Discrete Geometry II

http://wikis.fu-berlin.de/display/discgeom/Discrete+Geometry+II+SS+2014

— Preliminary Lecture Notes (without any guarantees) —

Prof. Günter M. Ziegler

Fachbereich Mathematik und Informatik, FU Berlin, 14195 Berlin
ziegler@math.fu-berlin.de

FU Berlin, Summer Term 2014

This is the second in a series of three courses on Discrete Geometry. We will get to know fascinating geometric structures such as configurations of points and lines, hyperplane arrangements, and in particular polytopes and polyhedra, and learn how to handle them using modern methods for computation and visualization and current analysis and proof techniques. A lot of this looks quite simple and concrete at first sight (and some of it is), but it also very quickly touches topics of current research.

For students with an interest in discrete mathematics and geometry, this is the starting point to specialize in discrete geometry. The topics addressed in the course supplement and deepen the understanding of discrete-geometric structures appearing in differential geometry, optimization, combinatorics, topology, and algebraic geometry. To follow the course, a solid background in linear algebra is necessary. Some knowledge of combinatorics and geometry is helpful.

–

Basic Literature

[1] Peter M. Gruber. *Convex and Discrete Geometry*, volume 336 of *Grundlehren Series*. Springer, 2007.

[2] Peter M. Gruber and Jörg Wills, editors. *Handbook of Convex Geometry*. North-Holland, Amsterdam, 1993. 2 Volumes.

[3] Branko Grünbaum. *Convex Polytopes*, volume 221 of *Graduate Texts in Math.* Springer-Verlag, New York, 2003. Second edition prepared by V. Kaibel, V. Klee and G. M. Ziegler (original edition: Interscience, London 1967).

[4] Jiří Matoušek and Bernd Gärtner. *Understanding and Using Linear Programming*. Universitext. Springer, 2007.

[5] Günter M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. Revised edition, 1998; seventh updated printing 2007.

Contents

A rough schedule, which we will adapt as we move along:

1. 1. Linear programming 1.1 On the diameter of polyhedra . . . . April 15
2. 1.2 Linear programming (Discrete geometry version) . . . . April 17
3. . . . and dual simplex algorithm . . . . April 22
4. 1.2.2 (Linear algebra version), 1.3 LP duality . . . . April 24
5. . . . . April 29
6. 2. Convex bodies, volumes, and roundness 2.1 Hilbert's 3rd problem . . . . May 6
7. . . . . (?/BMS) May 8
8. . . . . May 13
9. . . . . May 15
10. . . . . (?/NYC) May 20
11. . . . . May 22
12. . . . . May 27
13. . . . . May 29
14. . . . . June 3
15. . . . . June 5
16. . . . . June 10
17. . . . . June 12
18. . . . . June 17
19. . . . . June 19
20. . . . . June 24
21. . . . . June 26
22. . . . . July 1
23. . . . . July 3
24. . . . . July 8
25. . . . . (?/SA) July 10
26. . . . . July 15
27. . . . . July 17

0 Introduction

What's the goal?

This is a second course in a large and interesting mathematical domain commonly known as "Discrete Geometry". This spans from very classical topics (such as regular polyhedra – see Euclid's *Elements*) to very current research topics (Discrete Geometry, Extremal Geometry, Computational Geometry, Convex Geometry) that are also of great industrial importance (for Computer Graphics, Visualization, Molecular Modelling, and many other topics).

My goal will be to develop these topics in a three-semester sequence of Graduate Courses in such a way that

- you get an **overview** of the field of Discrete Geometry and its manifold connections,
- you learn to understand, analyze, visualize, and confidently/competently argue about the basic **structures** of Discrete Geometry, which includes
  - point configurations/hyperplane arrangements,
  - frameworks
  - subspace arrangements, and
  - polytopes and polyhedra,
- you learn to know (and appreciate) the most important **results** in Discrete Geometry, which includes both simple & basic as well as striking key results,
- you get to learn and practice important **ideas and techniques** from Discrete Geometry (many of which are interesting also for other domains of Mathematics), and
- You learn about current **research topics** and problems treated in Discrete Geometry.

In this second course of the sequence, we will in particular treat the relationship between

- "discrete objects" (such as polytopes and polyhedra, but also lattices and lattice points) and
- "general objects" (such as convex bodies)

in terms of various notions of diameter, volume, and roundness.

This will not only be interesting *per se*, but also lead us to some major theorems and insight (e.g. on such fundamental notions as *volume*), but also to major applications (e.g. on sphere packings, which is in turn important for coding theory).

1 Linear programming and some applications

1.1 On the diameter of polyhedra

Let's consider a polyhedron of dimension d with n facets; let's call it an (d, n)-polyhedron.

Careful: Want to look at *pointed* polyhedron, $n \geq d$, which has a vertex, so the *lineality space* is trivial.

The *Hirsch conjecture* from 1957 is the *false* (!) statement that the edge-graph of any (d, n)-polyhedron has diameter at most $n - d$. This was disproved for unbounded polyhedra by Klee & Walkup [3] in 1967 and in general by Santos [4] in 2012. The *polynomial Hirsch conjecture* remains open: It might still be that the maximal diameter, $\Delta(d, n)$, satisfies $\Delta(d, n) \leq d(n-d)$ for all $n \geq d \geq 1$.

We will, nevertheless, see why from a "linear programming point of view" the bound $n - d$ looks natural, and even more so, why this is a relevant parameter.

Exercise 1.1. Show that $\Delta(2, n) \leq n - 2$ and $\Delta(3, n) \leq n - 3$, and that both inequalities are sharp (that is, hold with equality for $n \geq 2$ resp. $n \geq 3$).

Up to recently, the best upper bound for the diameters of polyhedra was provided by Kalai & Kleitman in a striking two page paper [2] in 1992:

$$\Delta(d, n) \leq n^{\log(d)+2},$$

which was improved only slightly by Kalai [1] to

$$\Delta(d, n) \leq n^{\log(d)+1},$$

where throughout "$\log$" denotes the binary logarithm (i.e., base 2). However, just a few weeks ago Mike Todd (Cornell University) in a 4-page paper [5] sharpened the Kalai–Kleitman analysis to obtain

$$\Delta(d, n) \leq (n-d)^{\log(d)} = d^{\log(n-d)},$$

which indeed is sharp for $d = 1$ and $d = 2$.

In class, we will go through the arguments of Todd [5] (and thus, in particular, the idea of Kalai & Kleitman [2]).

[1] Gil Kalai. Linear programming, the simplex algorithm and simple polytopes. *Math. Programming, Ser. B*, 79:217–233, 1997. Proc. Int. Symp. Mathematical Programming (Lausanne 1997).

[2] Gil Kalai and Daniel J. Kleitman. A quasi-polynomial bound for the diameter of graphs of polyhedra. *Bulletin Amer. Math. Soc.*, 26:315–316, 1992.

[3] Victor Klee and David W. Walkup. The d-step conjecture for polyhedra of dimension $d < 6$. *Acta Math.*, 117:53–78, 1967.

[4] Francisco Santos. A counterexample to the Hirsch conjecture. *Annals of Math.*, 176:383–412, 2012.

[5] Michael J. Todd. An improved Kalai–Kleitman bound for the diameter of a polyhedron. Preprint, April 2014, 4 pages, `http://arxiv.org/abs/1402.3579`.

End of class on April 15

1.2 Geometry of linear programming and pivot rules

1.2.1 Linear programming (Discrete Geometry version)

Any system $Ax \leq b$ with $A \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$ defines a polyhedron $P \subseteq \mathbb{R}^d$ with $\dim P \leq d$ and $\#$facets $\leq n$.
Without loss of generality we may assume that $\operatorname{rank} A = d$, that is the system $Ax \leq b$ has a subsystem that defines an orthant, so in particular P is either pointed (has a vertex), or is empty.
Without loss of generality (theoretically, this may be harder to compute) we may assume that $\dim P = d$, so the polyhedron is full-dimensional. Moreover, we want to get our system into the form

$$Ax \leq b,\ -x \leq 0$$

with $b > 0$ componentwise. For this we have to solve a "Phase I" problem that finds a vertex x_0 of the polyhedron, and then do a coordinate transformation that moves the vertex x_0 to 0 and transforms a system of inequalities that are tight at x_0 to the positive orthant system $x \geq 0$.
With a linear objective function we have a system of the form

$$\begin{array}{rcl} \max\ c^t x & & \\ Ax & \leq & b \\ x. & & \end{array}$$

Example:

$$\begin{array}{rrcl} \max & y & & \\ & x - y & \leq & 2 \\ & -x + y & \leq & 1 \\ & x + 2y & \leq & 7 \\ & -x \quad & \leq & 0 \\ & y & \leq & 0. \end{array}$$

Geometric description of the polyhedron

- P is a full-dimensional polyhedron, with $\leq n$ facets, given in $\mathcal{H}$-description-
- We have a linear objective function, which might be assumed to be the last coordinate x_d, to be maximized (or in other situations: minimized).

- We assume that the polyhedron is simple, the system is in general position (this may be achieved by perturbing the right-hand sides: Exercise!).
- Any $d \times d$ full rank subsystem $A'x \leq b'$ defines a *generalized orthant*, which up to an affine transformation is equivalent to the standard positive orthant "$x \geq 0$."
- Any generalized orthant defines a point (the unique solution of $A'x = b'$) and d *rays* (by fixing all the d inequalities by one, and letting the *slack* in the last one get large).
- A generalized orthant is *feasible* if the point it defines by $A'x = b'$ is *feasible* (defines all inequalities, not only those in the subsystem). Note that this does not depend on the objective function.
- A generalized orthant is *dual feasible* if sliding along any of its rays does not improve the objective function. Note that this does not depend on the right-hand side vector b.
- A generalized orthant is *optimal* if it is both feasible and dual feasible.
- Any optimal generalized orthant defines an optimal solution of the linear program.

… and what the *primal simplex algorithm* does on it:

- We assume that after preprocessing (known as "Phase I") we have $-x \leq 0$ as a feasible generalized orthant, and in particular $x_0 = 0$ as a feasible starting vertex.
- If the generalized orthant is dual feasible, DONE with optimal solution.
- Select an improving ray, and slide along the ray. (Along the ray one inequality of the orthant is not tight any more; the objective function improves along the ray.)
- If the objective function improves without bound along the ray, DONE with optimal solution.
- Otherwise along the way we hit a bound, that is, a new facet, whose inequality completes a new feasible generalized orthant. REPEAT.

The process stops in finite time, since in every step we improve the objective function (no cycles) and there are only finitely many orthants — not more than $\binom{n}{d}$. (A better bound is obtained from the upper bound theorem — need a version for unbounded polyhedra: Exercise!)

End of class on April 17

Alternatively, here is what the *dual simplex algorithm* does on a linear program:

- We assume that after preprocessing (known as "Phase I") we have found a dual feasible generalized orthant, which in particular defines a current solution (vertex of the system, but not necessarily of the polyhedron).
- If the generalized orthant is feasible, DONE with optimal solution.
- Select an inequality violated by the current solution.
- If the violating inequality hits none of the rays of the current generalized orthant, then DONE with proof that the system is infasible.
- Otherwise construct a new dual feasible generalized orthant whose current solution gives a better upper bound on the maximum of the system. REPEAT.

The process stops in finite time, if we take care that in every step we improve the current upper bound on the objective function values on the polyhedron (no cycles) and there are only finitely many generalized orthants.

End of class on April 22

1.2.2 Linear programming (Numerical Linear Algebra version)

We write down two linear programs, in the following form.
The *primal linear program* is

$$\begin{array}{lrcl} (P) & \max\ c^t x & & \\ & Ax & \le & b \\ & x & \ge & 0. \end{array}$$

The associated *dual linear program* is

$$\begin{array}{lrcl} (D) & \min\ b^t y & & \\ & A^t y & \ge & c \\ & y & \ge & 0. \end{array}$$

Lemma 1.2 (Weak Duality Theorem)**.** *If for a primal-dual pair of linear programs x_0 is a feasible solution for the primal (P) and y_0 is a feasible solution for the dual (D), then*

$$c^t x_0 \le b^t y_0.$$

In particular, the maximum of (P) is smaller or equals to the minimum of (D).

Proof. We compute

$$c^t x_0 \le (A^t y_0)^t x_0 = y_0^t (A x_0) \le y_0^t b = b^t y_0.$$

□

The linear programs are then, by introduction of *slack variables*, converted into systems of linear equations, to be solved in non-negative variables.
Thus the primal linear program becomes

$$\begin{array}{lrcl} (P) & \max\ c^t x + 0^t \hat{x} & = & \gamma \\ & Ax + I_n \hat{x} & = & b \\ & x \ge 0,\ \hat{x} \ge 0 & & \end{array}$$

This system has an "obvious" current solution, given by $x \equiv 0$ (the "non-basic variables" are set to 0: these correspond to the inequalities that define the current generalized orthant), $\hat{x} = b$ (the "basic variables" are uniquely determined). This starting solution has the value $\gamma = 0$.
These systems are manipulated by *row operations*, which do not change the solution space. Thus after a number of steps we still have the system in the form

$$\begin{array}{lrcl} (P) & \max\ \bar{c}^t x_N + 0^t x_B & = & \bar{\gamma} \\ & \bar{A}_N x_N + I_n x_B & = & \bar{b} \\ & x_N \ge 0,\ x_B \ge 0 & & \end{array}$$

Here the columns have been resorted, to keep the "basic variables" and the "non-basic variables" together, that is, the index sets B and N together give the set of all columns labelled by $B \cup N =$

$\{1, 2, \ldots, d{+}n\}$. The coefficients in the system are $\bar{A}_N = A_B^{-1}A_N$, and $\bar{b} = A_B^{-1}b$. The objective function has been rewritten in terms of the non-basic variables. Its coefficients

$$\bar{c}_N^t = c_N^t - c_B^t A_B^{-1} A_N$$

are known as the *reduced costs*: in the geometric interpretation they give the slopes of the rays of the current generalized orthant.
The current solution is given by $x_N \equiv 0$, which uniquely determines the non-basic variables to be $x_B = \bar{b} = A_B^{-1}b$.
Thus the (current solution of the) system is *feasible* if $\bar{b} \geq 0$, and it is *dual feasible* if $\bar{c}_N \leq 0$.
A similar treatment/computation can be done for the dual system (D).

Lemma 1.3. *For any pair of primal linear program (P) and its dual program (D) in the equation form given above,*

- *the bases B for the system (P) are in bijection with the non-bases N of the system (D);*
- *the feasible bases for (P) are in bijection with the dual-feasible non-bases for (D);*
- *etc.*

Proof. This rests on the observation that in the $(n+d) \times (n+d)$ matrix

$$\begin{pmatrix} A & I_n \\ -I_d & A^t \end{pmatrix}$$

the row space spanned by the first n rows is the orthogonal complement of the space spanned by the last d rows. □

Theorem 1.4 (Duality Theorem for Linear Programming)**.** *If a primal linear program (P) and its dual (D) are both feasible, then they have optimal solutions x^* and $y*$, and these have the same optimal value.*
If one of the programs is not feasible, then the other one is either infeasible as well, or it is unbounded.

Proof. The optimal solutions exist, since the Simplex Algorithm will find it! □

From the geometry of an optimal basis/optimal generalized orthant, we also get *complementary slackness*: If in the optimal solution an inequality is not tight, then the corresponding variable in the dual program is zero; if a variable is positive, then the corresponding dual inequality has to be tight. This can also be seen from analysis of the inequalities in the proof of the Weak Duality Theorem.
The optimal solution to a linear program can be computed *efficiently*:

In Practice there are commercial, as well as non-commercial, software libraries for linear programming, which include implementations of the Primal Simplex Algorithm, the Dual Simplex Algorithm, as well as other methods (such as *Interior Point Methods*) which will solve to optimality practically every linear program that appears in practice.

In Theory there are two different computational models:

In the bit model the "Ellipsoid Method" (which will appear later in this course) is a polynomial time method for solving linear programs, whose running time is polynomial in the bit-size of the input. This method is theoretically very important, but has not been implemented in practice.

In the unit cost model the Simplex Algorithm with a suitable choice of variable selections ("pivot rule") may be polynomial — but this has not been proven. Indeed, we do not even know whether in general there is any short (i.e. polynomially many edges) path from a given starting vertex of the program to the optimal vertex. The best upper bound is the $n^{\log_2 d}$ upper bound discussed at the beginning of this course — and this bound is *not* a polynomial in n and d. An upper bound of the type $d(n-d)$ might exist, but has not been proven.

Thus the complexity of Linear Programming, and in particular of the Simplex Algorithm, is a major open problem both for Optimization, and for Discrete Geometry!

End of class on April 24

1.3 Further Notes on Linear Programming

Let's step away from the simplex algorithm, and let's look at the problem itsself — and let's assume we have a solution method (algorithm, perhaps software) that solves the problem, but which we can treat as a "black box." This is the *oracle* view, which has become popular in optimization, with grave consequences for (computational) discrete and convex geometry: well-defined input, well-defined output; estimate complexity
Examples:
LP-OPTIMIZATION problem/oracle:

INPUT: $d \geq 1$, $n \geq 1$, $A \in \mathbb{Q}^{n\times d}$, $b \in \mathbb{Q}^n$, $c \in \mathbb{Q}^d$

TASK: $\max c^t x :\ Ax \leq b,\ x \geq 0$

OUTPUT: optimal solution $x^* \in \mathbb{Q}^d$, with certificate (basis)
or information that problem is infeasible, with certificate (basis & inequality),
or information that problem is unbounded, with certificate (basis & ray).

LP-FEASIBILITY problem/algorithms/oracle:

INPUT: $d \geq 1$, $n \geq 1$, $A \in \mathbb{Q}^{n\times d}$, $b \in \mathbb{Q}^n$

TASK: find $x :\ Ax \leq b,\ x \geq 0$

OUTPUT: feasible solution $x^* \in \mathbb{Q}^d$, with certificate (basis)
or information that problem is infeasible, with certificate (basis & inequality).

Note: Any algorithm for solving **LP-OPTIMIZATION** can be used to solve **LP-FEASIBILITY**. We will see that the other direction "works as well."

Note: Two algorithms we know/could work out for **LP-OPTIMIZATION**: Fourier–Motzkin elimination (see Discrete Geometry I), and the Simplex Algorithm.

1.3.1 Complexity issues

Could it be that the solution exists, but it is too large (or too small) to write down in reasonable time?

Real input/solutions don't make sense, or need work to make sense of.

Recommended reading: Lovász' lecture notes [**?**].

Could get answer from Fourier–Motzkin elimination.

Here: get answer from simplex and Cramer's rule and Hadamard inequality.

Lemma 1.5 (Hadamard inequality)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a matrix with columns $A = (A_1, \dots, A_n)$. Then*

$$|\det A| \ \le\ |A_1| \cdots |A_n|.$$

Lemma 1.6 (The Cramer's rule estimate)**.** *Let $A \in \mathbb{Z}^{n\times n}$, $b \in \mathbb{Z}^n$, $\det A \neq 0$ (integer data!). Then the (rational!) solution for the system of equations $Ax = b$ satisfies*

$$|x_i| \le |A_1| \cdots |A_n| \cdot |b|.$$

Proof. Cramer's rule, together with the observation that the denominator, $\det A$, is an integer, so its absolute value is at least 1. The same is true for the length of each column $|A_i|$. □

1.3.2 Feasibility

First, we should discuss the problem how to find a feasible generalized orthant for $Ax \le b$, $x \ge 0$, in order to even *start* the simplex algorithm. Here are two solutions to that problem:

- Use the complexity estimates to get explicit upper bounds for the variables, and thus have a starting basis for the dual simplex algorithm (that is, a feasible basis for the simplex algorithm applied to the dual program).
- Phase I: Write down an artificial OPTIMIZATION program, which is feasible, and whose optimal solution (basis) will give a feasible solution (and a feasible basis!) for the FEASIBILITY problem: For example

$$\min x_0 : \ Ax - x_0\mathbf{1} \le b,\ x \ge 0, x_0 \ge 0.$$

It is trivial that if we can solve **LP-OPTIMIZATION** then we can solve **LP-FEASIBILITY**, in a way that is completely independent of the the specific algorithm used to "implement" **LP-OPTIMIZATION**; that is, we can use any **LP-OPTIMIZATION** *oracle* to "simulate" an **LP-FEASIBILITY** algorithm; in other words, we can program a (fast) algorithm for **LP-FEASIBILITY** if we can use a (fast) subroutine for **LP-OPTIMIZATION** (e.g. by putting objective function zero).

However, note that the converse is also true: If we know how to solve **LP-FEASIBILITY**, then we can also solve **LP-OPTIMIZATION**, that is,

LP-FEASIBILITY $\Longrightarrow$ **LP-OPTIMIZATION**.

For this, note that any *feasible* solution (x, y) for the primal-dual program

$$\begin{array}{cccc} & & c^t x \ge b^t y & \\ (PD) & Ax \le b & & A^t y \ge c \\ & x \ge 0 & & y \ge 0 \end{array}$$

1.3.3 Modelling issues

Conversion of programs from equality form to inequality form, and conversely. See the Exercises.

1.3.4 Perturbation techniques

If we replace the right-hand sides b_i by $b_i + \varepsilon^i$, for a suitably small ε, then

- the perturbed problem will be feasible if *and only if* the original problem is feasible,
- the perturbed problem will be primally *non-degenerate*, that is, it describes a simple polyhedron, and at any generalized orthant (basis), no extra inequalities are tight (that is, the non-basic variables are non-zero).

(see Exercise).

Moreover,

- similarly, by perturbing the objective function the program can be made *dually non-degenerate*, so that in particular the optimal solution is unique (if it exists), and
- the suitable $\varepsilon > 0$ can be estimated explicitly.

1.3.5 Integral solutions? An example

In general, the optimal solutions will not be integral, although many applications ask for integral solutions. Even if we find the best integral solution, this will come without a certificate, as there may be not dual constraints that are tight at the best integer solution.

However, in many combinatorial situations, we are lucky. Here is one example.

Example 1.7 (Network flows). If the bounds on each arc are integral, then the optimal solution will be integral.

(This may be seen from an algorithm by successive improvement, or from a matrix argument, see exercise.)

Interpretation of dual solutions: Max cut!

Max-Flow-Min-Cut theorem!

Exercise 1.8. Let $A \in \{0, 1, -1\}^{n\times n}$ be a $0/\pm 1$ matrix. Show that

(i) The determinant of a $0/1$-matrix A can be large, even if there are only two 1s per row.
(ii) The determinant of A is not large if there is at most one 1 and at most one -1 per row.
(iii) Use the Hadamard inequality to give an upper bound on $|\det A|$
(iv) For $A \in \{0, 1\}^{n\times n}$ give a much better upper bound, by
— Multiplying the matrix by 2,
— Adding a column of 0's and then a row of 1's,
— subtracting the first row from all others
and then applying Hadamard to the resulting ± 1-matrix.
(v) Give examples where this bound is tight.

End of class on April 29