Randomization and Abstraction

Useful Tools for Optimization

Bernd Gärtner

BRICS Notes Series NS-00-1

ISSN 0909-3206 February 2000

Randomization and Abstraction – Useful Tools in Optimzation

Bernd Gärtner[*]

Abstract

This report contains notes for a course on randomized methods in geometric optimization, held first at BRICS, Aarhus University, Denmark and later at the Institute for Computer Science, ETH Zürich, Switzerland.

[*]work supported by the Swiss Science Foundation (SNF), Project No. 21-50647.97

Introduction

This report contains the course notes (copies of the slides), exercises and solutions for the course *Randomization and Abstraction—Useful Tools for Optimization.* I have given this course first at BRICS in Aarhus, Denmark,[1] and later at ETH in Zürich.[2].

The BRICS course notes did not yet contain solutions. For the ETH course, I added them. In this process, I replaced exercises that did not make sense or were plain wrong (which I only noticed when I wrote up the solutions, of course...).

The course deals with two approaches to Linear Programming (and related problems): the "abstract" one via LP-type problems, due to Matoušek, Sharir and Welzl [3], and the "concrete" one via the well-known simplex method under Kalai's pivot rule *Random-Facet* [2]. In the course, I put both approaches in relation and show that they are dual to each other, under the concept of LP duality. This was first observed by Goldwasser [1].

I also review the best theoretical complexity bounds that are known in either of the approaches, and I point out that they were developed independently almost at the same time for both of them [2, 3].

Postscript files containing the sources for the course are available from my homepage at `www.inf.ethz.ch/personal/gaertner`. The course consists of about 85 slides; at BRICS, I tried to manage them in two slots, 90 minutes each, which was a little too ambitious, although the audience was fairly advanced. I still did all the slides, but it took me about 45 minutes longer in total.

At ETH, I had to face a less advanced audience, and I made it through roughly two thirds of the slides in 200 minutes. This means, I omitted some material. In part I, I only got to the first linear bound for the expected performance of the LP-type algorithm.

I now think that three or even four 90 minutes slots would be ideal to cover the material without any haste, even for a less advanced audience (undergraduates, say).

Acknowledgment

This course goes back to an initiative from Devdatt Dubhashi whom I first met at the RANDOM'98 workshop in Barcelona. Devdatt invited me to BRICS to give a mini-course, and he made my stay in Aarhus very enjoyable.

[1]BRICS Mini-course, held at BRICS Research Center, University of Aarhus, July 6–7, 1999

[2]Late Summer School "Facets of the Polytope World", held together with Jürgen Richter-Gebert and Emo Welzl at the computer science department of ETH Zürich, September 13–16, 1999

References

[1] M. Goldwasser. A survey of linear programming in randomized subexponential time. *ACM-SIGACT News*, 26(2):96–104, 1995.

[2] G. Kalai. A subexponential randomized simplex algorithm. In *Proc. 24th annu. ACM Symp. on Theory of Computing.*, pages 475–482, 1992.

[3] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. In *Proc. 8th annu. ACM Symp. on Comput. Geom.*, pages 1–8, 1992.

Facets of the Polytope World

Late Summer School · ETH Zürich · 1999

Randomization and Abstraction – Useful Tools for Optimization

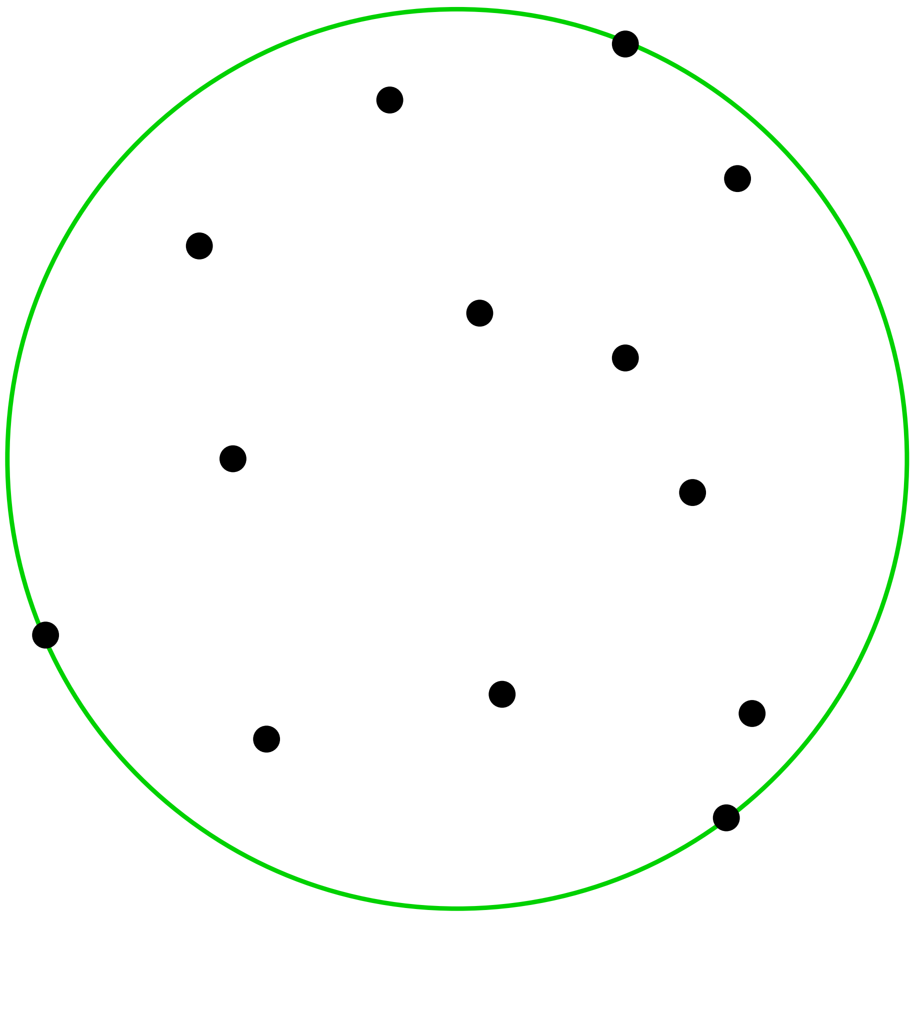
Part I

by Bernd Gärtner

A few geometric optimization problems. . .

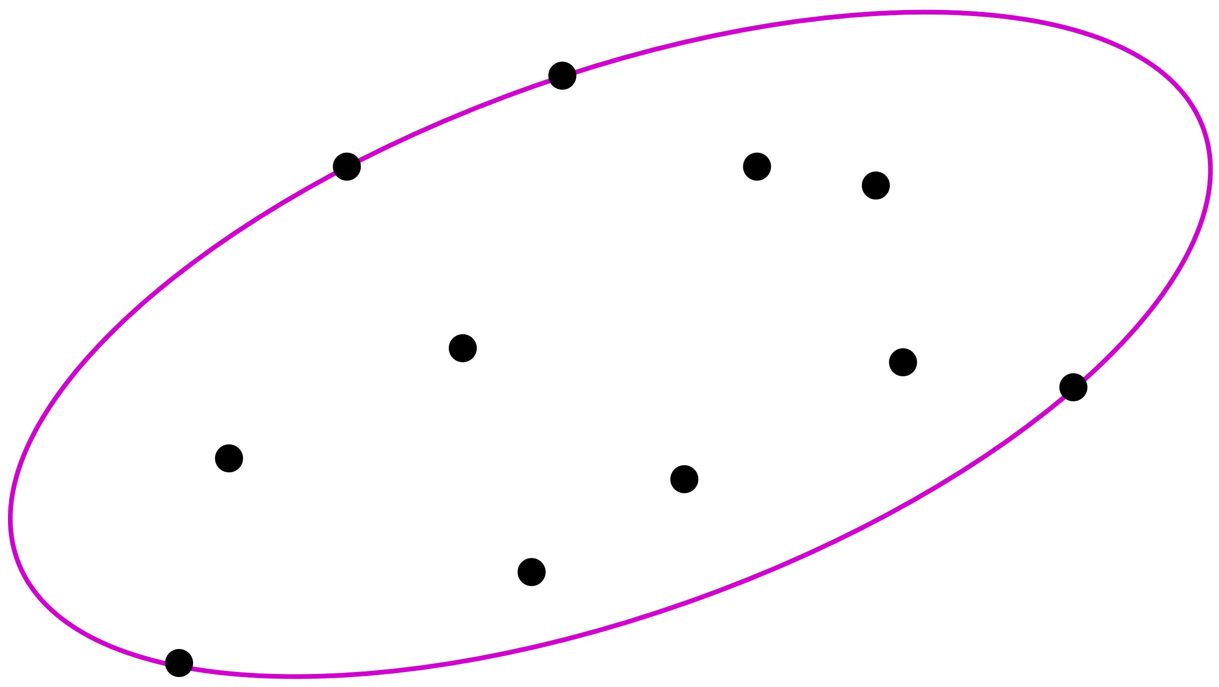
Smallest enclosing ball

Given n points in $\mathbb{R}^d$, find the ball of smallest volume covering all the points.



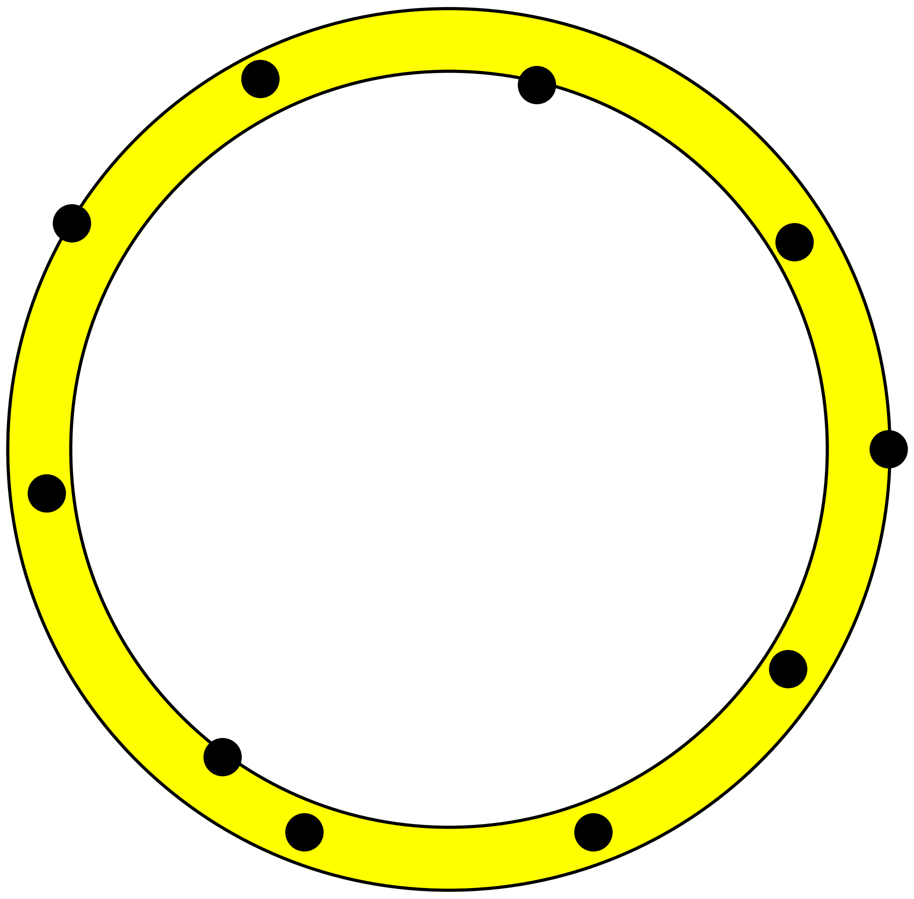
Smallest enclosing ellipsoid

Given n points in $\mathbb{R}^d$, find the ellipsoid of smallest volume covering all the points.



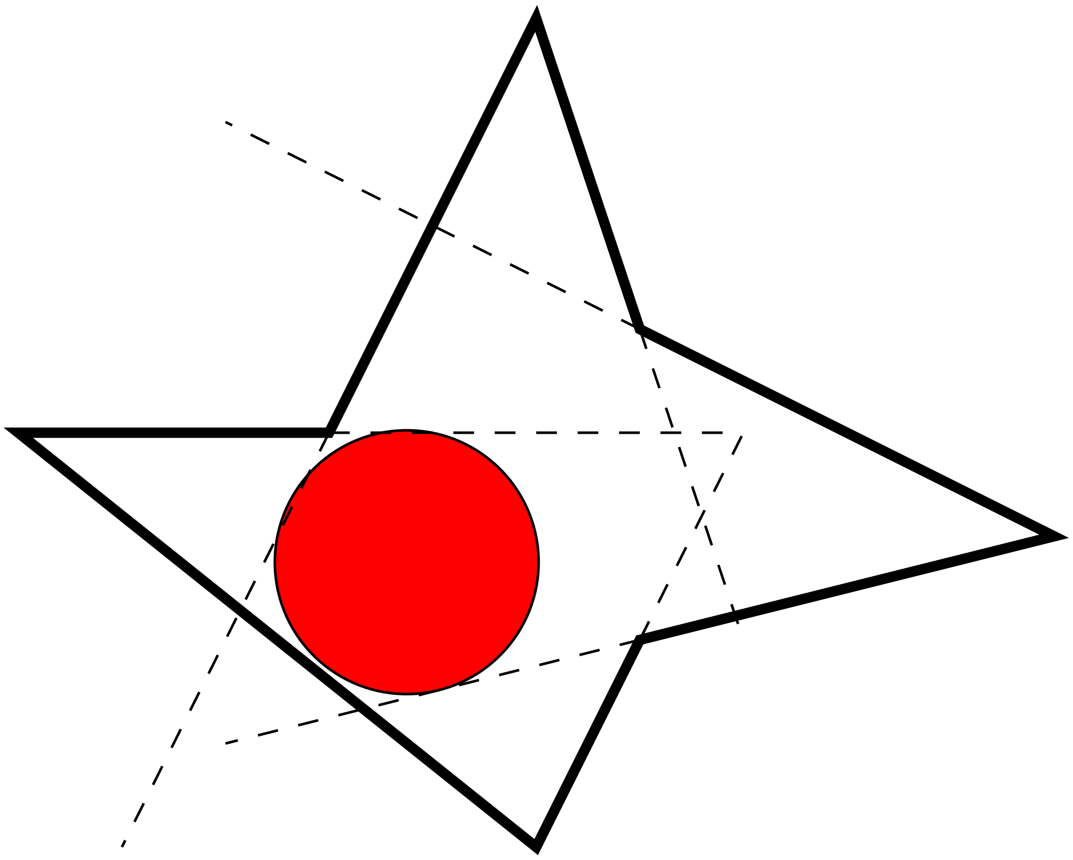
Smallest enclosing annulus

Given n points in the plane, find the annulus of smallest area covering all the points.



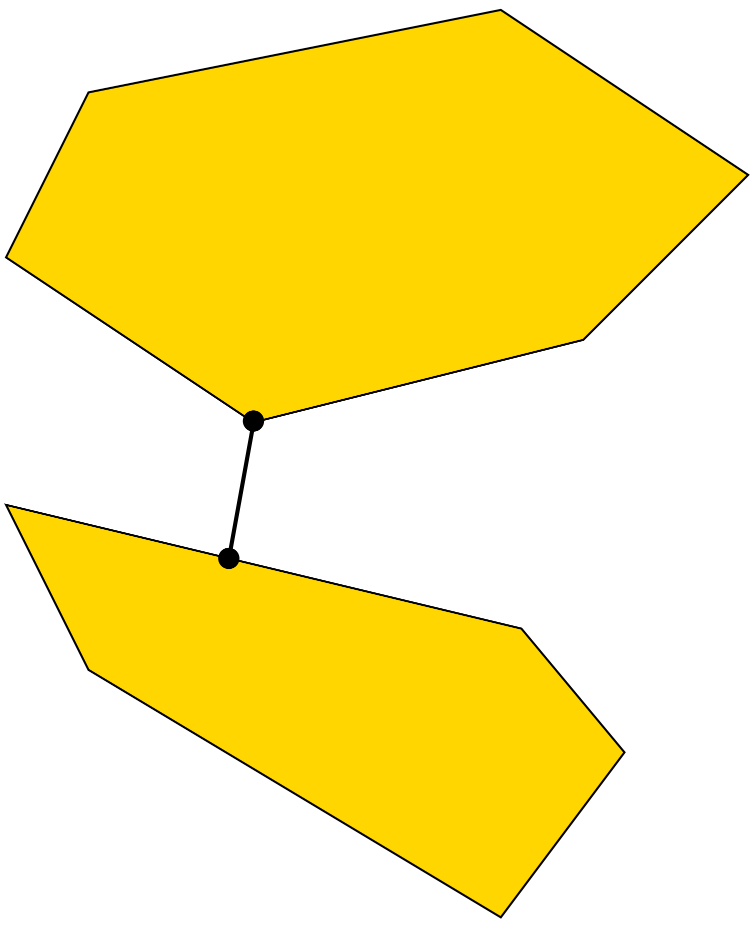
Largest disk in kernel

Given a starshaped polygon with n edges, find the largest disk in the kernel of the polygon.



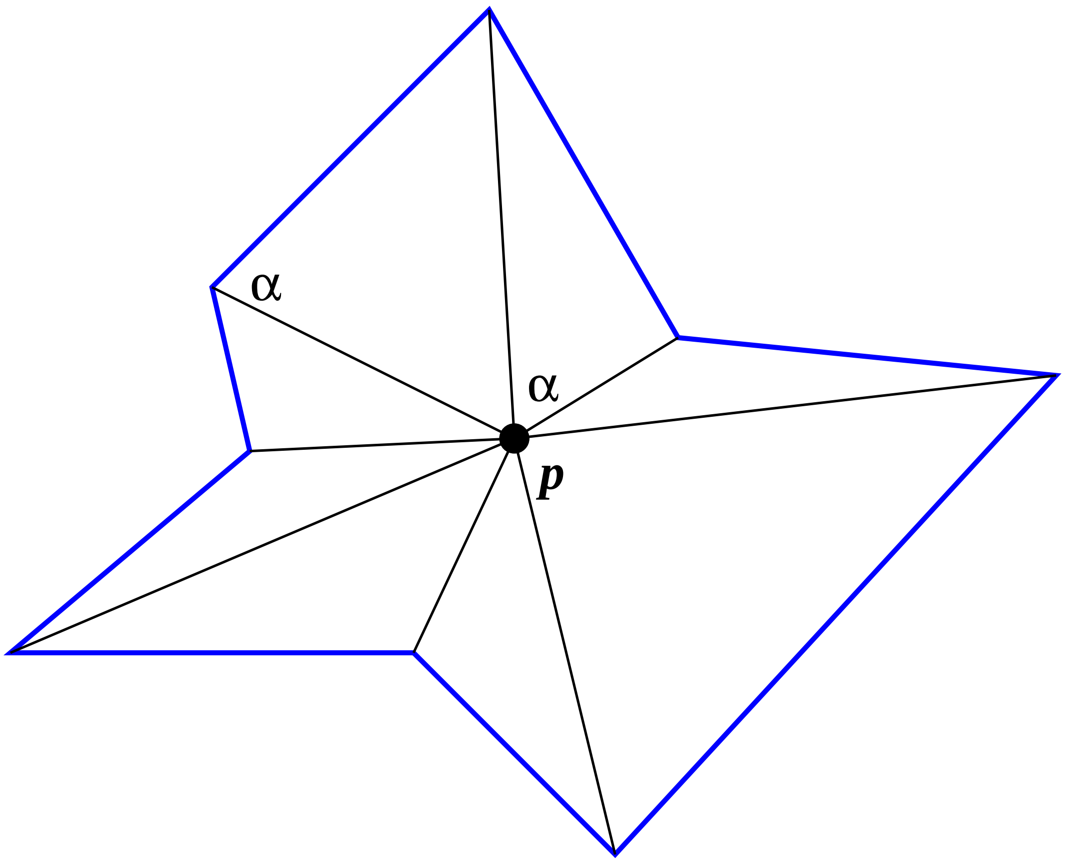
Distance of polytopes

Given two polytopes defined by overall n halfs-paces (or n points), find the shortest distance between them.



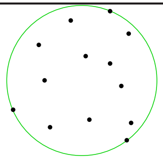
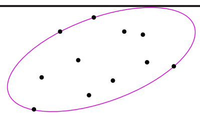
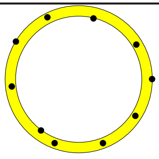
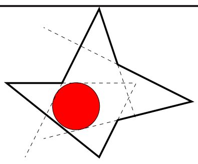
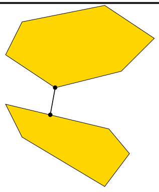
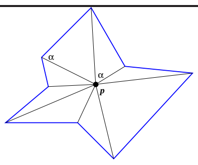
Angle-optimal placement

Given a starshaped polygon with n edges, find the point p in its kernel such that the smallest angle in the graph consisting of the polygon edges and the connections between p and the vertices is maximized.

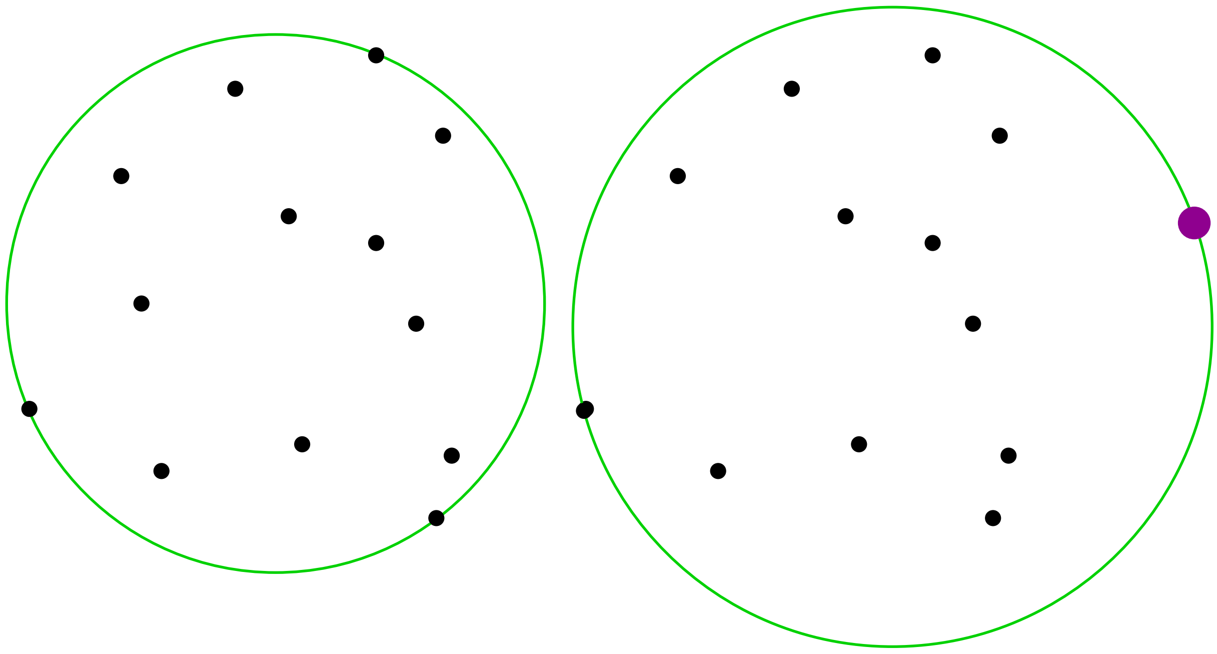


What is common to all these problems?

(1) Small basis size: Solution determined by small number δ of input objects (points, polygon edges, halfspaces, ...), independent of the number n of objects!

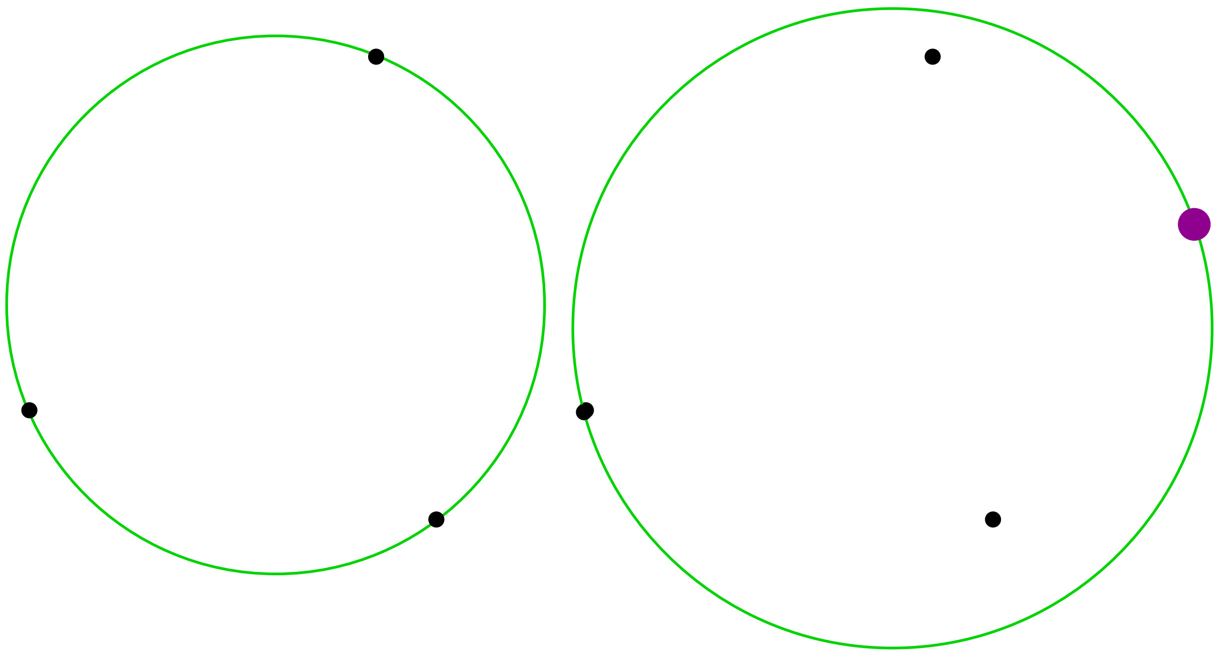
Problem	δ
	$d + 1$
	$d(d + 3)/2$
	4
	3
	$d + 2$
	3

(2) Monotonicity: solution does not get better if more input objects are added!



Status: trivial.

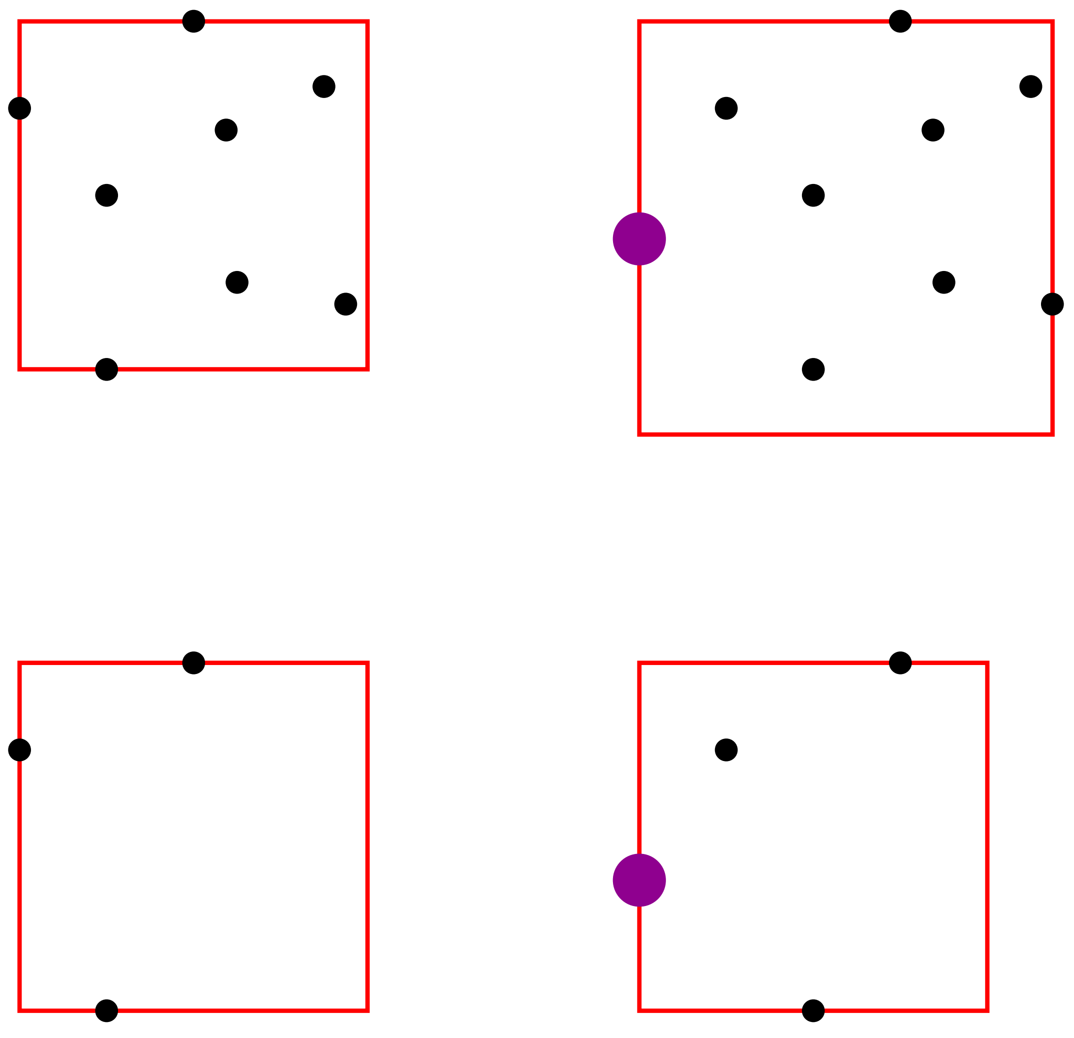
(3) Locality: if solution gets worse by adding more input objects, it already gets worse by adding them to the at most δ objects that determine it!



Status: not so trivial anymore.

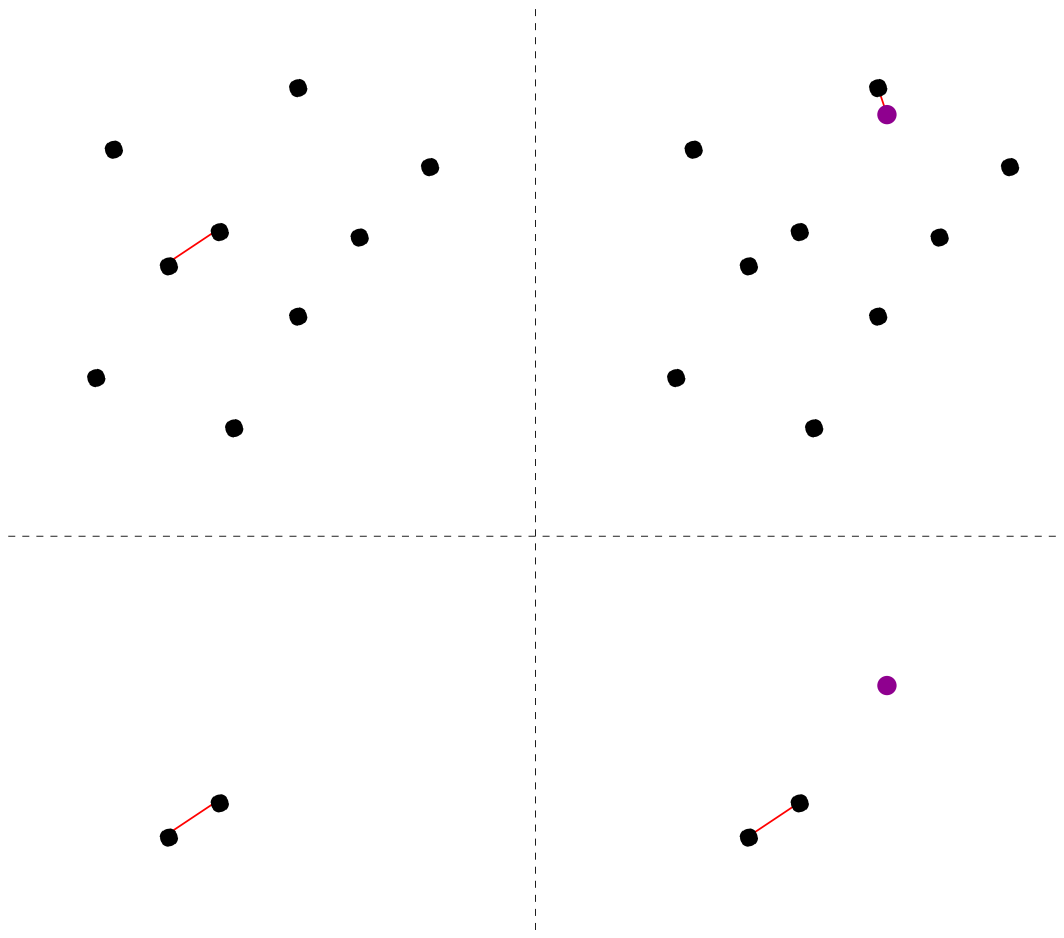
Non-local problems (I)

Smallest enclosing axis-parallel square



Non-local problems (II)

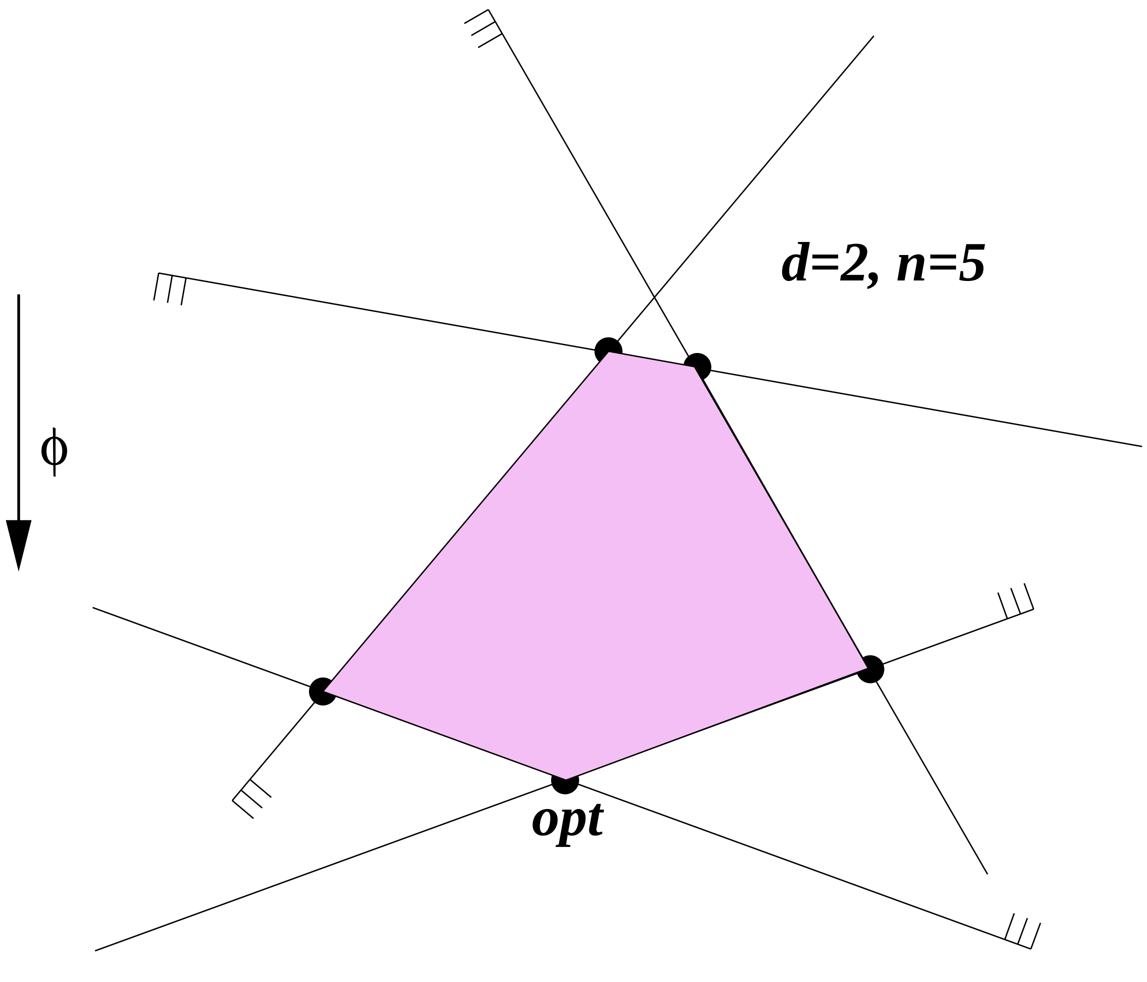
Closest pair of points



One more problem...

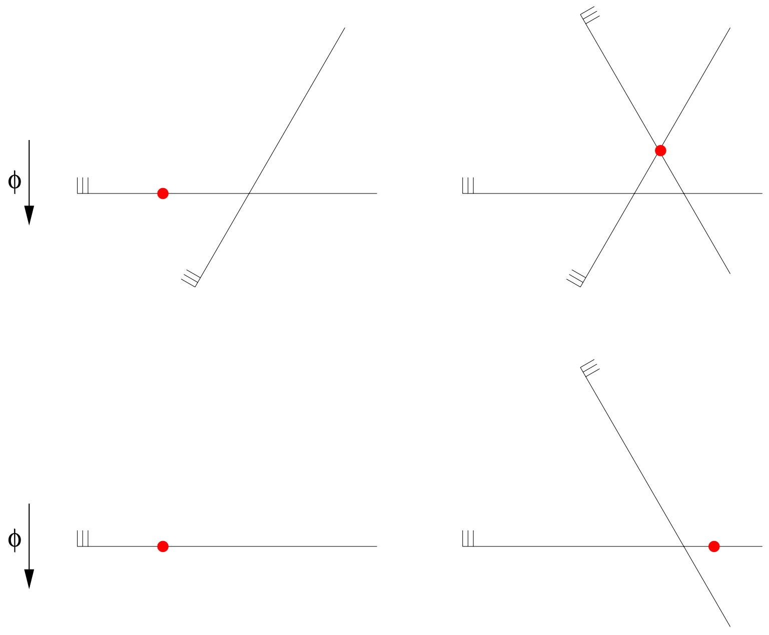
Linear Programming (LP).

Minimize a linear function ϕ in d variables subject to n linear inequalities (here: $n \gg d$).



LP and locality

- Locality does *not* hold. . .



- . . . but that can be fixed!

  minimizing

$$\phi(x) + \varepsilon x_1 + \varepsilon^2 x_2 + \cdots + \varepsilon^d x_d, \quad \varepsilon \text{ small}$$

  makes solution vector unique *and* determines it by d constraints (locality follows), or the problem becomes unbounded

LP-type Problems

An *LP-type problem* is a pair (H, w), H a finite set of *constraints*, $w : 2^H \to \mathcal{W} \cup \{-\infty, \infty\}$ ($\mathcal{W}$ some ordered set) an *objective function*, such that for any $F \subseteq G \subseteq H$

- $w(F) \leq w(G)$ (**monotonicity**)

- if $w(F) = w(G) \neq -\infty$ and

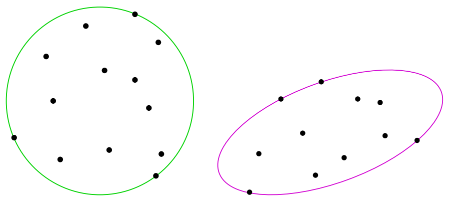
  $w(G \cup \{h\}) > w(G)$ for $h \in H$, then also
  $w(F \cup \{h\}) > w(F)$ (**locality**)

Interpretation:

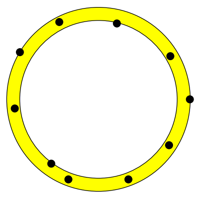
- Minimization problem; $-\infty \approx$ unbounded, $\infty \approx$ infeasible

- $w(G)$ is the minimum value *subject to the constraints* in G

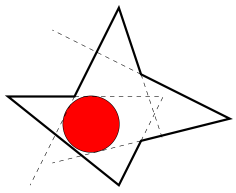
Problems revisited (I)



H set of point; $w(F)$ the radius of the smallest ball (ellipsoid) containing all points in F

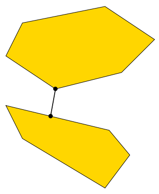


H set of points; $w(F)$ the area of the smallest annulus containing all points in F



H a set of halfplanes; $w(F)$ the negative radius of the largest ball in the intersection of all halfspaces in F (can be $-\infty, \infty$)

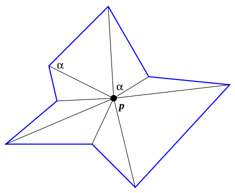
Problems revisited (II)



$H = H_1 \cup H_2$ a set of halfspaces; $w(F_1 \cup F_2)$ the smallest distance between the two polytopes defined by F_1 and F_2 (can be ∞; locality needs to be enforced like in LP)

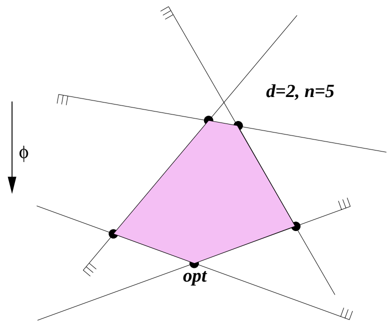
or

$H_1 \cup H_2$ a set of points; $w(F_1 \cup F_2)$ the *negative* smallest distance between the two polytopes defined by F_1 and F_2 (can be $-\infty$)



. . . Exercise!

Problems revisited (III)



Linear Programming

H a set of halfspaces in $\mathbb{R}^d$; $w(F)$ the minimum value of $\phi(x) + \varepsilon x_1 + \varepsilon^2 x_2 + \cdots + \varepsilon^d x_d$ over the intersection of the halfspaces in F (can be $-\infty, \infty$)

Unique optimal solution x^* is the lexicographically smallest point that minimizes $\phi(x)$.

Bases and combinatorial dimension

(H, w) LP-type problem

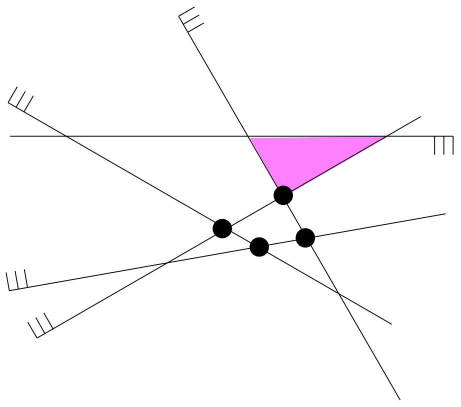
- *Basis* of $F \subseteq H$: inclusion-minimal subset $B \subseteq F$ with
$$w(B) = w(F).$$

- *Combinatorial dimension* δ: size of largest basis

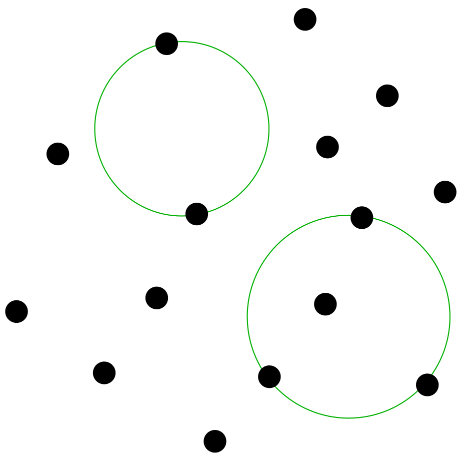
- (H, w) *basis-regular* if all bases of sets F with $|F| \geq \delta$ and $w(F) \neq -\infty$ have size exactly δ

- **Note:** even then, the basis B of F need not be unique!

Bases in LP / smallest ball

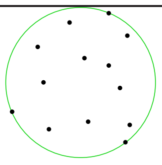
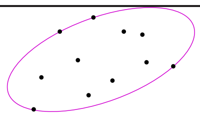
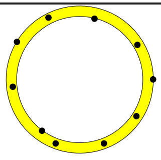
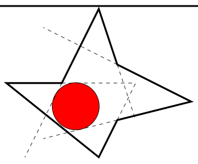
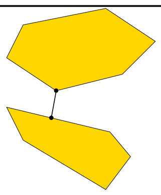
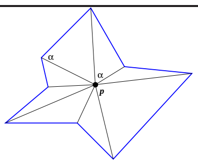
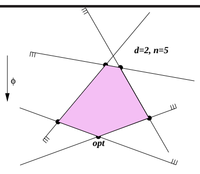


Bases $\sim$ local minima determined by d half-spaces $\Rightarrow$ basis-regular

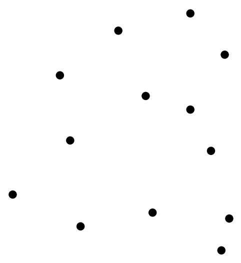


Bases $\sim$ locally minimal balls determined by $\leq d+1$ points $\Rightarrow$ not basis-regular

One more table. . .

Problem	δ	basis-regular?
	$d+1$	no
	$d(d+3)/2$	no
	4	yes
	3	yes
	$d+2$	no
	3	no
	d	yes

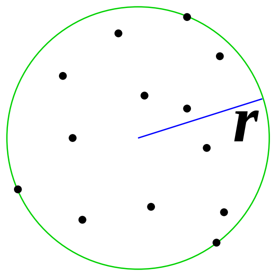
Solving LP-type problems



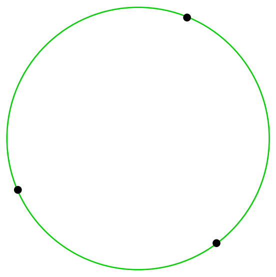
Given: (H, w)

Wanted:

- The optimal value $w_{opt} := w(H)$



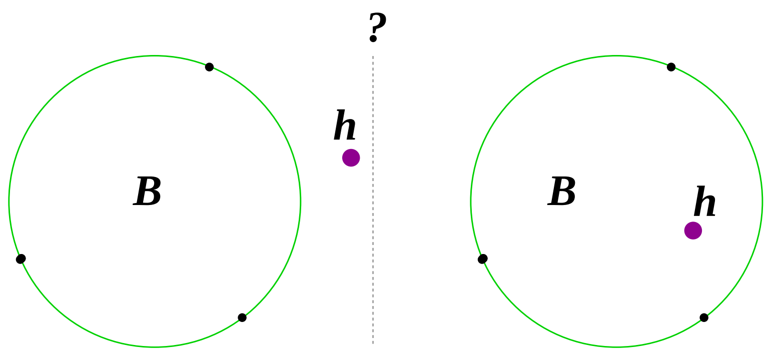
- A basis B of H



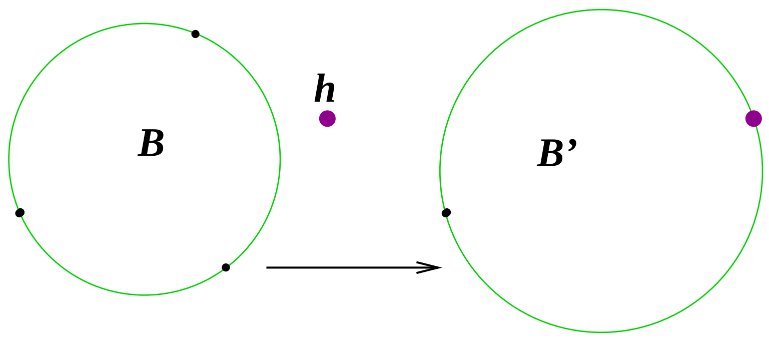
The LP-type algorithm (I)

Two primitives needed (B a Basis, h a constraint).

- *violation test*: $w(B \cup \{h\}) > w(B)$?



- *basis computation*: compute a basis B' of $B \cup \{h\}$ $\quad$ ($B' =$ basis(B, h))



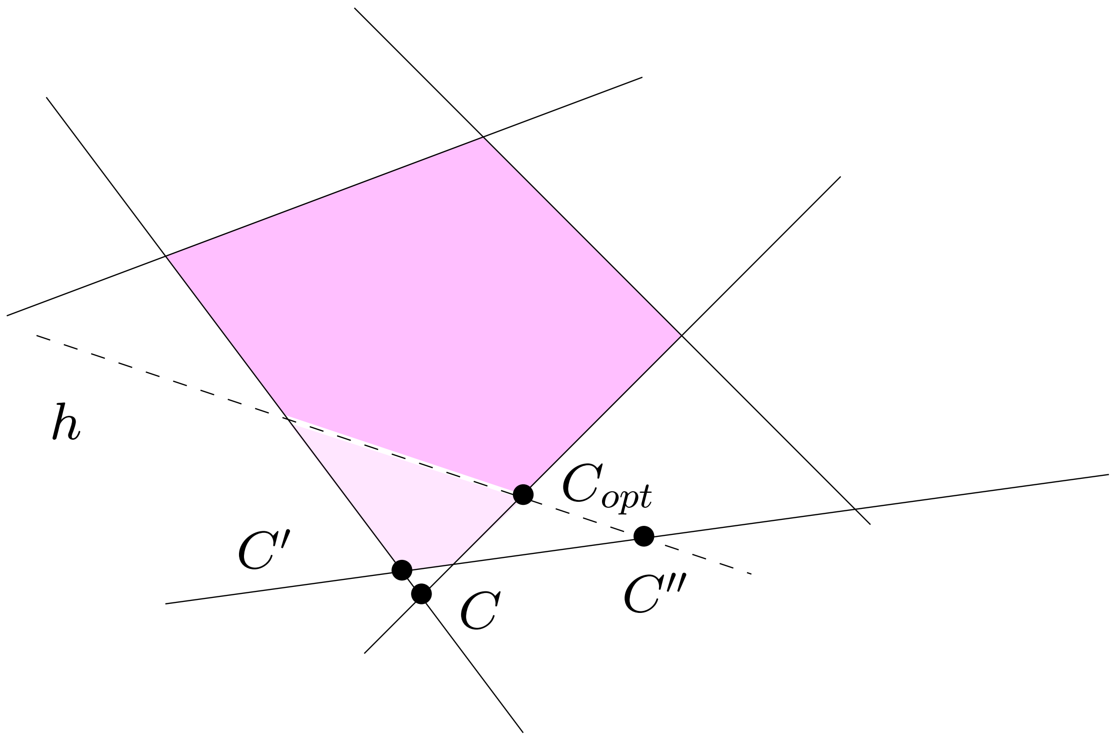
The LP-type algorithm (II)

Input: LP-type problem (H, w), *basis-regular*

Output: A Basis B of H (so $w_{opt} = w(B)$)

```
lp-type($G, C$): (* $G \subseteq H, C \subseteq G$ some basis *)
    IF $G = C$ THEN
        RETURN $C$
    ELSE
        choose $h \in G \setminus C$ at random
        $C' :=$ lp-type($G \setminus \{h\}, C$)
        IF $w(C' \cup \{h\}) > w(C')$ THEN
            $C'' :=$ basis($C', h$)
            RETURN lp-type($G, C''$)
        ELSE
            RETURN $C'$
        END
    END
```

lp-type on LP



- remove h not contributing to C and recursively solve the subproblem ($\Rightarrow C'$)
- if optimum not reached yet, compute the basis of $C \cup \{h\}$ ($\Rightarrow C''$)
- repeat from C''

lp-type – Correctness

Induction on

$$|G| \text{ (“problem size”)}$$

and

$$w(G) - w(C) \text{ (“optimality gap”)}.$$

Induction basis.

- $|G| = \delta$ ($\Leftrightarrow w(C) = w(G)$) implies that C is a basis of G (basis-regularity!) $\Rightarrow$ return value C correct

Inductive step:

- In first recursice call, problem size decreases
- In second recursive call, optimality gap decreases

lp-type – Analysis (I)

Observation: $h \in C''$, and h is contained in *all* bases appearing during the call to

lp-type(G, C'').

Proof: Let $C_0 \subseteq G \setminus \{h\}$ be any basis not containing h, C_1 any basis traced during the call to **lp-type**(G, C''). Then

$$w(C_1) \geq w(C'') > w(C') = w(G \setminus \{h\}) \geq w(C_0).$$

It follows that $h \in C_1$.

Definition 1: h is *enforced* in $(G, C), C \subseteq G$ if $w(C) > w(G \setminus \{h\})$. (It follows that $h \in C$.)

Definition 2: The *hidden dimension* ("degree of freedom") of (G, C) is the number

$$\delta(G, C) := \delta - \#\{h \in G \mid h \text{ enforced in } (G, C)\}.$$

lp-type – Analysis (II)

$t(n,k)$:= maximum expected number of violation tests in a call to **lp-type**(G,C) with $|G| = n$ and $\delta(G,C) \leq k$.

Lemma 1:

$$\begin{aligned} t(\delta,k) &= 0, \\ t(n,0) &= n-\delta, \\ t(n,k) &\leq t(n-1,k)+1+\frac{k}{n-\delta}t(n,k-1). \end{aligned}$$

Proof: (i) e enforced in (G,C) implies e enforced in $(G\setminus\{h\},C)$, (G,C'') and $e \in B$ for any basis B of G. (ii) h is enforced in (G,C''). (iii) Second recursive call only occurs for elements $h \in B$ which are not yet in C (there are at most k).

Theorem 1:

$$t(n,k) \leq \sum_{j=0}^{k} \frac{1}{j!}\ k!\ (n-\delta) \leq e\ k!\ (n-\delta).$$

LP – Randomized complexity

Theorem: Any linear program with n constraints in d variables can be solved in expected time $O(n)$ if d is fixed.

Proof: lp-type performs an expected number of at most $t(n, d)$

- violation tests $\rightarrow O(d)$
- basis computations $\rightarrow O(d^3)$

Expected runtime

$$O(d^3\ d!\ n).$$

Still an exponential algorithm if $n = O(d)$!

lp-type – Analysis (III)

Have seen: hidden dimension decreases by 1 from (G, C) to (G, C'').

Now show: hidden dimension halves on average from (G, C) to (G, C'')!

Consider the (at most) k elements in $B \setminus C$ leading to a second recursive call, ordered such that

$$w(G \setminus \{h_1\}) \leq w(G \setminus \{h_2\}) \leq \cdots \leq w(G \setminus \{h_k\}).$$

With $h = h_i$, we get

$$w(C'') > w(C') = w(G \setminus \{h_i\}) \geq \cdots \geq w(G \setminus \{h_1\})$$

$\Rightarrow h_1, \ldots, h_i$ are enforced in (G, C'')
$\Rightarrow k(G, C'') \leq k(G, C) - i$.

Lemma 2:

$$t(n,k) \leq t(n-1,k) + 1 + \frac{1}{n-\delta} \sum_{i=1}^{\min(k,n-\delta)} t(n, k-i).$$

lp-type – Analysis (IV)

Theorem 2:

$$t(n,k) \leq 2^k(n-\delta).$$

Improvement in the "big-Oh":

$$e\ k! \quad \rightarrow \quad 2^k.$$

For n not too large, the following is the best possible result.

Theorem 3:

$$t(n,k) \leq \exp\left(2\sqrt{k \ln n} + O(\sqrt{k} + \ln n)\right).$$

Subexponential algorithm (Matoušek, Sharir, Welzl '92)!!

The situation $n = 2\delta$...

...is in a sense the most difficult.

- **Theorem 1:**
$$t(n,\delta) \le e\ \delta\ \delta!.$$

- **Theorem 2:**
$$t(n,\delta) \le \delta\ 2^{\delta}.$$

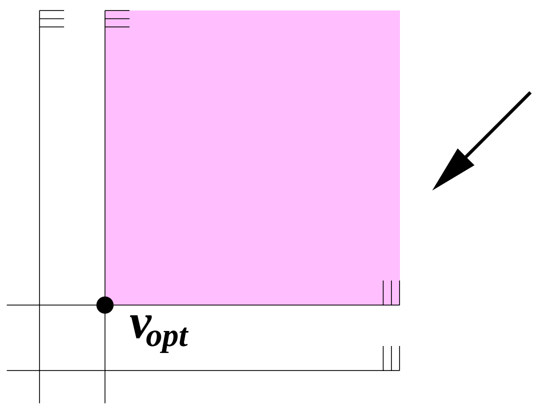
- **Theorem 3:**
$$t(n,\delta) \le e^{O(\sqrt{d \ln \delta})}.$$

Dual cubes

or: understanding the subexponential bound in a special scenario

A toy LP:

$$\begin{array}{lll} \text{min} & x_1 + \ldots + x_d & \\ \text{subject to} & \begin{array}{l} x_i \geq 0 \\ x_i \geq 1 \end{array}, & i = 1 \ldots d. \end{array}$$



$h_i := \{x_i \geq 0\}, \quad \bar{h}_i := \{x_i \geq 1\}$

$H := \{h_i \mid i = 1 \ldots d\} \cup \{\bar{h}_i \mid i = 1 \ldots d\}$

$$w(G) = \begin{cases} -\infty, & \text{if } \exists i : G \cap \{h_i, \bar{h}_i\} = \emptyset, \\ \#\{i \mid \bar{h}_i \in G\}, & \text{otherwise} \end{cases}$$

Definition 3: A *dual cube of dimension* δ is a basis-regular LP-type problem $\mathcal{L} = (H, w)$ with

- $|H| = 2\delta$
- there is an involution $\bar{\ }: H \to H$, i.e.

$$\begin{array}{l} \bar{h} \neq h \\ \bar{\bar{h}} = h \end{array}, \quad \text{for all } h \in H$$

- $w(G) = -\infty$ if and only if $G \cap \{h, \bar{h}\} = \emptyset$ for some h

Implications:

- B is basis if $|B| = \delta$ and $|B \cap \{h, \bar{h}\}| = 1, h \in H$. Thus, there are 2^δ bases
- the toy LP defines a dual cube

lp-type on dual cubes (I)

$b(n,k)$:= maximum expected number of *basis computations* in a call to **lp-type**(G,C) with $|G| = n$ and $\delta(G,C) \le k$.

$$\beta(k) := \max_{m \le n} b(m,k).$$

Lemma 3:

$$\begin{aligned} \beta(0) &= 0, \\ \beta(k) &\le \beta(k-1) + \frac{1}{k}\sum_{i=1}^{k}(1+\beta(k-i)). \end{aligned}$$

lp-type on dual cubes (II)

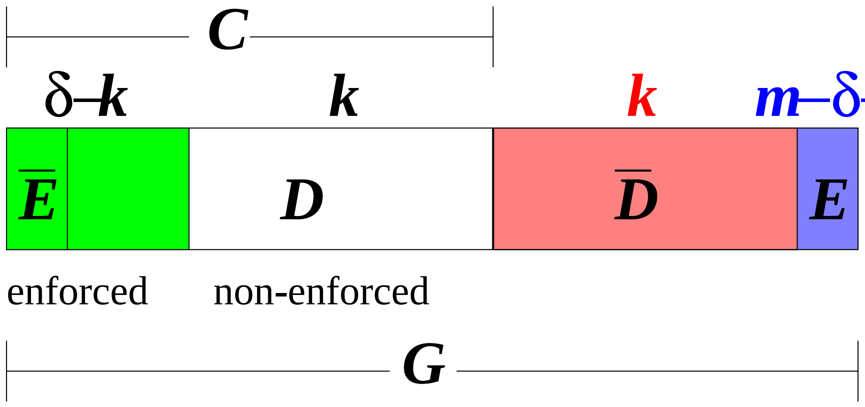
Proof: Let (G, C), $|G| = m, \delta(G, C) \leq k$ be the "worst-case problem" defining $\beta(k)$.

Case 1: $\delta(G, C) < k$. Then $\beta(k) = \beta(k-1)$.

Case 2: $\delta(G, C) = k$. Then

$$\begin{aligned} \beta(k) &\leq \frac{m-\delta-k}{m-\delta}\beta(k) \\ &+ \frac{k}{m-\delta}\left(\beta(k-1) + 1 + \frac{1}{k}\sum_{i=1}^{k}\beta(k-i)\right). \end{aligned}$$

$\Rightarrow$ Lemma 3.



lp-type on dual cubes (II)

Case 2 (contd.):

Let $h \in G \setminus C$ be the element removed for the first recursive call.

Case (a): $h \in E$. Because $\bar{h}$ is enforced, h is 'redundant' $\Rightarrow \delta(G \setminus \{h\}, C) = \delta(G, C) = k$, and no second recursive call.

Case (b): $h \in \bar{D}$. Then $\bar{h}$ is enforced in $(G \setminus \{h\}, C) \Rightarrow \delta(G \setminus \{h\}, C) \leq k-1$. Let $h_1, \ldots h_k \in \bar{D}$ be ordered such that

$$w(G \setminus \{h_1\}) \leq \cdots \leq w(G \setminus \{h_k\}).$$

If $h = h_i$, then $h_1, \ldots h_i$ are enforced in (G, C'') (because $w(C'') \geq w(G \setminus \{h_i\})) \Rightarrow \delta(G, C'') \leq k - i$.

lp-type on dual cubes (IV)

Theorem: The expected number of basis computations needed to solve a dual cube of dimension δ is bounded by

$$b(2\delta, \delta) \leq \beta(\delta).$$

Corollary: The expected number of violation tests is bounded by

$$t(2\delta, \delta) \leq \delta \cdot (\beta(\delta) + 1).$$

Proof: Any basis computation is followed by at most δ violation tests in a row, and there might be δ violation tests charged to the initial basis.

Analysis of $\beta(k)$

Define

$$\begin{aligned} f(0) &= 1, \\ f(k) &= f(k-1) + \frac{1}{k}\sum_{i=1}^{k} f(k-i). \end{aligned}$$

Then $\beta(k) \le f(k) - 1$.

Theorem 4:

$$f(k) = \sum_{i=0}^{k} \frac{1}{i!}\binom{k}{i}.$$

The asymptotics of $f(k)$

$$\begin{aligned}
f(k) &= \sum_{i=0}^{k} \binom{k}{i} \frac{1}{i!} \\
&\le \sum_{i=0}^{k} \frac{k^i}{i!} \frac{1}{i!} \\
&= \sum_{i=0}^{k} \frac{k^i}{i!^2} \\
&= \sum_{i=0}^{k} (\frac{\sqrt{k}^i}{i!})^2 \\
&\le (\sum_{i=0}^{k} \frac{\sqrt{k}^i}{i!})^2 \\
&\le (\sum_{i=0}^{\infty} \frac{\sqrt{k}^i}{i!})^2 = e^{2\sqrt{k}}.
\end{aligned}$$

Theorem: the expected number of basis computations needed to solve a dual cube is bounded by

$$b(2\delta, \delta) \le \beta(\delta) \le f(\delta) - 1 \le e^{2\sqrt{\delta}} - 1.$$

Exercise 1.1 Prove that the problem *angle-optimal placement* can be formulated as an LP-type problem. You may assume general position in the sense that there is always a unique angle-optimal placement, in the problem itself and in subproblems you define.

Furthermore, prove that the combinatorial dimension of the problem is bounded by 3.

Hint: You may assume *Helly's Theorem*:

> Let C be a collection of $n \geq d+1$ convex sets in $\mathbb{R}^d$. If any $d+1$ members of C have a point in common, then there is a point in common to all members of C.

Exercise 1.2 Show that the problems *smallest enclosing annulus* and *largest disk in kernel* are LP-type problems by showing that they can be formulated as Linear Programs.

Exercise 1.3 Prove the "easy" bounds

$$t(n,k) \leq \sum_{j=0}^{k} \frac{1}{j!}\, k!\ (n-\delta)$$

and

$$t(n,k) \leq 2^k (n-\delta)$$

(Theorem 1 and 2) for the number of violation tests in a call to **lp-type**(G,C) with $|G| = n$ and $\delta(G,C) \leq k$.

Exercise 1.4 Prove the explicit formula

$$f(k) = \sum_{i=0}^{k} \frac{1}{i!} \binom{k}{i}$$

(Theorem 4) for the function $f(k)$ defined on page 41.

Exercise 1.5 What is the expected number of basis computations resp. violation tests performed by **lp-type** in solving the "toy LP" (page 35)?

Exercise 1.6 Consider LP-type problems that are not necessarily basis-regular. Does algorithm **lp-type** still work? If not, how can we make it work again? Prove that the $O(n)$ bound for fixed combinatorial dimension δ still holds.

Exercise 1.7 What is the combinatorial dimension of LP in the LP-type framework when also infeasible problems are considered? What about basis-regularity in this case?

Exercise 1.8 Consider the non-local problems *smallest enclosing axis-parallel square* and *closest pair of points*. Are they inherently non-local, or can one achieve locality by a trick similar to the one we used to make LP local?

Facets of the Polytope World

Late Summer School · ETH Zürich · 1999

Randomization and Abstraction – Useful Tools for Optimization

Part II

by Bernd Gärtner

Games on natural numbers

Game $A(n)$:

```
    $x := n$
    WHILE $x > 0$ DO
        set $x$ to a *random* number in $[0, x-1]$
    END
```

$$\begin{tikzcd}
\mathbf{457} \arrow[d] & \\
\mathbf{266} \arrow[d] & \\
\mathbf{43} \arrow[d] & \\
\mathbf{7} \arrow[d] & \text{6 rounds} \\
\mathbf{6} \arrow[d] & \\
\mathbf{1} \arrow[d] & \\
\mathbf{0} &
\end{tikzcd}$$

$a(n) :=$expected number of rounds in $A(n)$

Game $B(n)$:

```
    $x := n = (b_{d-1}, \ldots, b_0)$ in binary encoding
    WHILE $x > 0$ DO
        choose *random* $k$ with $b_k = 1$
        FOR $i := 0$ TO $k$ DO
            $b_i := 1 - b_i$
        END
    END
```

$$
\begin{array}{cccccr l}
1 & 1 & 0 & 1 & 1 & \mathbf{27} & \\
 & & & \downarrow & & & \\
1 & 1 & 0 & 0 & 0 & \mathbf{24} & \\
\downarrow & & & & & & \\
0 & 0 & 1 & 1 & 1 & \mathbf{7} & \\
 & & & \downarrow & & & \text{5 rounds} \\
0 & 0 & 1 & 0 & 0 & \mathbf{4} & \\
 & & \downarrow & & & & \\
0 & 0 & 0 & 1 & 1 & \mathbf{3} & \\
 & & & \downarrow & & & \\
0 & 0 & 0 & 0 & 0 & \mathbf{0} &
\end{array}
$$

$b(n)$:=expected number of rounds in $B(n)$

Analysis of Game A

$$\begin{aligned} a(0) &= 0 \\ a(n) &= 1 + \frac{1}{n}\sum_{i=0}^{n-1} a(i) \end{aligned}$$

This implies

$$\begin{aligned} a(n) &= H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \\ &\approx \ln n. \end{aligned}$$

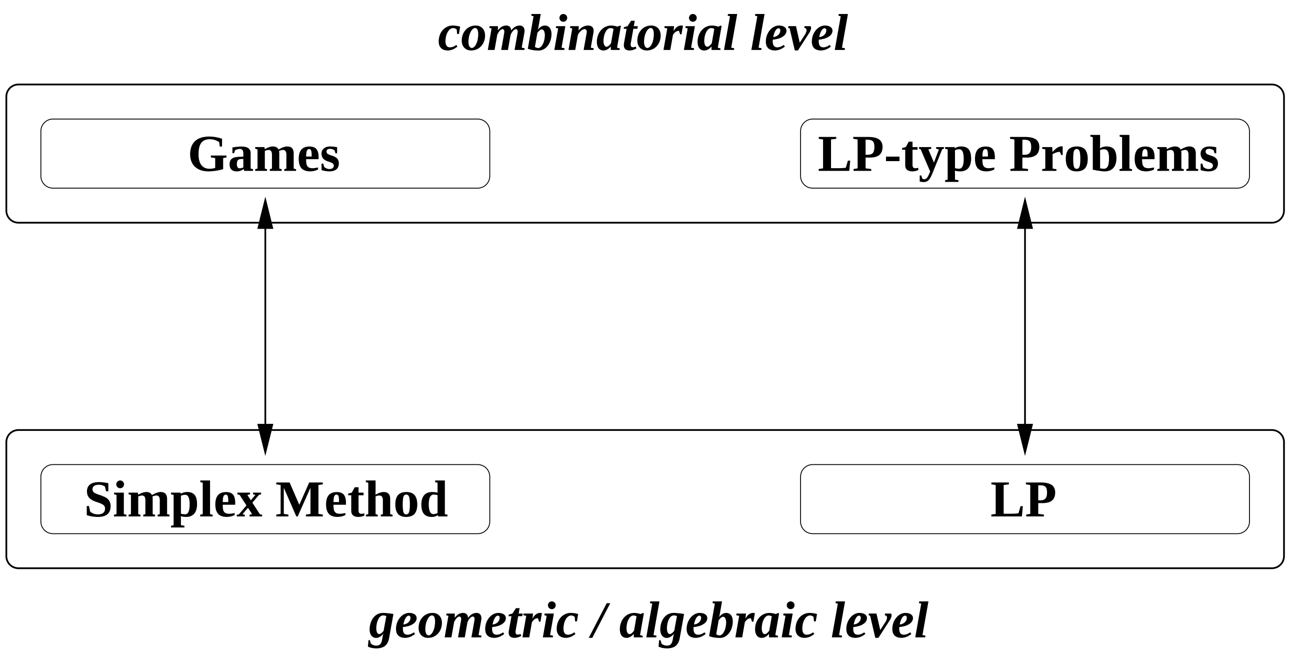
Analysis of Game B

$d = \lfloor \log_2 n \rfloor + 1$, number of significant bits

- $b(n) \le d(d+1)/2 \approx \log^2 n$
- Papadimitriou & Steiglitz (Exercise 8.10*):
$$b(n) = O(d)$$
- Papadimitriou (personal communication):
$$b(n) = O(d \log d) \quad \text{(maybe)}$$
- Kelly (conjecture, based on empirical tests):
$$b(n) = O(d \log^2 d)$$
- Gärtner, Ziegler (1994):
$$\exists n \ (d \text{ bits}) : b(n) \ge \frac{d^2}{4(H_{d+1} - 1)} \approx \frac{d^2}{\log d}$$

Do the games mean anything?

Just like **LP-type problems** are a combinatorial abstraction of **linear programming** . . .



. . . the **games** are a combinatorial abstraction of the **simplex method** for solving linear programming problems!

The simplex method

Linear Programming (Standard form).

$$
\text{(LP)} \quad \begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax \le b, \\ & x \ge 0, \end{array}
$$

Example.

$$
\begin{array}{lrcrcl}
\text{maximize} & x_1 + x_2 & & & & \\
\text{subject to} & -x_1 & + & x_2 & \le & 1, \\
& x_1 & & & \le & 3, \\
& & & x_2 & \le & 2, \\
& x_1, x_2 \ge 0. & & & &
\end{array}
$$

After introducing *slack* variables:

$$
\begin{array}{lrclcrcl}
\text{maximize} & x_1 + x_2 & & & & & & \\
\text{subject to} & x_3 & = & 1 & + & x_1 & - & x_2, \\
& x_4 & = & 3 & - & x_1, & & \\
& x_5 & = & 2 & & & - & x_2, \\
& x_1, \ldots, x_5 \ge 0. & & & & & &
\end{array}
$$

Tableaus

Rewrite the LP in *tableau* form.

$$\begin{array}{lclclcl}
x_3 & = & 1 & + & x_1 & - & x_2 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 2 & & & - & x_2 \\
\hline
z & = & & & x_1 & + & x_2
\end{array}$$

- Tableau expresses *basic* variables x_B and the objective function value z in terms of *nonbasic* variables x_N

- Any tableau solution with $x \geq 0$ is an LP solution

- $x_N = 0 \Rightarrow$ *basic feasible solution* (BFS)

The pivot step (I)

Given: Tableau

$$\begin{array}{rcrcrcr} x_3 & = & 1 & + & x_1 & - & x_2 \\ x_4 & = & 3 & - & x_1 & & \\ x_5 & = & 2 & & & - & x_2 \\ \hline z & = & & & x_1 & + & x_2 \end{array}$$

and associated basic feasible solution

$$x^* = (0, 0, 1, 3, 2).$$

Goal: increase z by making some nonbasic variable positive, keeping the basic ones non-negative

Candidates: the ones with positive coefficient in the z-row.

If x_2 is chosen:

$$\underbrace{\begin{array}{rcl} x_3 & = & 1 + x_1 - x_2 \quad \Rightarrow \quad x_2 \leq 1, \\ x_4 & = & 3 - x_1 \quad \Rightarrow \quad \text{no limit for } x_2 \\ x_5 & = & 2 - x_2 \quad \Rightarrow \quad x_2 \leq 2. \end{array}}_{\Rightarrow \text{ increase } x_2 \text{ by } 1,\ x_3 \text{ becomes zero}}$$

The pivot step (II)

Rewriting the tableau:

$$\begin{array}{lcl} x_2 & \sim & \textit{entering}\text{ variable} \\ x_3 & \sim & \textit{leaving}\text{ variable} \end{array}$$

Solve the equation

$$\mathbf{x_3} = 1 + x_1 - \mathbf{x_2}$$

for x_2 and subsitute

$$\mathbf{x_2} = 1 + x_1 - \mathbf{x_3}$$

on the right hand side of the tableau.

New tableau:

$$\begin{array}{lcccccc} x_2 & = & 1 & + & x_1 & - & x_3 \\ x_4 & = & 3 & - & x_1 & & \\ x_5 & = & 1 & - & x_1 & + & x_3 \\ \hline z & = & 1 & + & 2x_1 & - & x_3 \end{array}$$

New BFS:

$$x^* = (0, 1, 0, 3, 1), \quad z = 1$$

The pivot step (III)

$$\begin{array}{rclcrcr}
x_2 & = & 1 & + & x_1 & - & x_3 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 1 & - & x_1 & + & x_3 \\
\hline
z & = & 1 & + & 2x_1 & - & x_3
\end{array}$$

$$x^* = (0, 1, 0, 3, 1), \quad z = 1$$

x_1 **enters,** x_5 **leaves:**

$$\begin{array}{rclcrcr}
x_2 & = & 2 & & & - & x_5 \\
x_4 & = & 2 & - & x_3 & + & x_5 \\
x_1 & = & 1 & + & x_3 & - & x_5 \\
\hline
z & = & 3 & + & x_3 & - & 2x_5
\end{array}$$

$$x^* = (1, 2, 0, 2, 0), \quad z = 3.$$

x_3 **enters,** x_4 **leaves:**

$$\begin{array}{rclcrcr}
x_2 & = & 2 & & & - & x_5 \\
x_3 & = & 2 & - & x_4 & + & x_5 \\
x_1 & = & 3 & - & x_4 & & \\
\hline
z & = & 5 & - & x_4 & - & x_5
\end{array}$$

$$x^* = (3, 2, 2, 0, 0), \quad z = 5$$

Termination

$$\begin{array}{rclcl}
x_2 & = & 2 & & - \; x_5 \\
x_3 & = & 2 & - \; x_4 & + \; x_5 \\
x_1 & = & 3 & - \; x_4 & \\
\hline
z & = & 5 & - \; x_4 & - \; x_5
\end{array}$$

$$x^* = (3, 2, 2, 0, 0), \qquad z = 5$$

z cannot be increased anymore by increasing nonbasic variables (local optimum).

Observation: Any local optimum is a global optimum.

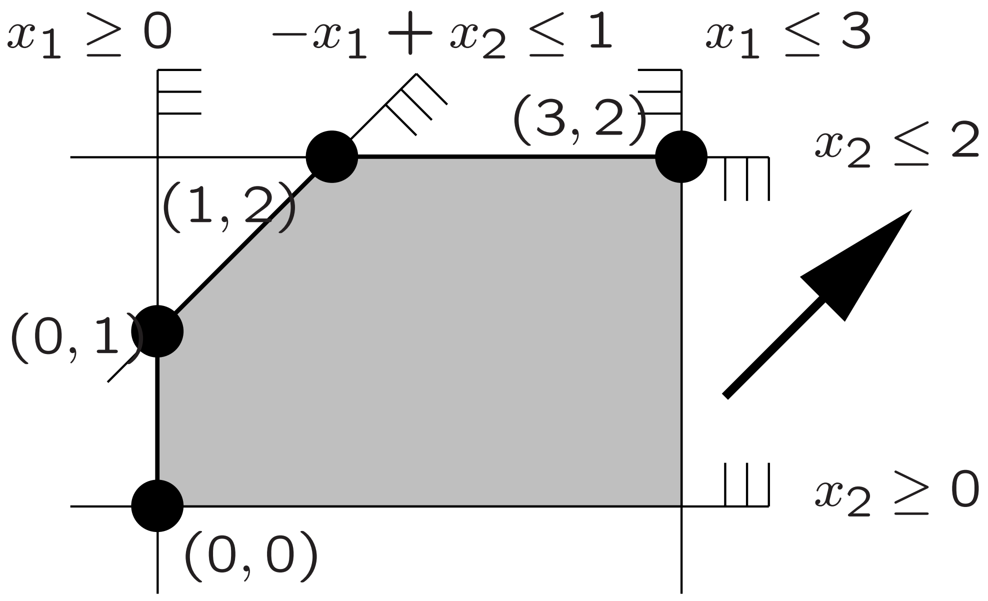
Proof:

$$\underbrace{\begin{array}{ccccc} z = 5 & - & x_4 & - & x_5 \\ & & x_4 \geq 0 & & \\ & & & & x_5 \geq 0 \end{array}}_{\text{implies } z \leq 5}$$

Current value $z = 5$ is optimal!

Geometric interpretation

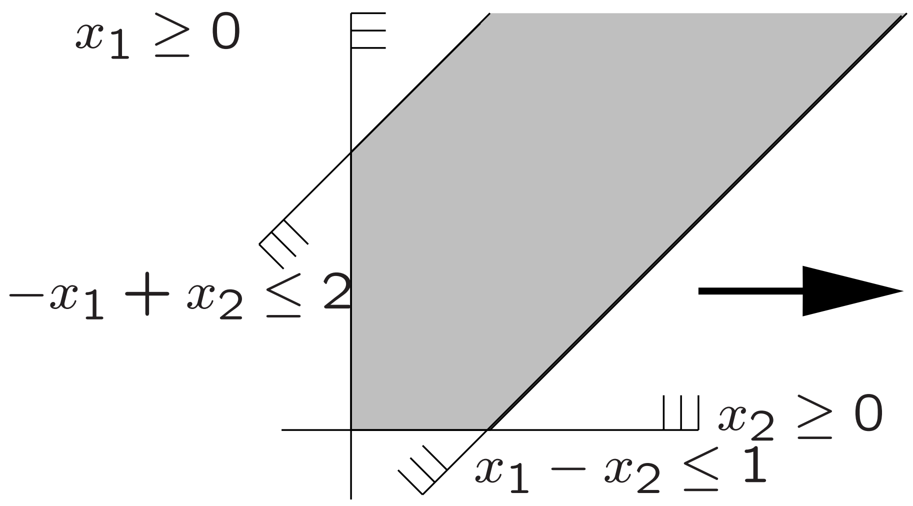
Feasible region	$\rightarrow$	*polyhedron* P
BFS	$\leftrightarrow$	*vertex* of P
Pivot step	$\leftrightarrow$	move along an *edge* of P, monotonically increasing z



BFS	vertex	$z = x_1 + x_2$
$(0, 0, 1, 3, 4)$	$(0, 0)$	0
$(0, 1, 0, 3, 1)$	$(0, 1)$	1
$(1, 2, 0, 2, 0)$	$(1, 2)$	3
$(3, 2, 2, 0, 0)$	$(3, 2)$	5

Unboundedness (I)

$$\begin{array}{llrcrcl} \text{maximize} & & x_1 & & & & \\ \text{subject to} & & x_1 & - & x_2 & \leq & 1, \\ & & -x_1 & + & x_2 & \leq & 2, \\ & x_1, x_2 \geq 0. & & & & & \end{array}$$



Initial tableau:

$$\begin{array}{lcrcrcr} x_3 & = & 1 & - & x_1 & + & x_2 \\ x_4 & = & 2 & + & x_1 & - & x_2 \\ \hline z & = & & & x_1 & & \end{array}$$

Unboundedness (II)

$$\begin{array}{rcccccc} x_3 & = & 1 & - & x_1 & + & x_2 \\ x_4 & = & 2 & + & x_1 & - & x_2 \\ \hline z & = & & & x_1 & & \end{array}$$

x_1 **enters,** x_3 **leaves:**

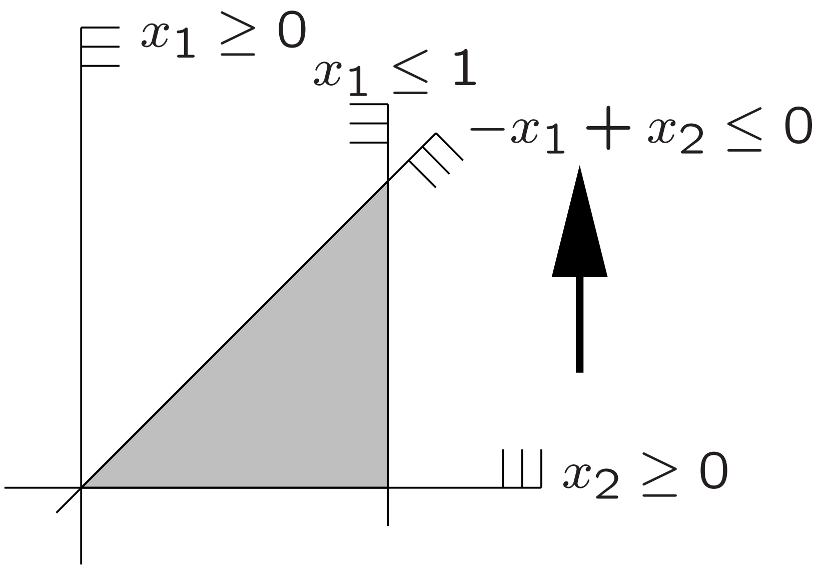
$$\begin{array}{rcccccc} x_1 & = & 1 & + & x_2 & - & x_3 \\ x_4 & = & 3 & & & - & x_3 \\ \hline z & = & 1 & + & x_2 & - & x_3 \end{array}$$

No bound on the increase of $x_2 \Rightarrow z_{opt} = \infty$

Geometric interpretation: Polyhedron is unbounded in the optimization direction.

Degeneracy (I)

$$\begin{array}{llcl} \text{maximize} & x_2 & & \\ \text{subject to} & -x_1 + x_2 & \le & 0, \\ & x_1 & \le & 2, \\ & x_1, x_2 \ge 0, & & \end{array}$$



Initial tableau:

$$\begin{array}{lcl} x_3 & = & x_1 - x_2 \\ x_4 & = & 2 - x_1 \\ \hline z & = & x_2 \end{array}$$

Degeneracy (II)

$$\begin{array}{rcrcrcr} x_3 & = & & & x_1 & - & x_2 \\ x_4 & = & 2 & - & x_1 & & \\ \hline z & = & & & & & x_2 \end{array}$$

x_2 enters, and . . . the increase in z is zero!

This does *not* imply global optimality!

x_2 enters, x_3 leaves:

$$\begin{array}{rcrcrcr} x_2 & = & & & x_1 & - & x_3 \\ x_4 & = & 2 & - & x_1 & & \\ \hline z & = & & & x_1 & - & x_3 \end{array}$$

$$x^* = (0, 0, 0, 2), \quad z = 0$$

x_1 enters, x_4 leaves:

$$\begin{array}{rcrcrcr} x_1 & = & 2 & & & - & x_4 \\ x_2 & = & 2 & - & x_3 & - & x_4 \\ \hline z & = & 2 & - & x_3 & - & x_4 \end{array}$$

$$x^* = (2, 2, 0, 0), \quad z = 2 \Rightarrow \text{optimal}$$

Degeneracy (III)

Geometric interpretation: vertex is overdetermined; more than d hyperplanes go through it.

Degeneracy can lead to *cycling*!

Fix: Symbolic perturbation: $Ax \leq b + (\varepsilon, \varepsilon^2, \ldots)^T$

$$\begin{array}{lcrcrcr} x_3 & = & \varepsilon & + & x_1 & - & x_2 \\ x_4 & = & 2 + \varepsilon^2 & - & x_1 & & \\ \hline z & = & & & & & x_2 \end{array}$$

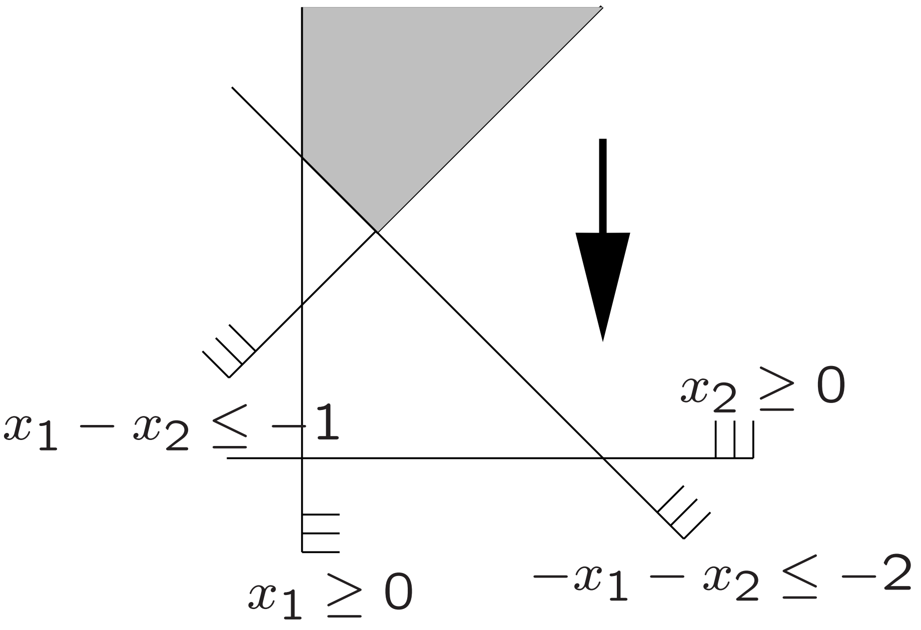
x_2 **enters,** x_3 **leaves:**

$$\begin{array}{lcrcrcr} x_2 & = & \varepsilon & + & x_1 & - & x_3 \\ x_4 & = & 2 + \varepsilon^2 & - & x_1 & & \\ \hline z & = & \varepsilon & + & x_1 & - & x_3 \end{array}$$

This time, we made ε-progress in z! In the end, ignore the ε's!

Infeasibility (I)

$$\begin{array}{llcll} \text{maximize} & -x_2 & & & \\ \text{subject to} & -x_1 & - & x_2 & \leq & -2, \\ & x_1 & - & x_2 & \leq & -1, \\ & x_1, x_2 \geq 0. \end{array}$$



Initial tableau:

$$\begin{array}{lcrcrcr} x_3 & = & -2 & + & x_1 & + & x_2 \\ x_4 & = & -1 & - & x_1 & + & x_2 \\ \hline z & = & & & & - & x_2 \end{array}$$

does not give a BFS!

Infeasibility (II)

Fix: set up auxiliary problem

$$\begin{array}{lrcrcrcr}
\text{maximize} & -x_0 & & & & & & \\
\text{subject to} & -x_1 & - & x_2 & - & x_0 & \leq & -2, \\
 & x_1 & - & x_2 & - & x_0 & \leq & -1, \\
 & \multicolumn{7}{l}{x_0, x_1, x_2 \geq 0.}
\end{array}$$

Observation:

(i) Auxiliary problem is feasible (choose x_0 large)

(ii) Original problem feasible $\Leftrightarrow$ auxiliary problem has optimal value $x_0 = 0$.

Infeasibility (III)

Initial tableau:

$$\begin{array}{rcrcrcrcr}
x_3 & = & -2 & + & x_1 & + & x_2 & + & x_0 \\
x_4 & = & -1 & - & x_1 & + & x_2 & + & x_0 \\
\hline
w & = & & & & & & - & x_0
\end{array}$$

Increase x_0 to obtain a BFS (w gets worse!)

x_0 enters, x_3 leaves:

$$\begin{array}{rcrcrcrcr}
x_0 & = & 2 & - & x_1 & - & x_2 & + & x_3 \\
x_4 & = & 1 & - & 2x_1 & & & + & x_3 \\
\hline
w & = & -2 & + & x_1 & + & x_2 & - & x_3
\end{array}$$

x_2 enters, x_0 leaves:

$$\begin{array}{rcrcrcrcr}
x_2 & = & 2 & - & x_1 & + & x_3 & - & x_0 \\
x_4 & = & 1 & - & 2x_1 & + & x_3 & & \\
\hline
w & = & & & & & & - & x_0
\end{array}$$

In the feasible case, x_0 is nonbasic in the end!

Infeasibility (IV)

To obtain a tableau for the original problem:

- Remove x_0 from the tableau

- Express z in the nonbasic variables

$$\begin{array}{rcrcrcrcr} x_2 & = & 2 & - & x_1 & + & x_3 & - & x_0 \\ x_4 & = & 1 & - & 2x_1 & + & x_3 & & \\ \hline w & = & & & & & & - & x_0 \end{array}$$

$$z = -x_2 = -2 + x_1 - x_3$$

Initial tableau for original problem is

$$\begin{array}{rcrcrcr} x_2 & = & 2 & - & x_1 & + & x_3 \\ x_4 & = & 1 & - & 2x_1 & + & x_3 \\ \hline z & = & -2 & + & x_1 & - & x_3 \end{array}$$

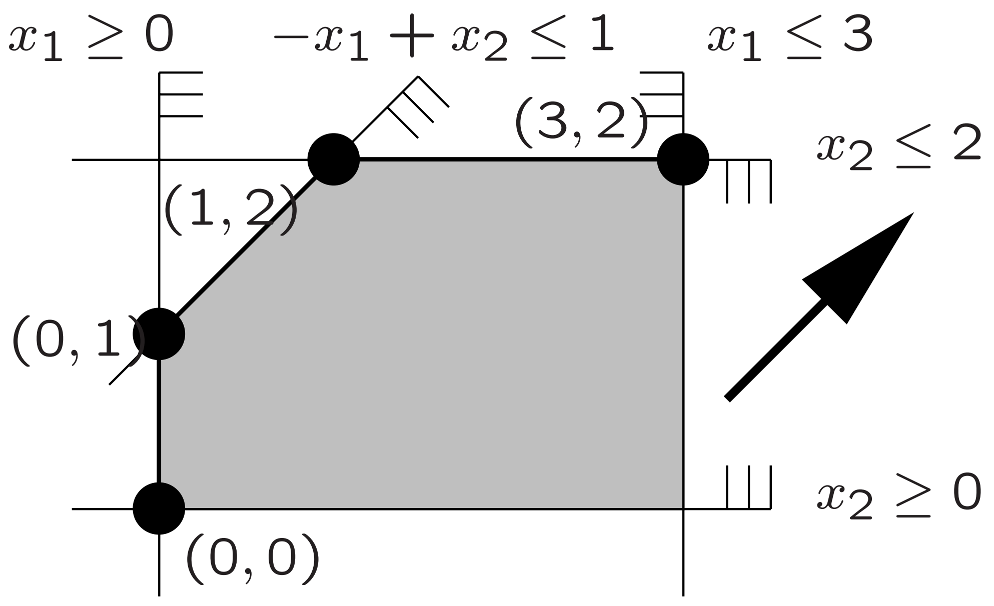
$$x^* = (0, 2, 0, 1)$$

Pivot rules (I)

How to choose the entering variable in case there are several choices?

$$\begin{array}{rcrcrcr} x_3 & = & 1 & + & x_1 & - & x_2 \\ x_4 & = & 3 & - & x_1 & & \\ x_5 & = & 2 & & & - & x_2 \\ \hline z & = & & & x_1 & + & x_2 \end{array}$$

Every choice corresponds to one z-increasing edge of the feasible region.



Two increasing edges in the initial vertex $(0, 0)$

Pivot rules (II)

Dantzig's rule: Choose the increasing variable with largest coefficient

Bland's rule: Choose the increasing variable with smallest index (according to some order)

Largest increase: Choose the increasing variable which leads to the largest increase in z

Steepest edge: Choose the increasing edge which is steepest w.r.t. the optimization direction

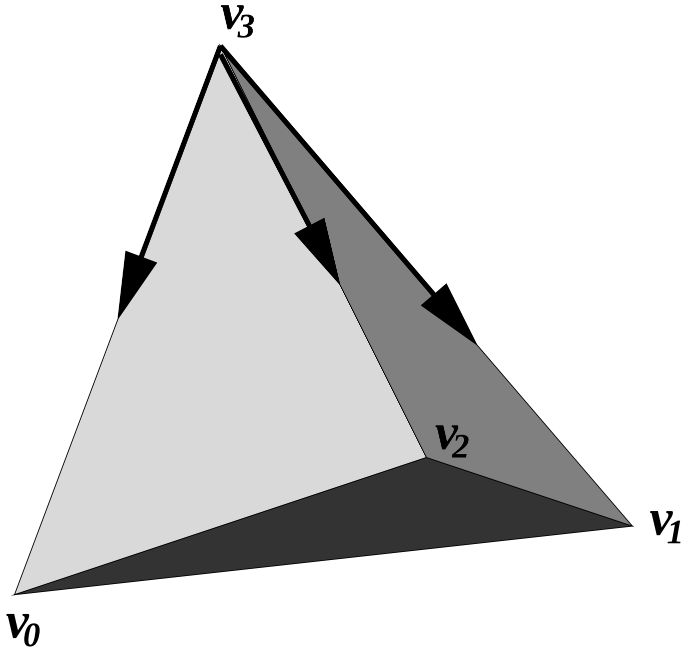
Random-Edge: Choose an increasing variable uniformly at random

Game A revisited (I)

$\{0, \ldots, n\} \leftrightarrow$ vertices $v_0, \ldots, v_n$ of n-simplex, such that

$$c^T v_0 < c^T v_1 < \cdots < c^T v_n$$

for some linear function $c^T x$.



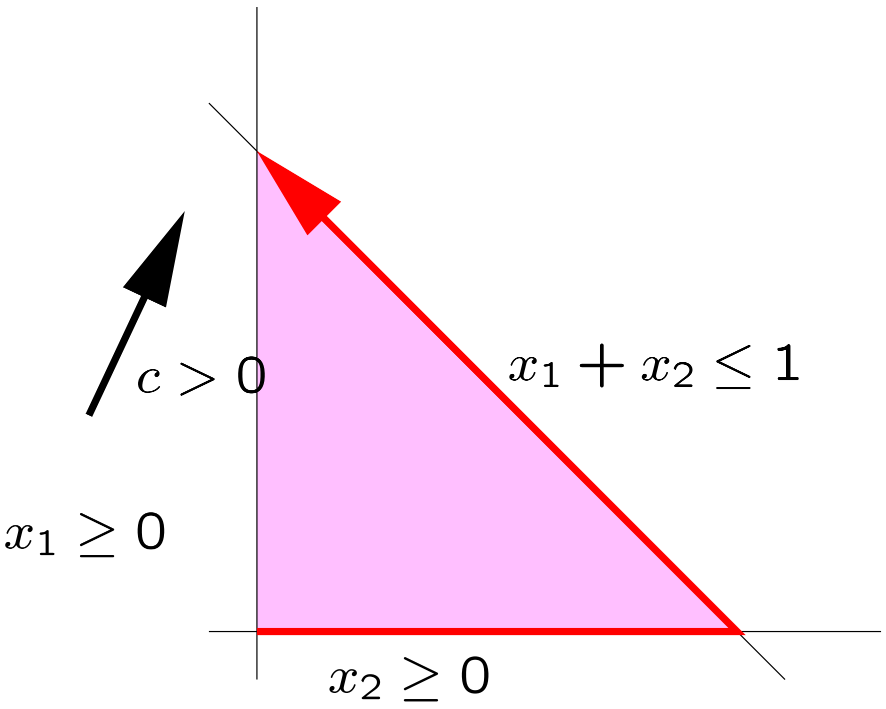
Game $A(n)$:

```
$v := v_n$
WHILE $v \neq v_0$ DO
    set $v$ to *random* neighbor $v', c^T v' < c^T v$
END
```

Game A revisited (II)

Game A models the simplex method with pivot rule **Random-Edge** on the LP

$$\begin{array}{ll} \text{maximize} & c_1x_1 + c_2x_2 + \cdots + c_dx_d \\ \text{subject to} & x_1 + x_2 + \cdots + x_d \leq 1, \\ & x_1, x_2, \ldots, x_d \geq 0. \end{array}$$



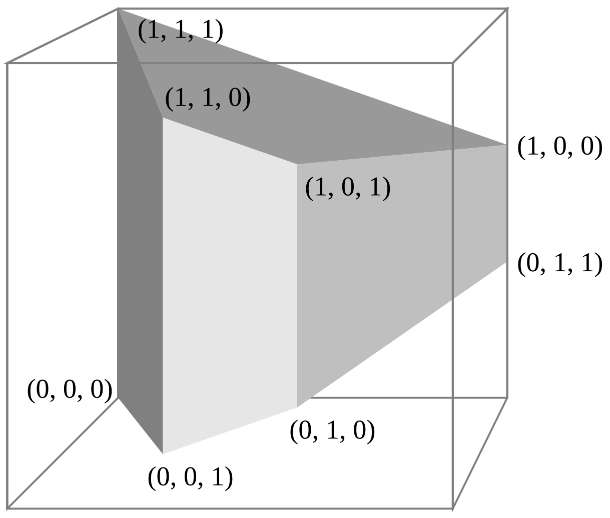
Simplex LP in standard form

Game B revisited (I)

$\{0,\ldots,2^d-1\} \leftrightarrow$ vertices $v_0,\ldots,v_{2^d-1}$ of d-dimensional *Klee-Minty cube*, such that

$$c^T v_0 < c^T v_1 < \cdots < c^T v_{2^d-1}$$

for some linear function $c^T x$.



minimize x_d
subject to

$0 \le x_1 \le 1$

$\varepsilon x_{i-1} \le x_i \le 1 - \varepsilon x_{i-1}$,

$i = 2,\ldots,d$

Game $B(n)$:

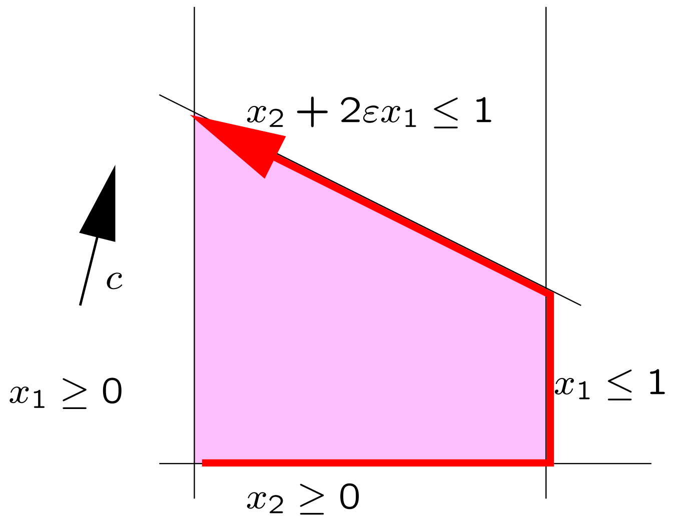
```
$v := v_n$
WHILE $v \neq v_0$ DO
    set $v$ to *random* neighbor $v', c^T v' < c^T v$
END
```

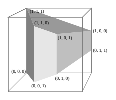
Game B revisited (II)

Game B models the simplex method with pivot rule **Random-Edge** on the LP

$$\begin{array}{lll}
\text{maximize} & x_d + \varepsilon x_{d-1} + \cdots & + \varepsilon^{d-1} x_1 \\
\text{subject to} & x_i + 2\varepsilon x_{i-1} + \cdots & + 2\varepsilon^{i-1} x_1 \leq 1, \\
 & & i = 1, \ldots, d \\
 & x_1, x_2, \ldots, x_d \geq 0. &
\end{array}$$

$(0 < \varepsilon < 1/2)$



 Klee-Minty cube LP in standard form, optimization direction reversed ($\varepsilon = 1/4$)

Simplex complexity (I)

Def.: Let $f_{\mathcal{P}}(n, d)$ denote the maximum (expected) number of *pivot steps* taken by the simplex method with pivot rule $\mathcal{P}$ on any linear program with n constraints and d variables.

- For $\mathcal{P} \in \{$*Dantzig's rule*, *Bland's rule*, *Largest increase*, *Steepest edge*,...$\}$

$$f_{\mathcal{P}}(2d, d) \geq c^d \quad \text{(exponential!)}$$

- $f_{\textit{Random-Edge}}(n, d) = \text{poly}(n, d)$???

- Is *some* pivot rule polynomial? This is the major open question in connection with the simplex method.

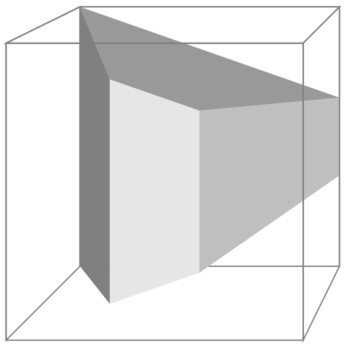
Simplex complexity (II)

Worst-case constructions:

- find polytope with *exponentially* long path of decreasing edges

- "Deform" the polytope such that the pivot rule is tricked into taking that path

"Default" polytope: Klee-Minty cube KM(d)

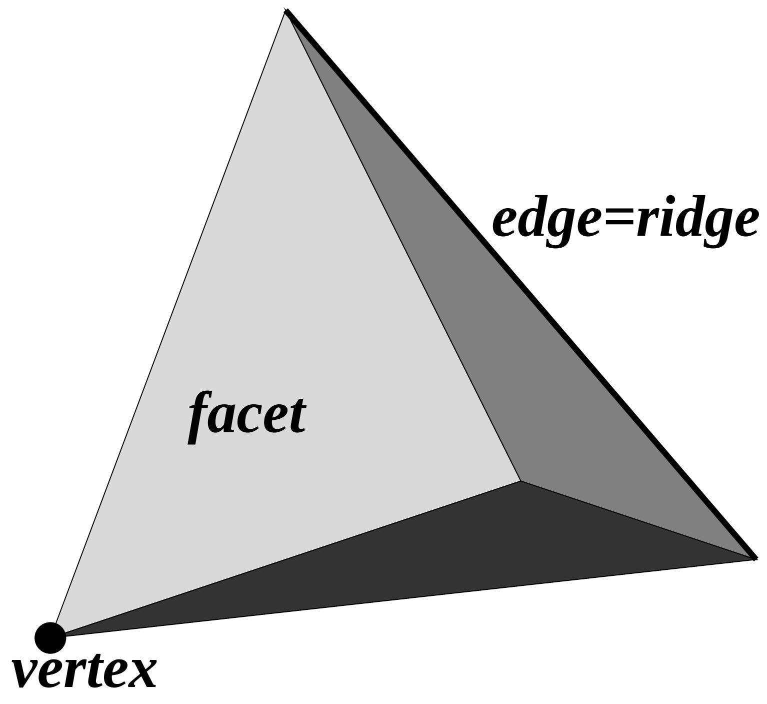


- $f_{\textit{Random-Edge}}(\text{KM}(d)) = O(d^2)$ (this was the analysis of Game B!)

Kalai’s pivot rule (I)

Polyhedra terminology:

k-dimensional	faces
0	vertices
1	edges
$d-2$	ridges
$d-1$	facets

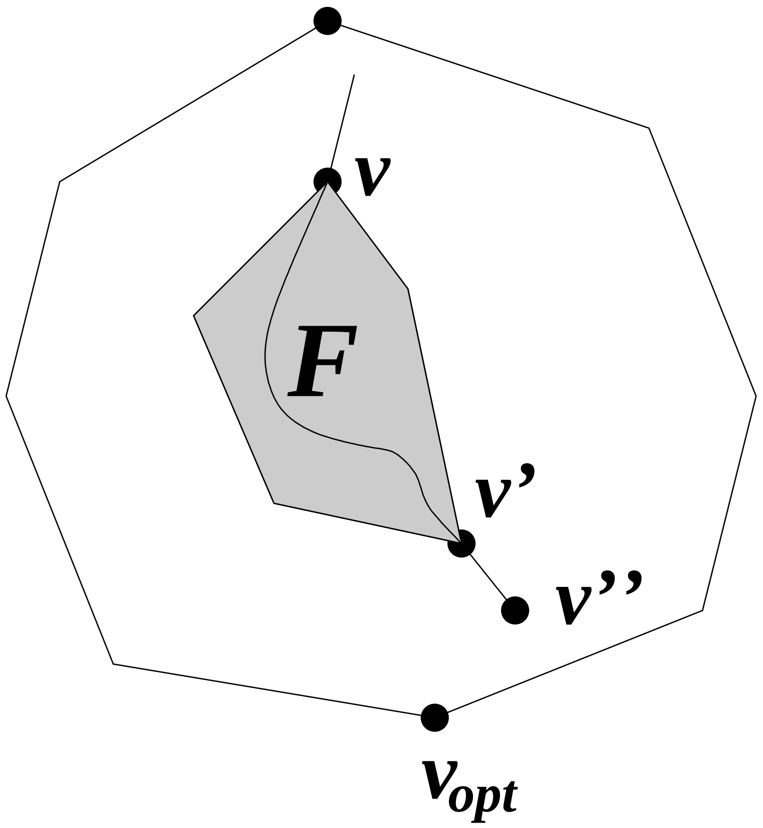


Every k-dimensional face is the intersection of $d-k$ facets.

Kalai's pivot rule (II)

Random-Facet(v): $v \sim$ current BFS

Choose a facet F containing v uniformly at random, and recursively optimize over F. If the vertex v' obtained is not the global optimum yet, move to its unique neighbor v'' with smaller objective function value, and repeat.



Kalai's pivot rule (III)

Thm.: (Kalai '92):

$$f_{\textit{Random-Facet}}(2d, d) = e^{O(\sqrt{d})}.$$

A **subexponential** pivot rule!

Flashback:

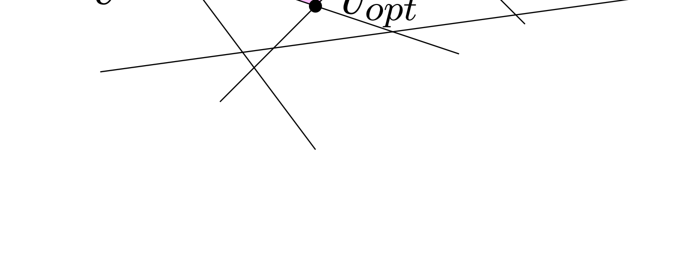
- Algorithm **lp-type**, dual cubes, . . .
- The expected number of basis computations needed to solve a dual cube is bounded by
$$b(2\delta, \delta) \leq e^{2\sqrt{\delta}} - 1.$$

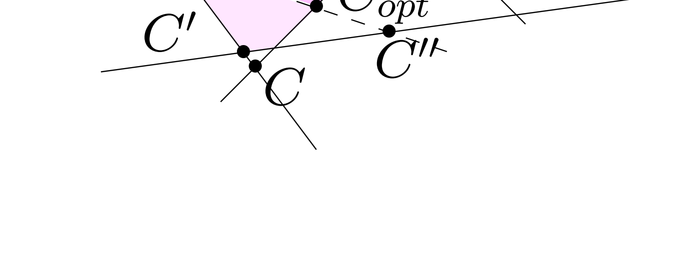
Random-Facet vs. lp-type(I)

Random-Facet(v):

Choose a facet F containing v uniformly at random, and recursively optimize over F. If the vertex v' obtained is not the global optimum yet, move to its unique neighbor v'' with smaller objective function value, and repeat.

```
lp-type$(G, C)$:
    IF $G = C$ THEN
        RETURN $C$
    ELSE
        choose $h \in G \setminus C$ at random
        $C' :=$ lp-type$(G \setminus \{h\}, C)$
        IF $w(C' \cup \{h\}) > w(C')$ THEN
            $C'' :=$ basis$(C', h)$
            RETURN lp-type$(G, C'')$
        ELSE
            RETURN $C'$
        END
    END
```





Random-Facet vs. lp-type(II)

- *Random-Facet* recursively **fixes** constraints; v is always *feasible* but only *(locally) optimal* in the end

- **lp-type** recursively **removes** constraints; C is always *(locally) optimal* but only *feasible* in the end

It's the same by LP duality!

LP duality (I)

lp-type setup

$$\Downarrow$$

$$\begin{array}{llll} \text{(LP)} & \min & c^T x & (d \text{ variables}) \\ & \text{s.t.} & Ax \geq b & (n \text{ constraints}) \end{array}$$

Let x be a feasible solution of (LP).

- for $y \in \mathbb{R}^n, y \geq 0$ we have $y^T A x \geq y^T b$
- if $y^T A = c^T$, then $c^T x \geq y^T b$

$$\begin{array}{llll} \text{(LP')} & \max & b^T y & (n \text{ variables}) \\ & \text{s.t.} & A^T y = c & (d \text{ constraints}) \\ & & y \geq 0 & \end{array}$$

$$\Uparrow$$

Random-Facet setup (standard form LP after introducing slack variables)

LP duality (II)

Assume (LP) is feasible and bounded ($\Rightarrow$ (LP) has an optimal solution).

Weak duality theorem: if (LP') is feasible, then

$$\max b^T y \leq \min c^T x.$$

Follows from the previous considerations.

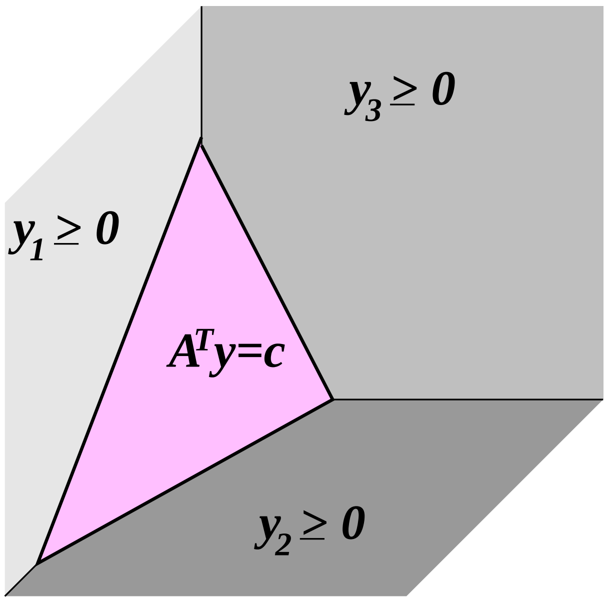
Strong duality theorem: (LP') is feasible, and

$$\max b^T y = \min c^T x.$$

(LP) and (LP') are *dual* to each other

LP duality (III)

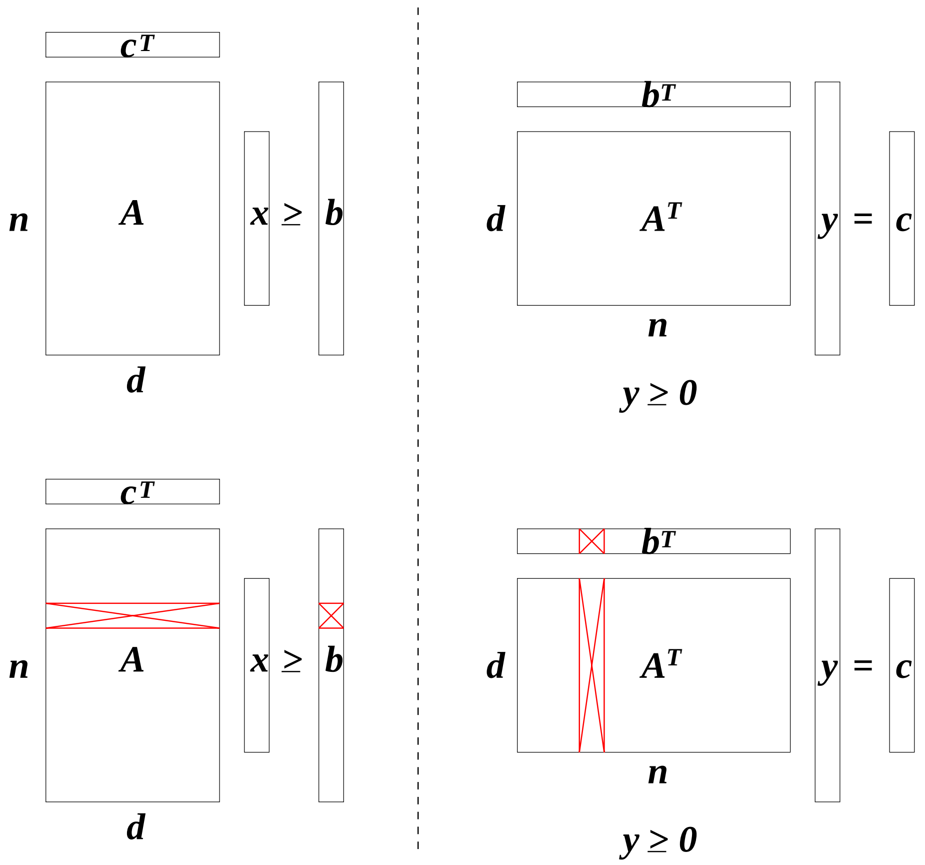
$$\begin{array}{llll} \text{(LP')} & \max & b^T y & (n \text{ variables}) \\ & \text{s.t.} & A^T y = c & (d \text{ constraints}) \\ & & y \geq 0 & \end{array}$$



Feasible region is a polyhedron

- of dimension $n - d$,
- with n facets $y_i = 0, \quad i = 1, \ldots, n$

LP duality (IV)



Removing constraint $A_i x \geq b_i$ in the primal
(lp-type)

$$\Updownarrow$$

Fixing a variable (facet) $y_i = 0$ in the dual
Random-Facet

Random-Facet vs. lp-type(III)

Kalai (working on polytopes)

$\Downarrow$

Subexponential **Random-Facet** rule on (LP')

$\updownarrow$

it's the same thing!

$\updownarrow$

Subexponential **lp-type** on (LP)

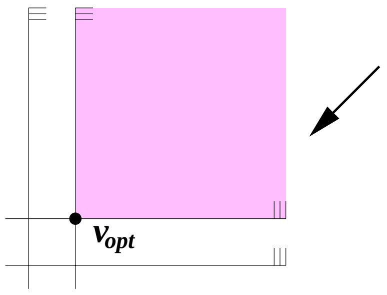
$\Uparrow$

Matoušek, Sharir and Welzl (working on geometric optimization problems)

Dual cubes revisited (I)

The toy LP (*dual cube*):

$$\begin{array}{ll} \text{min} & x_1 + \ldots + x_d \\ \text{subject to} & \begin{array}{c} x_i \geq 0 \\ x_i \geq 1 \end{array}, i = 1 \ldots d. \end{array}$$



Parameters:

$$c^T = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}$$

$$A = \left(\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \\ \hline 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{array}\right), \quad b = \left(\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ \hline 1 \\ 1 \\ \vdots \\ 1 \end{array}\right).$$

Dual cubes revisited (II)

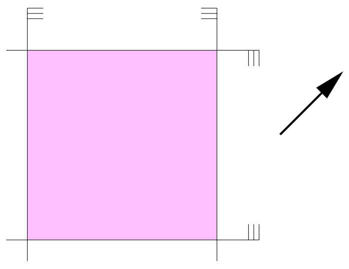
Dual Parameters:

$$b^T = \left(\begin{array}{cccc|cccc} 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \end{array} \right)$$

$$A^T = \left(\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 \end{array} \right), c^T = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Dual toy LP:

$$\begin{array}{ll} \text{min} & z_1 + \ldots + z_d \\ \text{subject to} & \begin{array}{l} y_i + z_i = 1 \\ y_i, z_i \geq 0 \end{array}, i = 1 \ldots d \end{array}$$



Equivalent to *cube*

$$\begin{array}{ll} \text{min} & z_1 + \ldots + z_d \\ \text{subject to} & \begin{array}{l} z_i \leq 1 \\ z_i \geq 0 \end{array}, i = 1 \ldots d \end{array}$$

Cubes

Theorem: The *Random-Facet* simplex algorithm solves any d-cube with an expected number of at most

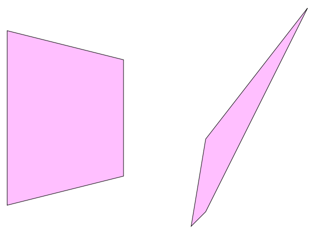
$$e^{2\sqrt{d}} - 1$$

pivot steps.

Follows from the bound for **lp-type** on dual cubes.

Remarks:

(i) Cubes may of course look more general!



(ii) Even for cubes, no better algorithm is known!

Exercise 2.1 Prove that the two formulations of the Klee-Minty cube LP on pages 27 and 28 are equivalent.

Exercise 2.2 Show that Game B indeed models the behavior of the simplex method with pivot rule **Random-Edge** on the standard form Klee-Minty cube LP of page 28.
Hint: Develop a formula for the z-row of the tableau, depending on the set of nonbasic variables.

Exercise 2.3 Which pivot rule forces the simplex method to take the worst-case number of pivot steps on the standard form Klee-Minty cube LP?

Exercise 2.4 Modify the Klee-Minty cube LP in such a way that *Dantzig's rule* leads to the worst-case behavior.

Exercise 2.5 Prove that every dual cube has exactly $\binom{\delta}{k}$ bases with k violating constraints, provided every set G with $w(G) > -\infty$ has a unique basis.

Exercise 2.6 Construct the dual of the Simplex LP on page 26. What is the dimension of its feasible region? Certify that primal and dual have the same optimal value (which one, depending on c?)

Exercise 2.7 Consider the following variant of Game A:

```
GAME $A'(n)$:
  $x := n$
  WHILE $x > 0$ DO
      choose a random number $i$ in $[0, n-1]$
      set $x$ to $\min(i, x)$
  END
```

(i) What is the expected number of rounds in this game?

(ii) Describe the pivot rule under which the simplex method applied to the simplex LP on page 26 corresponds to Game A', and give a scenario where this rule might make sense, although it apparantly leads to more pivot steps than **Random-Edge**.

Solutions to Exercises for Part I

Exercise 1.1 We let H be the set of polygon edges. We assume them to be oriented, so that every edge has an "inner" side with respect to the polygon.
For $G \subseteq H$ and a point p, the *quality* of p with respect to G is the smallest angle over all triangles formed by p and edges of G, with p on the inner side of the edge (see Figure 1). If p is on some line spanning an edge, the quality is zero, and the quality of p with respect to the empty set is defined to be ∞.

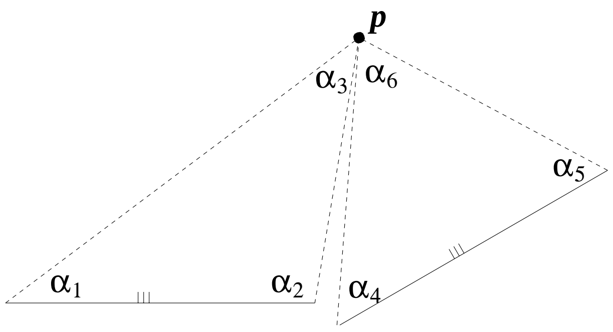


Figure 1: Quality of a point with respect to a set of edges

Now we define $w(G)$ to be the value α_G, where α_G is the largest quality of any point with respect to G. Let p_G be the point of quality α_G—by our nondegeneracy assumption, there is only one such point for every set G.
The ordering on the values α is simply the decreasing one, meaning that larger angles come first.
From this definition, it is clear that $w(H) = \alpha_H$ is the smallest angle in an optimal placement with respect to the whole polygon, and p_H is the unique point that defines this optimal placement.
To see that w indeed defines an LP-type problem, we have to check monotonicity and locality. Monotonicity is obvious by definition: if more edges are added, the largest possible quality can at most become smaller.
For locality, we have to characterize violation: what does it mean that $w(G \cup \{h\}) > w(G)$? We have $w(G \cup \{h\}) = w(G)$ if and only if the smallest angle in the new triangle formed by the inner side of h and p_G is no smaller than α_G. Because exactly in this case, p_G is still the angle-optimal point of quality α_G with respect to the larger set $G \cup \{h\}$.
This means, violation only depends on the relative position of the new "constraint" h with respect to the optimal point p_G. By nondegeneracy, if $\alpha_F = \alpha_G$, then also $p_F = p_G$, so $w(G \cup \{h\}) > w(G)$ holds if and only if $w(F \cup \{h\}) > w(F)$ holds, which is the locality property.

To prove that the combinatorial dimension is bounded by three, we apply Helly's Theorem.
Namely, the observation is that the region of points with quality at least some value α is the intersection of simple convex regions and is therefore convex itself. What are these simple regions? Consider an edge e. Then we know that any point p which achieves quality at least α has to satisfy three constraints: all three angles in the triangle formed by e and p must be at least α. By Thales' Theorem, the region of points p where the angle incident to p is at least α, is a spherical cap over e. If in addition, the two angles incident to e must be at least α, this cap gets intersected with two halfplanes, see Figure 2.

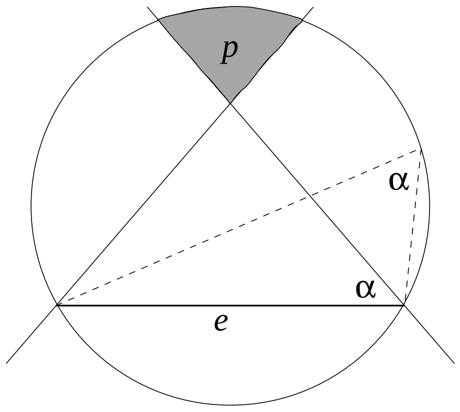


Figure 2: Region of quality at least α

The angle-optimal point p thus is the unique point in the intersection of $3n$ convex regions associated with the optimal angle α_{opt}.
Now we prove that there is a subset of at most three edges wich does not allow for a better solution than α_{opt}. For this, consider the $3n$ convex regions associated with the angle $\alpha_{opt} + \varepsilon$, for some very small $\varepsilon > 0$. Because α_{opt} was the optimal angle, the intersection of those $3n$ sets is empty. By Helly's Theorem, we then have a subset of at most three sets whose intersection is empty. The three sets come from at most three edges, and those edges do not allow for a solution with quality $\alpha_{opt} + \varepsilon$. In other words, the three edges contain some basis, which proves that no basis can have size more than three; equivalently, the combinatorial dimension of the problem is bounded by three.

Exercise 1.2 The *smallest enclosing annulus* problem can be formalized as follows. Let $p_i = (x_i, y_i), i = 1, \ldots, n$ be the n points. An annulus can be specified by its center $p = (x, y)$, the large radius R and the small radius r. The area is then proportional to $R^2 - r^2$. The smallest enclosing annulus is obtained as the solution of the following optimization problem.

$$\begin{array}{ll} \text{minimize} & R^2 - r^2 \\ \text{subject to} & r^2 \leq \|p_i - c\|^2 \leq R^2, \quad i = 1, \dots, n. \end{array}$$

This can be rewritten as

$$\begin{array}{lll} \text{minimize} & R^2 - r^2 & \\ \text{subject to} & r^2 - x^2 - y^2 + 2x_i x + 2y_i y \leq x_i^2 + y_i^2, & i = 1, \dots, n \\ & R^2 - x^2 - y^2 + 2x_i x + 2y_i y \geq x_i^2 + y_i^2, & i = 1, \dots, n \end{array}$$

Defining new variables $a := r^2 - x^2 - y^2$ and $b := R^2 - x^2 - y^2$, the problem becomes a linear program in four variables a, b, x, y and $2n$ constraints:

$$\begin{array}{lll} \text{minimize} & b - a & \\ \text{subject to} & a + 2x_i x + 2y_i y \leq x_i^2 + y_i^2, & i = 1, \dots, n \\ & b + 2x_i x + 2y_i y \geq x_i^2 + y_i^2, & i = 1, \dots, n \end{array}$$

It is clear that a, b, x, y form an optimal solution to this linear program if and only if r^2, R^2, x, y form an optimal solution to the original problem, where

$$r^2 = a + x^2 + y^2,$$
$$R^2 = b + x^2 + y^2.$$

To formulate the *largest disk in kernel* problem as a linear program, we do not want to give formulas but rather argue geometrically. What we are looking for is a point p which is on the inner side of every edge and maximizes the smallest distance to any edge.

Now consider "raising a roof" above the polygon by erecting n planes, each one going through an edge and tilting upwards at an angle of 45 degrees (see Figure 3).

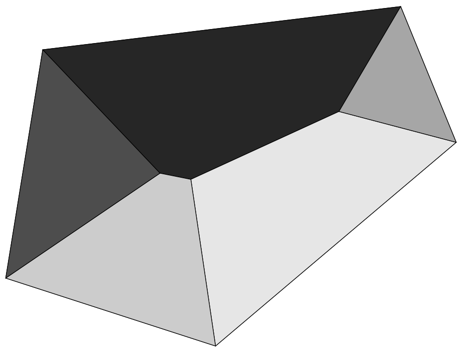


Figure 3: Roof above a polygon

The requirement that p is inside the polygon translates to the requirement that p is "under the roof" in the sense that it is below all the n planes. Because of the 45-degree angles, the vertical distance of any point to the roof (i.e. to the lowest plane above the point) equals the smallest distance to any edge. This means, the point p we are looking for is located under the highest point of the roof. To find this highest point is nothing else than a linear program in three variables x_1, x_2, x_3, the latter being the height coordinate: we need to maximize x_3 under the constraint that the point (x_1, x_2, x_3) lies under the roof, i.e. below all the n planes. This condition boils down to a system of n linear inequalities.

Exercise 1.3 This is simply by induction. Consider first the recurrence relation

$$\begin{aligned}
t(\delta, k) &= 0, \\
t(n, 0) &= n - \delta, \\
t(n, k) &\leq t(n-1, k) + 1 + \frac{k}{n-\delta} t(n, k-1), \quad n > \delta, k > 0.
\end{aligned}$$

The claim is that

$$t(n, k) \leq \sum_{j=0}^{k} \frac{1}{j!} k!(n-\delta),$$

which is obviously true for $n = \delta$ (where we get 0) and $k = 0$ (where we get $n - \delta$). For $n > \delta, k > 0$ we inductively get

$$\begin{aligned}
t(n, k) &\leq t(n-1, k) + 1 + \frac{k}{n-\delta} t(n, k-1) \\
&\leq \sum_{j=0}^{k} \frac{1}{j!} k!(n-1-\delta) + 1 + \frac{k}{n-\delta} \sum_{j=0}^{k-1} \frac{1}{j!} (k-1)!(n-\delta) \\
&= \sum_{j=0}^{k} \frac{1}{j!} k!(n-1-\delta) + 1 + \sum_{j=0}^{k-1} \frac{1}{j!} k! \\
&= \sum_{j=0}^{k} \frac{1}{j!} k!(n-1-\delta) + \sum_{j=0}^{k} \frac{1}{j!} k! \\
&= \sum_{j=0}^{k} \frac{1}{j!} k!(n-\delta).
\end{aligned}$$

The second bound of

$$t(n, k) \leq 2^k (n-\delta)$$

is derived from the recurrence relation

$$\begin{aligned}
t(\delta, k) &= 0, \\
t(n, 0) &= n - \delta, \\
t(n, k) &\leq t(n-1, k) + 1 + \frac{1}{n-\delta} \sum_{i=1}^{\min(k, n-\delta)} t(n, k-i), \quad n > \delta, k > 0.
\end{aligned}$$

As before, for $n = \delta$ or $k = 0$, the bound holds with equality. For $n > \delta, k > 0$ we inductively get

$$\begin{aligned}
t(n,k) &\leq t(n-1,k) + 1 + \frac{1}{n-\delta} \sum_{i=1}^{\min(k,n-\delta)} t(n,k-i) \\
&\leq 2^k(n-1-\delta) + 1 + \frac{1}{n-\delta} \sum_{i=1}^{\min(k,n-\delta)} 2^{k-i}(n-\delta) \\
&\leq 2^k(n-1-\delta) + 1 + \sum_{i=1}^{k} 2^{k-i} \\
&= 2^k(n-1-\delta) + 2^k \\
&= 2^k(n-\delta).
\end{aligned}$$

Exercise 1.4 This is also an easy induction, applying standard binomial coefficient identities.
For $k = 0$, the bound holds. For $k > 0$, we inductively get

$$\begin{aligned}
f(k) &= f(k-1) + \frac{1}{k} \sum_{i=1}^{k} f(k-i) \\
&= \sum_{i=0}^{k-1} \frac{1}{i!}\binom{k-1}{i} + \frac{1}{k} \sum_{i=1}^{k} \sum_{j=0}^{k-i} \frac{1}{j!}\binom{k-i}{j} \\
&= \sum_{i=0}^{k-1} \frac{1}{i!}\binom{k-1}{i} + \frac{1}{k} \sum_{j=0}^{k-1} \frac{1}{j!} \sum_{i=1}^{k-j} \binom{k-i}{j} \\
&= \sum_{i=0}^{k-1} \frac{1}{i!}\binom{k-1}{i} + \frac{1}{k} \sum_{j=0}^{k-1} \frac{1}{j!} \sum_{i=j}^{k-1} \binom{i}{j} \\
&= \sum_{i=0}^{k-1} \frac{1}{i!}\binom{k-1}{i} + \frac{1}{k} \sum_{j=0}^{k-1} \frac{1}{j!}\binom{k}{j+1}.
\end{aligned}$$

The latter equation uses the "summation on upper index" identity

$$\sum_{i=0}^{m} \binom{i}{j} = \binom{m+1}{j+1}.$$

Using the absorption identity

$$\frac{1}{k}\binom{k}{j+1} = \frac{1}{j+1}\binom{k-1}{j},$$

this further gives

$$\begin{aligned}
f(k) &= \sum_{i=0}^{k-1} \frac{1}{i!}\binom{k-1}{i} + \sum_{j=0}^{k-1} \frac{1}{(j+1)!}\binom{k-1}{j} \\
&= \sum_{i=0}^{k-1} \frac{1}{i!}\binom{k-1}{i} + \sum_{i=1}^{k} \frac{1}{i!}\binom{k-1}{i-1}.
\end{aligned}$$

We can extend the first sum to $i = k$ and the second sum to $i = 0$ without changing the value; this means, we get

$$\begin{aligned}
f(k) &= \sum_{i=0}^{k} \frac{1}{i!}\binom{k-1}{i} + \sum_{i=0}^{k} \frac{1}{i!}\binom{k-1}{i-1} \\
&= \sum_{i=0}^{k} \frac{1}{i!}\left(\binom{k-1}{i} + \binom{k-1}{i-1}\right) \\
&= \sum_{i=0}^{k} \frac{1}{i!}\binom{k}{i},
\end{aligned}$$

by the most basic binomial coefficient identity.

Exercise 1.5 Let's start with the number of basis computations. By construction of the toy LP, every basis computation replaces some constraint $h_i \equiv \{x_i \geq 0\}$ with its counterpart $\bar{h} \equiv \{x_i \geq 1\}$. Because we start with the basis $C = \{h_1, \ldots, h_d\}$ and end up with the basis $C_{opt} = \{\bar{h}_1, \ldots, \bar{h}_d\}$, the algorithm performs exactly d basis computations, regardless of the random choices.
Now let $t(d)$ be the expected number of violation tests performed with start basis $C = \{h_1, \ldots, h_d\}$. In the first recursive call, we have a subproblem which is isomorphic to a toy LP of dimension $d - 1$. The basis we get back is $C' = \{\bar{h}_1, \ldots, \bar{h}_{i-1}, h_i, \bar{h}_{i+1}, \ldots, \bar{h}_d\}$, where $\bar{h}_i$ was the constraint removed for the first recursive call. Then there is one violation test, followed by a basis computation which already gives the optimal basis $C'' = \{\bar{h}_1, \ldots, \bar{h}_d\}$. It is easy to see that the second recursive call then performs exactly d violation tests, one for every level of recursion.
Thus we get

$$\begin{aligned}
t(0) &= 0, \\
t(d) &= t(d-1) + 1 + d.
\end{aligned}$$

This solves to

$$t(d) = (d+1) + d + \ldots + 2 = \binom{d+2}{2} - 1.$$

Again, this number does not depend on the random choices performed by the algorithm.

Exercise 1.6 First of all, the algorithm still works. The only problem could be the return statement in case of $G = C$. But the invariant is that C is always a basis of some $F \subseteq G$. Thus, if $G = C$, C must be the basis of G, and it is correct to return C in this case.
To see that the linear runtime still holds, let us reconsider the recurrence relation for the expected runtime. Lemma 2 gives a bound for the expected number of violation tests in the basis-regular case. For the general case, the recurrence has to be changed. As it turns out, $t(\delta, k)$ is no longer zero in any case. But $t(\delta, k)$ can be no more than $\delta 2^k$: $\delta - k$ out of the δ elements are already fixed and will never leave the basis again (see the definition of hidden dimension). This means, at most 2^k bases will ever be considered in the call, each of which might be followed by up to d violation tests.
This gives the recurrence relation

$$\begin{aligned} t(\delta, k) &\leq \delta 2^k, \\ t(n, 0) &= n - \delta, \\ t(n, k) &\leq t(n-1, k) + 1 + \frac{1}{n-\delta} \sum_{i=1}^{\min(k, n-\delta)} t(n, k-i). \end{aligned}$$

We claim that

$$t(n, k) \leq 2^{2k} \delta (n - \delta + 1),$$

which is still linear in n. This is not the best possible bound, but an easily provable one.
For $n = \delta$ or $k = 0$ the bound holds, and for $n > \delta, k > 0$ we inductively get

$$\begin{aligned} t(n, k) &\leq 2^{2k} \delta (n - \delta) + 1 + \delta \frac{1}{n-\delta} \sum_{i=1}^{k} 2^{2(k-i)} (n - \delta + 1) \\ &\leq 2^{2k} \delta (n - \delta) + 1 + \delta \frac{n - \delta + 1}{n - \delta} (2^{2k-1} - 1) \\ &\leq 2^{2k} \delta (n - \delta) + \delta \frac{n - \delta + 1}{n - \delta} 2^{2k-1} \\ &\leq 2^{2k} \delta (n - \delta) + \delta 2^{2k} \\ &= 2^{2k} \delta (n - \delta + 1). \end{aligned}$$

Exercise 1.7 The combinatorial dimension becomes $d+1$ instead of d. Because by Helly's Theorem, if the problem on constraint set G is infeasible (meaning $w(G) = \infty$), there is a subset C of at most $d+1$ constraints which already defines an infeasible problem (meaning $w(C) = \infty$). Figure 4 ("inverted simplex") shows that less than $d + 1$ constraints won't do in general.
The problem is no longer basis-regular, because some sets G might define feasible subproblems (and have bases of size d), while others may define infeasible subproblems with bases of size $d + 1$.

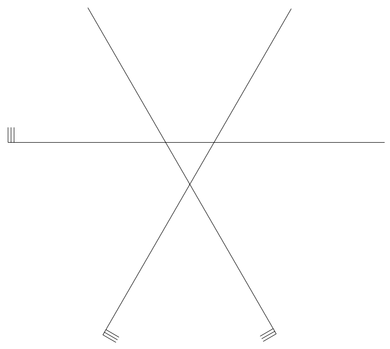


Figure 4: Infeasible LP on $d+1$ constraints

Exercise 1.8 The *smallest enclosing axis-parallel square* problem is actually a linear programming problem in disguise. With variables x, y, α (representing lower left corner and side length), it can be formulated as

$$\begin{array}{ll} \text{minimize} & \alpha \\ \text{subject to} & x_i \leq x \leq x_i + \alpha, \ldots i = 1, \ldots, n \\ & y_i \leq y \leq y_i + \alpha, \ldots i = 1, \ldots, n. \end{array}$$

This means, the problem can be made local just like LP, by slightly perturbing the objective function. Equivalently, one looks for the smallest square covering all the points that has the lexicographically smallest lower left corner.
The *closest pair* problem cannot be made into an LP-type problem with constant combinatorial dimension, so it is a truly non-local problem. The reason is that there is an $\Omega(n \log n)$ lower bound for that problem in the algebraic decision tree model. As an LP-type problem, it would have to be solvable in $O(n)$ time.

Solutions to Exercises for Part II

Exercise 2.1 To rewrite the LP on page 27 into that on page 28, we substitute

$$\begin{aligned} y_1 &:= x_1, \\ y_i &:= x_i - \varepsilon x_{i-1}, \quad i = 2, \ldots, d. \end{aligned}$$

Then the d constraints

$$\begin{aligned} 0 &\leq x_1, \\ \varepsilon x_{i-1} &\leq x_i \end{aligned}$$

become nonnegativity constraints

$$y_i \geq 0, \quad i = 1, \ldots, d.$$

To rewrite the other constraints, one easily proves by induction that

$$x_i = \sum_{j=1}^{i} \varepsilon^{i-j} y_j, \quad i = 1, \ldots, d, \tag{1}$$

which yields

$$x_i + \varepsilon x_{i-1} = y_i + 2 \sum_{j=1}^{i-1} \varepsilon^{i-j} y_j, \quad i = 2, \ldots, d.$$

In other words, the constraints

$$\begin{aligned} x_1 &\leq 1, \\ x_i &\leq 1 - \varepsilon x_{i-1}, \quad i = 2, \ldots, d \end{aligned}$$

become

$$y_i + 2 \sum_{j=1}^{i-1} \varepsilon^{i-j} y_j \leq 1, \quad i = 1, \ldots, d.$$

The objective function x_d translates to the desired form via (1), with the only difference that on page 28, we want to maximize, while on page 27, we started with a minimization problem.
Both versions are equivalent, though: substituting $x_d := 1 - z_d$, transforms the LP on page 27 into

$$\begin{array}{ll} \text{minimize } 1 - z_d & \\ \text{subject to} & \\ \quad 0 \leq x_1 \leq 1 & \\ \quad \varepsilon x_{i-1} \leq x_i \leq 1 - \varepsilon x_{i-1}, & i = 2, \ldots, d-1, \\ \quad \varepsilon x_{d-1} \leq z_d \leq 1 - \varepsilon x_{d-1}, & \end{array}$$

which has the same feasible region, but now the last coordinate z_d gets *maximized.*

Exercise 2.2 Starting with the LP on page 28, the simplex method introduces slack variables

$$s_i := 1 - x_i - 2\sum_{j=1}^{i-1} \varepsilon^{i-j} x_j, \quad i = 1, \ldots, d, \tag{2}$$

Then the z-row of the initial tableau (with the original variables being nonbasic) reads as

$$\begin{aligned} z &= \sum_{j=1}^{d} \varepsilon^{d-j} x_j \\ &= x_d + \varepsilon x_{d-1} + \cdots + \varepsilon^{d-1} x_1. \end{aligned}$$

The cube structure implies that for every j, exactly one of the variables x_j and s_j is nonbasic (we cannot have $x_j = s_j = 0$ for $\varepsilon < 1/2$). Let's analyze how the z-row changes when x_i is replaced with s_i for some i.
When we apply (2), we get

$$\begin{aligned} z &= x_d + \cdots + \varepsilon^{d-i-1} x_{i+1} + \varepsilon^{d-i} x_i + \varepsilon^{d-i+1} x_{i-1} + \cdots + \varepsilon^{d-1} x_1 \\ &= x_d + \cdots + \varepsilon^{d-i-1} x_{i+1} + \varepsilon^{d-i} \left(1 - s_i - 2\sum_{j=1}^{i-1} \varepsilon^{i-j} x_j\right) + \varepsilon^{d-i+1} x_{i-1} + \cdots \varepsilon^{d-1} x_1 \\ &= x_d + \cdots + \varepsilon^{d-i-1} x_{i+1} + \left(\varepsilon^{d-i} - \varepsilon^{d-i} s_i - 2\sum_{j=1}^{i-1} \varepsilon^{d-j} x_j\right) + \varepsilon^{d-i+1} x_{i-1} + \cdots \varepsilon^{d-1} x_1 \\ &= x_d + \cdots + \varepsilon^{d-i-1} x_{i+1} - \varepsilon^{d-i} s_i - \varepsilon^{d-i+1} x_{i-1} - \cdots - \varepsilon^{d-1} x_1 \quad + \varepsilon^{d-i}. \end{aligned}$$

This means, the coefficient of the j-th variable (x_j resp. s_j) keeps its absolute value but flips its sign for $j \leq i$.
The same happens when we replace another variable x_k by s_k, where $k < i$ (i.e. we replace a variable further to the right in the z-row as we have written it down in the previous equations): again, the coefficient of x_j resp. s_j flips its sign for $j \leq k$, but its absolute value stays the same.
The pattern we get when substituting x_j's by s_j's from left to right is therefore the following: the absolute value of the coefficient of the j-th variable is always ε^{d-j}, and its sign is determined by the number of s-variables to the left of it (including position j). If this number is even, the sign is positive, otherwise the sign is negative.
If we associate positive signs with the digit 1 and negative signs with the digit zero, we get a binary number representing the current z-row (in the beginning, it's the number 11...1). A pivot step choses a variable at some position j with positive coefficient and replaces it with its partner variable. By the above pattern, this flips the signs of all coefficients further to the right, including the coefficient at position j. In the binary number representation, this is exactly the behavior of game B, when the variable with positive coefficient is chosen among all the candidates at random.

Exercise 2.3 Bland's rule leads to the worst-case behavior, if the ordering of the variables is

$$x_1, s_1, x_2, s_2, \ldots, x_d, s_d.$$

Namely, in the z-row ordering of the previous solution, the rule then always chooses the rightmost variable with positive coefficient as the entering variable. In the binary number representation this means that in every step the rightmost one-entry gets flipped (together with all the zero-entries to the right of it). In other words, the number gets smaller by exactly one in every step. Starting with $11\ldots1$ thus generates the worst possible number of $2^d - 1$ pivot steps.

Exercise 2.4 If we had $\varepsilon > 1$, Dantzig's rule (choose the variable with largest positive coefficient) would behave exactly like Bland's rule with the variable ordering as before: namely, as we have seen, the coefficient of the j-th variable x_j resp. s_j is always of absolute value ε^{d-j}, which means that the variables are nicely ordered from left to right by incresing absolute coefficient in case of $\varepsilon > 1$.
Unfortunately, the problem is no longer a cube for such values of ε—for this we needed $\varepsilon < 1/2$. It can happen that there are feasible solutions with $x_i = s_i = 0$ for some i, in which case the pattern developed above is messed up.
To fix this, we manipulate the right-hand sides of the problem to make it a cube again. Changing the input to

$$\begin{array}{lll} \text{maximize} & x_d + \varepsilon x_{d-1} + \cdots & +\varepsilon^{d-1}x_1 \\ \text{subject to} & x_i + 2\varepsilon x_{i-1} + \cdots & +2\varepsilon^{i-1}x_1 \leq \varepsilon^{2i}, \\ & & i = 1, \ldots, d \\ & x_1, x_2, \ldots, x_d \geq 0 & \end{array}$$

will do it. What we need is that for any i, we can independently choose one of the variables x_i or s_i as nonbasic (meaning it has value zero), and the other one will automatically be basic. For this we compute

$$\begin{aligned} s_i + x_i &= \varepsilon^{2i} - 2\sum_{j=1}^{i-1} \varepsilon^{i-j} x_j \\ &\geq \varepsilon^{2i} - 2\sum_{j=1}^{i-1} \varepsilon^{i-j}\varepsilon^{2j} \\ &= \varepsilon^{2i} - 2\sum_{j=1}^{i-1} \varepsilon^{i+j} \\ &> \varepsilon^{2i} - 2\sum_{k=0}^{2i-1} \varepsilon^{k} \\ &= \varepsilon^{2i} - 2\frac{\varepsilon^{2i}-1}{\varepsilon-1} \end{aligned}$$

$$> \quad 0,$$

if $\varepsilon \geq 3$. This means $x_i = 0$ (nonbasic) implies $s_i > 0$ (basic) and vice versa.

Exercise 2.5 This is a nice application of *double counting.* For every j, we count the number of pairs (G, B), where $|G| = \delta + j$, $w(G) > -\infty$ and B is a basis of G. On the one hand, there are f_j such pairs, where f_j is the number of possible G's, i.e. subsets of H with $\delta + j$ elements that contain at least one element of $h, \bar{h}$ for every $h \in H$. This follows from the assumption that every set G has a unique basis B.
Let's consider a fixed basis B. How many sets G, $|G| = \delta + j$ are there such that B is a basis of G? By the LP-type axioms, B is a basis of G if and only if B has no violating constraints in G. Assume B has k violating constraints in total. Then G has B as a basis if and only if its j extra elements (the ones that don't belong to B) are chosen from the $\delta - k$ constraints not violating B. There are

$$\binom{\delta - k}{j}$$

possibilities to do this. If h_k denotes the number of bases with k violating constraints, then the overall number of pairs (G, B) for a fixed j is

$$f_j = \sum_k h_k \binom{\delta - k}{j} = \sum_k h_{\delta - k} \binom{k}{j}. \tag{3}$$

Remember that it's the h_k we are trying to compute. As a first step, we can evaluate the f_j: to choose a G that is counted for f_j, we must specify j pairs $h, \bar{h}$ representing all elements in G whose "partner" is also in G. There are

$$\binom{\delta}{j}$$

possibilities to do this. For every such choice, we have exactly

$$2^{\delta - j}$$

ways to pick the remaining $\delta - j$ elements without partner, one from every remaining pair of elements. This means, we get

$$f_j = \binom{\delta}{j} 2^{\delta - j}.$$

Now that we know the f_j, we can consider (3) as a system of $\delta + 1$ linear equations (one for every $0 \leq j \leq \delta$) in $\delta + 1$ unknowns $h_0, \ldots, h_\delta$, and we might hope for a unique solution.

Indeed, there is such a solution. To get it without grinding, we apply the following generating function trick. Define

$$\begin{aligned} F(x) &:= \sum_j f_j x^j, \\ H(x) &:= \sum_k h_{\delta-k} x^k. \end{aligned} \tag{4}$$

We are going to develop a closed form for $H(x)$ from which we can read off the coefficients $h_{\delta-k}$. By (3) we can write

$$\begin{aligned} F(x) &= \sum_j f_j x^j \\ &= \sum_j \sum_k h_{\delta-k} \binom{k}{j} x^j \\ &= \sum_k h_{\delta-k} \sum_j \binom{k}{j} x^j \\ &= \sum_k h_{\delta-k} (x+1)^k \\ &= H(x+1). \end{aligned}$$

Furthermore, we get

$$\begin{aligned} F(x) &= \sum_j \binom{\delta}{j} 2^{\delta-j} x^j \\ &= (x+2)^\delta, \end{aligned}$$

which gives

$$\begin{aligned} H(x) &= F(x-1) \\ &= (x+1)^\delta \\ &= \sum_k \binom{\delta}{k} x^k. \end{aligned}$$

By comparing coefficients with (4), we obtain

$$h_{\delta-k} = \binom{\delta}{k} = \binom{\delta}{\delta-k} = h_k,$$

which finishes the proof.

The proof exhibits a more general pattern, known as the *inversion formula.* If we have sequences (a_j) and (b_k), related by the formula

$$a_j = \sum_k \binom{k}{j} b_k,$$

then we can express the b_k by the a_j via the formula

$$b_k = \sum_j (-1)^{j-k} \binom{j}{k} a_j.$$

To prove this, we define as above

$$\begin{aligned} A(x) &:= \sum_j a_j x^j, \\ B(x) &:= \sum_k b_k x^k, \end{aligned}$$

and observe that $A(x) = B(x+1)$ holds, which in turn yields

$$\begin{aligned} B(x) &= A(x-1) \\ &= \sum_j a_j (x-1)^j, \\ &= \sum_j a_j \sum_k \binom{j}{k} (-1)^{j-k} x^k \\ &= \sum_k \left(\sum_j (-1)^{j-k} \binom{j}{k} a_j \right) x^k, \end{aligned}$$

from which the formula for the b_k follows by comparing coefficients.

Exercise 2.6 To construct the dual, we write the Simplex LP in the form

$$\begin{array}{lll} \min & -c^T x & (d \text{ variables}) \\ \text{s.t.} & Ax \geq b & (d+1 \text{ constraints}), \end{array}$$

where

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \\ -1 & \cdots & -1 & -1 \end{pmatrix}, \quad b = (0, \ldots, 0, -1)^T.$$

Then the dual problem has $d+1$ variables $y_1, \ldots, y_d$, and reads as

$$\begin{array}{ll} \max & -y_{d+1} \\ \text{s.t.} & y_i - y_{d+1} = -c_i, \quad i = 1, \ldots, d, \\ & y \geq 0. \end{array}$$

The feasible region consists of the line

$$(y_1, \ldots, y_{d+1})^T = (-c_1, \ldots, -c_d, 0)^T + \lambda(1, \ldots, 1)^T,$$

intersected with the positive orthant $\{y \geq 0\}$. It is therefore of dimension 1.
The optimal solution of the primal problem is $\max_i c_i$, achieved by setting $x_i = 1$ and $x_j = 0, j \neq i$. No larger value is achievable, because any objective function value attained over the feasible region is a convex combination of the c_i and therefore lies between $\min_i c_i$ and $\max_i c_i$.
The dual tries to minimize y_{d+1}, but the nonnegativity constraints imply that $y_{d+1} \geq c_i$ for all i. The smallest possible value of y_{d+1} (the optimal solution of the dual) is therefore $y_{d+1} = \max_i c_i$.

Exercise 2.7
(i) In every round, a random number between 0 and $n-1$ is drawn, and the game ends exactly at the first time the number 0 is drawn. We can model this as a sequence of independent Bernoulli trials with probability of success equal to $p = 1/n$. It is well known that the expected number of trials needed until the first success is $1/p = n$.
So, the expected number of rounds in game $A'(n)$ is exactly n.

(ii) The pivot rule is the following: choose a random nonbasic variable; if its coefficient in the z-row is positive, let it enter the basis, otherwise ignore the choice and choose again. Let us call this rule **Random-Variable**.
A scenario in which this rule makes sense arises when it is expensive to compute coefficients in the z-row. For example, the LP might be given implicitly, and the coefficients are computed only on demand. In fact, this is exactly the case in the *revised* simplex method where the tableau is not explicitly kept. Counting the complexity in terms of coefficient evaluations ("violation tests" in the abstract LP-type settings), **Random-variable** wins over **Random-Edge**. The former performs one violation test per round, leading to an expected total of n violation tests. **Random-Edge** on the other hand performs n violation tests per round, leading to an expected total of $nH_n = \Theta(n \log n)$ violation tests. This means, if the violation test is for some resaon much more expensive than the pivot step itself (basis computation), **Random-variable** will outperform **Random-Edge**.