

## Part II

# Linear Programming

## Brewery Problem

$\bar{U}$  Brewery brews ale and beer.

- ▶ Production limited by supply of corn, hops and barley malt
- ▶ Recipes for ale and beer require different amounts of resources

	<i>Corn (kg)</i>	<i>Hops (kg)</i>	<i>Malt (kg)</i>	<i>Profit (€)</i>
ale (barrel)	5	4	35	13
beer (barrel)	15	4	20	23
supply	480	160	1190	

## Brewery Problem

	<i>Corn (kg)</i>	<i>Hops (kg)</i>	<i>Malt (kg)</i>	<i>Profit (€)</i>
ale (barrel)	5	4	35	13
beer (barrel)	15	4	20	23
supply	480	160	1190	

How can brewer maximize profits?

- ▶ only brew ale: 34 barrels of ale  $\Rightarrow$  442 €
- ▶ only brew beer: 32 barrels of beer  $\Rightarrow$  736 €
- ▶ 7.5 barrels ale, 29.5 barrels beer  $\Rightarrow$  776 €
- ▶ 12 barrels ale, 28 barrels beer  $\Rightarrow$  800 €

## Brewery Problem

Linear Program

- ▶ Introduce **variables**  $a$  and  $b$  that define how much ale and beer to produce.
- ▶ Choose the variables in such a way that the **objective function** (profit) is maximized.
- ▶ Make sure that no **constraints** (due to limited supply) are violated.

$$\begin{aligned}
 \max \quad & 13a + 23b \\
 \text{s.t.} \quad & 5a + 15b \leq 480 \\
 & 4a + 4b \leq 160 \\
 & 35a + 20b \leq 1190 \\
 & a, b \geq 0
 \end{aligned}$$

## Standard Form LPs

### LP in standard form:

- ▶ input: numbers  $a_{ij}, c_j, b_i$
- ▶ output: numbers  $x_j$
- ▶  $n = \#$ decision variables,  $m = \#$ constraints
- ▶ maximize linear objective function subject to linear inequalities

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^n a_{ij} x_j = b_i \quad 1 \leq i \leq m \\ & x_j \geq 0 \quad 1 \leq j \leq n \end{aligned}$$

$$\begin{aligned} \max \quad & c^t x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

## Standard Form LPs

### Original LP

$$\begin{aligned} \max \quad & 13a + 23b \\ \text{s.t.} \quad & 5a + 15b \leq 480 \\ & 4a + 4b \leq 160 \\ & 35a + 20b \leq 1190 \\ & a, b \geq 0 \end{aligned}$$

### Standard Form

Add a **slack variable** to every constraint.

$$\begin{aligned} \max \quad & 13a + 23b \\ \text{s.t.} \quad & 5a + 15b + s_c = 480 \\ & 4a + 4b + s_h = 160 \\ & 35a + 20b + s_m = 1190 \\ & a, b, s_c, s_h, s_m \geq 0 \end{aligned}$$

## Standard Form LPs

There are different standard forms:

#### standard form

$$\begin{aligned} \max \quad & c^t x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & c^t x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

#### standard maximization form

$$\begin{aligned} \max \quad & c^t x \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

#### standard minimization form

$$\begin{aligned} \min \quad & c^t x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

## Standard Form LPs

It is easy to transform variants of LPs into (any) standard form:

- ▶ **less or equal to equality:**

$$a - 3b + 5c \leq 12 \Rightarrow a - 3b + 5c + s = 12 \\ s \geq 0$$

- ▶ **greater or equal to equality:**

$$a - 3b + 5c \geq 12 \Rightarrow a - 3b + 5c - s = 12 \\ s \geq 0$$

- ▶ **min to max:**

$$\min a - 3b + 5c \Rightarrow \max -a + 3b - 5c$$

## Standard Form LPs

It is easy to transform variants of LPs into (any) standard form:

► **equality to less or equal:**

$$a - 3b + 5c = 12 \Rightarrow \begin{aligned} a - 3b + 5c &\leq 12 \\ -a + 3b - 5c &\leq -12 \end{aligned}$$

► **equality to greater or equal:**

$$a - 3b + 5c = 12 \Rightarrow \begin{aligned} a - 3b + 5c &\geq 12 \\ -a + 3b - 5c &\geq -12 \end{aligned}$$

► **unrestricted to nonnegative:**

$$x \text{ unrestricted} \Rightarrow x = x^+ - x^-, x^+ \geq 0, x^- \geq 0$$

## Standard Form LPs

**Observations:**

- a linear program does not contain  $x^2$ ,  $\cos(x)$ , etc.
- transformations between standard forms can be done efficiently and only change the size of the LP by a small constant factor
- for the standard minimization or maximization LPs we could include the nonnegativity constraints into the set of ordinary constraints; this is of course not possible for the standard form

## Fundamental Questions

**Definition 1 (Linear Programming Problem (LP))**

Let  $A \in \mathbb{Q}^{m \times n}$ ,  $b \in \mathbb{Q}^m$ ,  $c \in \mathbb{Q}^n$ ,  $\alpha \in \mathbb{Q}$ . Does there exist  $x \in \mathbb{Q}^n$  s.t.  $Ax = b$ ,  $x \geq 0$ ,  $c^t x \geq \alpha$ ?

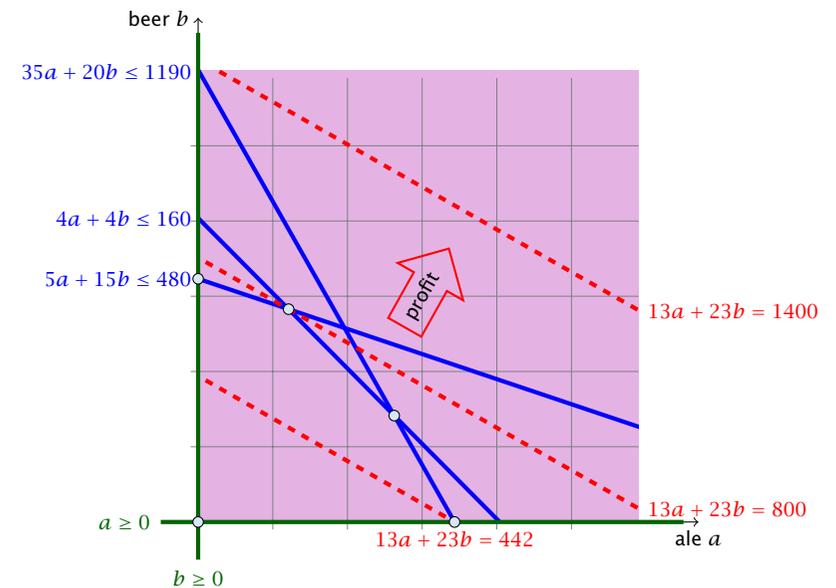
**Questions:**

- Is LP in NP?
- Is LP in co-NP?
- Is LP in P?

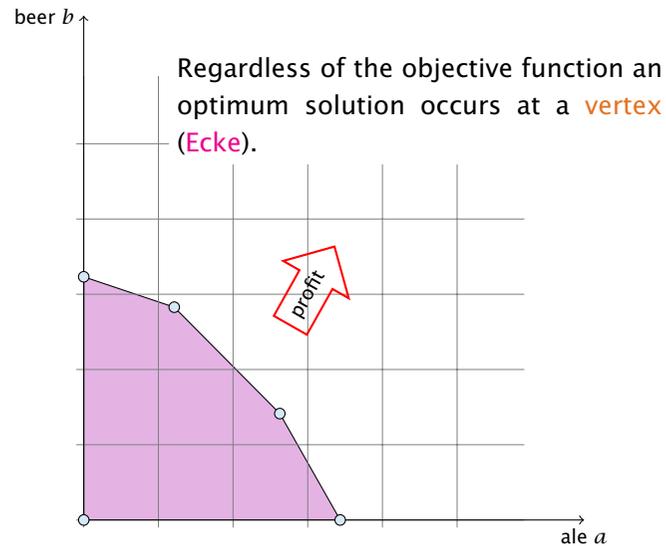
**Input size:**

- $n$  number of variables,  $m$  constraints,  $L$  number of bits to encode the input

## Geometry of Linear Programming



## Geometry of Linear Programming



## Convex Sets

A set  $S \subseteq \mathbb{R}^n$  is **convex** if for all  $x, y \in S$  also  $\lambda x + (1 - \lambda)y \in S$  for all  $0 \leq \lambda \leq 1$ .

A point in  $x \in S$  that can't be written as a convex combination of two other points in the set is called a **vertex**.

## Definitions

Let for a Linear Program in standard form

$$P = \{x \mid Ax = b, x \geq 0\}.$$

- ▶  $P$  is called the **feasible region** (Lösungsraum) of the LP.
- ▶ A point  $x \in P$  is called a **feasible point** (gültige Lösung).
- ▶ If  $P \neq \emptyset$  then the LP is called **feasible** (erfüllbar). Otherwise, it is called **infeasible** (unerfüllbar).
- ▶ An LP is **bounded** (beschränkt) if it is feasible and
  - ▶  $c^t x < \infty$  for all  $x \in P$  (for maximization problems)
  - ▶  $c^t x > -\infty$  for all  $x \in P$  (for minimization problems)

## Definitions

### Definition 2

A **polytop** is a set  $P \subseteq \mathbb{R}^n$  that is the convex hull of a **finite** set of points, i.e.,  $P = \text{conv}(X)$  where

$$\text{conv}(X) = \left\{ \sum_{i=1}^{\ell} \lambda_i x_i \mid \ell \in \mathbb{N}, x_1, \dots, x_{\ell} \in X, \lambda_i \geq 0, \sum_i \lambda_i = 1 \right\}$$

and  $|X| = c$ .

## Definitions

### Definition 3

A **polyhedron** is a set  $P \subseteq \mathbb{R}^n$  that can be represented as the intersection of finitely many half-spaces

$\{H(a_1, b_1), \dots, H(a_m, b_m)\}$ , where

$$H(a_i, b_i) = \{x \in \mathbb{R}^n \mid a_i x \leq b_i\} .$$

## Definitions

### Theorem 4

$P$  is a bounded polyhedron *iff*  $P$  is a polytop.

### Definition 5

Let  $P \subseteq \mathbb{R}^n$ ,  $a \in \mathbb{R}^n$  and  $b \in \mathbb{R}$ . The hyperplane

$$H(a, b) = \{x \in \mathbb{R}^n \mid ax = b\}$$

is a **supporting hyperplane** of  $P$  if  $\max\{ax \mid x \in P\} = b$ .

### Definition 6

Let  $P \subseteq \mathbb{R}^n$ .  $F$  is a **face** of  $P$  if  $F = P$  or  $F = P \cap H$  for some supporting hyperplane  $H$ .

### Definition 7

Let  $P \subseteq \mathbb{R}^n$ .

- ▶  $v$  is a **vertex** of  $P$  if  $\{v\}$  is a face of  $P$ .
- ▶  $e$  is an **edge** of  $P$  if  $e$  is a face and  $\dim(e) = 1$ .
- ▶  $F$  is a **facet** of  $P$  if  $F$  is a face and  $\dim(e) = \dim(P) - 1$ .

### Observation

The feasible region of an LP is a Polyhedron.

## Convex Sets

### Theorem 8

If there exists an optimal solution to an LP then there exists an optimum solution that is a vertex.

#### Proof

- ▶ suppose  $x$  is optimal solution that is not a vertex
- ▶ there exists direction  $d \neq 0$  such that  $x \pm d \in P$
- ▶  $Ad = 0$  because  $A(x \pm d) = b$
- ▶ Wlog. assume  $c^t d \geq 0$  (by taking either  $d$  or  $-d$ )
- ▶ Consider  $x + \lambda d, \lambda > 0$

## Convex Sets

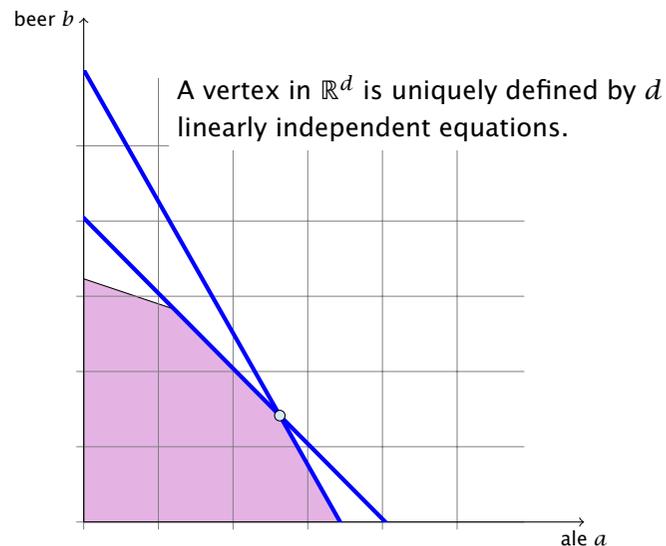
### Case 1. [ $\exists j$ s.t. $d_j < 0$ ]

- ▶ increase  $\lambda$  to  $\lambda'$  until first component of  $x + \lambda d$  hits 0
- ▶  $x + \lambda' d$  is feasible. Since  $A(x + \lambda' d) = b$  and  $x + \lambda' d \geq 0$
- ▶  $x + \lambda' d$  has one more zero-component ( $d_k = 0$  for  $x_k = 0$  as  $x \pm d \in P$ )
- ▶  $c^t x' = c^t(x + \lambda' d) = c^t x + \lambda' c^t d \geq c^t x$

### Case 2. [ $d_j \geq 0$ for all $j$ and $c^t d > 0$ ]

- ▶  $x + \lambda d$  is feasible for all  $\lambda \geq 0$  since  $A(x + \lambda d) = b$  and  $x + \lambda d \geq x \geq 0$
- ▶ as  $\lambda \rightarrow \infty, c^t(x + \lambda d) \rightarrow \infty$  as  $c^t d > 0$

## Algebraic View



### Notation

Suppose  $B \subseteq \{1 \dots n\}$  is a set of column-indices. Define  $A_B$  as the subset of columns of  $A$  indexed by  $B$ .

### Theorem 9

Let  $P = \{x \mid Ax = b, x \geq 0\}$ . For  $x \in P$ , define  $B = \{j \mid x_j > 0\}$ . Then  $x$  is a vertex **iff**  $A_B$  has linearly independent columns.

### Theorem 9

Let  $P = \{x \mid Ax = b, x \geq 0\}$ . For  $x \in P$ , define  $B = \{j \mid x_j > 0\}$ . Then  $x$  is a vertex **iff**  $A_B$  has linearly independent columns.

#### Proof ( $\Leftarrow$ )

- ▶ assume  $x$  is not a vertex
- ▶ there exists direction  $d$  s.t.  $x \pm d \in P$
- ▶  $Ad = 0$  because  $A(x \pm d) = b$
- ▶ define  $B' = \{j \mid d_j \neq 0\}$
- ▶  $A_{B'}$  has linearly dependent columns as  $Ad = 0$
- ▶  $d_j = 0$  for all  $j$  with  $x_j = 0$  as  $x \pm d \geq 0$
- ▶ Hence,  $B' \subseteq B$ ,  $A_{B'}$  is sub-matrix of  $A_B$

### Theorem 9

Let  $P = \{x \mid Ax = b, x \geq 0\}$ . For  $x \in P$ , define  $B = \{j \mid x_j > 0\}$ . Then  $x$  is a vertex **iff**  $A_B$  has linearly independent columns.

#### Proof ( $\Rightarrow$ )

- ▶ assume  $A_B$  has linearly dependent columns
- ▶ there exists  $d \neq 0$  such that  $A_B d = 0$
- ▶ extend  $d$  to  $\mathbb{R}^n$  by adding 0-components
- ▶ now,  $Ad = 0$  and  $d_j = 0$  whenever  $x_j = 0$
- ▶ for sufficiently small  $\lambda$  we have  $x \pm \lambda d \in P$
- ▶ hence,  $x$  is not a vertex

### Observation

For an LP we can assume wlog. that the matrix  $A$  has full row-rank. This means  $\text{rank}(A) = m$ .

- ▶ assume that  $\text{rank}(A) < m$
- ▶ assume wlog. that the first row  $A_1$  lies in the span of the other rows  $A_2, \dots, A_m$ ; this means

$$A_1 = \sum_{i=2}^m \lambda_i \cdot A_i, \text{ for suitable } \lambda_i$$

- C1** if now  $b_1 = \sum_{i=2}^m \lambda_i \cdot b_i$  then for all  $x$  with  $A_i x = b_i$  we also have  $A_1 x = b_1$ ; hence the first constraint is superfluous
- C2** if  $b_1 \neq \sum_{i=2}^m \lambda_i \cdot b_i$  then the LP is infeasible, since for all  $x$  that fulfill constraints  $A_2, \dots, A_m$  we have

$$A_1 x = \sum_{i=2}^m \lambda_i \cdot A_i x = \sum_{i=2}^m \lambda_i \cdot b_i \neq b_1$$

**From now on we will always assume that the constraint matrix of a standard form LP has full row rank.**

### Theorem 10

Given  $P = \{x \mid Ax = b, x \geq 0\}$ .  $x$  is a vertex iff there exists  $B \subseteq \{1, \dots, n\}$  with  $|B| = m$  and

- ▶  $A_B$  is non-singular
- ▶  $x_B = A_B^{-1}b \geq 0$
- ▶  $x_N = 0$

where  $N = \{1, \dots, n\} \setminus B$ .

### Proof

Take  $B = \{j \mid x_j > 0\}$  and augment with linearly independent columns until  $|B| = m$ ; always possible since  $\text{rank}(A) = m$ .

## Basic Feasible Solutions

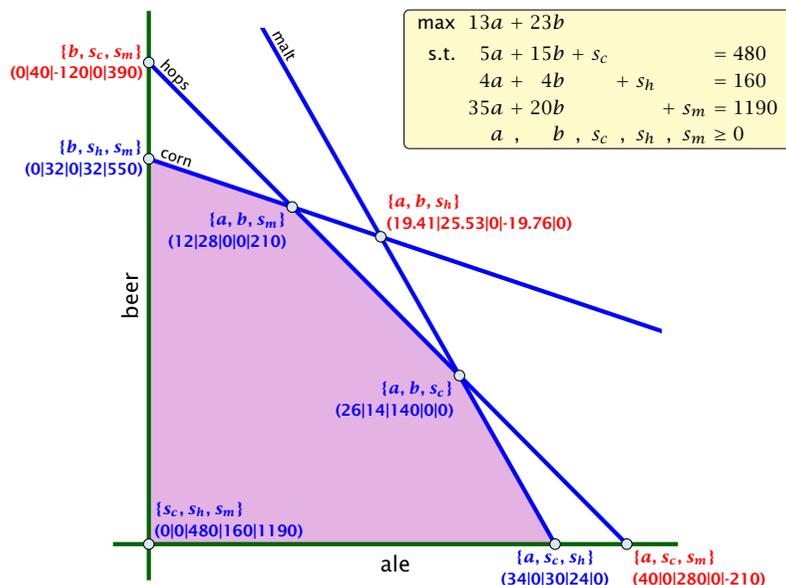
$x \in \mathbb{R}^n$  is called **basic solution (Basislösung)** if  $Ax = b$  and  $\text{rank}(A_J) = |J|$  where  $J = \{j \mid x_j \neq 0\}$ ;

$x$  is a **basic feasible solution (gültige Basislösung)** if in addition  $x \geq 0$ .

A **basis (Basis)** is an index set  $B \subseteq \{1, \dots, n\}$  with  $\text{rank}(A_B) = m$  and  $|B| = m$ .

$x \in \mathbb{R}^n$  with  $A_B x = b$  and  $x_j = 0$  for all  $j \notin B$  is the **basic solution associated to basis B (die zu B assoziierte Basislösung)**

## Algebraic View



## Fundamental Questions

### Linear Programming Problem (LP)

Let  $A \in \mathbb{Q}^{m \times n}$ ,  $b \in \mathbb{Q}^m$ ,  $c \in \mathbb{Q}^n$ ,  $\alpha \in \mathbb{Q}$ . Does there exist  $x \in \mathbb{Q}^n$  s.t.  $Ax = b$ ,  $x \geq 0$ ,  $c^t x \geq \alpha$ ?

### Questions:

- ▶ Is LP in NP? yes!
- ▶ Is LP in co-NP?
- ▶ Is LP in P?

### Proof:

- ▶ Given a basis  $B$  we can compute the associated basis solution by calculating  $A_B^{-1}b$  in polynomial time; then we can also compute the profit.

### Observation

We can compute an optimal solution to a linear program in time  $\mathcal{O}\left(\binom{n}{m} \cdot \text{poly}(n, m)\right)$ .

- ▶ there are only  $\binom{n}{m}$  different bases.
- ▶ compute the profit of each of them and take the maximum

## 4 Simplex Algorithm

Enumerating all basic feasible solutions (BFS), in order to find the optimum is slow.

**Simplex Algorithm** [George Dantzig 1947]

Move from BFS to **adjacent** BFS, without decreasing objective function.

Two BFSs are called **adjacent** if the bases just differ in one variable.

## 4 Simplex Algorithm

$$\begin{array}{ll} \max & 13a + 23b \\ \text{s.t.} & 5a + 15b + s_c = 480 \\ & 4a + 4b + s_h = 160 \\ & 35a + 20b + s_m = 1190 \\ & a, b, s_c, s_h, s_m \geq 0 \end{array}$$

$$\begin{array}{ll} \max Z & \\ 13a + 23b & - Z = 0 \\ 5a + 15b + s_c & = 480 \\ 4a + 4b + s_h & = 160 \\ 35a + 20b + s_m & = 1190 \\ a, b, s_c, s_h, s_m & \geq 0 \end{array}$$

$$\begin{array}{l} \text{basis} = \{s_c, s_h, s_m\} \\ A = B = 0 \\ Z = 0 \\ s_c = 480 \\ s_h = 160 \\ s_m = 1190 \end{array}$$

## Pivoting Step

$$\begin{array}{ll} \max Z & \\ 13a + 23b & - Z = 0 \\ 5a + 15b + s_c & = 480 \\ 4a + 4b + s_h & = 160 \\ 35a + 20b + s_m & = 1190 \\ a, b, s_c, s_h, s_m & \geq 0 \end{array}$$

$$\begin{array}{l} \text{basis} = \{s_c, s_h, s_m\} \\ a = b = 0 \\ Z = 0 \\ s_c = 480 \\ s_h = 160 \\ s_m = 1190 \end{array}$$

- ▶ choose variable to bring into the basis
- ▶ chosen variable should have positive coefficient in objective function
- ▶ apply **min-ratio** test to find out by how much the variable can be increased
- ▶ pivot on row found by min-ratio test
- ▶ the existing basis variable in this row leaves the basis

$$\begin{array}{rcl}
 \max Z & & \\
 13a + 23b & - Z = & 0 \\
 5a + 15b + s_c & = & 480 \\
 4a + 4b + s_h & = & 160 \\
 35a + 20b + s_m & = & 1190 \\
 a, b, s_c, s_h, s_m & \geq & 0
 \end{array}$$

$$\begin{array}{l}
 \text{basis} = \{s_c, s_h, s_m\} \\
 a = b = 0 \\
 Z = 0 \\
 s_c = 480 \\
 s_h = 160 \\
 s_m = 1190
 \end{array}$$

- ▶ Choose variable with coefficient  $\geq 0$  as **entering variable**.
- ▶ If we keep  $a = 0$  and increase  $b$  from 0 to  $\theta > 0$  s.t. all constraints ( $Ax = b, x \geq 0$ ) are still fulfilled the objective value  $Z$  will strictly increase.
- ▶ For maintaining  $Ax = b$  we need e.g. to set  $s_c = 480 - 15\theta$ .
- ▶ Choosing  $\theta = \min\{480/15, 160/4, 1190/20\}$  ensures that in the new solution one current basic variable becomes 0, and no variable goes negative.
- ▶ The basic variable in the row that gives  $\min\{480/15, 160/4, 1190/20\}$  becomes the **leaving variable**.

$$\begin{array}{rcl}
 \max Z & & \\
 13a + 23b & - Z = & 0 \\
 5a + 15b + s_c & = & 480 \\
 4a + 4b + s_h & = & 160 \\
 35a + 20b + s_m & = & 1190 \\
 a, b, s_c, s_h, s_m & \geq & 0
 \end{array}$$

$$\begin{array}{l}
 \text{basis} = \{s_c, s_h, s_m\} \\
 a = b = 0 \\
 Z = 0 \\
 s_c = 480 \\
 s_h = 160 \\
 s_m = 1190
 \end{array}$$

Substitute  $b = \frac{1}{15}(480 - 5a - s_c)$ .

$$\begin{array}{rcl}
 \max Z & & \\
 \frac{16}{3}a - \frac{23}{15}s_c & - Z = & -736 \\
 \frac{1}{3}a + b + \frac{1}{15}s_c & = & 32 \\
 \frac{8}{3}a - \frac{4}{15}s_c + s_h & = & 32 \\
 \frac{85}{3}a - \frac{4}{3}s_c + s_m & = & 550 \\
 a, b, s_c, s_h, s_m & \geq & 0
 \end{array}$$

$$\begin{array}{l}
 \text{basis} = \{b, s_h, s_m\} \\
 a = s_c = 0 \\
 Z = 736 \\
 b = 32 \\
 s_h = 32 \\
 s_m = 550
 \end{array}$$

$$\begin{array}{rcl}
 \max Z & & \\
 \frac{16}{3}a - \frac{23}{15}s_c & - Z = & -736 \\
 \frac{1}{3}a + b + \frac{1}{15}s_c & = & 32 \\
 \frac{8}{3}a - \frac{4}{15}s_c + s_h & = & 32 \\
 \frac{85}{3}a - \frac{4}{3}s_c + s_m & = & 550 \\
 a, b, s_c, s_h, s_m & \geq & 0
 \end{array}$$

$$\begin{array}{l}
 \text{basis} = \{b, s_h, s_m\} \\
 a = s_c = 0 \\
 Z = 736 \\
 b = 32 \\
 s_h = 32 \\
 s_m = 550
 \end{array}$$

Choose variable  $a$  to bring into basis.

Computing  $\min\{3 \cdot 32, 3 \cdot 32/8, 3 \cdot 550/85\}$  means pivot on line 2.

Substitute  $a = \frac{3}{8}(32 + \frac{4}{15}s_c - s_h)$ .

$$\begin{array}{rcl}
 \max Z & & \\
 -s_c - 2s_h & - Z = & -800 \\
 b + \frac{1}{10}s_c - \frac{1}{8}s_h & = & 28 \\
 a - \frac{1}{10}s_c + \frac{3}{8}s_h & = & 12 \\
 \frac{3}{2}s_c - \frac{85}{8}s_h + s_m & = & 210 \\
 a, b, s_c, s_h, s_m & \geq & 0
 \end{array}$$

$$\begin{array}{l}
 \text{basis} = \{a, b, s_m\} \\
 s_c = s_h = 0 \\
 Z = 800 \\
 b = 28 \\
 a = 12 \\
 s_m = 210
 \end{array}$$

## 4 Simplex Algorithm

Pivoting stops when all coefficients in the objective function are non-positive.

**Solution is optimal:**

- ▶ any feasible solution satisfies all equations in the tableaux
- ▶ in particular:  $Z = 800 - s_c - 2s_h, s_c \geq 0, s_h \geq 0$
- ▶ hence optimum solution value is at most 800
- ▶ the current solution has value 800

## Matrix View

Let our linear program be

$$\begin{aligned} c_B^t x_B + c_N^t x_N &= Z \\ A_B x_B + A_N x_N &= b \\ x_B, x_N &\geq 0 \end{aligned}$$

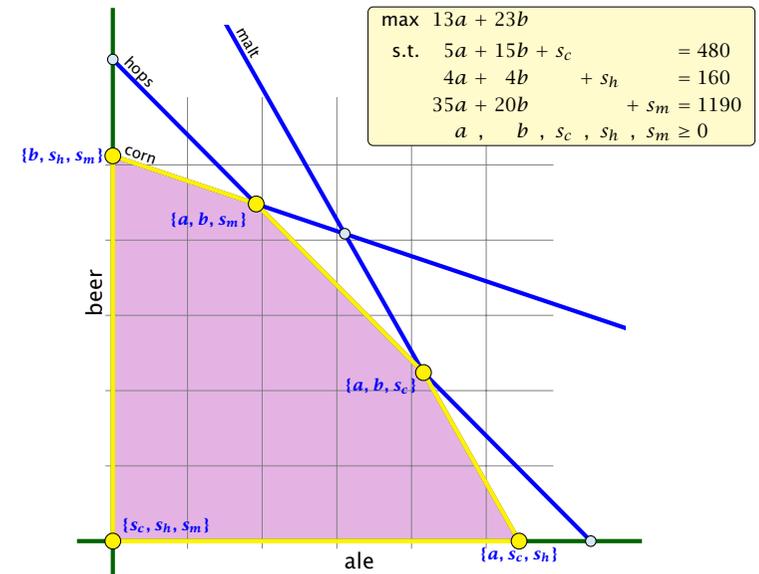
The simplex tableaux for basis  $B$  is

$$\begin{aligned} (c_N^t - c_B^t A_B^{-1} A_N) x_N &= Z - c_B^t A_B^{-1} b \\ I x_B + A_B^{-1} A_N x_N &= A_B^{-1} b \\ x_B, x_N &\geq 0 \end{aligned}$$

The BFS is given by  $x_N = 0, x_B = A_B^{-1} b$ .

If  $(c_N^t - c_B^t A_B^{-1} A_N) \leq 0$  we know that we have an optimum solution.

## Geometric View of Pivoting



## Algebraic Definition of Pivoting

- ▶ Given basis  $B$  with BFS  $x^*$ .
- ▶ Choose index  $j \notin B$  in order to increase  $x_j^*$  from 0 to  $\theta > 0$ .
  - ▶ Other non-basis variables should stay at 0.
  - ▶ Basis variables change to maintain feasibility.
- ▶ Go from  $x^*$  to  $x^* + \theta \cdot d$ .

Requirements for  $d$ :

- ▶  $d_j = 1$  (normalization)
- ▶  $d_\ell = 0, \ell \notin B, \ell \neq j$
- ▶  $A(x^* + \theta d) = b$  must hold. Hence  $Ad = 0$ .
- ▶ Altogether:  $A_B d_B + A_{*j} = Ad = 0$ , which gives  $d_B = -A_B^{-1} A_{*j}$ .

## Algebraic Definition of Pivoting

### Definition 11 ( $j$ -th basis direction)

Let  $B$  be a basis, and let  $j \notin B$ . The vector  $d$  with  $d_j = 1$  and  $d_\ell = 0, \ell \notin B, \ell \neq j$  and  $d_B = -A_B^{-1} A_{*j}$  is called the  $j$ -th basis direction for  $B$ .

Going from  $x^*$  to  $x^* + \theta \cdot d$  the objective function changes by

$$\theta \cdot c^t d = \theta (c_j - c_B^t A_B^{-1} A_{*j})$$

## Algebraic Definition of Pivoting

### Definition 12 (Reduced Cost)

For a basis  $B$  the value

$$\tilde{c}_j = c_j - c_B^t A_B^{-1} A_{*j}$$

is called the **reduced cost** for variable  $x_j$ .

Note that this is defined for every  $j$ . If  $j \in B$  then the above term is 0.

## Algebraic Definition of Pivoting

Let our linear program be

$$\begin{aligned} c_B^t x_B + c_N^t x_N &= Z \\ A_B x_B + A_N x_N &= b \\ x_B, x_N &\geq 0 \end{aligned}$$

The simplex tableau for basis  $B$  is

$$\begin{aligned} (c_N^t - c_B^t A_B^{-1} A_N) x_N &= Z - c_B^t A_B^{-1} b \\ I x_B + A_B^{-1} A_N x_N &= A_B^{-1} b \\ x_B, x_N &\geq 0 \end{aligned}$$

The BFS is given by  $x_N = 0, x_B = A_B^{-1} b$ .

If  $(c_N^t - c_B^t A_B^{-1} A_N) \leq 0$  we know that we have an optimum solution.

## 4 Simplex Algorithm

### Questions:

- ▶ What happens if the min ratio test fails to give us a value  $\theta$  by which we can safely increase the entering variable?
- ▶ How do we find the initial basic feasible solution?
- ▶ Is there always a basis  $B$  such that

$$(c_N^t - c_B^t A_B^{-1} A_N) \leq 0 ?$$

Then we can terminate because we know that the solution is optimal.

- ▶ If yes how do we make sure that we reach such a basis?

## Min Ratio Test

The min ratio test computes a value  $\theta \geq 0$  such that after setting the entering variable to  $\theta$  the leaving variable becomes 0 and all other variables stay non-negative.

For this one computes  $b_i/A_{ie}$  for all constraints  $i$  and calculates the minimum positive value.

What does it mean that the ratio  $b_i/A_{ie}$  (and hence  $A_{ie}$ ) is negative for a constraint?

This means that the corresponding basic variable will increase if we increase  $b$ . Hence, there is no danger of this basic variable becoming negative

What happens if **all**  $b_i/A_{ie}$  are negative? Then we do not have a leaving variable. **Then the LP is unbounded!**

## Termination

The objective function does not decrease during one iteration of the simplex-algorithm.

Does it always increase?

## Termination

The objective function may not increase!

Because a variable  $x_\ell$  with  $\ell \in B$  is already 0.

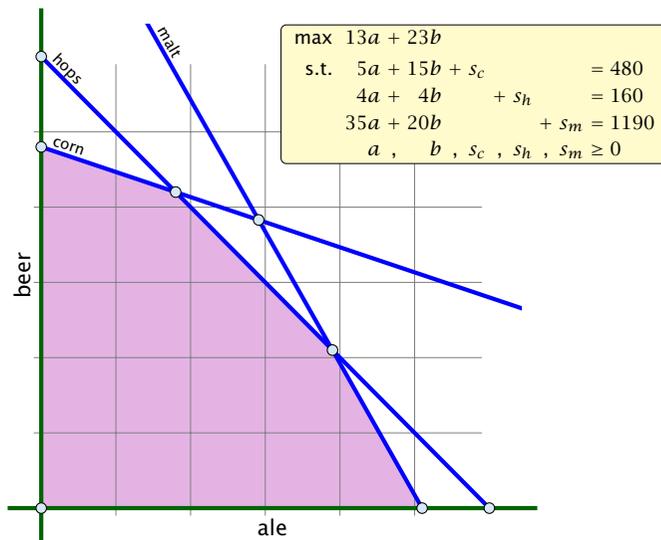
The set of inequalities is **degenerate** (also the basis is degenerate).

### Definition 13 (Degeneracy)

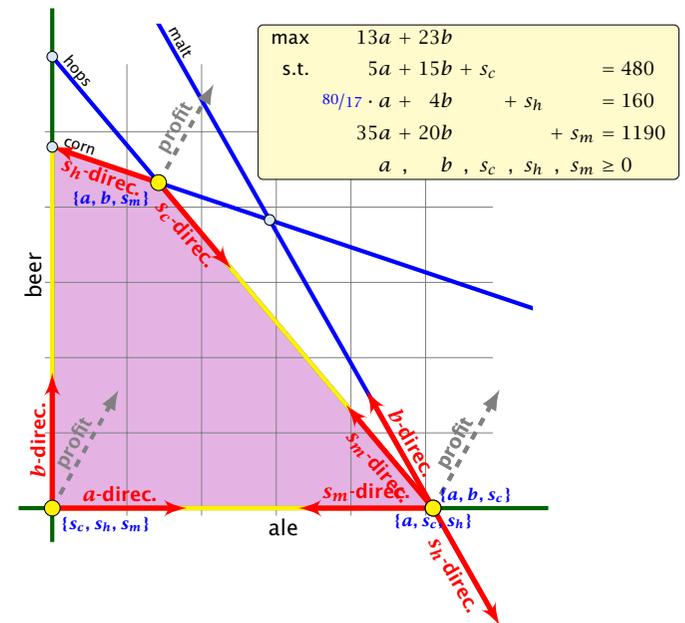
A BFS  $x^*$  is called **degenerate** if the set  $J = \{j \mid x_j^* > 0\}$  fulfills  $|J| < m$ .

It is possible that the algorithm **cycles**, i.e., it cycles through a sequence of different bases without ever terminating. Happens, very rarely in practise.

## Non Degenerate Example



## Degenerate Example



## Summary: How to choose pivot-elements

- ▶ We can choose a column  $e$  as an entering variable if  $\tilde{c}_e > 0$  ( $\tilde{c}_e$  is reduced cost for  $x_e$ ).
- ▶ The standard choice is the column that maximizes  $\tilde{c}_e$ .
- ▶ If  $A_{ie} \leq 0$  for all  $i \in \{1, \dots, m\}$  then the maximum is not bounded.
- ▶ Otw. choose a leaving variable  $\ell$  such that  $b_\ell / A_{\ell e}$  is minimal among all variables  $i$  with  $A_{ie} > 0$ .
- ▶ If several variables have minimum  $b_\ell / A_{\ell e}$  you reach a **degenerate** basis.
- ▶ Depending on the choice of  $\ell$  it may happen that the algorithm runs into a cycle where it does not escape from a degenerate vertex.

## Termination

### What do we have so far?

Suppose we are given an initial feasible solution to an LP. If the LP is non-degenerate then Simplex will terminate.

Note that we either terminate because the min-ratio test fails and we can conclude that the LP is **unbounded**, or we terminate because the vector of reduced cost is non-positive. In the latter case we have an **optimum solution**.

### How do we come up with an initial solution?

- ▶  $Ax \leq b, x \geq 0$ , and  $b \geq 0$ .
- ▶ The standard slack form for this problem is  $Ax + Is = b, x \geq 0, s \geq 0$ , where  $s$  denotes the vector of slack variables.
- ▶ Then  $s = b, x = 0$  is a basic feasible solution (how?).
- ▶ We directly can start the simplex algorithm.

How do we find an initial basic feasible solution for an arbitrary problem?

## Two phase algorithm

Suppose we want to maximize  $c^t x$  s.t.  $Ax = b, x \geq 0$ .

1. Multiply all rows with  $b_i < 0$  by  $-1$ .
2. maximize  $-\sum_i v_i$  s.t.  $Ax + Iv = b, x \geq 0, v \geq 0$  using Simplex.  $x = 0, v = b$  is initial feasible.
3. If  $\sum_i v_i > 0$  then the original problem is **infeasible**.
4. Otw. you have  $x \geq 0$  with  $Ax = b$ .
5. From this you can get basic feasible solution.
6. Now you can start the Simplex for the original problem.

## Optimality

### Lemma 14

Let  $B$  be a basis and  $x^*$  a BFS corresponding to basis  $B$ .  $\tilde{c} \leq 0$  implies that  $x^*$  is an optimum solution to the LP.

## Duality

How do we get an upper bound to a maximization LP?

$$\begin{aligned} \max \quad & 13a + 23b \\ \text{s.t.} \quad & 5a + 15b \leq 480 \\ & 4a + 4b \leq 160 \\ & 35a + 20b \leq 1190 \\ & a, b \geq 0 \end{aligned}$$

Note that a lower bound is easy to derive. Every choice of  $a, b \geq 0$  gives us a lower bound (e.g.  $a = 12, b = 28$  gives us a lower bound of 800).

If you take a conic combination of the rows (multiply the  $i$ -th row with  $y_i \geq 0$ ) such that  $\sum_i y_i a_{ij} \geq c_j$  then  $\sum_i y_i b_i$  will be an upper bound.

## Duality

### Definition 15

Let  $z = \max\{c^t x \mid Ax \leq b, x \geq 0\}$  be a linear program  $P$  (called the primal linear program).

The linear program  $D$  defined by

$$w = \min\{b^t y \mid A^t y \geq c, y \geq 0\}$$

is called the **dual problem**.

## Duality

### Lemma 16

The dual of the dual problem is the primal problem.

**Proof:**

- ▶  $w = \min\{b^t y \mid A^t y \geq c, y \geq 0\}$
- ▶  $w = -\max\{-b^t y \mid -A^t y \leq -c, y \geq 0\}$

The dual problem is

- ▶  $z = -\min\{-c^t x \mid -Ax \geq -b, x \geq 0\}$
- ▶  $z = \max\{c^t x \mid Ax \leq b, x \geq 0\}$

## Weak Duality

Let  $z = \max\{c^t x \mid Ax \leq b, x \geq 0\}$  and  $w = \min\{b^t y \mid A^t y \geq c, y \geq 0\}$  be a primal dual pair.

$x$  is primal feasible iff  $x \in \{x \mid Ax \leq b, x \geq 0\}$

$y$  is dual feasible, iff  $y \in \{y \mid A^t y \geq c, y \geq 0\}$ .

### Theorem 17 (Weak Duality)

Let  $\hat{x}$  be primal feasible and let  $\hat{y}$  be dual feasible. Then

$$c^t \hat{x} \leq z \leq w \leq b^t \hat{y} .$$

## Weak Duality

$$A^t \hat{y} \geq c \Rightarrow \hat{x}^t A^t \hat{y} \geq \hat{x}^t c \quad (\hat{x} \geq 0)$$

$$A \hat{x} \leq b \Rightarrow y^t A \hat{x} \leq y^t b \quad (y \geq 0)$$

This gives

$$c^t \hat{x} \leq y^t A \hat{x} \leq b^t y .$$

Since, there exists primal feasible  $\hat{x}$  with  $c^t \hat{x} = z$ , and dual feasible  $\hat{y}$  with  $b^t \hat{y} = w$  we get  $z \leq w$ .

If  $P$  is unbounded then  $D$  is infeasible.

The following linear programs form a primal dual pair:

$$z = \max\{c^t x \mid Ax = b, x \geq 0\}$$
$$w = \min\{b^t y \mid A^t y \geq c\}$$

This means for computing the dual of a standard form LP, we do not have non-negativity constraints for the dual variables.

## Proof

**Primal:**

$$\begin{aligned} \max\{c^t x \mid Ax = b, x \geq 0\} \\ &= \max\{c^t x \mid Ax \leq b, -Ax \leq -b, x \geq 0\} \\ &= \max\{c^t x \mid \begin{bmatrix} A \\ -A \end{bmatrix} x \leq \begin{bmatrix} b \\ -b \end{bmatrix}, x \geq 0\} \end{aligned}$$

**Dual:**

$$\begin{aligned} \min\{[b^t \ -b^t] y \mid [A^t \ -A^t] y \geq c, y \geq 0\} \\ &= \min \left\{ [b^t \ -b^t] \cdot \begin{bmatrix} y^+ \\ y^- \end{bmatrix} \mid [A^t \ -A^t] \cdot \begin{bmatrix} y^+ \\ y^- \end{bmatrix} \geq c, y^- \geq 0, y^+ \geq 0 \right\} \\ &= \min \{ b^t \cdot (y^+ - y^-) \mid A^t \cdot (y^+ - y^-) \geq c, y^- \geq 0, y^+ \geq 0 \} \\ &= \min \{ b^t y' \mid A^t y' \geq c \} \end{aligned}$$

## Proof of Optimality Criterion for Simplex

Suppose that we have a basic feasible solution with **reduced cost**

$$\tilde{c} = c^t - c_B^t A_B^{-1} A \leq 0$$

This is equivalent to  $A^t (A_B^{-1})^t c_B \geq c$

$y^* = (A_B^{-1})^t c_B$  is solution to the **dual**  $\min\{b^t y \mid A^t y \geq c\}$ .

$$\begin{aligned} b^t y^* &= (A x^*)^t y^* = (A_B x_B^*)^t y^* \\ &= (A_B x_B^*)^t (A_B^{-1})^t c_B = (x_B^*)^t A_B^t (A_B^{-1})^t c_B \\ &= c^t x^* \end{aligned}$$

Hence, the solution is optimal.

## Strong Duality

### Theorem 18 (Strong Duality)

Let  $P$  and  $D$  be a primal dual pair of linear programs, and let  $z^*$  and  $w^*$  denote the optimal solution to  $P$  and  $D$ , respectively. Then

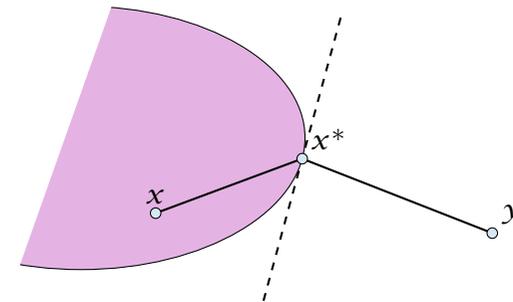
$$z^* = w^*$$

### Lemma 19 (Weierstrass)

Let  $X$  be a compact set and let  $f(x)$  be a continuous function on  $X$ . Then  $\min\{f(x) : x \in X\}$  exists.

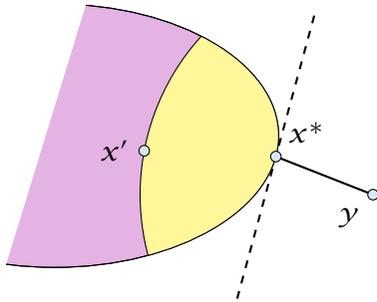
### Lemma 20 (Projection Lemma)

Let  $X \subseteq \mathbb{R}^m$  be a non-empty convex set, and let  $y \notin X$ . Then there exist  $x^* \in X$  with minimum distance from  $y$ . Moreover for all  $x \in X$  we have  $(y - x^*)^t (x - x^*) \leq 0$ .



## Proof of the Projection Lemma

- ▶ Define  $f(x) = \|y - x\|$ .
- ▶ We want to apply Weierstrass but  $X$  may not be bounded.
- ▶  $X \neq \emptyset$ . Hence, there exists  $x' \in X$ .
- ▶ Define  $X' = \{x \in X \mid \|y - x\| \leq \|y - x'\|\}$ . This set is closed and bounded.
- ▶ Applying Weierstrass gives the existence.



## Proof of the Projection Lemma (continued)

$x^*$  is minimum. Hence  $\|y - x^*\|^2 \leq \|y - x\|^2$  for all  $x \in X$ .

By **convexity**:  $x \in X$  then  $x^* + \epsilon(x - x^*) \in X$  for all  $0 \leq \epsilon \leq 1$ .

$$\begin{aligned} \|y - x^*\|^2 &\leq \|y - x^* - \epsilon(x - x^*)\|^2 \\ &= \|y - x^*\|^2 + \epsilon^2\|x - x^*\|^2 - 2\epsilon(y - x^*)^t(x - x^*) \end{aligned}$$

Hence,  $(y - x^*)^t(x - x^*) \leq \frac{1}{2}\epsilon\|x - x^*\|^2$ .

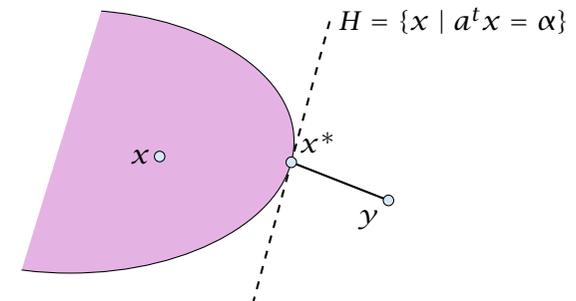
Letting  $\epsilon \rightarrow 0$  gives the result.

## Theorem 21 (Separating Hyperplane)

Let  $X \subseteq \mathbb{R}^m$  be a non-empty closed convex set, and let  $y \notin X$ . Then there exists a **separating hyperplane**  $\{x \in \mathbb{R}^m : a^t x = \alpha\}$  where  $a \in \mathbb{R}^m$ ,  $\alpha \in \mathbb{R}$  that **separates**  $y$  from  $X$ . ( $a^t y < \alpha$ ;  $a^t x \geq \alpha$  for all  $x \in X$ )

## Proof of the Hyperplane Lemma

- ▶ Let  $x^* \in X$  be closest point to  $y$  in  $X$ .
- ▶ By previous lemma  $(y - x^*)^t(x - x^*) \leq 0$  for all  $x \in X$ .
- ▶ Choose  $a = (x^* - y)$  and  $\alpha = a^t x^*$ .
- ▶ For  $x \in X$ :  $a^t(x - x^*) \geq 0$ , and, hence,  $a^t x \geq \alpha$ .
- ▶ Also,  $a^t y = a^t(x^* - a) = \alpha - \|a\|^2 < \alpha$



### Lemma 22 (Farkas Lemma)

Let  $A$  be an  $m \times n$  matrix,  $b \in \mathbb{R}^m$ . Then *exactly one* of the following statements holds.

1.  $\exists x \in \mathbb{R}^n$  with  $Ax = b$ ,  $x \geq 0$
2.  $\exists y \in \mathbb{R}^m$  with  $A^t y \geq 0$ ,  $b^t y < 0$

Assume  $\hat{x}$  satisfies 1. and  $\hat{y}$  satisfies 2. Then

$$0 > \hat{y}^t b = \hat{y}^t A \hat{x} \geq 0$$

Hence, at most one of the statements can hold.

## Proof of Farkas Lemma

Now, assume that 1. does not hold.

Consider  $S = \{Ax : x \geq 0\}$  so that  $S$  closed, convex,  $b \notin S$ .

We want to show that there is  $y$  with  $A^t y \geq 0$ ,  $b^t y < 0$ .

Let  $y$  be a hyperplane that separates  $b$  from  $S$ . Hence,  $y^t b < \alpha$  and  $y^t s \geq \alpha$  for all  $s \in S$ .

$$0 \in S \Rightarrow \alpha \leq 0 \Rightarrow y^t b < 0$$

$y^t Ax \geq \alpha$  for all  $x \geq 0$ . Hence,  $y^t A \geq 0$  as we can choose  $x$  arbitrarily large.

### Lemma 23 (Farkas Lemma; different version)

Let  $A$  be an  $m \times n$  matrix,  $b \in \mathbb{R}^m$ . Then *exactly one* of the following statements holds.

1.  $\exists x \in \mathbb{R}^n$  with  $Ax \leq b$ ,  $x \geq 0$
2.  $\exists y \in \mathbb{R}^m$  with  $A^t y \geq 0$ ,  $b^t y < 0$ ,  $y \geq 0$

Rewrite the conditions:

1.  $\exists x \in \mathbb{R}^n$  with  $\begin{bmatrix} A & I \end{bmatrix} \cdot \begin{bmatrix} x \\ s \end{bmatrix} = b$ ,  $x \geq 0$ ,  $s \geq 0$
2.  $\exists y \in \mathbb{R}^m$  with  $\begin{bmatrix} A^t \\ I \end{bmatrix} y \geq 0$ ,  $b^t y < 0$

## Proof of Strong Duality

$$P: z = \max\{c^t x \mid Ax \leq b, x \geq 0\}$$

$$D: w = \min\{b^t y \mid A^t y \geq c, y \geq 0\}$$

### Theorem 24 (Strong Duality)

Let  $P$  and  $D$  be a primal dual pair of linear programs, and let  $z$  and  $w$  denote the optimal solution to  $P$  and  $D$ , respectively (i.e.,  $P$  and  $D$  are non-empty). Then

$$z = w .$$

## Proof of Strong Duality

$z \leq w$ : follows from weak duality

$z \geq w$ :

We show  $z < \alpha$  implies  $w < \alpha$ .

$\begin{aligned} \exists x \in \mathbb{R}^n \\ \text{s.t. } Ax &\leq b \\ -c^t x &\leq -\alpha \\ x &\geq 0 \end{aligned}$	$\begin{aligned} \exists y \in \mathbb{R}^m; v \in \mathbb{R} \\ \text{s.t. } A^t y - cv &\geq 0 \\ b^t y - \alpha v &< 0 \\ y, v &\geq 0 \end{aligned}$
--	--

From the definition of  $\alpha$  we know that the first system is infeasible; hence the second must be feasible.

## Proof of Strong Duality

$$\begin{aligned} \exists y \in \mathbb{R}^m; v \in \mathbb{R} \\ \text{s.t. } A^t y - v &\geq 0 \\ b^t y - \alpha v &< 0 \\ y, v &\geq 0 \end{aligned}$$

If the solution  $y, v$  has  $v = 0$  we have that

$$\begin{aligned} \exists y \in \mathbb{R}^m \\ \text{s.t. } A^t y &\geq 0 \\ b^t y &< 0 \\ y &\geq 0 \end{aligned}$$

is feasible. By Farkas lemma this gives that LP  $P$  is infeasible. Contradiction to the assumption of the lemma.

## Proof of Strong Duality

Hence, there exists a solution  $y, v$  with  $v > 0$ .

We can rescale this solution (scaling both  $y$  and  $v$ ) s.t.  $v = 1$ .

Then  $y$  is feasible for the dual but  $b^t y < \alpha$ . This means that  $w < \alpha$ .

## Fundamental Questions

### Definition 25 (Linear Programming Problem (LP))

Let  $A \in \mathbb{Q}^{m \times n}$ ,  $b \in \mathbb{Q}^m$ ,  $c \in \mathbb{Q}^n$ ,  $\alpha \in \mathbb{Q}$ . Does there exist  $x \in \mathbb{Q}^n$  s.t.  $Ax = b$ ,  $x \geq 0$ ,  $c^t x \geq \alpha$ ?

### Questions:

- ▶ Is LP in NP?
- ▶ Is LP in co-NP? **yes!**
- ▶ Is LP in P?

### Proof:

- ▶ Given a primal maximization problem  $P$  and a parameter  $\alpha$ . Suppose that  $\alpha > \text{opt}(P)$ .
- ▶ We can prove this by providing an optimal basis for the dual.
- ▶ A verifier can check that the associated dual solution fulfills all dual constraints and that it has dual cost  $< \alpha$ .

## Complementary Slackness

### Lemma 26

Assume a linear program  $P = \max\{c^t x \mid Ax \leq b; x \geq 0\}$  has solution  $x^*$  and its dual  $D = \min\{b^t y \mid A^t y \geq c; y \geq 0\}$  has solution  $y^*$ .

1. If  $x_j^* > 0$  then the  $j$ -th constraint in  $D$  is tight.
2. If the  $j$ -th constraint in  $D$  is not tight than  $x_j^* = 0$ .
3. If  $y_i^* > 0$  then the  $i$ -th constraint in  $P$  is tight.
4. If the  $i$ -th constraint in  $P$  is not tight than  $y_i^* = 0$ .

If we say that a variable  $x_j^*$  ( $y_i^*$ ) has slack if  $x_j^* > 0$  ( $y_i^* > 0$ ), (i.e., the corresponding variable restriction is not tight) and a constraint has slack if it is not tight, then the above says that for a primal-dual solution pair it is not possible that a constraint **and** its corresponding (dual) variable has slack.

## Proof: Complementary Slackness

Analogous to the proof of weak duality we obtain

$$c^t x^* \leq y^{*t} A x^* \leq b^t y^*$$

Because of strong duality we then get

$$c^t x^* = y^{*t} A x^* = b^t y^*$$

This gives e.g.

$$\sum_j (y^t A - c^t)_j x_j^* = 0$$

From the constraint of the dual it follows that  $y^t A \geq c^t$ . Hence the left hand side is a sum over the product of non-negative numbers. Hence, if e.g.  $(y^t A - c^t)_j > 0$  (the  $j$ -th constraint in the dual is not tight) then  $x_j = 0$  (2.). The result for (1./3./4.) follows similarly.

## Interpretation of Dual Variables

- ▶ Brewer: find mix of ale and beer that maximizes profits

$$\begin{aligned} \max \quad & 13a + 23b \\ \text{s.t.} \quad & 5a + 15b \leq 480 \\ & 4a + 4b \leq 160 \\ & 35a + 20b \leq 1190 \\ & a, b \geq 0 \end{aligned}$$

- ▶ Entrepreneur: buy resources from brewer at minimum cost  
 $C, H, M$ : unit price for corn, hops and malt.

$$\begin{aligned} \min \quad & 480C + 160H + 1190M \\ \text{s.t.} \quad & 5C + 4H + 35M \geq 13 \\ & 15C + 4H + 20M \geq 23 \\ & C, H, M \geq 0 \end{aligned}$$

Note that brewer won't sell (at least not all) if e.g.  $5C + 4H + 35M < 13$  as then brewing ale would be advantageous.

## Interpretation of Dual Variables

### Marginal Price:

- ▶ How much money is the brewer willing to pay for additional amount of Corn, Hops, or Malt?
- ▶ We are interested in the marginal price, i.e., what happens if we increase the amount of Corn, Hops, and Malt by  $\varepsilon_C, \varepsilon_H$ , and  $\varepsilon_M$ , respectively.

The profit increases to  $\max\{c^t x \mid Ax \leq b + \varepsilon; x \geq 0\}$ . Because of strong duality this is equal to

$$\begin{aligned} \min \quad & (b^t + \varepsilon^t) y \\ \text{s.t.} \quad & A^t y \geq c \\ & y \geq 0 \end{aligned}$$

## Interpretation of Dual Variables

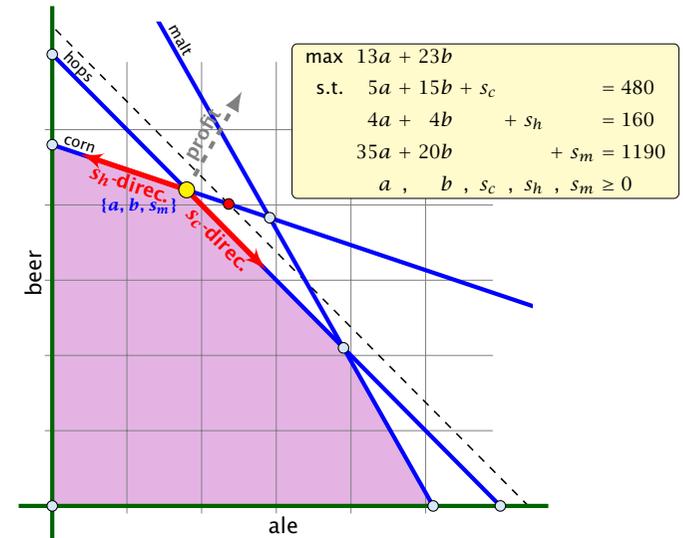
If  $\epsilon$  is “small” enough then the optimum dual solution  $y^*$  might not change. Therefore the profit increases by  $\sum_i \epsilon_i y_i^*$ .

Therefore we can interpret the dual variables as **marginal prices**.

Note that with this interpretation, complementary slackness becomes obvious.

- ▶ If the brewer has slack of some resource (e.g. corn) then he is not willing to pay anything for it (corresponding dual variable is zero).
- ▶ If the dual variable for some resource is non-zero, then an increase of this resource increases the profit of the brewer. Hence, it makes no sense to have left-overs of this resource. Therefore its slack must be zero.

## Example



The change in profit when increasing hops by one unit is  
 $= \underbrace{c_B^t A_B^{-1}}_{y^*} e_h.$

Of course, the previous argument about the increase in the primal objective only holds for the non-degenerate case.

If the optimum basis is degenerate then increasing the supply of one resource may not allow the objective value to increase.

## Flows

### Definition 27

An  $(s, t)$ -flow in a (complete) directed graph  $G = (V, V \times V, c)$  is a function  $f: V \times V \rightarrow \mathbb{R}_0^+$  that satisfies

1. For each edge  $(x, y)$

$$0 \leq f_{xy} \leq c_{xy} .$$

(capacity constraints)

2. For each  $v \in V \setminus \{s, t\}$

$$\sum_x f_{vx} = \sum_x f_{xv} .$$

(flow conservation constraints)

## Flows

### Definition 28

The value of an  $(s, t)$ -flow  $f$  is defined as

$$\text{val}(f) = \sum_x f_{sx} - \sum_x f_{xs}.$$

### Maximum Flow Problem:

Find an  $(s, t)$ -flow with maximum value.

## LP-Formulation of Maxflow

$$\begin{aligned} \max \quad & \sum_z f_{sz} - \sum_z f_{zs} \\ \text{s.t.} \quad & \forall (z, w) \in V \times V \quad f_{zw} \leq c_{zw} \quad \ell_{zw} \\ & \forall w \neq s, t \quad \sum_z f_{zw} - \sum_z f_{wz} = 0 \quad p_w \\ & f_{zw} \geq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & \sum_{(xy)} c_{xy} \ell_{xy} \\ \text{s.t.} \quad & f_{xy} (x, y \neq s, t): \quad 1\ell_{xy} - 1p_x + 1p_y \geq 0 \\ & f_{sy} (y \neq s, t): \quad 1\ell_{sy} \quad + 1p_y \geq 1 \\ & f_{xs} (x \neq s, t): \quad 1\ell_{xs} - 1p_x \geq -1 \\ & f_{ty} (y \neq s, t): \quad 1\ell_{ty} \quad + 1p_y \geq 0 \\ & f_{xt} (x \neq s, t): \quad 1\ell_{xt} - 1p_x \geq 0 \\ & f_{st}: \quad 1\ell_{st} \geq 1 \\ & f_{ts}: \quad 1\ell_{ts} \geq -1 \\ & \ell_{xy} \geq 0 \end{aligned}$$

## LP-Formulation of Maxflow

$$\begin{aligned} \min \quad & \sum_{(xy)} c_{xy} \ell_{xy} \\ \text{s.t.} \quad & f_{xy} (x, y \neq s, t): \quad 1\ell_{xy} - 1p_x + 1p_y \geq 0 \\ & f_{sy} (y \neq s, t): \quad 1\ell_{sy} - \quad 1 + 1p_y \geq 0 \\ & f_{xs} (x \neq s, t): \quad 1\ell_{xs} - 1p_x + \quad 1 \geq 0 \\ & f_{ty} (y \neq s, t): \quad 1\ell_{ty} - \quad 0 + 1p_y \geq 0 \\ & f_{xt} (x \neq s, t): \quad 1\ell_{xt} - 1p_x + \quad 0 \geq 0 \\ & f_{st}: \quad 1\ell_{st} - \quad 1 + \quad 0 \geq 0 \\ & f_{ts}: \quad 1\ell_{ts} - \quad 0 + \quad 1 \geq 0 \\ & \ell_{xy} \geq 0 \end{aligned}$$

## LP-Formulation of Maxflow

$$\begin{aligned} \min \quad & \sum_{(xy)} c_{xy} \ell_{xy} \\ \text{s.t.} \quad & f_{xy} (x, y \neq s, t): \quad 1\ell_{xy} - 1p_x + 1p_y \geq 0 \\ & f_{sy} (y \neq s, t): \quad 1\ell_{sy} - \quad p_s + 1p_y \geq 0 \\ & f_{xs} (x \neq s, t): \quad 1\ell_{xs} - 1p_x + \quad p_s \geq 0 \\ & f_{ty} (y \neq s, t): \quad 1\ell_{ty} - \quad p_t + 1p_y \geq 0 \\ & f_{xt} (x \neq s, t): \quad 1\ell_{xt} - 1p_x + \quad p_t \geq 0 \\ & f_{st}: \quad 1\ell_{st} - \quad p_s + \quad p_t \geq 0 \\ & f_{ts}: \quad 1\ell_{ts} - \quad p_t + \quad p_s \geq 0 \\ & \ell_{xy} \geq 0 \end{aligned}$$

with  $p_t = 0$  and  $p_s = 1$ .

## LP-Formulation of Maxflow

$$\begin{array}{ll}
 \min & \sum_{(x,y)} c_{xy} l_{xy} \\
 \text{s.t.} & f_{xy} : 1l_{xy} - 1p_x + 1p_y \geq 0 \\
 & l_{xy} \geq 0 \\
 & p_s = 1 \\
 & p_t = 0
 \end{array}$$

We can interpret the  $l_{xy}$  value as assigning a length to every edge.

The value  $p_x$  for a variable, then can be seen as the distance of  $x$  to  $t$  (where the distance from  $s$  to  $t$  is required to be 1 since  $p_s = 1$ ).

The constraint  $p_x \leq l_{xy} + p_y$  then simply follows from triangle inequality ( $d(x,t) \leq d(x,y) + d(y,t) \Rightarrow d(x,t) \leq l_{xy} + d(y,t)$ ).

One can show that there is an optimum LP-solution for the dual problem that gives an integral assignment of variables.

This means  $p_x = 1$  or  $p_x = 0$  for our case. This gives rise to a cut in the graph with vertices having value 1 on one side and the other vertices on the other side. The objective function then evaluates the capacity of this cut.

This shows that the Maxflow/Mincut theorem follows from linear programming duality.

## Degeneracy Revisited

If a basis variable is 0 in the basic feasible solution then we may not make progress during an iteration of simplex.

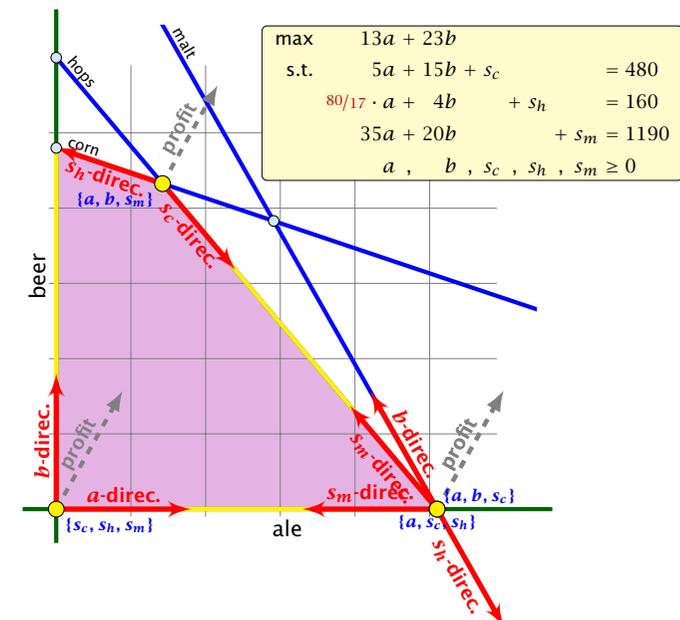
### Idea:

Change  $LP := \max\{c^t x, Ax = b; x \geq 0\}$  into

$LP' := \max\{c^t x, Ax = b', x \geq 0\}$  such that

- I. LP is feasible
- II. If a set  $B$  of basis variables corresponds to an **infeasible** basis (i.e.  $A_B^{-1}b \not\geq 0$ ) then  $B$  corresponds to an infeasible basis in  $LP'$  (note that columns in  $A_B$  are linearly independent).
- III. LP has no degenerate basic solutions

## Degenerate Example



## Degeneracy Revisited

If a basis variable is 0 in the basic feasible solution then we may not make progress during an iteration of simplex.

### Idea:

Given feasible LP :=  $\max\{c^t x, Ax = b; x \geq 0\}$ . Change it into LP' :=  $\max\{c^t x, Ax = b', x \geq 0\}$  such that

- I. LP' is feasible
- II. If a set  $B$  of basis variables corresponds to an **infeasible** basis (i.e.  $A_B^{-1}b \not\geq 0$ ) then  $B$  corresponds to an infeasible basis in LP' (note that columns in  $A_B$  are linearly independent).
- III. LP' has no degenerate basic solutions

## Perturbation

Let  $B$  be index set of **some** basis with basic solution

$$x_B^* = A_B^{-1}b \geq 0, x_N^* = 0 \quad (\text{i.e. } B \text{ is feasible})$$

Fix

$$b' := b + A_B \begin{pmatrix} \varepsilon \\ \vdots \\ \varepsilon^m \end{pmatrix} \text{ for } \varepsilon > 0 .$$

This is the perturbation that we are using.

## Property I

The new LP is feasible because the set  $B$  of basis variables provides a feasible basis:

$$A_B^{-1} \left( b + A_B \begin{pmatrix} \varepsilon \\ \vdots \\ \varepsilon^m \end{pmatrix} \right) = x_B^* + \begin{pmatrix} \varepsilon \\ \vdots \\ \varepsilon^m \end{pmatrix} \geq 0 .$$

## Property II

Let  $\tilde{B}$  be a non-feasible basis. This means  $(A_{\tilde{B}}^{-1}b)_i < 0$  for some row  $i$ .

Then for small enough  $\varepsilon > 0$

$$\left( A_{\tilde{B}}^{-1} \left( b + A_B \begin{pmatrix} \varepsilon \\ \vdots \\ \varepsilon^m \end{pmatrix} \right) \right)_i = (A_{\tilde{B}}^{-1}b)_i + \left( A_{\tilde{B}}^{-1}A_B \begin{pmatrix} \varepsilon \\ \vdots \\ \varepsilon^m \end{pmatrix} \right)_i < 0$$

Hence,  $\tilde{B}$  is not feasible.

## Property III

Let  $\tilde{B}$  be a basis. It has an associated solution

$$x_{\tilde{B}}^* = A_{\tilde{B}}^{-1}b + A_{\tilde{B}}^{-1}A_B \begin{pmatrix} \varepsilon \\ \vdots \\ \varepsilon^m \end{pmatrix}$$

in the perturbed instance.

We can view each component of the vector as a polynomial with variable  $\varepsilon$  of degree at most  $m$ .

$A_{\tilde{B}}^{-1}A_B$  has rank  $m$ . Therefore no polynomial is 0.

A polynomial of degree at most  $m$  has at most  $m$  roots (Nullstellen).

Hence,  $\varepsilon > 0$  small enough gives that no component of the above vector is 0. Hence, no degeneracies.

Since, there are no degeneracies Simplex will terminate when run on  $LP'$ .

- ▶ If it terminates because the reduced cost vector fulfills

$$\tilde{c} = (c^t - c_B^t A_B^{-1}A) \leq 0$$

then we have found an optimal basis. Note that this basis is also optimal for LP, as the above constraint does not depend on  $b$ .

- ▶ If it terminates because it finds a variable  $x_j$  with  $\tilde{c}_j > 0$  for which the  $j$ -th basis direction  $d$ , fulfills  $d \geq 0$  we know that  $LP'$  is unbounded. The basis direction does not depend on  $b$ . Hence, we also know that LP is unbounded.

## Lexicographic Pivoting

Doing calculations with perturbed instances may be costly. Also the right choice of  $\varepsilon$  is difficult.

### Idea:

Simulate behaviour of  $LP'$  without explicitly doing a perturbation.

## Lexicographic Pivoting

We choose the entering variable arbitrarily as before ( $\tilde{c}_e > 0$ , of course).

If we do not have a choice for the leaving variable then  $LP'$  and LP do the same (i.e., choose the same variable).

Otherwise we have to be careful.

## Lexicographic Pivoting

In the following we assume that  $b \geq 0$ . This can be obtained by replacing the initial system  $(A_B \mid b)$  by  $(A_B^{-1}A \mid A_B^{-1}b)$  where  $B$  is the index set of a feasible basis (found e.g. by the first phase of the Two-phase algorithm).

Then the perturbed instance is

$$b' = b + \begin{pmatrix} \varepsilon \\ \vdots \\ \varepsilon^m \end{pmatrix}$$

## Matrix View

Let our linear program be

$$\begin{aligned} c_B^t x_B + c_N^t x_N &= Z \\ A_B x_B + A_N x_N &= b \\ x_B, x_N &\geq 0 \end{aligned}$$

The simplex tableaux for basis  $B$  is

$$\begin{aligned} (c_N^t - c_B^t A_B^{-1} A_N) x_N &= Z - c_B^t A_B^{-1} b \\ I x_B + A_B^{-1} A_N x_N &= A_B^{-1} b \\ x_B, x_N &\geq 0 \end{aligned}$$

The BFS is given by  $x_N = 0, x_B = A_B^{-1}b$ .

If  $(c_N^t - c_B^t A_B^{-1} A_N) \leq 0$  we know that we have an optimum solution.

## Lexicographic Pivoting

LP chooses an arbitrary leaving variable that has  $\hat{A}_{\ell e} > 0$  and minimizes

$$\theta_\ell = \frac{\hat{b}_\ell}{\hat{A}_{\ell e}} = \frac{(A_B^{-1}b)_\ell}{(A_B^{-1}A_{*e})_\ell}.$$

$\ell$  is the index of a leaving variable within  $B$ . This means if e.g.  $B = \{1, 3, 7, 14\}$  and leaving variable is 3 then  $\ell = 2$ .

## Lexicographic Pivoting

### Definition 29

$u \leq_{\text{lex}} v$  if and only if the first component in which  $u$  and  $v$  differ fulfills  $u_i \leq v_i$ .

## Lexicographic Pivoting

LP' chooses an index that minimizes

$$\theta_\ell = \frac{\left( A_B^{-1} \left( b + \begin{pmatrix} \varepsilon \\ \vdots \\ \varepsilon^m \end{pmatrix} \right) \right)_\ell}{(A_B^{-1} A_{*e})_\ell} = \frac{\left( A_B^{-1} (b | I) \begin{pmatrix} 1 \\ \varepsilon \\ \vdots \\ \varepsilon^m \end{pmatrix} \right)_\ell}{(A_B^{-1} A_{*e})_\ell}$$
$$= \frac{\ell\text{-th row of } A_B^{-1} (b | I)}{(A_B^{-1} A_{*e})_\ell} \begin{pmatrix} 1 \\ \varepsilon \\ \vdots \\ \varepsilon^m \end{pmatrix}$$

## Lexicographic Pivoting

This means you can choose the variable/row  $\ell$  for which the vector

$$\frac{\ell\text{-th row of } A_B^{-1} (b | I)}{(A_B^{-1} A_{*e})_\ell}$$

is lexicographically minimal.

Of course only including rows with  $(A_B^{-1} A_{*e})_\ell > 0$ .

This technique guarantees that your pivoting is the same as in the perturbed case. This guarantees that cycling does not occur.

## Number of Simplex Iterations

Each iteration of Simplex can be implemented in polynomial time.

If we use lexicographic pivoting we know that Simplex requires at most  $\binom{n}{m}$  iterations, because it will not visit a basis twice.

The input size is  $L \cdot n \cdot m$ , where  $n$  is the number of **variables**,  $m$  is the number of **constraints**, and  $L$  is the length of the binary representation of the largest coefficient in the matrix  $A$ .

If we really require  $\binom{n}{m}$  iterations then Simplex is not a polynomial time algorithm.

**Can we obtain a better analysis?**

## Number of Simplex Iterations

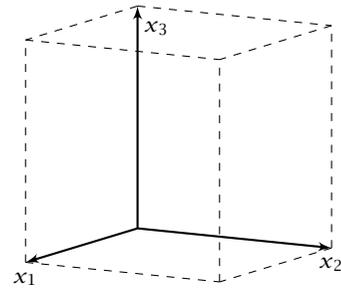
### Observation

Simplex visits every **feasible** basis at most once.

However, also the number of feasible bases can be very large.

## Example

$$\begin{aligned} \max \quad & c^t x \\ \text{s.t.} \quad & 0 \leq x_1 \leq 1 \\ & 0 \leq x_2 \leq 1 \\ & \vdots \\ & 0 \leq x_n \leq 1 \end{aligned}$$

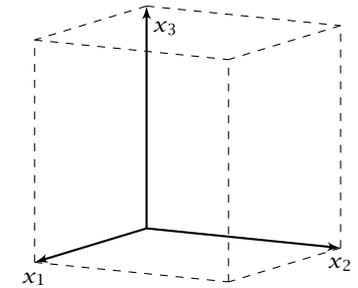


$2n$  constraint on  $n$  variables define an  $n$ -dimensional hypercube as feasible region.

The feasible region has  $2^n$  vertices.

## Example

$$\begin{aligned} \max \quad & c^t x \\ \text{s.t.} \quad & 0 \leq x_1 \leq 1 \\ & 0 \leq x_2 \leq 1 \\ & \vdots \\ & 0 \leq x_n \leq 1 \end{aligned}$$



However, Simplex may still run quickly as it usually does not visit all feasible bases.

In the following we give an example of a feasible region for which there is a bad **Pivoting Rule**.

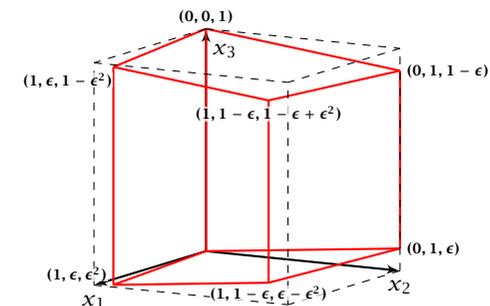
## Pivoting Rule

A Pivoting Rule defines how to choose the entering and leaving variable for an iteration of Simplex.

In the non-degenerate case after choosing the entering variable the leaving variable is unique.

## Klee Minty Cube

$$\begin{aligned} \max \quad & x_n \\ \text{s.t.} \quad & 0 \leq x_1 \leq 1 \\ & \epsilon x_1 \leq x_2 \leq 1 - \epsilon x_1 \\ & \epsilon x_2 \leq x_3 \leq 1 - \epsilon x_2 \\ & \vdots \\ & \epsilon x_{n-1} \leq x_n \leq 1 - \epsilon x_{n-1} \\ & x_i \geq 0 \end{aligned}$$



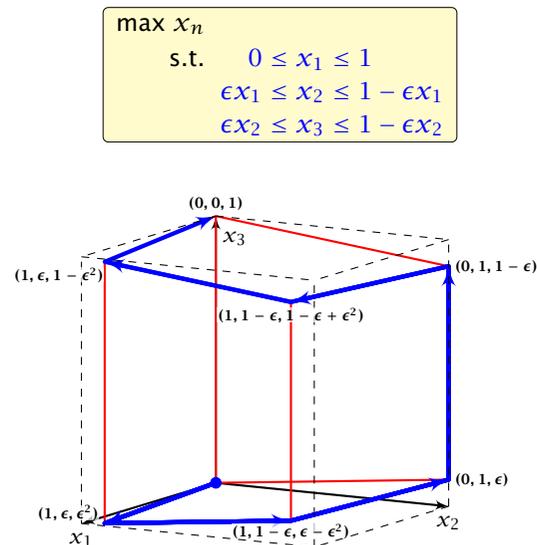
## Observations

- ▶ We have  $2n$  constraints, and  $3n$  variables (after adding slack variables to every constraint).
- ▶ Every basis is defined by  $2n$  variables, and  $n$  non-basic variables.
- ▶ There exist degenerate vertices.
- ▶ The degeneracies come from the non-negativity constraints, which are superfluous.
- ▶ In the following all variables  $x_i$  stay in the basis at all times.
- ▶ Then, we can uniquely specify a basis by choosing for each variable whether it should be equal to its lower bound, or equal to its upper bound (the slack variable corresponding to the non-tight constraint is part of the basis).
- ▶ We can also simply identify each basis/vertex with the corresponding hypercube vertex obtained by letting  $\epsilon \rightarrow 0$ .

## Analysis

- ▶ In the following we specify a sequence of bases (identified by the corresponding hypercube node) along which the objective function strictly increases.
- ▶ The basis  $(0, \dots, 0, 1)$  is the unique optimal basis.
- ▶ Our sequence  $S_n$  starts at  $(0, \dots, 0)$  ends with  $(0, \dots, 0, 1)$  and visits every node of the hypercube.
- ▶ An unfortunate Pivoting Rule may choose this sequence, and, hence, require an exponential number of iterations.

## Klee Minty Cube



## Analysis

The sequence  $S_n$  that visits every node of the hypercube is defined recursively

$$\begin{array}{c} (0, \dots, 0, 0, 0) \\ \downarrow S_{n-1} \\ (0, \dots, 0, 1, 0) \\ \downarrow \\ (0, \dots, 0, 1, 1) \\ \downarrow S_{n-1}^{\text{rev}} \\ (0, \dots, 0, 0, 1) \end{array} \quad \begin{array}{c} \downarrow S_n \\ \downarrow \\ \downarrow \end{array}$$

The non-recursive case is  $S_1 = 0 \rightarrow 1$

## Analysis

### Lemma 30

The objective value  $x_n$  is increasing along path  $S_n$ .

#### Proof by induction:

$n = 1$ : obvious, since  $S_1 = 0 \rightarrow 1$ , and  $1 > 0$ .

$n - 1 \rightarrow n$

- ▶ For the first part the value of  $x_n = \epsilon x_{n-1}$ .
- ▶ By induction hypothesis  $x_{n-1}$  is increasing along  $S_{n-1}$ , hence, also  $x_n$ .
- ▶ Going from  $(0, \dots, 0, 1, 0)$  to  $(0, \dots, 0, 1, 1)$  increases  $x_n$  for small enough  $\epsilon$ .
- ▶ For the remaining path  $S_{n-1}^{\text{rev}}$  we have  $x_n = 1 - \epsilon x_{n-1}$ .
- ▶ By induction hypothesis  $x_{n-1}$  is increasing along  $S_{n-1}$ , hence  $-\epsilon x_{n-1}$  is increasing along  $S_{n-1}^{\text{rev}}$ .

## Remarks about Simplex

### Observation

The simplex algorithm takes at most  $\binom{n}{m}$  iterations. Each iteration can be implemented in time  $\mathcal{O}(mn)$ .

In practise it usually takes a linear number of iterations.

## Remarks about Simplex

### Theorem

For almost all known **deterministic** pivoting rules (rules for choosing entering and leaving variables) there exist lower bounds that require the algorithm to have exponential running time ( $\Omega(2^{\Omega(n)})$ ) (e.g. Klee Minty 1972).

## Remarks about Simplex

### Theorem

For some standard **randomized** pivoting rules there exist subexponential lower bounds ( $\Omega(2^{\Omega(n^\alpha)})$  for  $\alpha > 0$ ) (Friedmann, Hansen, Zwick 2011).

## Remarks about Simplex

### Conjecture (Hirsch 1957)

The edge-vertex graph of an  $m$ -facet polytope in  $d$ -dimensional Euclidean space has diameter no more than  $m - d$ .

The conjecture has been proven wrong in 2010.

But the question whether the diameter is perhaps of the form  $\mathcal{O}(\text{poly}(m, d))$  is open.

## 8 Seidels LP-algorithm

- ▶ Suppose we want to solve  $\min\{c^t x \mid Ax \geq b; x \geq 0\}$ , where  $x \in \mathbb{R}^d$  and we have  $m$  constraints.
- ▶ In the worst-case Simplex runs in time roughly  $\mathcal{O}(m(m+d)\binom{m+d}{m}) \approx (m+d)^m$ . (slightly better bounds on the running time exist, but will not be discussed here).
- ▶ If  $d$  is much smaller than  $m$  one can do a lot better.
- ▶ In the following we develop an algorithm with running time  $\mathcal{O}(d! \cdot m)$ , i.e., **linear in  $m$** .

## 8 Seidels LP-algorithm

### Setting:

- ▶ We assume an LP of the form

$$\begin{array}{ll} \min & c^t x \\ \text{s.t.} & Ax \geq b \\ & x \geq 0 \end{array}$$

- ▶ We assume that the LP is **bounded**.

## Ensuring Conditions

Given a **standard minimization LP**

$$\begin{array}{ll} \min & c^t x \\ \text{s.t.} & Ax \geq b \\ & x \geq 0 \end{array}$$

how can we obtain an LP of the required form?

- ▶ **Compute a lower bound on  $c^t x$  for any basic feasible solution.**

## Computing a Lower Bound

Let  $s$  denote the smallest common multiple of all denominators of entries in  $A, b$ .

Multiply entries in  $A, b$  by  $s$  to obtain integral entries. **This does not change the feasible region.**

Add slack variables to  $A$ ; denote the resulting matrix with  $\bar{A}$ .

If  $B$  is an optimal basis then  $x_B$  with  $\bar{A}_B x_B = b$ , gives an optimal assignment to the basis variables (non-basic variables are 0).

## Theorem 31 (Cramers Rule)

Let  $M$  be a matrix with  $\det(M) \neq 0$ . Then the solution to the system  $Mx = b$  is given by

$$x_j = \frac{\det(M_j)}{\det(M)},$$

where  $M_j$  is the matrix obtained from  $M$  by replacing the  $j$ -th column by the vector  $b$ .

### Proof:

- ▶ Define

$$X_j = \begin{pmatrix} | & & | & | & | \\ e_1 & \cdots & e_{j-1} & x & e_{j+1} & \cdots & e_n \\ | & & | & | & | \end{pmatrix}$$

Note that expanding along the  $j$ -th column gives that  $\det(X_j) = x_j$ .

- ▶ Further, we have

$$MX_j = \begin{pmatrix} | & & | & | & | \\ Me_1 & \cdots & Me_{j-1} & Mx & Me_{j+1} & \cdots & Men \\ | & & | & | & | \end{pmatrix} = M_j$$

- ▶ Hence,

$$x_j = \det(X_j) = \frac{\det(M_j)}{\det(M)}$$

## Bounding the Determinant

Let  $Z$  be the maximum absolute entry occurring in  $\bar{A}, \bar{b}$  or  $c$ . Let  $C$  denote the matrix obtained from  $\bar{A}_B$  by replacing the  $j$ -th column with vector  $\bar{b}$ .

Observe that

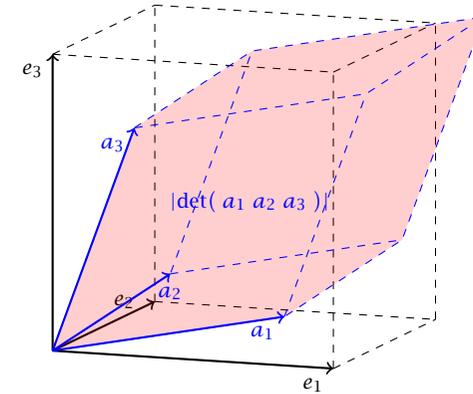
$$\begin{aligned} |\det(C)| &= \left| \sum_{\pi \in S_m} \prod_{1 \leq i \leq m} \operatorname{sgn}(\pi) C_{i\pi(i)} \right| \\ &\leq \sum_{\pi \in S_m} \prod_{1 \leq i \leq m} |C_{i\pi(i)}| \\ &\leq m! \cdot Z^m. \end{aligned}$$

## Bounding the Determinant

Alternatively, Hadamard's inequality gives

$$|\det(C)| \leq \prod_{i=1}^m \|C_{*i}\| \leq \prod_{i=1}^m (\sqrt{m}Z) \\ \leq m^{m/2} Z^m .$$

## Hadamard's Inequality



Hadamard's inequality says that the volume of the red parallelepiped (Spat) is smaller than the volume in the black cube (if  $\|e_1\| = \|a_1\|$ ,  $\|e_2\| = \|a_2\|$ ,  $\|e_3\| = \|a_3\|$ ).

## Ensuring Conditions

Given a **standard minimization LP**

$$\begin{array}{ll} \min & c^t x \\ \text{s.t.} & Ax \geq b \\ & x \geq 0 \end{array}$$

how can we obtain an LP of the required form?

- ▶ **Compute a lower bound on  $c^t x$  for any basic feasible solution.** Add the constraint  $c^t x \geq -mZ(m! \cdot Z^m) - 1$ . Note that this constraint is **superfluous unless the LP is unbounded.**

## Ensuring Conditions

Compute an optimum basis for the new LP.

- ▶ If the cost is  $c^t x = -(mZ)(m! \cdot Z^m) - 1$  we know that the original LP is unbounded.
- ▶ Otw. we have an optimum basis.

In the following we use  $\mathcal{H}$  to denote the set of all constraints apart from the constraint  $c^t x \geq -mZ(m! \cdot Z^m) - 1$ .

We give a routine  $\text{SeidelLP}(\mathcal{H}, d)$  that is given a set  $\mathcal{H}$  of **explicit, non-degenerate** constraints over  $d$  variables, and minimizes  $c^t x$  over all feasible points.

In addition it obeys the implicit constraint  $c^t x \geq -(mZ)(m! \cdot Z^m) - 1$ .

#### Algorithm 1 $\text{SeidelLP}(\mathcal{H}, d)$

```

1: if  $d = 1$  then solve 1-dimensional problem and return;
2: if  $\mathcal{H} = \emptyset$  then return  $x$  on implicit constraint hyperplane
3: choose random constraint  $h \in \mathcal{H}$ 
4:  $\hat{\mathcal{H}} \leftarrow \mathcal{H} \setminus \{h\}$ 
5:  $\hat{x}^* \leftarrow \text{SeidelLP}(\hat{\mathcal{H}}, d)$ 
6: if  $\hat{x}^* = \text{infeasible}$  then return infeasible
7: if  $\hat{x}^*$  fulfills  $h$  then return  $\hat{x}^*$ 
8: // optimal solution fulfills  $h$  with equality, i.e.,  $A_h x = b_h$ 
9: solve  $A_h x = b_h$  for some variable  $x_\ell$ ;
10: eliminate  $x_\ell$  in constraints from  $\hat{\mathcal{H}}$  and in implicit constr.;
11:  $\hat{x}^* \leftarrow \text{SeidelLP}(\hat{\mathcal{H}}, d - 1)$ 
12: if  $\hat{x}^* = \text{infeasible}$  then
13:   return infeasible
14: else
15:   add the value of  $x_\ell$  to  $\hat{x}^*$  and return the solution

```

## 8 Seidels LP-algorithm

- ▶ If  $d = 1$  we can solve the 1-dimensional problem in time  $\mathcal{O}(m)$ .
- ▶ If  $d > 1$  and  $m = 0$  we take time  $\mathcal{O}(d)$  to return  $d$ -dimensional vector  $x$ .
- ▶ The first recursive call takes time  $T(m - 1, d)$  for the call plus  $\mathcal{O}(d)$  for checking whether the solution fulfills  $h$ .
- ▶ If we are unlucky and  $\hat{x}^*$  does not fulfill  $h$  we need time  $\mathcal{O}(d(m + 1)) = \mathcal{O}(dm)$  to eliminate  $x_\ell$ . Then we make a recursive call that takes time  $T(m - 1, d - 1)$ .
- ▶ The probability of being unlucky is at most  $d/m$  as there are at most  $d$  constraints whose removal will decrease the objective function

## 8 Seidels LP-algorithm

This gives the recurrence

$$T(m, d) = \begin{cases} \mathcal{O}(m) & \text{if } d = 1 \\ \mathcal{O}(d) & \text{if } d > 1 \text{ and } m = 0 \\ \mathcal{O}(d) + T(m - 1, d) + \\ \frac{d}{m} (\mathcal{O}(dm) + T(m - 1, d - 1)) & \text{otw.} \end{cases}$$

Note that  $T(m, d)$  denotes the **expected running time**.

## 8 Seidels LP-algorithm

Let  $C$  be the largest constant in the  $\mathcal{O}$ -notations.

$$T(m, d) = \begin{cases} Cm & \text{if } d = 1 \\ Cd & \text{if } d > 1 \text{ and } m = 0 \\ Cd + T(m-1, d) + \frac{d}{m}(Cdm + T(m-1, d-1)) & \text{otw.} \end{cases}$$

Note that  $T(m, d)$  denotes the **expected running time**.

## 8 Seidels LP-algorithm

Let  $C$  be the largest constant in the  $\mathcal{O}$ -notations.

We show  $T(m, d) \leq Cf(d) \max\{1, m\}$ .

**$d = 1$ :**

$$T(m, 1) \leq Cm \leq Cf(1) \max\{1, m\} \text{ for } f(1) \geq 1$$

**$d > 1; m = 0$ :**

$$T(0, d) \leq \mathcal{O}(d) \leq Cd \leq Cf(d) \max\{1, m\} \text{ for } f(d) \geq d$$

**$d > 1; m = 1$ :**

$$\begin{aligned} T(1, d) &= \mathcal{O}(d) + T(0, d) + d(\mathcal{O}(d) + T(0, d-1)) \\ &\leq Cd + Cd + Cd^2 + dCf(d-1) \\ &\leq Cf(d) \max\{1, m\} \text{ for } f(d) \geq 3d^2 + df(d-1) \end{aligned}$$

## 8 Seidels LP-algorithm

**$d > 1; m > 1$ :**

(by induction hypothesis statm. true for  $d' < d, m' \geq 0$ ;  
and for  $d' = d, m' < m$ )

$$\begin{aligned} T(m, d) &= \mathcal{O}(d) + T(m-1, d) + \frac{d}{m}(\mathcal{O}(dm) + T(m-1, d-1)) \\ &\leq Cd + Cf(d)(m-1) + Cd^2 + \frac{d}{m}Cf(d-1)(m-1) \\ &\leq 2Cd^2 + Cf(d)(m-1) + dCf(d-1) \\ &\leq Cf(d)m \end{aligned}$$

**if  $f(d) \geq df(d-1) + 2d^2$ .**

## 8 Seidels LP-algorithm

► Define  $f(1) = 3 \cdot 1^2$  and  $f(d) = df(d-1) + 3d^2$  for  $d > 1$ .

Then

$$\begin{aligned} f(d) &= 3d^2 + df(d-1) \\ &= 3d^2 + d[3(d-1)^2 + (d-1)f(d-2)] \\ &= 3d^2 + d[3(d-1)^2 + (d-1)[3(d-2)^2 + (d-2)f(d-3)]] \\ &= 3d^2 + 3d(d-1)^2 + 3d(d-1)(d-2)^2 + \dots \\ &\quad + 3d(d-1)(d-2) \cdot \dots \cdot 4 \cdot 3 \cdot 1^2 \\ &= 3d! \left( \frac{d^2}{d!} + \frac{(d-1)^2}{(d-1)!} + \frac{(d-2)^2}{(d-2)!} + \dots \right) \\ &= \mathcal{O}(d!) \end{aligned}$$

since  $\sum_{i \geq 1} \frac{i^2}{i!}$  is a constant.

## Complexity

### LP Feasibility Problem (LP feasibility)

- ▶ Given  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$ . Does there exist  $x \in \mathbb{R}$  with  $Ax = b$ ,  $x \geq 0$ ?
- ▶ Note that allowing  $A, b$  to contain rational numbers does not make a difference, as we can multiply every number by a suitable large constant so that everything becomes integral but the **feasible region** does not change.

Is this problem in NP or even in P?

## The Bit Model

### Input size

- ▶ The number of bits to represent a number  $a \in \mathbb{Z}$  is

$$\lceil \log_2(|a|) \rceil + 1$$

- ▶ Let for an  $m \times n$  matrix  $M$ ,  $L(M)$  denote the number of bits required to encode all the numbers in  $M$ .

$$L(M) := \sum_{i,j} \lceil \log_2(|m_{ij}|) + 1 \rceil$$

- ▶ In the following we assume that input matrices are encoded in a standard way, where each number is encoded in binary and then suitable separators are added in order to separate distinct number from each other.
- ▶ Then the input length is  $\Theta(L([A|b]))$ .

- ▶ In the following we sometimes refer to  $L := L([A|b])$  as the input size (even though the real input size is something in  $\Theta(L([A|b]))$ ).
- ▶ In order to show that LP-decision is in NP we show that if there is a solution  $x$  then there exists a small solution for which feasibility can be verified in polynomial time (polynomial in  $L([A|b])$ ).

Suppose that  $Ax = b$ ;  $x \geq 0$  is feasible.

Then there exists a basic feasible solution. This means a set  $B$  of basic variables such that

$$x_B = A_B^{-1}b$$

and all other entries in  $x$  are 0.

## Size of a Basic Feasible Solution

### Lemma 32

Let  $M \in \mathbb{Z}^{m \times m}$  be an invertible matrix and let  $b \in \mathbb{Z}^m$ . Further define  $L' = L([M \mid b]) + n \log_2 n$ . Then a solution to  $Mx = b$  has rational components  $x_j$  of the form  $\frac{D_j}{D}$ , where  $|D_j| \leq 2^{L'}$  and  $|D| \leq 2^{L'}$ .

### Proof:

Cramer's rule says that we can compute  $x_j$  as

$$x_j = \frac{\det(M_j)}{\det(M)}$$

where  $M_j$  is the matrix obtained from  $M$  by replacing the  $j$ -th column by the vector  $b$ .

## Bounding the Determinant

Let  $X = A_B$ . Then

$$\begin{aligned} |\det(X)| &= \left| \sum_{\pi \in \mathcal{S}_n} \prod_{1 \leq i \leq n} \operatorname{sgn}(\pi) X_{i\pi(i)} \right| \\ &\leq \sum_{\pi \in \mathcal{S}_n} \prod_{1 \leq i \leq n} |X_{i\pi(i)}| \\ &\leq n! \cdot 2^{L([A|b])} \leq n^n 2^L \leq 2^{L'} . \end{aligned}$$

Analogously for  $\det(M_j)$ .

This means if  $Ax = b$ ,  $x \geq 0$  is feasible we only need to consider vectors  $x$  where an entry  $x_j$  can be represented by a rational number with encoding length polynomial in the input length  $L$ .

Hence, the  $x$  that we have to guess is of length polynomial in the input-length  $L$ .

For a given vector  $x$  of polynomial length we can check for feasibility in polynomial time.

Hence, LP feasibility is in NP.

## Reducing LP-solving to LP decision.

Given an LP  $\max\{c^t x \mid Ax = b; x \geq 0\}$  do a **binary search** for the optimum solution

(Add constraint  $c^t x - \delta = M$ ;  $\delta \geq 0$  or  $(c^t x \geq M)$ . Then checking for feasibility shows whether optimum solution is larger or smaller than  $M$ ).

If the LP is feasible then the binary search finishes in at most

$$\log_2 \left( \frac{2n2^{2L'}}{1/2^{L'}} \right) = \mathcal{O}(L') ,$$

as the range of the search is at most  $-n2^{2L'}, \dots, n2^{2L'}$  and the distance between two adjacent values is at least  $\frac{1}{\det(A)} \geq \frac{1}{2^{L'}}$ .

Here we use  $L' = L([A \mid b \mid c]) + n \log_2 n$  (it also includes the encoding size of  $c$ ).

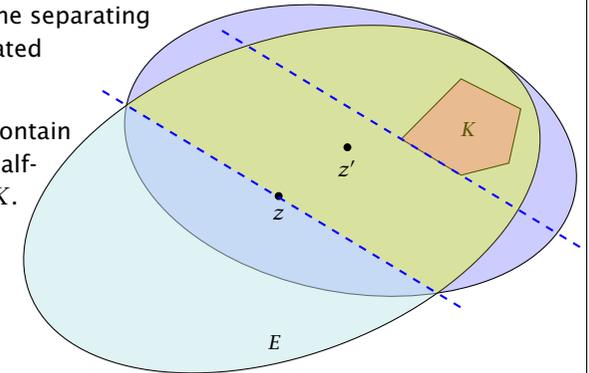
### How do we detect whether the LP is unbounded?

Let  $M_{\max} = n2^{2L'}$  be an upper bound on the objective value of a **basic feasible solution**.

We can add a constraint  $c^t x \geq M_{\max} + 1$  and check for feasibility.

## Ellipsoid Method

- ▶ Let  $K$  be a convex set.
- ▶ Maintain ellipsoid  $E$  that is guaranteed to contain  $K$  provided that  $K$  is non-empty.
- ▶ If center  $z \in K$  STOP.
- ▶ Otw. find a hyperplane separating  $K$  from  $z$  (e.g. a violated constraint in the LP).
- ▶ Shift hyperplane to contain node  $z'$ .  $H$  denotes half-space that contains  $K$ .
- ▶ Compute (smallest) ellipsoid  $E'$  that contains  $K \cap H$ .
- ▶ REPEAT



### Issues/Questions:

- ▶ How do you choose the first Ellipsoid? What is its volume?
- ▶ What if the polytop  $K$  is unbounded?
- ▶ How do you measure progress? By how much does the volume decrease in each iteration?
- ▶ When can you stop? What is the minimum volume of a non-empty polytop?

### Definition 33

A mapping  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  with  $f(x) = Lx + t$ , where  $L$  is an invertible matrix is called an **affine transformation**.

### Definition 34

A ball in  $\mathbb{R}^n$  with center  $c$  and radius  $r$  is given by

$$\begin{aligned} B(c, r) &= \{x \mid (x - c)^t(x - c) \leq r^2\} \\ &= \{x \mid \sum_i (x - c)_i^2 / r^2 \leq 1\} \end{aligned}$$

$B(0, 1)$  is called the **unit ball**.

### Definition 35

An affine transformation of the unit ball is called an **ellipsoid**.

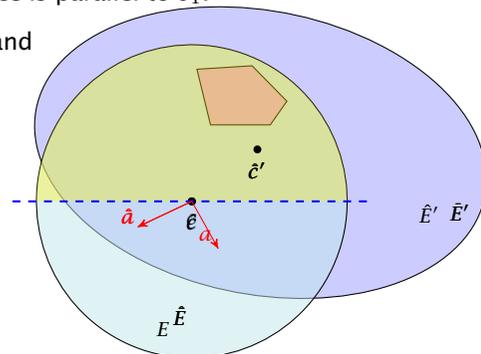
From  $f(x) = Lx + t$  follows  $x = L^{-1}(f(x) - t)$ .

$$\begin{aligned} f(B(0, 1)) &= \{f(x) \mid x \in B(0, 1)\} \\ &= \{y \in \mathbb{R}^n \mid L^{-1}(y - t) \in B(0, 1)\} \\ &= \{y \in \mathbb{R}^n \mid (y - t)^t L^{-1t} L^{-1}(y - t) \leq 1\} \\ &= \{y \in \mathbb{R}^n \mid (y - t)^t Q^{-1}(y - t) \leq 1\} \end{aligned}$$

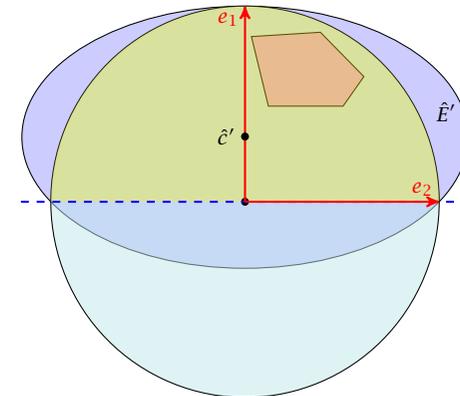
where  $Q = LL^t$  is an invertible matrix.

## How to Compute the New Ellipsoid

- ▶ Use  $f^{-1}$  (recall that  $f = Lx + t$  is the affine transformation of the unit ball) to rotate/distort the ellipsoid (back) into the unit ball.
- ▶ Use a rotation  $R^{-1}$  to rotate the unit ball such that the normal vector of the halfspace is parallel to  $e_1$ .
- ▶ Compute the new center  $\hat{c}'$  and the new matrix  $\hat{Q}'$  for this simplified setting.
- ▶ Use the transformations  $R$  and  $f$  to get the new center  $c'$  and the new matrix  $Q'$  for the original ellipsoid  $E$ .



## The Easy Case



- ▶ The new center lies on axis  $x_1$ . Hence,  $\hat{c}' = te_1$  for  $t > 0$ .
- ▶ The vectors  $e_1, e_2, \dots$  have to fulfill the ellipsoid constraint with equality. Hence  $(e_i - \hat{c}')^t \hat{Q}'^{-1}(e_i - \hat{c}') = 1$ .

## The Easy Case

- ▶ To obtain the matrix  $\hat{Q}'^{-1}$  for our ellipsoid  $\hat{E}'$  note that  $\hat{E}'$  is **axis-parallel**.
- ▶ Let  $a$  denote the radius along the  $x_1$ -axis and let  $b$  denote the (common) radius for the other axes.
- ▶ The matrix

$$\hat{L}' = \begin{pmatrix} a & 0 & \dots & 0 \\ 0 & b & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & b \end{pmatrix}$$

maps the unit ball (via function  $\hat{f}'(x) = \hat{L}'x$ ) to an axis-parallel ellipsoid with radius  $a$  in direction  $x_1$  and  $b$  in all other directions.

## The Easy Case

- ▶ As  $\hat{Q}' = \hat{L}'\hat{L}'^t$  the matrix  $\hat{Q}'^{-1}$  is of the form

$$\hat{Q}'^{-1} = \begin{pmatrix} \frac{1}{a^2} & 0 & \dots & 0 \\ 0 & \frac{1}{b^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{b^2} \end{pmatrix}$$

## The Easy Case

- ▶  $(e_1 - \hat{c}')^t \hat{Q}'^{-1} (e_1 - \hat{c}') = 1$  gives

$$\begin{pmatrix} 1-t \\ 0 \\ \vdots \\ 0 \end{pmatrix}^t \cdot \begin{pmatrix} \frac{1}{a^2} & 0 & \dots & 0 \\ 0 & \frac{1}{b^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{b^2} \end{pmatrix} \cdot \begin{pmatrix} 1-t \\ 0 \\ \vdots \\ 0 \end{pmatrix} = 1$$

- ▶ This gives  $(1-t)^2 = a^2$ .

## The Easy Case

- ▶ For  $i \neq 1$  the equation  $(e_i - \hat{c}')^t \hat{Q}'^{-1} (e_i - \hat{c}') = 1$  gives

$$\begin{pmatrix} -t \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}^t \cdot \begin{pmatrix} \frac{1}{a^2} & 0 & \dots & 0 \\ 0 & \frac{1}{b^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{b^2} \end{pmatrix} \cdot \begin{pmatrix} -t \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = 1$$

- ▶ This gives  $\frac{t^2}{a^2} + \frac{1}{b^2} = 1$ , and hence

$$\frac{1}{b^2} = 1 - \frac{t^2}{a^2} = 1 - \frac{t^2}{(1-t)^2} = \frac{1-2t}{(1-t)^2}$$

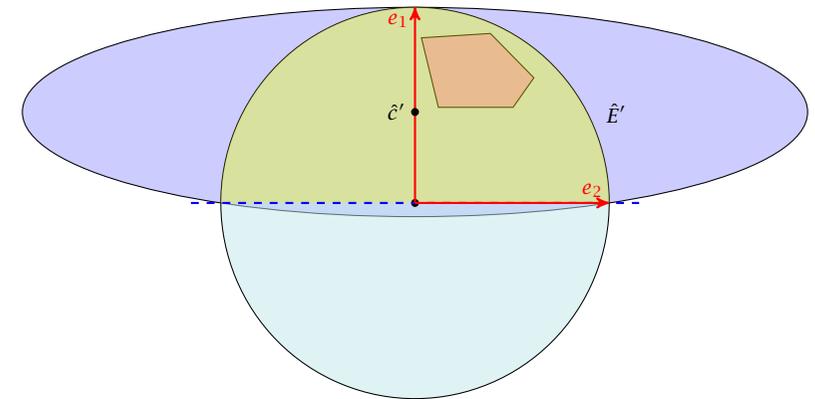
## Summary

So far we have

$$a = 1 - t \quad \text{and} \quad b = \frac{1 - t}{\sqrt{1 - 2t}}$$

## The Easy Case

We still have many choices for  $t$ :



Choose  $t$  such that the volume of  $\hat{E}'$  is minimal!!!

## The Easy Case

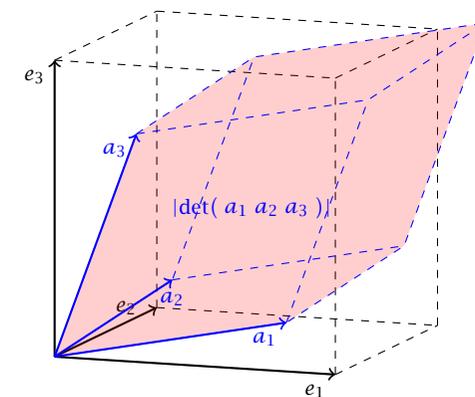
We want to choose  $t$  such that the volume of  $\hat{E}'$  is minimal.

### Lemma 36

Let  $L$  be an affine transformation and  $K \subseteq \mathbb{R}^n$ . Then

$$\text{vol}(L(K)) = |\det(L)| \cdot \text{vol}(K) .$$

## n-dimensional volume



## The Easy Case

- ▶ We want to choose  $t$  such that the volume of  $\hat{E}'$  is minimal.

$$\text{vol}(\hat{E}') = \text{vol}(B(0, 1)) \cdot |\det(\hat{L}')| ,$$

where  $\hat{Q}' = \hat{L}'\hat{L}'^t$ .

- ▶ We have

$$\hat{L}'^{-1} = \begin{pmatrix} \frac{1}{a} & 0 & \dots & 0 \\ 0 & \frac{1}{b} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{b} \end{pmatrix} \text{ and } \hat{L}' = \begin{pmatrix} a & 0 & \dots & 0 \\ 0 & b & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & b \end{pmatrix}$$

- ▶ Note that  $a$  and  $b$  in the above equations depend on  $t$ , by the previous equations.

## The Easy Case

$$\begin{aligned} \text{vol}(\hat{E}') &= \text{vol}(B(0, 1)) \cdot |\det(\hat{L}')| \\ &= \text{vol}(B(0, 1)) \cdot ab^{n-1} \\ &= \text{vol}(B(0, 1)) \cdot (1-t) \cdot \left(\frac{1-t}{\sqrt{1-2t}}\right)^{n-1} \\ &= \text{vol}(B(0, 1)) \cdot \frac{(1-t)^n}{(\sqrt{1-2t})^{n-1}} \end{aligned}$$

## The Easy Case

$$\begin{aligned} \frac{d \text{vol}(\hat{E}')}{dt} &= \frac{d}{dt} \left( \frac{(1-t)^n}{(\sqrt{1-2t})^{n-1}} \right) \\ &= \frac{1}{N^2} \cdot \left( \underbrace{(-1) \cdot n(1-t)^{n-1}}_{\text{derivative of numerator}} \cdot \underbrace{(\sqrt{1-2t})^{n-1}}_{\text{denominator}} \right) \\ &= \frac{1}{N^2} \cdot \underbrace{(\sqrt{1-2t})^{n-3}}_{\text{outer derivative}} \cdot \underbrace{\frac{1}{2\sqrt{1-2t}}}_{\text{inner derivative}} \cdot \underbrace{(-2)}_{\text{numerator}} \cdot (1-t)^n \\ &= \frac{1}{N^2} \cdot (\sqrt{1-2t})^{n-3} \cdot (1-t)^{n-1} \cdot \left( (n-1)(1-t) - n(1-2t) \right) \\ &= \frac{1}{N^2} \cdot (\sqrt{1-2t})^{n-3} \cdot (1-t)^{n-1} \cdot \left( (n+1)t - 1 \right) \end{aligned}$$

## The Easy Case

- ▶ We obtain the minimum for  $t = \frac{1}{n+1}$ .
- ▶ For this value we obtain

$$a = 1 - t = \frac{n}{n+1} \text{ and } b = \frac{1-t}{\sqrt{1-2t}} = \frac{n}{\sqrt{n^2-1}}$$

To see the equation for  $b$ , observe that

$$b^2 = \frac{(1-t)^2}{1-2t} = \frac{\left(1 - \frac{1}{n+1}\right)^2}{1 - \frac{2}{n+1}} = \frac{\left(\frac{n}{n+1}\right)^2}{\frac{n-1}{n+1}} = \frac{n^2}{n^2-1}$$

## The Easy Case

Let  $\gamma_n = \frac{\text{vol}(\hat{E}')}{\text{vol}(B(0,1))} = ab^{n-1}$  be the ratio by which the volume changes:

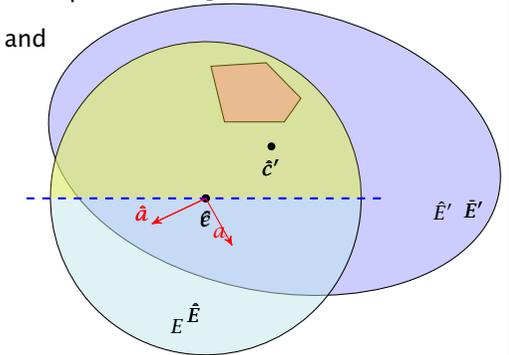
$$\begin{aligned} \gamma_n^2 &= \left(\frac{n}{n+1}\right)^2 \left(\frac{n^2}{n^2-1}\right)^{n-1} \\ &= \left(1 - \frac{1}{n+1}\right)^2 \left(1 + \frac{1}{(n-1)(n+1)}\right)^{n-1} \\ &\leq e^{-2\frac{1}{n+1}} \cdot e^{\frac{1}{n+1}} \\ &= e^{-\frac{1}{n+1}} \end{aligned}$$

where we used  $(1+x)^a \leq e^{ax}$  for  $x \in \mathbb{R}$  and  $a > 0$ .

This gives  $\gamma_n \leq e^{-\frac{1}{2(n+1)}}$ .

## How to Compute the New Ellipsoid

- ▶ Use  $f^{-1}$  (recall that  $f = Lx + t$  is the affine transformation of the unit ball) to rotate/distort the ellipsoid (back) into the unit ball.
- ▶ Use a rotation  $R^{-1}$  to rotate the unit ball such that the normal vector of the halfspace is parallel to  $e_1$ .
- ▶ Compute the new center  $\hat{c}'$  and the new matrix  $\hat{Q}'$  for this simplified setting.
- ▶ Use the transformations  $R$  and  $f$  to get the new center  $c'$  and the new matrix  $Q'$  for the original ellipsoid  $E$ .



Our progress is the same:

$$\begin{aligned} e^{-\frac{1}{2(n+1)}} &\geq \frac{\text{vol}(\hat{E}')}{\text{vol}(B(0,1))} = \frac{\text{vol}(\hat{E}')}{\text{vol}(\hat{E})} = \frac{\text{vol}(R(\hat{E}'))}{\text{vol}(R(\hat{E}))} \\ &= \frac{\text{vol}(\hat{E}')}{\text{vol}(\hat{E})} = \frac{\text{vol}(f(\hat{E}'))}{\text{vol}(f(\hat{E}))} = \frac{\text{vol}(E')}{\text{vol}(E)} \end{aligned}$$

Here it is important that mapping a set with affine function  $f(x) = Lx + t$  changes the volume by factor  $\det(L)$ .

## The Ellipsoid Algorithm

**How to Compute The New Parameters?**

The transformation function of the (old) ellipsoid:  $f(x) = Lx + c$ ;

The halfspace to be intersected:  $H = \{x \mid a^t(x - c) \leq 0\}$ ;

$$\begin{aligned} f^{-1}(H) &= \{f^{-1}(x) \mid a^t(x - c) \leq 0\} \\ &= \{f^{-1}(f(y)) \mid a^t(f(y) - c) \leq 0\} \\ &= \{y \mid a^t(f(y) - c) \leq 0\} \\ &= \{y \mid a^t(Ly + c - c) \leq 0\} \\ &= \{y \mid (a^tL)y \leq 0\} \end{aligned}$$

This means  $\bar{a} = L^t a$ .

## The Ellipsoid Algorithm

After rotating back (applying  $R^{-1}$ ) the normal vector of the halfspace points in negative  $x_1$ -direction. Hence,

$$R^{-1}\left(\frac{L^t a}{\|L^t a\|}\right) = -e_1 \quad \Rightarrow \quad -\frac{L^t a}{\|L^t a\|} = R \cdot e_1$$

Hence,

$$\bar{c}' = R \cdot \hat{c}' = R \cdot \frac{1}{n+1} e_1 = -\frac{1}{n+1} \frac{L^t a}{\|L^t a\|}$$

$$\begin{aligned} c' &= f(\bar{c}') = L \cdot \bar{c}' + c \\ &= -\frac{1}{n+1} L \frac{L^t a}{\|L^t a\|} + c \\ &= c - \frac{1}{n+1} \frac{Qa}{\sqrt{a^t Q a}} \end{aligned}$$

For computing the matrix  $Q'$  of the new ellipsoid we assume in the following that  $\hat{E}'$ ,  $\bar{E}'$  and  $E'$  refer to the ellipsoids centered in the origin.

Recall that

$$\hat{Q}' = \begin{pmatrix} a^2 & 0 & \dots & 0 \\ 0 & b^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & b^2 \end{pmatrix}$$

This gives

$$\hat{Q}' = \frac{n^2}{n^2 - 1} \left( I - \frac{2}{n+1} e_1 e_1^t \right)$$

because for  $a = n/n+1$  and  $b = n/\sqrt{n^2-1}$

$$\begin{aligned} b^2 - b^2 \frac{2}{n+1} &= \frac{n^2}{n^2 - 1} - \frac{2n^2}{(n-1)(n+1)^2} \\ &= \frac{n^2(n+1) - 2n^2}{(n-1)(n+1)^2} = \frac{n^2(n-1)}{(n-1)(n+1)^2} = a^2 \end{aligned}$$

## 9 The Ellipsoid Algorithm

$$\begin{aligned} \bar{E}' &= R(\hat{E}') \\ &= \{R(x) \mid x^t \hat{Q}'^{-1} x \leq 1\} \\ &= \{y \mid (R^{-1}y)^t \hat{Q}'^{-1} R^{-1}y \leq 1\} \\ &= \{y \mid y^t (R^t)^{-1} \hat{Q}'^{-1} R^{-1}y \leq 1\} \\ &= \{y \mid y^t \underbrace{(R \hat{Q}' R^t)^{-1}}_{\hat{Q}'} y \leq 1\} \end{aligned}$$

## 9 The Ellipsoid Algorithm

Hence,

$$\begin{aligned}\bar{Q}' &= R\hat{Q}'R^t \\ &= R \cdot \frac{n^2}{n^2-1} \left( I - \frac{2}{n+1} e_1 e_1^t \right) \cdot R^t \\ &= \frac{n^2}{n^2-1} \left( R \cdot R^t - \frac{2}{n+1} (R e_1)(R e_1)^t \right) \\ &= \frac{n^2}{n^2-1} \left( I - \frac{2}{n+1} \frac{L^t a a^t L}{\|L^t a\|^2} \right)\end{aligned}$$

## 9 The Ellipsoid Algorithm

$$\begin{aligned}E' &= L(\bar{E}') \\ &= \{L(x) \mid x^t \bar{Q}'^{-1} x \leq 1\} \\ &= \{y \mid (L^{-1}y)^t \bar{Q}'^{-1} L^{-1}y \leq 1\} \\ &= \{y \mid y^t (L^t)^{-1} \bar{Q}'^{-1} L^{-1}y \leq 1\} \\ &= \{y \mid y^t \underbrace{(L\bar{Q}'L^t)^{-1}}_{Q'} y \leq 1\}\end{aligned}$$

## 9 The Ellipsoid Algorithm

Hence,

$$\begin{aligned}Q' &= L\bar{Q}'L^t \\ &= L \cdot \frac{n^2}{n^2-1} \left( I - \frac{2}{n+1} \frac{L^t a a^t L}{a^t Q a} \right) \cdot L^t \\ &= \frac{n^2}{n^2-1} \left( Q - \frac{2}{n+1} \frac{Q a a^t Q}{a^t Q a} \right)\end{aligned}$$

## Incomplete Algorithm

### Algorithm 1 ellipsoid-algorithm

- 1: **input:** point  $c \in \mathbb{R}^n$ , convex set  $K \subseteq \mathbb{R}^n$
- 2: **output:** point  $x \in K$  or “ $K$  is empty”
- 3:  $Q \leftarrow ???$
- 4: **repeat**
- 5:     **if**  $c \in K$  **then return**  $c$
- 6:     **else**
- 7:         choose a violated hyperplane  $a$
- 8:          $c \leftarrow c - \frac{1}{n+1} \frac{Qa}{\sqrt{a^t Q a}}$
- 9:          $Q \leftarrow \frac{n^2}{n^2-1} \left( Q - \frac{2}{n+1} \frac{Q a a^t Q}{a^t Q a} \right)$
- 10:     **endif**
- 11: **until**  $???$
- 12: **return** “ $K$  is empty”

## Repeat: Size of basic solutions

### Lemma 37

Let  $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$  be a bounded polytop. Let  $\langle a_{\max} \rangle$  be the maximum encoding length of an entry in  $A, b$ . Then every entry  $x_j$  in a basic solution fulfills  $|x_j| = \frac{D_j}{D}$  with  $D_j, D \leq 2^{2n\langle a_{\max} \rangle + 2n \log_2 n}$ .

In the following we use  $\delta := 2^{2n\langle a_{\max} \rangle + 2n \log_2 n}$ .

Note that here we have  $P = \{x \mid Ax \leq b\}$ . The previous lemmas we had about the size of feasible solutions were slightly different as they were for different polytopes.

## Repeat: Size of basic solutions

### Proof:

Let  $\bar{A} = \begin{bmatrix} A & -A & I_m \\ -A & A & I_m \end{bmatrix}$ ,  $\bar{b} = \begin{pmatrix} b \\ -b \end{pmatrix}$ , be the matrix and right-hand vector after transforming the system to standard form.

The determinant of the matrices  $\bar{A}_B$  and  $\bar{M}_j$  (matrix obt. when replacing the  $j$ -th column of  $\bar{A}_B$  by  $\bar{b}$ ) can become at most

$$\det(\bar{A}_B), \det(\bar{M}_j) \leq \|\vec{\ell}_{\max}\|^{2n} \leq (\sqrt{2n} \cdot 2^{\langle a_{\max} \rangle})^{2n} \leq 2^{2n\langle a_{\max} \rangle + 2n \log_2 n},$$

where  $\vec{\ell}_{\max}$  is the longest column-vector that can be obtained after deleting all but  $2n$  rows and columns from  $\bar{A}$ .

This holds because columns from  $I_m$  selected when going from  $\bar{A}$  to  $\bar{A}_B$  do not increase the determinant. Only the at most  $2n$  columns from matrices  $A$  and  $-A$  that  $\bar{A}$  consists of contribute.

## How do we find the first ellipsoid?

For feasibility checking we can assume that the polytop  $P$  is bounded; it is sufficient to consider basic solutions.

Every entry  $x_i$  in a basic solution fulfills  $|x_i| \leq \delta$ .

Hence,  $P$  is contained in the cube  $-\delta \leq x_i \leq \delta$ .

A vector in this cube has at most distance  $R := \sqrt{n}\delta$  from the origin.

Starting with the ball  $E_0 := B(0, R)$  ensures that  $P$  is completely contained in the initial ellipsoid. This ellipsoid has volume at most  $R^n B(0, 1) \leq (n\delta)^n B(0, 1)$ .

## When can we terminate?

Let  $P := \{x \mid Ax \leq b\}$  with  $A \in \mathbb{Z}$  and  $b \in \mathbb{Z}$  be a bounded polytop. Let  $\langle a_{\max} \rangle$  be the encoding length of the largest entry in  $A$  or  $b$ .

Consider the following polytope

$$P_\lambda := \left\{ x \mid Ax \leq b + \frac{1}{\lambda} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \right\},$$

where  $\lambda = \delta^2 + 1$ .

### Lemma 38

$P_\lambda$  is feasible if and only if  $P$  is feasible.

$\Leftarrow$ : obvious!

$\Rightarrow$ :

Consider the polytopes

$$\bar{P} = \left\{ x \mid \begin{bmatrix} A & -A & I_m \\ -A & A & I_m \end{bmatrix} x = \begin{pmatrix} b \\ -b \end{pmatrix}; x \geq 0 \right\}$$

and

$$\bar{P}_\lambda = \left\{ x \mid \begin{bmatrix} A & -A & I_m \\ -A & A & I_m \end{bmatrix} x = \begin{pmatrix} b \\ -b \end{pmatrix} + \frac{1}{\lambda} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}; x \geq 0 \right\} .$$

$P$  is feasible if and only if  $\bar{P}$  is feasible, and  $P_\lambda$  feasible if and only if  $\bar{P}_\lambda$  feasible.

$\bar{P}_\lambda$  is bounded since  $P_\lambda$  and  $P$  are bounded.

$$\text{Let } \bar{A} = \begin{bmatrix} A & -A & I_m \\ -A & A & I_m \end{bmatrix}, \text{ and } \bar{b} = \begin{pmatrix} b \\ -b \end{pmatrix}.$$

$\bar{P}_\lambda$  feasible implies that there is a basic feasible solution represented by

$$x_B = \bar{A}_B^{-1} \bar{b} + \frac{1}{\lambda} \bar{A}_B^{-1} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

(The other  $x$ -values are zero)

The only reason that this basic feasible solution is not feasible for  $\bar{P}$  is that one of the basic variables becomes negative.

Hence, there exists  $i$  with

$$(\bar{A}_B^{-1} \bar{b})_i < 0 \leq (\bar{A}_B^{-1} \bar{b})_i + \frac{1}{\lambda} (\bar{A}_B^{-1} \bar{1})_i$$

By Cramers rule we get

$$(\bar{A}_B^{-1} \bar{b})_i < 0 \implies (\bar{A}_B^{-1} \bar{b})_i \leq -\frac{1}{\det(\bar{A}_B)}$$

and

$$(\bar{A}_B^{-1} \bar{1})_i \leq \det(\bar{M}_j) ,$$

where  $\bar{M}_j$  is obtained by replacing the  $j$ -th column of  $\bar{A}_B$  by  $\bar{1}$ .

However, we showed that the determinants of  $\bar{A}_B$  and  $\bar{M}_j$  can become at most  $\delta$ .

Since, we chose  $\lambda = \delta^2 + 1$  this gives a contradiction.

### Lemma 39

If  $P_\lambda$  is feasible then it contains a ball of radius  $r := 1/\delta^3$ . This has a volume of at least  $r^n \text{vol}(B(0, 1)) = \frac{1}{\delta^{3n}} \text{vol}(B(0, 1))$ .

#### Proof:

If  $P_\lambda$  feasible then also  $P$ . Let  $x$  be feasible for  $P$ .

This means  $Ax \leq b$ .

Let  $\vec{\ell}$  with  $\|\vec{\ell}\| \leq r$ . Then

$$\begin{aligned}(A(x + \vec{\ell}))_i &= (Ax)_i + (A\vec{\ell})_i \leq b_i + A_i \vec{\ell} \\ &\leq b_i + \|A_i\| \cdot \|\vec{\ell}\| \leq b_i + \sqrt{n} \cdot 2^{\langle a_{\max} \rangle} \cdot r \\ &\leq b_i + \frac{\sqrt{n} \cdot 2^{\langle a_{\max} \rangle}}{\delta^3} \leq b_i + \frac{1}{\delta^2 + 1} \leq b_i + \frac{1}{\lambda}\end{aligned}$$

Hence,  $x + \vec{\ell}$  is feasible for  $P_\lambda$  which proves the lemma.

How many iterations do we need until the volume becomes too small?

$$e^{-\frac{i}{2(n+1)}} \cdot \text{vol}(B(0, R)) < \text{vol}(B(0, r))$$

Hence,

$$\begin{aligned}i &> 2(n+1) \ln \left( \frac{\text{vol}(B(0, R))}{\text{vol}(B(0, r))} \right) \\ &= 2(n+1) \ln \left( n^n \delta^n \cdot \delta^{3n} \right) \\ &= 8n(n+1) \ln(\delta) + 2(n+1)n \ln(n) \\ &= \mathcal{O}(\text{poly}(n, \langle a_{\max} \rangle))\end{aligned}$$

### Algorithm 1 ellipsoid-algorithm

```
1: input: point  $c \in \mathbb{R}^n$ , convex set  $K \subseteq \mathbb{R}^n$ , radii  $R$  and  $r$ 
2:   with  $K \subseteq B(c, R)$ , and  $B(x, r) \subseteq K$  for some  $x$ 
3: output: point  $x \in K$  or “ $K$  is empty”
4:  $Q \leftarrow \text{diag}(R^2, \dots, R^2)$  // i.e.,  $L = \text{diag}(R, \dots, R)$ 
5: repeat
6:   if  $c \in K$  then return  $c$ 
7:   else
8:     choose a violated hyperplane  $a$ 
9:      $c \leftarrow c - \frac{1}{n+1} \frac{Qa}{\sqrt{a^t Q a}}$ 
10:     $Q \leftarrow \frac{n^2}{n^2-1} \left( Q - \frac{2}{n+1} \frac{Qaa^t Q}{a^t Q a} \right)$ 
11:   endif
12: until  $\det(Q) \leq r^{2n}$  // i.e.,  $\det(L) \leq r^n$ 
13: return “ $K$  is empty”
```

### Separation Oracle:

Let  $K \subseteq \mathbb{R}^n$  be a convex set. A separation oracle for  $K$  is an algorithm  $A$  that gets as input a point  $x \in \mathbb{R}^n$  and either

- ▶ certifies that  $x \in K$ ,
- ▶ or finds a hyperplane separating  $x$  from  $K$ .

We will usually assume that  $A$  is a polynomial-time algorithm.

In order to find a point in  $K$  we need

- ▶ a guarantee that a ball of radius  $r$  is contained in  $K$ ,
- ▶ an initial ball  $B(c, R)$  with radius  $R$  that contains  $K$ ,
- ▶ a separation oracle for  $K$ .

The Ellipsoid algorithm requires  $\mathcal{O}(\text{poly}(n) \cdot \log(R/r))$  iterations. Each iteration is polytime for a polynomial-time Separation oracle.

## 10 Karmarkars Algorithm

We want to solve the following linear program:

- ▶  $\min v = c^t x$  subject to  $Ax = 0$  and  $x \in \Delta$ .
- ▶ Here  $\Delta = \{x \in \mathbb{R}^n \mid e^t x = 1, x \geq 0\}$  with  $e^t = (1, \dots, 1)$  denotes the **standard simplex** in  $\mathbb{R}^n$ .

**Further assumptions:**

1.  $A$  is an  $m \times n$ -matrix with rank  $m$ .
2.  $Ae = 0$ , i.e., the center of the simplex is feasible.
3. The optimum solution is 0.

## 10 Karmarkars Algorithm

Suppose you start with  $\max\{c^t x \mid Ax = b; x \geq 0\}$ .

- ▶ Multiply  $c$  by  $-1$  and do a minimization.  $\Rightarrow$  **minimization problem**
- ▶ We can check for feasibility by using the two phase algorithm.  $\Rightarrow$  **can assume that LP is feasible.**
- ▶ Compute the dual; pack primal and dual into one LP and minimize the duality gap.  $\Rightarrow$  **optimum is 0**
- ▶ Add a new variable pair  $x_\ell, x'_\ell$  (both restricted to be positive) and the constraint  $\sum_i x_i = 1$ .  $\Rightarrow$  **solution in simplex**
- ▶ Add  $-(\sum_i x_i)b_i = -b_i$  to every constraint.  $\Rightarrow$  **vector  $b$  is 0**
- ▶ If  $A$  does not have full row rank we can delete constraints (or conclude that the LP is infeasible).  
 $\Rightarrow$   **$A$  has full row rank**

We still need to make  $e/n$  feasible.

## 10 Karmarkars Algorithm

The algorithm computes **strictly feasible** interior points

$$x^{(0)} = \frac{e}{n}, x^{(1)}, x^{(2)}, \dots \text{ with}$$

$$c^t x^{(k)} \leq 2^{-\Theta(L)} c^t x^{(0)}$$

For  $k = \Theta(L)$ . A point  $x$  is **strictly feasible** if  $x > 0$ .

If my objective value is close enough to 0 (the optimum!!) I can “snap” to an optimum vertex.

## 10 Karmarkars Algorithm

**Iteration:**

1. Distort the problem by mapping the simplex onto itself so that the **current point  $\bar{x}$**  moves to the center.
2. Project the optimization direction  $c$  onto the feasible region. Determine a distance to travel along this direction such that you do not leave the simplex (and you do not touch the border).  $\hat{x}_{\text{new}}$  is the point you reached.
3. Do a backtransformation to transform  $\hat{x}$  into your new point  $\bar{x}_{\text{new}}$ .

## The Transformation

Let  $\tilde{Y} = \text{diag}(\tilde{x})$  the diagonal matrix with entries  $\tilde{x}$  on the diagonal.

Define

$$F_{\tilde{x}} : x \mapsto \frac{\tilde{Y}^{-1}x}{e^t \tilde{Y}^{-1}x}.$$

The inverse function is

$$F_{\tilde{x}}^{-1} : \hat{x} \mapsto \frac{\tilde{Y}\hat{x}}{e^t \tilde{Y}\hat{x}}.$$

Note that  $\tilde{x} > 0$  in every coordinate. Therefore the above is well defined.

## Properties

$F_{\tilde{x}}^{-1}$  really is the inverse of  $F_{\tilde{x}}$ :

$$F_{\tilde{x}}(F_{\tilde{x}}^{-1}(\hat{x})) = \frac{\tilde{Y}^{-1} \frac{\tilde{Y}\hat{x}}{e^t \tilde{Y}\hat{x}}}{e^t \tilde{Y}^{-1} \frac{\tilde{Y}\hat{x}}{e^t \tilde{Y}\hat{x}}} = \frac{\hat{x}}{e^t \hat{x}} = \hat{x}$$

because  $\hat{x} \in \Delta$ .

Note that in particular every  $\hat{x} \in \Delta$  has a preimage (Urbild) under  $F_{\tilde{x}}$ .

## Properties

$\tilde{x}$  is mapped to  $e/n$

$$F_{\tilde{x}}(\tilde{x}) = \frac{\tilde{Y}^{-1}\tilde{x}}{e^t \tilde{Y}^{-1}\tilde{x}} = \frac{e}{e^t e} = \frac{e}{n}$$

## Properties

A unit vectors  $e_i$  is mapped to itself:

$$F_{\tilde{x}}(e_i) = \frac{\tilde{Y}^{-1}e_i}{e^t \tilde{Y}^{-1}e_i} = \frac{(0, \dots, 0, 1/\tilde{x}_i, 0, \dots, 0)^t}{e^t (0, \dots, 0, 1/\tilde{x}_i, 0, \dots, 0)^t} = e_i$$

## Properties

All nodes of the simplex are mapped to the simplex:

$$F_{\bar{x}}(x) = \frac{\bar{Y}^{-1}x}{e^t \bar{Y}^{-1}x} = \frac{\left(\frac{x_1}{\bar{x}_1}, \dots, \frac{x_n}{\bar{x}_n}\right)^t}{e^t \left(\frac{x_1}{\bar{x}_1}, \dots, \frac{x_n}{\bar{x}_n}\right)^t} = \frac{\left(\frac{x_1}{\bar{x}_1}, \dots, \frac{x_n}{\bar{x}_n}\right)^t}{\sum_i \frac{x_i}{\bar{x}_i}} \in \Delta$$

## The Transformation

Easy to check:

- ▶  $F_{\bar{x}}^{-1}$  really is the inverse of  $F_{\bar{x}}$ .
- ▶  $\bar{x}$  is mapped to  $e/n$ .
- ▶ A unit vectors  $e_i$  is mapped to itself.
- ▶ All nodes of the simplex are mapped to the simplex.

## 10 Karmarkars Algorithm

We have the problem

$$\begin{aligned} & \min\{c^t x \mid Ax = 0; x \in \Delta\} \\ & = \min\{c^t F_{\bar{x}}^{-1}(\hat{x}) \mid AF_{\bar{x}}^{-1}(\hat{x}) = 0; F_{\bar{x}}^{-1}(\hat{x}) \in \Delta\} \\ & = \min\{c^t F_{\bar{x}}^{-1}(\hat{x}) \mid AF_{\bar{x}}^{-1}(\hat{x}) = 0; \hat{x} \in \Delta\} \\ & = \min\left\{\frac{c^t \bar{Y} \hat{x}}{e^t \bar{Y} \hat{x}} \mid \frac{A \bar{Y} \hat{x}}{e^t \bar{Y} \hat{x}} = 0; \hat{x} \in \Delta\right\} \end{aligned}$$

Since the optimum solution is 0 this problem is the same as

$$\min\{\hat{c}^t \hat{x} \mid \hat{A} \hat{x} = 0, \hat{x} \in \Delta\}$$

with  $\hat{c} = \bar{Y}^t c = \bar{Y} c$  and  $\hat{A} = A \bar{Y}$ .

Note that  $e^t \bar{Y} x > 0$  for  $x \in \Delta$ .

We still need to make  $e/n$  feasible.

- ▶ We know that our LP is feasible. Let  $\bar{x}$  be a feasible point.
- ▶ Apply  $F_{\bar{x}}$ , and solve

$$\min\{\hat{c}^t x \mid \hat{A} x = 0; x \in \Delta\}$$

- ▶ The feasible point is moved to the center.

## 10 Karmarkars Algorithm

When computing  $\hat{x}_{\text{new}}$  we do not want to leave the simplex or touch its boundary (why?).

For this we compute the radius of a ball that completely lies in the simplex.

$$B\left(\frac{e}{n}, \rho\right) = \left\{x \in \mathbb{R}^n \mid \left\|x - \frac{e}{n}\right\| \leq \rho\right\}.$$

We are looking for the largest radius  $r$  such that

$$B\left(\frac{e}{n}, r\right) \cap \{x \mid e^t x = 1\} \subseteq \Delta.$$

## 10 Karmarkars Algorithm

This holds for  $r = \left\| \frac{e}{n} - (e - e_1) \frac{1}{n-1} \right\|$ . ( $r$  is the distance between the center  $e/n$  and the center of the  $(n-1)$ -dimensional simplex obtained by intersecting a side ( $x_i = 0$ ) of the unit cube with  $\Delta$ .)

This gives  $r = \frac{1}{\sqrt{n(n-1)}}$ .

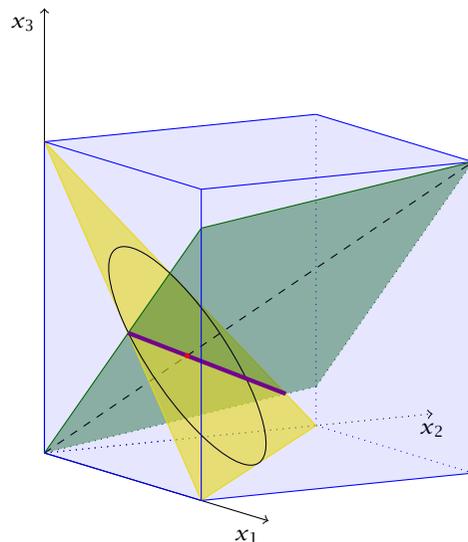
Now we consider the problem

$$\min\{\hat{c}^t x \mid \hat{A}x = 0, x \in B(e/n, r) \cap \Delta\}$$

This problem is easy to solve!!!

$$r^2 = (n-1) \cdot \left(\frac{1}{n} - \frac{1}{n-1}\right)^2 + \frac{1}{n^2} = \frac{1}{n^2(n-1)} + \frac{1}{n^2} = \frac{1}{n(n-1)}$$

## The Simplex



## 10 Karmarkars Algorithm

Ideally we would like to go in direction of  $-\hat{c}$  (starting from the center of the simplex).

However, doing this may violate constraints  $\hat{A}\hat{x} = 0$  or the constraint  $\hat{x} \in \Delta$ .

Therefore we first project  $\hat{c}$  on the nullspace of

$$B = \begin{pmatrix} \hat{A} \\ e^t \end{pmatrix}$$

We use

$$P = I - B^t(BB^t)^{-1}B$$

Then

$$\hat{d} = P\hat{c}$$

is the required projection.

## 10 Karmarkars Algorithm

We get the new point

$$\hat{x}(\rho) = \frac{e}{n} - \rho \frac{\hat{d}}{\|\hat{d}\|}$$

for  $\rho < r$ .

Choose  $\rho = \alpha r$  with  $\alpha = 1/4$ .

## Iteration of Karmarkars Algorithm

- ▶ Current solution  $\bar{x}$ .  $\bar{Y} := \text{diag}(\bar{x}_1, \dots, \bar{x}_n)$ .
- ▶ Transform problem via  $F_{\bar{x}}(x) = \frac{\bar{Y}^{-1}x}{e^t \bar{Y}^{-1}x}$ .  
Let  $\hat{c} = \bar{Y}c$ , and  $\hat{A} = A\bar{Y}$ .

- ▶ Compute

$$\hat{d} = (I - B^t(BB^t)^{-1}B)\hat{c},$$

where  $B = \begin{pmatrix} \hat{A} \\ e^t \end{pmatrix}$ .

- ▶ Set

$$\hat{x}_{\text{new}} = \frac{e}{n} - \rho \frac{\hat{d}}{\|\hat{d}\|},$$

with  $\rho = \alpha r$  with  $\alpha = 1/4$  and  $r = 1/\sqrt{n(n-1)}$ .

- ▶ Compute  $\bar{x}_{\text{new}} = F_{\bar{x}}^{-1}(\hat{x}_{\text{new}})$ .

### Lemma 40

The new point  $\hat{x}_{\text{new}}$  in the transformed space is the point that minimizes the cost  $\hat{c}^t \hat{x}$  among all feasible points in  $B(\frac{e}{n}, \rho)$ .

**Proof:** Let  $\hat{z}$  be another feasible point in  $B(\frac{e}{n}, \rho)$ .

As  $\hat{A}\hat{z} = 0$ ,  $\hat{A}\hat{x}_{\text{new}} = 0$ ,  $e^t \hat{z} = 1$ ,  $e^t \hat{x}_{\text{new}} = 1$  we have

$$B(\hat{x}_{\text{new}} - \hat{z}) = 0.$$

Further,

$$\begin{aligned} (\hat{c} - \hat{d})^t &= (\hat{c} - P\hat{c})^t \\ &= (B^t(BB^t)^{-1}B\hat{c})^t \\ &= \hat{c}^t B^t (BB^t)^{-1} B \end{aligned}$$

Hence, we get

$$(\hat{c} - \hat{d})^t (\hat{x}_{\text{new}} - \hat{z}) = 0 \text{ or } \hat{c}^t (\hat{x}_{\text{new}} - \hat{z}) = \hat{d}^t (\hat{x}_{\text{new}} - \hat{z})$$

which means that the cost-difference between  $\hat{x}_{\text{new}}$  and  $\hat{z}$  is the same measured w.r.t. the cost-vector  $\hat{c}$  or the projected cost-vector  $\hat{d}$ .

But

$$\frac{\hat{d}^t}{\|\hat{d}\|} (\hat{x}_{\text{new}} - \hat{z}) = \frac{\hat{d}^t}{\|\hat{d}\|} \left( \frac{e}{n} - \rho \frac{\hat{d}}{\|\hat{d}\|} - \hat{z} \right) = \frac{\hat{d}^t}{\|\hat{d}\|} \left( \frac{e}{n} - \hat{z} \right) - \rho < 0$$

as  $\frac{e}{n} - \hat{z}$  is a vector of length at most  $\rho$ .

This gives  $\hat{d}^t (\hat{x}_{\text{new}} - \hat{z}) \leq 0$  and therefore  $\hat{c} \hat{x}_{\text{new}} \leq \hat{c} \hat{z}$ .

In order to measure the progress of the algorithm we introduce a **potential function**  $f$ :

$$f(x) = \sum_j \ln\left(\frac{c^t x}{x_j}\right) = n \ln(c^t x) - \sum_j \ln(x_j) .$$

- ▶ The function  $f$  is invariant to scaling (i.e.,  $f(kx) = f(x)$ ).
- ▶ The potential function essentially measures **cost** (note the term  $n \ln(c^t x)$ ) but it penalizes us for choosing  $x_j$  values very small (by the term  $-\sum_j \ln(x_j)$ ; note that  $-\ln(x_j)$  is always positive).

For a point  $\hat{z}$  in the transformed space we use the potential function

$$\begin{aligned} \hat{f}(\hat{z}) &:= f(F_{\hat{x}}^{-1}(\hat{z})) = f\left(\frac{\bar{Y}\hat{z}}{e^t \bar{Y}\hat{z}}\right) = f(\bar{Y}\hat{z}) \\ &= \sum_j \ln\left(\frac{c^t \bar{Y}\hat{z}}{\bar{x}_j \hat{z}_j}\right) = \sum_j \ln\left(\frac{\hat{c}^t \hat{z}}{\hat{z}_j}\right) - \sum_j \ln \bar{x}_j \end{aligned}$$

**Observation:**

This means the potential of a point in the transformed space is simply the potential of its pre-image under  $F$ .

Note that if we are interested in **potential-change** we can ignore the additive term above. Then  $f$  and  $\hat{f}$  have the same form; only  $c$  is replaced by  $\hat{c}$ .

The basic idea is to show that one iteration of Karmarkar results in a constant decrease of  $\hat{f}$ . This means

$$\hat{f}(\hat{x}_{\text{new}}) \leq \hat{f}\left(\frac{e}{n}\right) - \delta ,$$

where  $\delta$  is a constant.

This gives

$$f(\bar{x}_{\text{new}}) \leq f(\bar{x}) - \delta .$$

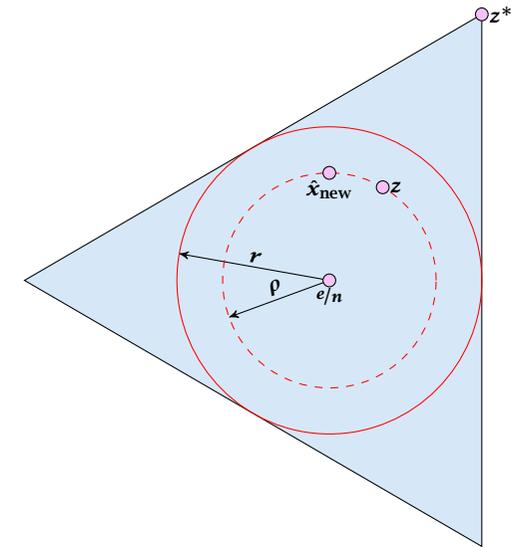
### Lemma 41

There is a feasible point  $z$  (i.e.,  $\hat{A}z = 0$ ) in  $B(\frac{e}{n}, \rho) \cap \Delta$  that has

$$\hat{f}(z) \leq \hat{f}\left(\frac{e}{n}\right) - \delta$$

with  $\delta = \ln(1 + \alpha)$ .

Note that this shows the existence of a good point within the ball. In general it will be difficult to find this point.



Let  $z^*$  be the feasible point in the transformed space where  $\hat{c}^t x$  is minimized. (Note that in contrast  $\hat{x}_{\text{new}}$  is the point in the **intersection of the feasible region and  $B(\frac{e}{n}, \rho)$**  that minimizes this function; in general  $z^* \neq \hat{x}_{\text{new}}$ )

$z^*$  must lie at the boundary of the simplex. This means  $z^* \notin B(\frac{e}{n}, \rho)$ .

The point  $z$  we want to use lies farthest in the direction from  $\frac{e}{n}$  to  $z^*$ , namely

$$z = (1 - \lambda)\frac{e}{n} + \lambda z^*$$

for some positive  $\lambda < 1$ .

Hence,

$$\hat{c}^t z = (1 - \lambda)\hat{c}^t \frac{e}{n} + \lambda \hat{c}^t z^*$$

The optimum cost (at  $z^*$ ) is zero.

Therefore,

$$\frac{\hat{c}^t \frac{e}{n}}{\hat{c}^t z} = \frac{1}{1 - \lambda}$$

The improvement in the potential function is

$$\begin{aligned}\hat{f}\left(\frac{e}{n}\right) - \hat{f}(z) &= \sum_j \ln\left(\frac{\hat{c}^t \frac{e}{n}}{\frac{1}{n}}\right) - \sum_j \ln\left(\frac{\hat{c}^t z}{z_j}\right) \\ &= \sum_j \ln\left(\frac{\hat{c}^t \frac{e}{n}}{\hat{c}^t z} \cdot \frac{z_j}{\frac{1}{n}}\right) \\ &= \sum_j \ln\left(\frac{n}{1-\lambda} z_j\right) \\ &= \sum_j \ln\left(\frac{n}{1-\lambda} \left((1-\lambda)\frac{1}{n} + \lambda z_j^*\right)\right) \\ &= \sum_j \ln\left(1 + \frac{n\lambda}{1-\lambda} z_j^*\right)\end{aligned}$$

We can use the fact that for non-negative  $s_i$

$$\sum_i \ln(1 + s_i) \geq \ln(1 + \sum_i s_i)$$

This gives

$$\begin{aligned}\hat{f}\left(\frac{e}{n}\right) - \hat{f}(z) &= \sum_j \ln\left(1 + \frac{n\lambda}{1-\lambda} z_j^*\right) \\ &\geq \ln\left(1 + \frac{n\lambda}{1-\lambda}\right)\end{aligned}$$

Suppose true for  $s_1, \dots, s_{k-1}$ . Then

$$\begin{aligned}\sum_{i=1}^k \ln(1 + s_i) &\geq \ln\left(1 + \sum_{i=1}^{k-1} s_i\right) + \ln(1 + s_k) = \ln\left(\left(1 + \sum_{i=1}^{k-1} s_i\right)(1 + s_k)\right) \\ &= \ln\left(1 + \sum_i s_i + s_k \sum_{i=1}^{k-1} s_i\right) \geq \ln(1 + \sum_i s_i)\end{aligned}$$

In order to get further we need a bound on  $\lambda$ :

$$\alpha r = \rho = \|z - e/n\| = \|\lambda(z^* - e/n)\| \leq \lambda R$$

Here  $R$  is the radius of the ball around  $\frac{e}{n}$  that contains the whole simplex.

$R = \sqrt{(n-1)/n}$ . Since  $r = 1/\sqrt{(n-1)n}$  we have  $R/r = n-1$  and

$$\lambda \geq \alpha \frac{r}{R} \geq \alpha/(n-1)$$

Then

$$1 + n \frac{\lambda}{1-\lambda} \geq 1 + \frac{n\alpha}{n-\alpha-1} \geq 1 + \alpha$$

This gives the lemma.

### Lemma 42

If we choose  $\alpha = 1/4$  and  $n \geq 4$  in Karmarkars algorithm the point  $\hat{x}_{\text{new}}$  satisfies

$$\hat{f}(\hat{x}_{\text{new}}) \leq \hat{f}\left(\frac{e}{n}\right) - \delta$$

with  $\delta = 1/10$ .

**Proof:**

Define

$$\begin{aligned} g(\hat{x}) &= n \ln \frac{\hat{c}^t \hat{x}}{\hat{c}^t \frac{e}{n}} \\ &= n(\ln \hat{c}^t \hat{x} - \ln \hat{c}^t \frac{e}{n}) . \end{aligned}$$

This is the change in the **cost part** of the potential function when going from the center  $\frac{e}{n}$  to the point  $\hat{x}$  in the **transformed space**.

Similar, the **penalty** when going from  $\frac{e}{n}$  to  $w$  increases by

$$h(\hat{x}) = \text{pen}(\hat{x}) - \text{pen}\left(\frac{e}{n}\right) = - \sum_j \ln \frac{\hat{x}_j}{\frac{1}{n}}$$

where  $\text{pen}(v) = - \sum_j \ln(v_j)$ .

We want to derive a lower bound on

$$\begin{aligned} \hat{f}\left(\frac{e}{n}\right) - \hat{f}(\hat{x}_{\text{new}}) &= [\hat{f}\left(\frac{e}{n}\right) - \hat{f}(z)] \\ &\quad + h(z) \\ &\quad - h(\hat{x}_{\text{new}}) \\ &\quad + [g(z) - g(\hat{x}_{\text{new}})] \end{aligned}$$

where  $z$  is the point in the ball where  $\hat{f}$  achieves its minimum.

We have

$$[\hat{f}\left(\frac{e}{n}\right) - \hat{f}(z)] \geq \ln(1 + \alpha)$$

by the previous lemma.

We have

$$[g(z) - g(\hat{x}_{\text{new}})] \geq 0$$

since  $\hat{x}_{\text{new}}$  is the point with minimum cost in the ball, and  $g$  is monotonically increasing with cost.

We show that the change  $h(w)$  in **penalty** when going from  $e/n$  to  $w$  fulfills

$$|h(w)| \leq \frac{\beta^2}{2(1-\beta)}$$

where  $\beta = n\alpha r$  and  $w$  is some point in the ball  $B(\frac{e}{n}, \alpha r)$ .

Hence,

$$\hat{f}\left(\frac{e}{n}\right) - \hat{f}(\hat{x}_{\text{new}}) \geq \ln(1+\alpha) - \frac{\beta^2}{(1-\beta)}.$$

### Lemma 43

For  $|x| \leq \beta < 1$

$$|\ln(1+x) - x| \leq \frac{x^2}{2(1-\beta)}.$$

For  $|x| < 1$

$$\ln(1+x) = \sum_{i \geq 1} (-1)^{i+1} \frac{x^i}{i} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$$

This gives

$$\begin{aligned} |\ln(1+x) - x| &\leq \left| -\frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots \right| \leq \left| \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \dots \right| \\ &\leq \frac{x^2}{2} |x^0 + x^1 + x^2 + \dots| = \frac{x^2}{2(1-|x|)}. \end{aligned}$$

This gives for  $w \in B(\frac{e}{n}, \rho)$

$$\begin{aligned} |h(w)| &= \left| \sum_j \ln \frac{w_j}{1/n} \right| \\ &= \left| \sum_j \ln \left( \frac{1/n + (w_j - 1/n)}{1/n} \right) - \sum_j n \left( w_j - \frac{1}{n} \right) \right| \\ &= \left| \sum_j \left[ \ln \left( 1 + n \left( w_j - \frac{1}{n} \right) \right) - n \left( w_j - \frac{1}{n} \right) \right] \right| \\ &\leq \sum_j \frac{n^2 (w_j - 1/n)^2}{2(1-\alpha nr)} \\ &\leq \frac{(\alpha nr)^2}{2(1-\alpha nr)} \end{aligned}$$

The decrease in potential is therefore at least

$$\ln(1+\alpha) - \frac{\beta^2}{1-\beta}$$

with  $\beta = n\alpha r = \alpha \sqrt{\frac{n}{n-1}}$ .

It can be shown that this is at least  $\frac{1}{10}$  for  $n \geq 4$  and  $\alpha = 1/4$ .

Let  $\tilde{x}^{(k)}$  be the current point after the  $k$ -th iteration, and let  $\tilde{x}^{(0)} = \frac{e}{n}$ .

Then  $f(\tilde{x}^{(k)}) \leq f(e/n) - k/10$ .

This gives

$$\begin{aligned} n \ln \frac{c^t \tilde{x}^{(k)}}{c^t \frac{e}{n}} &\leq \sum_j \ln \tilde{x}_j^{(k)} - \sum_j \ln \frac{1}{n} - k/10 \\ &\leq n \ln n - k/10 \end{aligned}$$

Choosing  $k = 10n(\ell + \ln n)$  with  $\ell = \Theta(L)$  we get

$$\frac{c^t \tilde{x}^{(k)}}{c^t \frac{e}{n}} \leq e^{-\ell} \leq 2^{-\ell}.$$

Hence,  $\Theta(nL)$  iterations are sufficient. One iteration can be performed in time  $\mathcal{O}(n^3)$ .