq G$ from consistency, furthermore $C \subseteq G$ from the assumption $Q \cap \mathsf{V}(P) = \emptyset$, moreover $\mathsf{V}(F) = \mathsf{V}(G)$ by Lemma 6.17, and finally $F \cap \mathsf{V}(C) = \emptyset$ from $P \subseteq Q$ and consistency. Therefore the assumptions of the Gärtner's condition are satisfied, and we infer

$$(H \setminus \mathsf{V}(P)) \cap \mathsf{V}(Q) = \emptyset,$$

therefore $\mathsf{V}(Q) \subseteq \mathsf{V}(P)$.

Now we use Gärtner's condition with $C := P$, $F := Q$, and $G := H \setminus \mathsf{V}(Q)$. We have $F \subseteq G$ from consistency; $C \subseteq G$ from $P \subseteq Q$ and consistency; $\mathsf{V}(F) = \mathsf{V}(G)$ by Lemma 6.17; and $F \cap \mathsf{V}(C) = Q \cap \mathsf{V}(P) = \emptyset$. Therefore we obtain

$$(H \setminus \mathsf{V}(Q)) \cap \mathsf{V}(P) = \emptyset,$$

hence $\mathsf{V}(P) \subseteq \mathsf{V}(Q)$.

Since we proved both inclusions between $\mathsf{V}(P)$ and $\mathsf{V}(Q)$, we have the equality. Therefore locality holds. ∎

Chapter 7

Oriented matroid programming

In this chapter we introduce oriented matroids and oriented matroid programming (often abbreviated to OM programming). Oriented matroids are a mathematical abstraction arising among others in studies of convexity, point configurations, topology, and theoretical chemistry. OM programming captures properties of oriented matroids related to optimization. Linear complementarity problems, some convex programming problems, etc., may be expressed in terms of oriented matroid programming.

We prove that a wide class of OM programs gives rise to violator spaces with minus infinity in such a way that solving an OM program corresponds to finding a basis of the related violator space. In particular, Clarkson's algorithm is applicable and for OM programs of fixed rank runs in expected linear time.

We present only the part of the oriented matroid theory relevant to OM programming and its relation to models studied in this thesis. We deliberately omit other aspects and applications of the theory. In particular, we present only one of the many axiomatic systems. We encourage a reader interested in more information to read the introduction chapter [RGZ97] in a handbook, or the monograph [BLVS$^+$99].

We start with a particular simple linear program and we transform it, step by step, into an OM program. We try to keep the number of definitions as small as possible and we illustrate most of the definitions on the linear program. We show how OM programming encompasses some terms familiar from linear programming, like bounding cones or duality. In this part we omit all proofs; the reader can find them in [BLVS$^+$99].

The model linear program is given geometrically by Figure 7.1. The set of constraints is $H = \{a, b, c, d\}$. We are to optimize the value of $w : \mathbb{R}^2 \to \mathbb{R}$ given by the arrow representing the direction of improving[1] w.

Note that the arrangement is in a general position, in particular, no two boundary lines are parallel to each other and no boundary line is orthogonal to the optimization direction. This is not necessary to construct the OM program. However, degeneracy causes some problems when creating the related violator space.

[1] Since the goal in OM programming is usually maximizing whereas in LP-type models minimizing, we thoroughly use neutral words like 'improving' or 'optimize'.

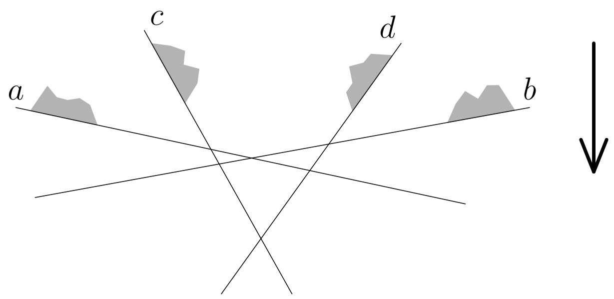

Figure 7.1. The model linear program

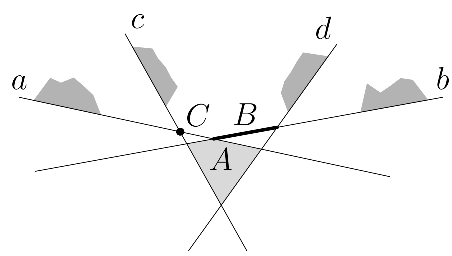

Figure 7.2. Faces of the arrangement

7.1. Oriented matroids

First we neglect the optimization nature of the problem and we keep only the arrangement of the constraints. We introduce oriented matroids as a combinatorial structure to model the arrangement.

Sign patterns. Every constraint h in the linear program corresponds to a hyperplane h_0 dividing the plane into two open halfspaces: the one where the constraint is satisfied and the one where the constraint is violated. Let us call the former one the positive side of the hyperplane and the latter one the negative side, denoted by h^+ and h^- respectively. Furthermore, let h_0^+ and h_0^- be the corresponding closed halfspaces.

The central idea is to identify every face of the arrangement by saying for each hyperplane which of the two halfspaces does contain the face or whether it is contained in the boundary hyperplane.

For instance, the face A in Figure 7.2 is on the positive side of the constraints c and d and on the negative side of the constraints a and b. We record this information by a sign pattern[2] whose components encode position of the face A with respect to a, b, c, and d, respectively. Thus the face A is represented by the pattern $(-,-,+,+)$, the edge B (which is a face as well) by the pattern $(+,0,+,+)$, the vertex C by the pattern $(0,+,0,+)$, etc. For a face X and a constraint $h \in H$, let X_h denote the h-component of the pattern representing X.

[2] By a sign pattern we mean a finite sequence whose elements are $+$, $-$, and 0. We intentionally avoid calling this a sign vector in order to prevent a confusion later.

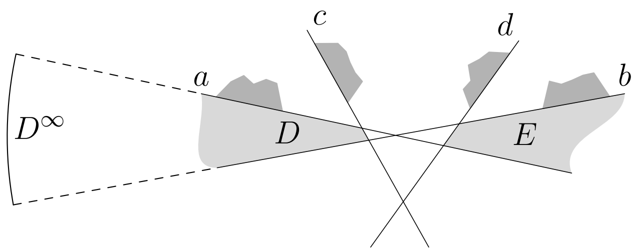

Figure 7.3. Unbounded and infinite faces

Extended sign patterns. The oriented matroid theory requires the following subtle modification. At first we present an informal description valid only for an arrangement in general position, and then we present a more precise description working in degenerate positions as well.

We extend the sign pattern of every face by one component, corresponding to a mythical constraint that we call g. For all faces of the arrangement, we set this component to $+$. Thus the face A will actually by represented by the extended pattern $(-,-,+,+,+)$ rather than by $(-,-,+,+)$.

Furthermore, for every unbounded face X of the arrangement we add a pattern having the last component 0. We interpret these patterns as faces in the infinity and we call them *infinite* faces (not to be confused with unbounded faces of the arrangement). For instance, the "very distant edge" of the face D is represented by the pattern $D^\infty = (-,+,-,+,0)$; see Figure 7.3.

Finally, we admit the existence of an "antiworld" containing the negative version of every face; for instance $-A = (+,+,-,-,-)$.

Note that the negative of an infinite face is an actual infinite face; for example, $-D^\infty = (+,-,+,-,0) = E^\infty$.

By convention, we define one more face consisting of all zeros, i.e., $(0,0,0,0,0)$.

Formally we introduce the additional constraint g in a way connected to projective geometry. Let us identify $\mathbb{R}^d$ with the affine subspace of $\mathbb{R}^{d+1}$ given by the equation $x_{d+1} = 1$. Now every constraint h determines a hyperplane $\bar{h}_0$ in $\mathbb{R}^{d+1}$ given as the affine span of $h_0 \cup \{\mathbf{0}\}$. This $\bar{h}_0$ divides $\mathbb{R}^{d+1}$ into halfspaces $\bar{h}^+$ and $\bar{h}^-$ containing the "halfhyperplanes" h^+ and h^- respectively. For the mythical constraint g we define

$$\bar{g}^+ := \big\{(x_1,\dots,x_{d+1}) : x_{d+1} > 0\big\}.$$

By extended sign patterns of the original arrangement we mean the sign patterns of faces of the new arrangement. In the new arrangement all the boundary hyperplanes intersect (in the point $\mathbf{0}$), which explains why we accept the all-zero pattern as a face.

For the two-dimensional case we can provide the following intuitive interpretation. We embed the plane carrying the arrangement into a three-dimensional space and we consider a sphere touching the plane by its north pole. We map every point X in the plane to a pair of antipodal points $\psi^{\mathrm{N}}(X), \psi^{\mathrm{S}}(X)$ on the sphere: we draw a line ℓ determined by X and the center of the sphere, and we define $\psi^{\mathrm{N}}(X)$

and $\psi^{\mathrm{S}}(X)$ as intersections of ℓ with the northern and the southern hemisphere respectively. As an image of a line in the arrangement we get a great circle. To a halfplane containing a point X we assign the hemisphere containing $\psi^{\mathrm{N}}(X)$. We obtained an arrangement of hemispheres; the mythical constraint g corresponds to the northern hemisphere.

We call the extended sign patterns of faces the *covectors*. Note that not all sign patterns are covectors. For instance $(+,-,-,+,+)$ is not a covector, since the open hyperplanes a^+, b^-, c^-, and d^+ have empty intersection. Similarly, $(0,0,0,+,+)$ is not a covector, since the lines a_0, b_0, and c_0 do not meet in a single point.

All the covectors of the model example are listed here:

$++-++$,	$+++++$,	$+++-+$,	$-+-++$,	$-++++$,	$+-+++$,	$+-+-+$,
$---++$,	$--+++$,	$--+-+$,	$----+$,	$0+-++$,	$++0++$,	$0++++$,
$+0+++$,	$+++0+$,	$+0+-+$,	$-0-++$,	$-+0++$,	$-0+++$,	$0-+++$,
$+-+0+$,	$0-+-+$,	$--0++$,	$--+0+$,	$--0-+$,	$---0+$,	$0+0++$,
$+0+0+$,	$00+++$,	$-00++$,	$0-+0+$,	$--00+$,	$++-+0$,	$++0+0$,
$++++0$,	$+++00$,	$+++-0$,	$+0+-0$,	$+-+-0$,	$0-+-0$,	$--+-0$,
$--0-0$,	$----0$,	$---00$,	$---+0$,	$-0-+0$,	$-+-+0$,	$0+-+0$,
$--+--$,	$-----$,	$---+-$,	$+-+--$,	$+----$,	$-+---$,	$-+-+-$,
$+++--$,	$++---$,	$++-+-$,	$++++-$,	$0-+--$,	$--0--$,	$0----$,
$-0---$,	$---0-$,	$-0-+-$,	$+0+--$,	$+-0--$,	$+0---$,	$0+---$,
$-+-0-$,	$0+-+-$,	$++0--$,	$++-0-$,	$++0+-$,	$+++0-$,	$0-0--$,
$-0-0-$,	$00---$,	$+00--$,	$0+-0-$,	$++00-$,	00000.	

Axioms for oriented matroids. The extended sign patterns of faces satisfy some important properties that are taken as axioms in the definition of oriented matroids. In the following, we state the properties and we give an interpretation for the nontrivial ones.

Axiom (A0). *The pattern* $(0,\ldots,0)$ *is a covector.*

Axiom (A1). *If* X *is a covector then its negative* $-X$ *is a covector.*

These two properties are trivial, they reflect the conventions mentioned above.

Axiom (A2). *If* X, Y *are covectors,* $X \circ Y$ *defined as follows is also a covector:*

$$(X \circ Y)_h = \begin{cases} X_h & \text{if } X_h \neq 0, \\ Y_h & \text{if } X_h = 0. \end{cases}$$

In other words, if we replace zero components in a covector by the corresponding components from another covector, we get a covector.

To interpret (A2), we consider a face X that is not of a full dimension (i.e., it is contained in some of the boundary hyperplanes, namely these corresponding to zero components of X), and we make a tiny step in the direction of a face Y; see Figure 7.4. The property (A2) guarantees that we enter a valid face Z.

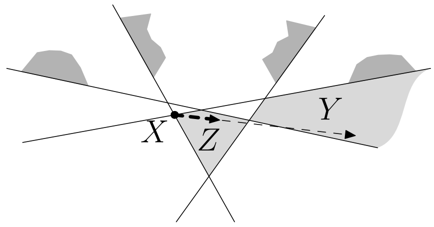

Figure 7.4. Interpretation of Axiom (A2)

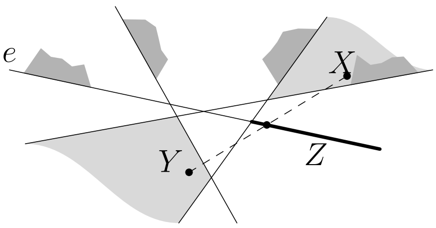

Figure 7.5. Interpretation of Axiom (A3)

Axiom (A3). *Let X, Y be covectors and let $e \in H$ be an index such that $X_e = +$, $Y_e = -$. Then there exists a covector Z such that*

- $Z_e = 0$,
- *for every $h \in H$ with $Z_h \neq 0$ we have $Z_h = X_h$ or $Z_h = Y_h$,*
- *for every $h \in H$ with $Z_h = 0$ we have $X_h = -Y_h$.*

The axiom (A3) applies in a situation when two faces X, Y are separated by a constraint e; that is, the face X is on the positive side of the boundary hyperplane and the face Y on the negative side. The axiom asserts that there exists a face Z contained in the boundary hyperplane (this is guaranteed by the condition $Z_e = 0$) and lying between X and Y. The exact meaning of "between" is specified in the statement.

Geometrically we can find a suitable face Z as follows. We connect any point in X with any point in Y with a straight line ℓ. We let Z be the face of the arrangement containing the intersection of ℓ with e_0; see Figure 7.5. Note that sometimes we can get different faces for different choices of points in X and Y.

In the theory of oriented matroids, there are several equivalent variants for the axiom (A3). The version presented here is called *covector elimination.*

We conclude with the formal definition of oriented matroid.

Definition 7.1. *Let E be a finite set. By a* sign pattern *over E we mean a mapping $F : E \to \{+, -, 0\}$. Let $\mathcal{F}$ be a set of sign patterns over E called* covectors. *If the axioms (A0), (A1), (A2), and (A3) are satisfied then we call the pair $(E, \mathcal{F})$ an* oriented matroid.

We remark that there is a wide class of oriented matroids called *non-realizable* that cannot be represented by an arrangement of oriented hyperplanes.

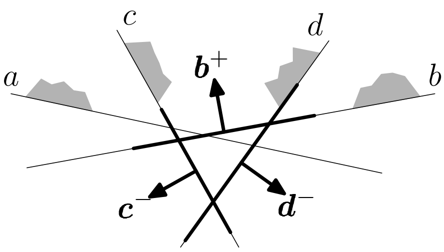

Figure 7.6. Minimum nonintersecting system of halfspaces representing the vector $(0, +, -, -, +)$

Dual oriented matroid. For every oriented matroid M there exists a dual oriented matroid M^*. This duality is related to the linear programming duality and it provides a background for some terminology useful for discussing oriented matroid programming. Our presentation introduces the duality as a tool for describing nonintersecting sets of halfspaces.

In our model example we see that the closed halfspaces b_0^+, c_0^-, and d_0^- do not intersect; see Figure 7.6. This means that the arrangement does not contain a face contained in all of b_0^+, c_0^-, and d_0^-. In other words, every face of the arrangement is contained in at least one of b^-, c^+, or d^+. In the oriented matroid language we can say that every covector F corresponding to an actual face (i.e., with $F_g = +$) satisfies $F_b = -$ or $F_c = +$ or $F_d = +$.

We represent the nonintersecting configuration by a sign pattern W. We set the components corresponding to the individual halfspaces in the configuration to $+$ or $-$ depending on which of the two halfspaces we have. In the components corresponding to hyperplanes absent in the configuration we take 0. In the last component we take $+$. For the our configuration we get $W = (0, +, -, -, +)$.

Now we can say that for every covector F with $F_g = +$ there exists a component h with $W_h \neq 0$ and $F_h = -W_h$. If we have a covector F with $F_g = -$, by negating F and referring to the previous case we can see that there exists a component h with $W_h \neq 0$ and $F_h = W_h$. We call such patterns F and W orthogonal. A pattern W orthogonal to all covectors is called a vector, and the vectors of M form the dual oriented matroid.

Definition 7.2. *We say that the sign patterns F, W are* orthogonal *if one of the following holds:*

- *there exists a component h with $F_h = W_h \neq 0$ and a component h' with $F_{h'} = -W_{h'} \neq 0$; or*
- *for every component h, either $F_h = 0$ or $W_h = 0$.*

By vectors *of an oriented matroid M we mean sign patterns orthogonal to all covectors of M. The* dual oriented matroid *M^* is the oriented matroid whose covectors are the vectors of M.*

In our example, the vectors of the oriented matroid are the following:

$$\begin{array}{ccccccc}
{+}{+}{-}{-}{+}, & {+}{-}{-}{+}{+}, & {+}{-}{-}{-}{+}, & {-}{+}{+}{-}{+}, & {-}{+}{-}{-}{+}, & 0{+}{-}{-}{+}, & {+}0{-}{-}{+}, \\
{-}{+}0{-}{+}, & {+}{-}{-}0{+}, & {-}{-}{+}{+}{-}, & {-}{+}{+}{-}{-}, & {-}{+}{+}{+}{-}, & {+}{-}{-}{+}{-}, & {+}{-}{+}{+}{-}, \\
0{-}{+}{+}{-}, & {-}0{+}{+}{-}, & {+}{-}0{+}{-}, & {-}{+}{+}0{-}, & {+}{-}{-}{+}0, & {-}{+}{+}{-}0, & 00000.
\end{array}$$

Rank. Consider an oriented matroid corresponding to an arrangement of at least d halfspaces in d-dimensional space. Let W be a nonzero vector with the minimum number k of nonzero components. For simplicity assume that $W_g = +$. Consider the set $\mathcal{H}$ of open halfspaces corresponding to the remaining nonzero components of W. We have $|\mathcal{H}| = k - 1$. Since W is a vector, the intersection of the halfspaces is empty. On the other hand, from minimality of W we get that after removing any single halfspace, the remaining $k-2$ have some point in common; see Figure 7.6. In this situation the famous Helly theorem from combinatorial geometry implies that $k - 2 \leq d$. This relates the dimension of the arrangement to the number of nonzero components in vectors, and justifies the following definition.

Definition 7.3. *The* rank *of an oriented matroid M is $k - 1$, where k is the minimum number of nonzero components of a nonzero vector of M.*

In particular, the oriented matroid representing an arrangement of many half-spaces in $\mathbb{R}^d$ in general position has rank $d + 1$.

Uniform oriented matroids. The oriented matroids corresponding to arrangements in general position are called *uniform*. Geometrically, in an arrangement of hyperplanes in $\mathbb{R}^d$ in general position at most d hyperplanes meet in a single point. Formally, we say that an oriented matroid $M = (E, \mathcal{F})$ is uniform if every nonzero covector has fewer than $\operatorname{rank}(M)$ zero components. From the theory of nonoriented matroids then follows that every nonzero vector has at least $(\operatorname{rank}(M) + 1)$ nonzero components; that is, at most $(|E| - \operatorname{rank}(M) - 1)$ zero components. The oriented matroid corresponding to our model example is uniform.

7.2. Oriented matroid programming

Now when we know how to translate some basic geometric terms to the oriented matroid language, let us proceed with showing how are oriented matroids related to optimization. Recall that our optimization problem is to find a feasible point x for which the value of $w(x)$ is optimized.

It is quite obvious how we can define the feasible region of the problem. Geometrically, it is the area contained in all of the closed halfspaces corresponding to the constraints in question. If a face is not contained in some halfspace, we have some $-$ component in the corresponding sign pattern. We therefore say that a covector is feasible if it does not contain any $-$. We are interested in the actual faces only, not in the antifaces or the infinite faces, so the extra g-th component in the extended sign pattern has to be $+$. If we want to admit the infinite faces, we allow the g-component to be 0.

Now we continue with the less obvious part. We need to express the optimization direction. To do this, we add a suitable auxiliary constraint f to the arrangement; see Figure 7.7. We choose f_0 to be orthogonal to the optimization direction, and we orient f so that the positive side corresponds to improving of w:

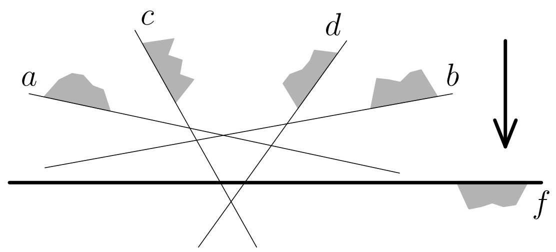

Figure 7.7. Auxiliary hyperplane representing the optimization direction

$$f_0 := \{x : w(x) = 0\},$$
$$f^+ := \{x : w(x) \text{ is better than } 0\}, \quad f^- := \{x : w(x) \text{ is worse than } 0\}.$$

We do not consider f when we determine feasibility of faces. We claim that combinatorial properties of the oriented hyperplane f carry enough information about the optimization direction so that we can identify the optimum solution.

Every line not parallel to f_0 has one "end" in f^+ (the *positive* end) and the other "end" in f^- (the *negative* end). Formally we can represent an end of a line by a suitable covector Z with $Z_g = 0$. The value of a point on the line improves as it moves towards the positive end. Thus having for instance two adjacent vertices of the arrangement, we can compare their value by considering the line determined by the vertices and favoring the vertex closer to the positive end of the line. If the value of w in a vertex is better than the values in all of its neighbors, we can say that the vertex represents the optimum solution.

More generally, we say that a covector Z is an improving direction for a face X if $Z_g = 0$, $Z_f = +$, and after making a small step from X in the direction of Z we still remain feasible; i.e., $X \circ Z$ is feasible. A covector X represents an optimum solution if it is feasible and does not have an improving direction. The problem is unbounded if there exists an infinite face Z on the positive side of f satisfying all constraints.

We formulate the following formal definition in a way that makes convenient to talk about solving problems determined by a subset of constraints.

Definition 7.4. *Let $M = (E, \mathcal{F})$ be an oriented matroid. Fix two elements $f, g \in E$. We call the elements of the set $H := E \setminus \{f, g\}$* constraints. *To avoid trivialities, we assume that there exist some covector X of M with $X_g \neq 0$, and some vector W of M with $W_f \neq 0$. Then the triple (M, g, f) is called an* oriented matroid program.

We say that a covector X is a feasible solution *with respect to constraints $G \subseteq H$, if $X_g = +$ and $X_h \geq 0$ for all $h \in G$. We say that a covector Z is an* improving direction *for a feasible solution X with respect to constraints $G \subseteq H$ if $Z_g = 0$, $Z_f = +$, and $Z_h \geq 0$ for all $h \in G$ with $X_h = 0$. We say that a covector X* optimizes *f with respect to constraints $G \subseteq H$, if X is feasible and admits no improving direction with respect to G.*

We say that a set of constraints $G \subseteq H$ is feasible *if there exists a covector X feasible with respect to G. We say that a covector Z is an* unbounded direction

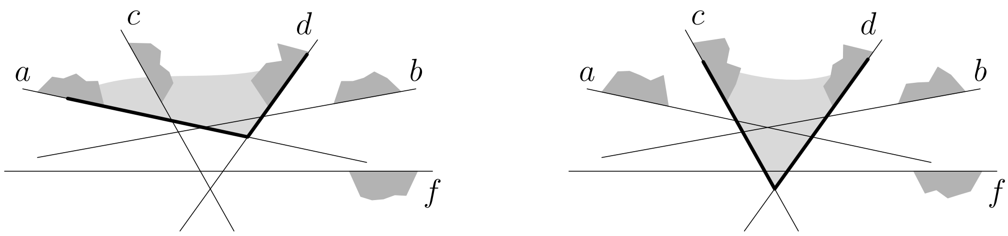

Figure 7.8. Bounding cones represented by vectors $(+, 0, 0, +, +, +)$ (left) and $(0, 0, +, +, -, +)$ (right)

with respect to constraints $G \subseteq H$ if $Z_g = 0$, $Z_f = +$, and $Z_h \geq 0$ for all $h \in G$.

The goal of the OM program is to find a covector X optimizing f with respect to $E \setminus \{f, g\}$, or to exhibit an unbounded direction, or to determine that H is infeasible.

We say that the OM program (M, g, f) is nondegenerate *if M is uniform.*

By an OM program dual *to $P = (M, g, f)$ we mean the OM program $P^* = (M^*, f, g)$.*

Bounding cones. An important role in the theory of duality in OM programming is played by bounding cones. These essentially correspond to dual feasible bases in linear programming. We think of a bounding cone as an inclusion-minimal system of constraints that does not admit any infinite face X with $X_f = +$. We represent bounding cones by vectors of the OM program.

For instance in our model problem, the constraints a and d form a bounding cone; see Figure 7.8 left. Since the intersection $a^+ \cap d^+ \cap f^+$ is empty, every face X with $X_a \geq 0$ and $X_d \geq 0$ has $X_f = -$. We represent this bounding cone by the vector $(+, 0, 0, +, +, +)$.

As another example take the bounding cone formed by the constraints c and d; see Figure 7.8 right. The representation of this cone by a vector is more complicated than in the previous example, since we need a suitable intersection to be empty, and here $c^+ \cap d^+ \cap f^+$ is nonempty. Fortunately, the intersection of the opposite halfspaces, that is $c^- \cap d^- \cap f^-$, is empty. The oriented matroid program therefore possesses a vector $W = (0, 0, -, -, +, -)$. By convention, we represent the bounding cone by $-W = (0, 0, +, +, -, +)$.

Formally, in an OM program (M, g, f) we define a bounding cone with respect to a set of constraints G as a vector W with inclusion-minimal set of nonzero components for which $W_h \geq 0$ for every $h \in H$, moreover $W_h = 0$ for every $h \in H \setminus G$, and finally $W_f = +$.

Note that in a nondegenerate OM program, every bounding cone W has exactly $(\mathrm{rank}(M) + 1)$ nonzero elements, two of them being W_f and W_g.

We say that an OM program is *bounded* if there is a bounding cone containing all feasible faces. We remark that an infeasible OM program is bounded if there exists some bounding cone at all; there are examples of both bounded infeasible and unbounded infeasible OM programs.

Note that the bounding cones in an OM program are exactly feasible solutions to the dual OM program.

Using bounding cones to prove optimality. Let W be a bounding cone and let X be the "tip" of the cone, that is, the face contained in all of the bounding hyperplanes of the cone. We can prove that if X is a feasible solution then it is the optimum. Formally, suppose that we have a feasible solution X and a bounding cone W such that for every $h \in H$ either X_h or W_h is 0; then X is an optimum solution. To prove this, we assume for contradiction that X has an improving direction, i.e., a covector Z with $Z_g = 0$, $Z_f = +$, and $Z_h \geq 0$ for all $h \in H$ with $X_h = 0$. Since W is a vector, it is orthogonal to Z; in particular, $Z_h = -W_h \neq 0$ for some $h \in H$ (note that $Z_f = + = W_f$ and $Z_g = 0$). Because W is nonnegative, we have $Z_h = -$ and $W_h = +$. Since $W_h = +$, the relation of W and X implies that $X_h = 0$, hence the properties of Z give $Z_h = +$, which is a contradiction.

Infeasible and unbounded OM programs. The OM program (M, g, f) is infeasible if for some set of constraints $G \subseteq H$ the intersection of closed halfspaces corresponding to G is empty. We know that such a situation is described by some vector W with $W_g = +$, furthermore $W_h = +$ for every constraint $h \in G$, and $W_h = 0$ for the remaining constraints h. Since the auxiliary hyperplane f is not involved in the nonintersecting system, we have $W_f = 0$. More formally, we have the following equivalence:

The OM program (M, g, f) is infeasible if and only if it has some vector W with all components nonnegative, $W_g = +$, and $W_f = 0$.

Note how infeasibility is dual to unboundedness. The OM program is unbounded if and only if it has some feasible infinite face with "infinitely good" value, in other words, if there exists a covector X with all components nonnegative and furthermore $X_f = +$ and $X_g = 0$.

Theorems on OM programming. Now we state two important theorems concerning OM programming.

Theorem 7.5 (Main theorem of OM programming). *For every OM program (M, g, f) exactly one of these conditions holds:*

(i) *(M, g, f) has an optimum solution,*
(ii) *(M, g, f) has an unbounded direction,*
(iii) *(M, g, f) is not feasible.*

Theorem 7.6 (Duality theorem for OM programming). *Let $P = (M, g, f)$ be an OM program. Then exactly one of the following two statements holds.*

(i) *The program P is infeasible, because there is a nonnegative vector W with $W_g = +$, $W_f = 0$; or the program P is unbounded, because there is a nonnegative covector X with $X_f = +$, $X_g = 0$; or both.*
(ii) *There exists a feasible solution X and a bounding cone W such that for every $h \in E(M) \setminus \{f, g\}$ we have $X_h = 0$ or $W_h = 0$, implying that X is an optimum solution to P and W is an optimum solution to P^*. If (M, g, f) is nondegenerate then X and W are determined uniquely.*

7.3. Relation between OM programs and violator spaces

In this section we show how we can use Clarkson's algorithm for violator spaces to solve nondegenerate oriented matroid programs. In particular, for a given nondegenerate OM program (M, g, f) we construct a violator space (H, V), we determine its combinatorial dimension, we present a relation between the basis of H in the violator space and the optimum solution to the OM program, and we show how to implement the computational Primitives 6.5 and 6.6.

We start with the construction of the violator space.

Definition 7.7. *Consider a nondegenerate oriented matroid program* (M, g, f), *where* $M = (E, \mathcal{F})$. *We define a violator space with minus infinity* $\mathrm{Vio}(M, g, f) := (H, \mathsf{V}, \mathcal{U})$ *by setting* $H := E \setminus \{f, g\}$ *and defining* $\mathcal{U}$ *and* V *as follows:*

$$\mathcal{U} := \{G \subseteq H : G \text{ has an unbounded direction}\},$$

$$\mathsf{V}(G) := \begin{cases} \emptyset & \text{if } G \text{ is not feasible,} \\ \{h \in H : G \cup \{h\} \notin \mathcal{U}\} & \text{if } G \in \mathcal{U}, \\ \{h \in H : X_h = -\} & \text{if the optimum solution } X \text{ wrt. } G \text{ exists.} \end{cases}$$

Note that in the last case the optimum X is determined uniquely, since we assume nondegeneracy.

We continue with a series of lemmas describing the relations between the violator space and the OM program.

Lemma 7.8. *The structure* $(H, \mathsf{V}, \mathcal{U})$ *defined above is indeed a violator space with minus infinity.*

Proof. Of the conditions in the definition of violator space with $-\infty$ we immediately see that V matches $\mathcal{U}$. It remains to prove consistency, monotonicity, and locality.

First let us prove consistency, that is, $G \cap \mathsf{V}(G) = \emptyset$ for every $G \subseteq H$. If G is unbounded, this follows from definition of V. If G is infeasible, $G \cap \mathsf{V}(G) \subseteq \mathsf{V}(G) = \emptyset$. If G has an optimum solution X then $X_h \geq 0$ for every $h \in G$ from feasibility of X, and $G \cap \mathsf{V}(G) = \emptyset$ follows from definition of V.

To prove monotonicity, consider a set $G \subseteq H$ admitting an unbounded direction Z, and a subset $F \subseteq G$. We have $Z_g = 0$, $Z_f = +$, and $Z_h \geq 0$ for all $h \in F \subseteq G$, therefore Z is an unbounded direction for F as well.

Now it remains to prove locality. Let $F \subseteq G \subseteq H$ with F bounded. We want to deduce that $\mathsf{V}(F) = \mathsf{V}(G)$. By monotonicity, G is bounded. The further action depends on the feasibility of F.

First let F be infeasible. In this case, a solution X feasible with respect to G would be feasible for F too. Therefore G is infeasible too. The definition of V now gives $\mathsf{V}(F) = \emptyset = \mathsf{V}(G)$.

Now let F be feasible. Let X be the optimum solution with respect to F. The assumption $G \cap \mathsf{V}(F) = \emptyset$ together with feasibility with respect to F imply that X is feasible with respect to G. From the assumption $F \subseteq G$ we infer that the bounding cone Y associated to X for F works as a bounding cone for G too. Therefore X is the optimum solution with respect to G. Since the definition of V depends only on the optimum solution X, we get $\mathsf{V}(F) = \mathsf{V}(G)$ as desired. ∎

Lemma 7.9. *If (M, g, f) is a nondegenerate OM program then the combinatorial dimension of* $\text{Vio}(M, g, f)$ *is* $(\text{rank}(M) - 1)$ *if (M, g, f) is bounded and feasible, and* $\text{rank}(M)$ *if (M, g, f) is infeasible. Moreover, in the latter case we have $|B| =$* $\text{rank}(M)$ *only for infeasible bases.*

Proof. First we prove that every bounded feasible basis has $(\text{rank}(M)-1)$ elements. Let B be such a basis. Let X denote the optimum solution with respect to B.

We claim that B is the inclusion-minimal system for which X is the optimum solution. Let C be a proper subset of B. Let Y by the optimum solution determined by C. By properties of bases in violator spaces we have $B \cap \mathsf{V}(C) \neq \emptyset$, therefore there is a constraint $h \in B$ with $Y_h = -$. This means that $X \neq Y$, which proves the claim in the beginning of this paragraph.

Now let W be the bounding cone proving optimality of X with respect to B. We have $h \in B$ for every $h \in H$ with $W_h = +$. We claim that $B = \{h \in H : W_h = +\}$. The inclusion $\supseteq$ follows from the definition of bounding cone. To prove the inclusion $\subseteq$ we proceed by contradiction. Assume that $C := \{h \in H : W_h = +\} \subset B$. Since B is a basis, we have $B \cap \mathsf{V}(C) \neq \emptyset$, hence there is a constraint $h \in B$ with $X_h = -$. However, this contradicts feasibility of X with respect to B.

We therefore have $B = \{h \in H : W_h = +\}$. Moreover $W_f = +$ since W is a bounding cone, and $W_g \neq 0$ from uniformity. This gives $|B| = \text{rank}(M) - 1$.

Now we prove that every infeasible basis has $\text{rank}(M)$ elements. Let B be such a basis. By properties of bases in violator spaces, B is an inclusion-minimal infeasible system of constraints. We define a sign pattern W as $W_h := +$ for all $h \in B$, furthermore $W_h := 0$ for all $h \in H \setminus B$, and finally $W_f := 0$ and $W_g := +$. Now W is a vector with an inclusion-minimal set of nonzero components; their number is $|B| + 1$ by definition of W and uniformity, and on the other hand $(\text{rank}(M) + 1)$ from properties of the rank. Therefore $|B| = \text{rank}(M)$. ∎

Lemma 7.10. *Let (M, g, f) be a nondegenerate OM program. Let B be a basis of H in the violator space $(H, \mathsf{V}, \mathcal{U}) =$* $\text{Vio}(M, g, f)$*. Then B is related to the solution to the OM program in the following way:*

- *$B \in \mathcal{U}$ if and only if the OM program is unbounded;*
- *$B \notin \mathcal{U}$ and B is infeasible if and only if the OM program is infeasible;*
- *$B \notin \mathcal{U}$ and B is feasible if and only if the program has the optimum solution. In this case, the optimum with respect to B is the optimum for the whole program.*

Proof. If the problem is unbounded then $B \in \mathcal{U}$, and vice versa, by the definition of $\mathcal{U}$.

In the remaining cases, we have $\mathsf{V}(B) = \mathsf{V}(H) = \emptyset$.

First let us examine the infeasible case. If B is infeasible then the whole OM program is clearly infeasible. In the other direction we proceed indirectly. If B is feasible, let X be the optimum with respect to B. Since $\mathsf{V}(B) = \emptyset$, the face X is feasible with respect to H, which proves that the OM program is feasible.

The case of bounded feasible program remains. We need to prove that the optimum for B is optimum for the whole program. Let X denote the optimum with respect to B. As before, we have $\mathsf{V}(B) = \emptyset$, hence X is feasible with respect to H. The duality theorem asserts the existence of a bounding cone W that proves the optimality of X with respect to B. Then W is a bounding cone for H since $B \subseteq H$. This concludes the proof of optimality of X. ∎

Implementing the primitives. To be able to use Clarkson's algorithm for violator spaces as described in Chapter 6, we need to implement violation and boundedness test. We show how to do this if the OM program is specified by the following subroutine.

Primitive 7.11. *For a given $G \subseteq H$ and $h \in H$ such that $|G| \leq \operatorname{rank}(M)$, solve the OM program with respect to G. If the problem is infeasible or unbounded, return this information. Otherwise return the h-th component of the optimum solution.*

To implement the boundedness test, i.e., to determine whether a set $F \subseteq H$ with $|F| \leq d$ is bounded, we simply invoke Primitive 7.11 with $G := F$ and arbitrary $h \in H$.

To implement the violation test, i.e., to determine whether a set $F \subseteq H$ with $|F| \leq d$ is violated by $e \in H \setminus F$, we invoke Primitive 7.11 with $G := F$ and $h := e$. If the problem with respect to F is unbounded, we call Primitive 7.11 once more with $G := F \cup \{e\}$ and arbitrary $h \in H$; we have $e \in \mathsf{V}(F)$ if and only if the problem is bounded. If the problem with respect to F is infeasible, we have $\mathsf{V}(F) = \emptyset$, hence $e \notin \mathsf{V}(F)$. If the problem is bounded and feasible, we have $e \in \mathsf{V}(F)$ if and only if $X_e = -$ (where X_e is returned by Primitive 7.11).

Note that from Lemma 7.9 follows that we call Primitive 7.11 with sets G of size at most $\operatorname{rank}(M)$.

Algorithm. Now we can apply Theorem 6.13 to the violator space corresponding to the oriented matroid program.

Proposition 7.12. *Let (M, g, f) be a nondegenerate OM program. Clarkson's algorithm can solve (M, g, f) using expected number of $O(dn + d^{O(d)})$ calls to Primitive 7.11, provided that an initial bounded basis is available.*

Proof. Clarkson's algorithm finds the basis B of H in $\operatorname{Vio}(M, g, f)$ in the stated time. By calling Primitive 7.11 once more with $G := B$ we determine whether (M, g, f) with respect to B is infeasible or we find the optimum solution. Note that the problem is not unbounded, since it possesses an initial bounded basis. By Lemma 7.10, the result with respect to B is correct for the whole OM program. ∎

Degenerate OM programs. We conjecture that the above result can be modified to work for degenerate OM programs. Fukuda (personal communication, May 2007) suggested using the method of lexicographic perturbations [FLN97].

Moreover we believe that if every bounded feasible set $G \subseteq H$ has a unique optimum then the definition of the violator space $\operatorname{Vio}(M, g, f)$ and the algorithm do not need to be modified at all, just the proofs of the Lemmas 7.8, 7.9 and 7.10 have to be changed.

Acyclicity. In some oriented matroid programs we can construct cyclic sequences of vertices with improving values. This is essentially the same phenomenon as cyclicity in violator spaces. OM programs that admit such cyclic sequences are called *noneuclidean*. An example of a noneuclidean oriented matroid program $\mathcal{P}$ was first exhibited by Fukuda [Fuk82] and Mandel [Man82]. We omit the formal definition needing some preparatory work.

The example of basis-regular nondegenerate cyclic violator space in Chapter 4 was obtained by a slight modification of $\mathrm{Vio}(\mathcal{P}^*)$, where $\mathcal{P}^*$ is the dual of the Fukuda's and Mandel's noneuclidean OM program $\mathcal{P}$. The small modification was necessary to remove the $-\infty$.

We conjecture that if an OM program (M, g, f) is Euclidean then the violator space $\mathrm{Vio}(M, g, f)$ is acyclic. The converse statement does not hold; as an counterexample one can take the primal Fukuda's and Mandel's problem $\mathcal{P}$.

This is the end of the thesis. Thank you for reading it. I am pleased if you liked it.

References

For the convenience of the reader, the PDF version of the thesis, which is available at `http://kam.mff.cuni.cz/~xofon/thesis/`, contains hypertext links to the referenced works if available on the Web, to conference versions, or to other appropriate sites.

[Ame94] Nina Amenta. Helly theorems and generalized linear programming. *Discrete and Computational Geometry*, 12:241–261, 1994. (Cited on pages vii, 9.)

[Ame96] Nina Amenta. A short proof of an interesting Helly-type theorem. *Discrete and Computational Geometry*, 15:423–427, 1996. (Cited on page vii.)

[BLVS+99] Anders Björner, Michel Las Vergnas, Bernd Sturmfels, Neil White, and Günter M. Ziegler. *Oriented matroids*. Cambridge University Press, Cambridge, 2nd edition, 1999. (Cited on pages 49, 71.)

[BSV03] Henrik Björklund, Sven Sandberg, and Sergei Vorobyov. A discrete subexponential algorithm for parity games. In *Proc. 20th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 663–674, 2003. (Cited on page vii.)

[Cha04] Timothy M. Chan. An optimal randomized algorithm for maximum Tukey depth. In *Proc. 15th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 423–429, 2004. (Cited on page vii.)

[Chv83] Vašek Chvátal. *Linear programming*. W. H. Freeman and Company, New York, 1983. (Cited on page 2.)

[Cla93] Kenneth L. Clarkson. A bound on local minima of arrangements that implies the upper bound theorem. *Discrete and Computational Geometry*, 10:427–433, 1993. (Cited on page 54.)

[Cla95] Kenneth L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. *Journal of the ACM*, 42:488–499, 1995. (Cited on pages vii, 62.)

[CM96] Bernard Chazelle and Jiří Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *Journal of Algorithms*, 21:579–597, 1996. (Cited on pages vii, 62.)

[Fis05] Kaspar Fischer. *Smallest enclosing balls of balls*. PhD thesis, ETH Zürich, 2005. (Cited on pages viii, 9.)

[FLN97] Komei Fukuda, Hans-Jakob Lüthi, and Makoto Namiki. The existence of a short sequence of admissible pivots to an optimal basis in LP and LCP. *International Transactions in Operational Research*, 4:273–284, 1997. (Cited on page 83.)

[Fuk82] Komei Fukuda. *Oriented matroid programming*. PhD thesis, University of Waterloo, 1982. (Cited on page 84.)

[Gär95] Bernd Gärtner. *Randomized Optimization by Simplex-Type Methods*. PhD thesis, Freie Universität Berlin, 1995. (Cited on page 8.)

[GKP89] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete mathematics: A foundation for computer science*. Addison-Wesley, 1989. (Cited on pages 54, 58.)

[GMR] Bernd Gärtner, Walter D. Morris, Jr., and Leo Rüst. Unique sink orientations of grids. Algorithmica, to appear. (Cited on pages viii, 49.)

[GMRŠ06] Bernd Gärtner, Jiří Matoušek, Leo Rüst, and Petr Škovroň. Violator spaces: Structure and algorithms. In *Proc. 14th European Symposium on Algorithms (ESA)*, pages 387–398, 2006. (Cited on pages vii, viii, ix, 10, 12, 18, 51.)

[GS06] Bernd Gärtner and Ingo Schurr. Linear programming and unique sink orientations. In *Proc. 17th Symposium on Discrete Algorithms (SODA)*, pages 749–757, 2006. (Cited on page viii.)

[GW96] Bernd Gärtner and Emo Welzl. Linear programming—randomization and abstract frameworks. In *Proc. 13th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 669–687, London, UK, 1996. (Cited on pages vii, 62.)

[GW01] Bernd Gärtner and Emo Welzl. A simple sampling lemma—analysis and applications in geometric optimization. *Discrete and Computational Geometry*, 25(4):569–590, 2001. (Cited on pages 54, 63.)

[Hal04] Nir Halman. On the power of discrete and lexicographic Helly theorems. In *Proc. 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 463–472, 2004. (Cited on page vii.)

[Man82] Arnaldo Mandel. *Topology of oriented matroids*. PhD thesis, University of Waterloo, 1982. (Cited on page 84.)

[Mat95] Jiří Matoušek. On geometric optimization with few violated constraints. *Discrete and Computational Geometry*, 14:365–384, 1995. (Cited on pages vii, 22, 23.)

[MN98] Jiří Matoušek and Jaroslav Nešetřil. *Invitation to Discrete Mathematics*. Oxford University Press, 1998. (Cited on page 53.)

[MŠ07] Jiří Matoušek and Petr Škovroň. Removing degeneracy may require unbounded dimension increase. Submitted. Extended abstract accepted to Eurocomb 2007, Sevilla, 2007. (Cited on page ix.)

[MSW96] Jiří Matoušek, Micha Sharir, and Emo Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996. (Cited on pages vii, 8.)

[RGZ97] Jürgen Richter-Gebert and Günter M. Ziegler. Oriented matroids. In *Handbook of Discrete and Computational Geometry*, pages 111–132. CRC Press, 1997. (Cited on page 71.)

[Sch04] Ingo Schurr. *Unique sink orientations of cubes*. PhD thesis, ETH Zürich, 2004. (Cited on page viii.)

[Ško02] Petr Škovroň. Generalized linear programming. Master's thesis, Charles University, Prague, Faculty of Mathematics and Physics, 2002. (Cited on pages viii, 10.)

[Ško05] Petr Škovroň. On the number of violator spaces. In *Proc. 14th Week of Doctoral Students (WDS)*, pages I:53–57. Matfyzpress, 2005. (Cited on page ix.)

[Ško06] Petr Škovroň. Removing degeneracies in LP-type problems may need to increase dimension. In *Proc. 15th Week of Doctoral Students (WDS)*, pages I:196–207. Matfyzpress, 2006. (Cited on page ix.)

[SS04] Ingo Schurr and Tibor Szabó. Finding the sink takes some time. *Discrete and Computational Geometry*, 31:627–642, 2004. (Cited on page viii.)

[SW78] Alan Stickney and Layne Watson. Digraph models of Bard-type algorithms for the linear complementarity problem. *Mathematics of Operations Research*, 3:322–333, 1978. (Cited on page viii.)

[SW92] Micha Sharir and Emo Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 569–579, 1992. (Cited on pages vii, 4.)

[SW01] Tibor Szabó and Emo Welzl. Unique sink orientations of cubes. In *Proc. 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 547–555, 2001. (Cited on page viii.)