Geometrische Grundlagen der Linearen Optimierung

Martin Henk & Martin Skutella

TU Berlin

WS 2015/16

Chapter 5:
Introduction to Linear Programming

Optimization Problems

Generic optimization problem

Given: set X, function $f : X \to \mathbb{R}$
Task: find $x^* \in X$ maximizing (minimizing) $f(x^*)$, i. e.,

$$f(x^*) \geq f(x) \quad (f(x^*) \leq f(x)) \qquad \text{for all } x \in X.$$

- An x^* with these properties is called optimal solution (optimum).
- Here, X is the set of feasible solutions, f is the objective function.

Short form:

$$\begin{aligned} \text{maximize} \quad & f(x) \\ \text{subject to} \quad & x \in X \end{aligned}$$

or simply: $\max\{f(x) \mid x \in X\}$.

Problem: Too general to say anything meaningful!

Reminder: Convex Optimization Problems

Definition. (cp. Def. 1.5, Def. 1.12)

Let $X \subseteq \mathbb{R}^n$ and $f : X \to \mathbb{R}$.

a X is convex if for all $x, y \in X$ and $0 \leq \lambda \leq 1$ it holds that

$$\lambda \cdot x + (1 - \lambda) \cdot y \in X \ .$$

b f is convex if for all $x, y \in X$ and $0 \leq \lambda \leq 1$ with $\lambda \cdot x + (1 - \lambda) \cdot y \in X$ it holds that

$$\lambda \cdot f(x) + (1 - \lambda) \cdot f(y) \geq f(\lambda \cdot x + (1 - \lambda) \cdot y) \ .$$

c If X and f are both convex, then $\min\{f(x) \mid x \in X\}$ is a convex optimization problem.

Note: $f : X \to \mathbb{R}$ is called concave if $-f$ is convex.

Reminder: Local and Global Optimality

Definition. (cp. Theorem 1.13)

Let $X \subseteq \mathbb{R}^n$ and $f : X \to \mathbb{R}$.
$x' \in X$ is a local optimum of the optimization problem $\min\{f(x) \mid x \in X\}$ if there is an $\varepsilon > 0$ such that

$$f(x') \leq f(x) \qquad \text{for all } x \in X \text{ with } \|x' - x\|_2 \leq \varepsilon.$$

Theorem 1.13.

For a convex optimization problem, every local optimum is a (global) optimum.

Optimization Problems Considered in this Course:

$$\begin{array}{ll} \text{maximize} & f(x) \\ \text{subject to} & x \in X \end{array}$$

- $X \subseteq \mathbb{R}^n$ polyhedron, f linear function
  $\longrightarrow$ linear optimization problem (in particular convex)
- $X \subseteq \mathbb{Z}^n$ integer points of a polyhedron, f linear function
  $\longrightarrow$ integer linear optimization problem
- X related to some combinatorial structure (e. g., graph)
  $\longrightarrow$ combinatorial optimization problem
- X finite (but usually huge)
  $\longrightarrow$ discrete optimization problem

Example: Shortest Path Problem

Given: directed graph $D = (V, A)$, weight function $w : A \to \mathbb{R}_{\geq 0}$,
start node $s \in V$, destination node $t \in V$.

Task: find s-t-path of minimum weight in D.

That is, $X = \{P \subseteq A \mid P \text{ is } s\text{-}t\text{-path in } D\}$ and $f : X \to \mathbb{R}$ is given by

$$f(P) = \sum_{a \in P} w(a) \ .$$

Remark.
Note that the finite set of feasible solutions X is only implicitly given by D.
This holds for all interesting problems in combinatorial optimization!

Example: Minimum Spanning Tree (MST) Problem

Given: undirected graph $G = (V, E)$, weight function $w : E \to \mathbb{R}_{\geq 0}$.

Task: find connected subgraph of G containing all nodes in V with minimum total weight.

That is, $X = \{E' \subseteq E \mid E' \text{ connects all nodes in } V\}$ and $f : X \to \mathbb{R}$ is given by

$$f(E') = \sum_{e \in E'} w(e) \ .$$

Remarks.

- Notice that there always exists an optimal solution without cycles.
- A connected graph without cycles is called a tree.
- A subgraph of G containing all nodes in V is called spanning.

Example: Minimum Cost Flow Problem

Given: directed graph $D = (V, A)$, with arc *capacities* $u : A \to \mathbb{R}_{\geq 0}$, arc costs $c : A \to \mathbb{R}$, and node *balances* $b : V \to \mathbb{R}$.

Interpretation:

- nodes $v \in V$ with $b(v) > 0$ ($b(v) < 0$) have *supply* (*demand*) and are called *sources* (*sinks*)
- the capacity $u(a)$ of arc $a \in A$ limits the amount of flow that can be sent through arc a.

Task: find a *flow* $x : A \to \mathbb{R}_{\geq 0}$ obeying capacities and satisfying all supplies and demands, that is,

$$\begin{aligned} 0 \leq x(a) \leq u(a) &\qquad \text{for all } a \in A, \\ \sum_{a \in \delta^+(v)} x(a) - \sum_{a \in \delta^-(v)} x(a) = b(v) &\qquad \text{for all } v \in V, \end{aligned}$$

such that x has minimum cost $c(x) := \sum_{a \in A} c(a) \cdot x(a)$.

Example: Minimum Cost Flow Problem (cont.)

Formulation as a linear program (LP):

$$\begin{aligned} \text{minimize} \quad & \sum_{a \in A} c(a) \cdot x(a) && && (5.1) \\ \text{subject to} \quad & \sum_{a \in \delta^+(v)} x(a) - \sum_{a \in \delta^-(v)} x(a) = b(v) && \text{for all } v \in V, && (5.2) \\ & x(a) \leq u(a) && \text{for all } a \in A, && (5.3) \\ & x(a) \geq 0 && \text{for all } a \in A. && (5.4) \end{aligned}$$

- Objective function given by (5.1). Set of feasible solutions:

$$X = \{x \in \mathbb{R}^A \mid x \text{ satisfies } (5.2), (5.3), \text{ and } (5.4)\} .$$

- Notice that (5.1) is a linear function of x and (5.2) – (5.4) are linear equations and linear inequalities, respectively. $\longrightarrow$ linear program

Example (cont.): Adding Fixed Cost

Fixed costs $w : A \to \mathbb{R}_{\geq 0}$.

If arc $a \in A$ shall be used (i. e., $x(a) > 0$), it must be bought at cost $w(a)$.

Add variables $y(a) \in \{0, 1\}$ with $y(a) = 1$ if arc a is used, 0 otherwise.

This leads to the following mixed-integer linear program (MIP):

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{a \in A} c(a) \cdot x(a) + \sum_{a \in A} w(a) \cdot y(a) & \\
\text{subject to} & \displaystyle\sum_{a \in \delta^+(v)} x(a) - \sum_{a \in \delta^-(v)} x(a) = b(v) & \text{for all } v \in V, \\
& x(a) \leq u(a) \cdot y(a) & \text{for all } a \in A, \\
& x(a) \geq 0 & \text{for all } a \in A. \\
& y(a) \in \{0, 1\} & \text{for all } a \in A.
\end{array}
$$

MIP: Linear program where some variables may only take integer values.

Example: Maximum Weighted Matching Problem

Given: undirected graph $G = (V, E)$, weight function $w : E \to \mathbb{R}$.

Task: find matching $M \subseteq E$ with maximum total weight.

($M \subseteq E$ is a matching if every node is incident to at most one edge in M.)

Formulation as an integer linear program (IP):

Variables: $x_e \in \{0, 1\}$ for $e \in E$ with $x_e = 1$ if and only if $e \in M$.

$$
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{e \in E} w(e) \cdot x_e & \\
\text{subject to} & \displaystyle\sum_{e \in \delta(v)} x_e \leq 1 & \text{for all } v \in V, \\
& x_e \in \{0, 1\} & \text{for all } e \in E.
\end{array}
$$

IP: Linear program where all variables may only take integer values.

Example: Traveling Salesperson Problem (TSP)

Given: complete graph K_n on n nodes, weight function $w : E(K_n) \to \mathbb{R}$.

Task: find a Hamiltonian circuit with minimum total weight.

(A Hamiltonian circuit visits every node exactly once.)

Formulation as an integer linear program? (later!)

Typical Questions

For a given optimization problem:

- How to find an optimal solution?
- How to find a feasible solution?
- Does there exist an optimal/feasible solution?
- How to prove that a computed solution is optimal?
- How difficult is the problem?
- Does there exist an *efficient algorithm* with "small" worst-case running time?
- How to formulate the problem as a (mixed integer) linear program?
- Is there a useful special structure of the problem?

Outline of the Remainder of this Course

- linear programming and the simplex algorithm
- geometric interpretation of the simplex algorithm
- LP duality, complementary slackness
- maybe: efficient algorithms for maximum flows and minimum cost flows
- ...

Literature on Linear Optimization (not complete)

- D. Bertsimas, J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena, 1997.
- V. Chvatal, *Linear Programming*, Freeman, 1983.
- G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, 1998 (1963).
- M. Grötschel, L. Lovàsz, A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.
- J. Matousek, B. Gärtner, *Using and Understanding Linear Programming*, Springer, 2006.
- M. Padberg, *Linear Optimization and Extensions*, Springer, 1995.
- A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, 1986.
- R. J. Vanderbei, *Linear Programming*, Springer, 2001.

Chapter 6:
Linear Programming Basics

(cp. Bertsimas & Tsitsiklis, Chapter 1)

Example of a Linear Program

$$
\begin{array}{lrcrcrcrl}
\text{minimize} & 2x_1 & - & x_2 & + & 4x_3 & & & \\
\text{subject to} & x_1 & + & x_2 & & & + & x_4 & \leq 2 \\
& & & 3x_2 & - & x_3 & & & = 5 \\
& & & & & x_3 & + & x_4 & \geq 3 \\
& x_1 & & & & & & & \geq 0 \\
& & & & & x_3 & & & \leq 0
\end{array}
$$

Remarks.

- objective function is linear in vector of variables $x = (x_1, x_2, x_3, x_4)^T$
- constraints are linear inequalities and linear equations
- last two constraints are special
  (non-negativity and non-positivity constraint, respectively)

General Linear Program

$$\begin{aligned}
\text{minimize} \quad & c^T \cdot x \\
\text{subject to} \quad & {a_i}^T \cdot x \geq b_i & \text{for } i \in M_1, && (6.1)\\
& {a_i}^T \cdot x = b_i & \text{for } i \in M_2, && (6.2)\\
& {a_i}^T \cdot x \leq b_i & \text{for } i \in M_3, && (6.3)\\
& x_j \geq 0 & \text{for } j \in N_1, && (6.4)\\
& x_j \leq 0 & \text{for } j \in N_2, && (6.5)
\end{aligned}$$

with $c \in \mathbb{R}^n$, $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i \in M_1 \,\dot\cup\, M_2 \,\dot\cup\, M_3$ (finite index sets), and $N_1, N_2 \subseteq \{1, \ldots, n\}$ given.

- $x \in \mathbb{R}^n$ satisfying constraints (6.1) – (6.5) is a feasible solution;
- set of feasible solutions $\{x \in \mathbb{R}^n \mid (6.1) - (6.5)\}$ is polyhedron in $\mathbb{R}^n$;
- feasible solution x^* is optimal solution if

$$c^T \cdot x^* \leq c^T \cdot x \qquad \text{for all feasible solutions } x;$$

- linear program is unbounded if, for all $k \in \mathbb{R}$, there is a feasible solution $x \in \mathbb{R}^n$ with $c^T \cdot x \leq k$.

Special Forms of Linear Programs

- maximizing $c^T \cdot x$ is equivalent to minimizing $(-c)^T \cdot x$.
- any linear program can be written in the form

$$\begin{aligned}
\text{minimize} \quad & c^T \cdot x \\
\text{subject to} \quad & A \cdot x \geq b
\end{aligned}$$

  for some $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$:
  - rewrite ${a_i}^T \cdot x = b_i$ as: ${a_i}^T \cdot x \geq b_i \quad \wedge \quad {a_i}^T \cdot x \leq b_i$,
  - rewrite ${a_i}^T \cdot x \leq b_i$ as: $(-a_i)^T \cdot x \geq -b_i$.
- Linear program in standard form:

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

  with $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$.

Example: Diet Problem

Given:

- n different foods, m different nutrients
- $a_{ij} :=$ amount of nutrient i in one unit of food j
- $b_i :=$ requirement of nutrient i in some ideal diet
- $c_j :=$ cost of one unit of food j

Task: find a cheapest ideal diet consisting of foods $1, \dots, n$.

LP formulation: Let $x_j :=$ number of units of food j in the diet:

$$\begin{aligned} \min \quad & c^T \cdot x \\ \text{s.t.} \quad & A \cdot x = b \\ & x \geq 0 \end{aligned} \qquad \text{or} \qquad \begin{aligned} \min \quad & c^T \cdot x \\ \text{s.t.} \quad & A \cdot x \geq b \\ & x \geq 0 \end{aligned}$$

with $A = (a_{ij}) \in \mathbb{R}^{m \times n}$, $b = (b_i) \in \mathbb{R}^m$, $c = (c_j) \in \mathbb{R}^n$.

Reduction to Standard Form

Any linear program can be brought into standard form:

- elimination of free (unbounded) variables x_j:

  replace x_j with $x_j^+, x_j^- \geq 0$: $\quad x_j = x_j^+ - x_j^-$

- elimination of non-positive variables x_j:

  replace $x_j \leq 0$ with $(-x_j) \geq 0$.

- elimination of inequality constraint $a_i{}^T \cdot x \leq b_i$:

  introduce slack variable $s \geq 0$ and rewrite: $a_i{}^T \cdot x + s = b_i$

- elimination of inequality constraint $a_i{}^T \cdot x \geq b_i$:

  introduce slack variable $s \geq 0$ and rewrite: $a_i{}^T \cdot x - s = b_i$

Example

The linear program

$$\begin{array}{rrcrl} \min & 2\,x_1 & + & 4\,x_2 & \\ \text{s.t.} & x_1 & + & x_2 & \geq 3 \\ & 3\,x_1 & + & 2\,x_2 & = 14 \\ & x_1 & & & \geq 0 \end{array}$$

is equivalent to the standard form problem

$$\begin{array}{rrcrcrcrl} \min & 2\,x_1 & + & 4\,x_2^+ & - & 4\,x_2^- & & & \\ \text{s.t.} & x_1 & + & x_2^+ & - & x_2^- & - & x_3 & = 3 \\ & 3\,x_1 & + & 2\,x_2^+ & - & 2\,x_2^- & & & = 14 \\ & & & & & & & x_1,\, x_2^+,\, x_2^-,\, x_3 & \geq 0 \end{array}$$

Affine Linear and Convex Functions

Lemma 6.1.

a) An affine linear function $f : \mathbb{R}^n \to \mathbb{R}$ given by $f(x) = c^T \cdot x + d$ with $c \in \mathbb{R}^n$, $d \in \mathbb{R}$, is both convex and concave.

b) If $f_1, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}$ are convex functions, then $f : \mathbb{R}^n \to \mathbb{R}$ defined by $f(x) := \max_{i=1,\ldots,k} f_i(x)$ is also convex.

Proof: . . . $\square$

Piecewise Linear Convex Objective Functions

Let $c_1, \ldots, c_k \in \mathbb{R}^n$ and $d_1, \ldots, d_k \in \mathbb{R}$.

Consider piecewise linear convex function: $x \mapsto \max_{i=1,\ldots,k} {c_i}^T \cdot x + d_i$:

$$\begin{array}{ll} \min & \max\limits_{i=1,\ldots,k} {c_i}^T \cdot x + d_i \\ \text{s.t.} & A \cdot x \geq b \end{array} \quad \longleftrightarrow \quad \begin{array}{lll} \min & z & \\ \text{s.t.} & z \geq {c_i}^T \cdot x + d_i & \text{for all } i \\ & A \cdot x \geq b & \end{array}$$

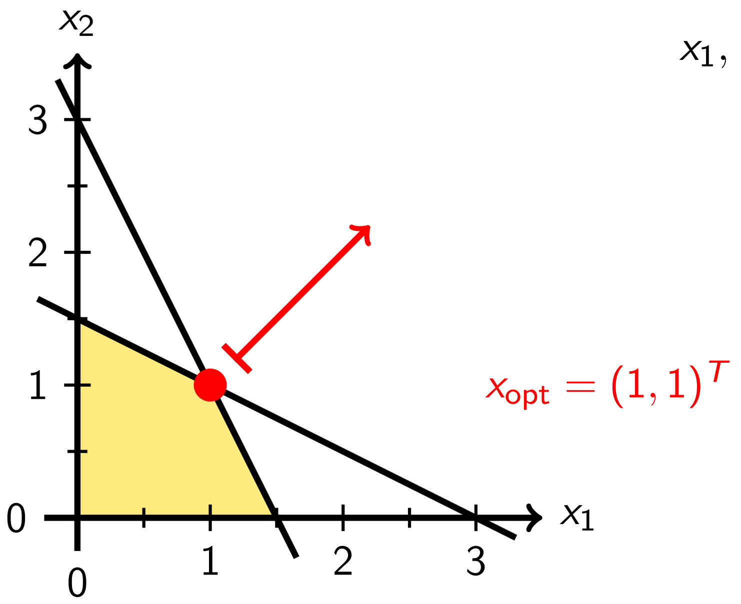
Example: let $c_1, \ldots, c_n \geq 0$

$$\begin{array}{ll} \min & \sum\limits_{i=1}^n c_i \cdot |x_i| \\ \text{s.t.} & A \cdot x \geq b \end{array} \quad \leftrightarrow \quad \begin{array}{ll} \min & \sum\limits_{i=1}^n c_i \cdot z_i \\ \text{s.t.} & z_i \geq x_i \\ & z_i \geq -x_i \\ & A \cdot x \geq b \end{array} \quad \leftrightarrow \quad \begin{array}{ll} \min & \sum\limits_{i=1}^n c_i \cdot (x_i^+ + x_i^-) \\ \text{s.t.} & A \cdot (x^+ - x^-) \geq b \\ & x^+, x^- \geq 0 \end{array}$$

Graphical Representation and Solution

2D example:

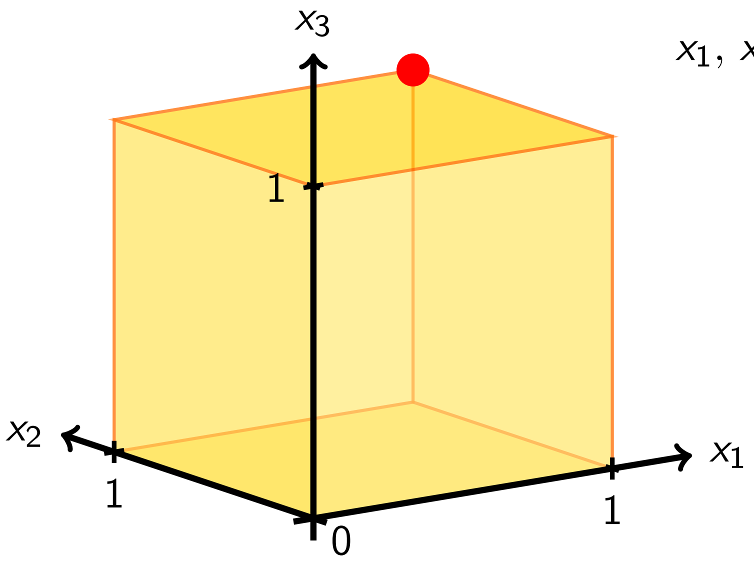
$$\begin{array}{lrcrl} \min & -x_1 & - & x_2 & \\ \text{s.t.} & x_1 & + & 2\,x_2 & \leq 3 \\ & 2\,x_1 & + & x_2 & \leq 3 \\ & & & x_1,\, x_2 & \geq 0 \end{array}$$



Graphical Representation and Solution (cont.)

3D example:

$$
\begin{array}{rrcrcrl}
\min & -x_1 & - & x_2 & - & x_3 & \\
\text{s.t.} & x_1 & & & & & \leq 1 \\
& & & x_2 & & & \leq 1 \\
& & & & & x_3 & \leq 1 \\
& & & & & x_1,\, x_2,\, x_3 & \geq 0
\end{array}
$$

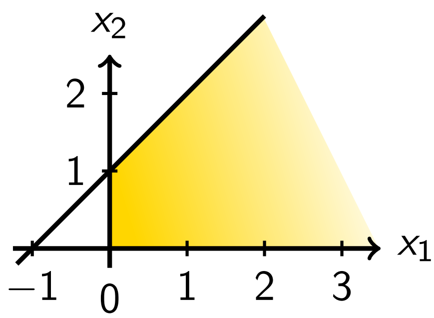


$x_{\text{opt}} = (1, 1, 1)^T$

Graphical Representation and Solution (cont.)

another 2D example:

$$
\begin{array}{rrcrl}
\min & c_1\, x_1 & + & c_2\, x_2 & \\
\text{s.t.} & -x_1 & + & x_2 & \leq 1 \\
& & & x_1,\, x_2 & \geq 0
\end{array}
$$



- for $c = (1, 1)^T$, the unique optimal solution is $x = (0, 0)^T$
- for $c = (1, 0)^T$, the optimal solutions are exactly the points
$$x = (0, x_2)^T \qquad \text{with } 0 \leq x_2 \leq 1$$
- for $c = (0, 1)^T$, the optimal solutions are exactly the points
$$x = (x_1, 0)^T \qquad \text{with } x_1 \geq 0$$
- for $c = (-1, -1)^T$, the problem is unbounded, optimal cost is $-\infty$
- if we add the constraint $x_1 + x_2 \leq -1$, the problem is infeasible

Properties of the Set of Optimal Solutions

In the last example, the following 5 cases occurred:

i there is a unique optimal solution

ii there exist infinitely many optimal solutions, but the set of optimal solutions is bounded

iii there exist infinitely many optimal solutions and the set of optimal solutions is unbounded

iv the problem is unbounded, i. e., the optimal cost is $-\infty$ and no feasible solution is optimal

v the problem is infeasible, i. e., the set of feasible solutions is empty

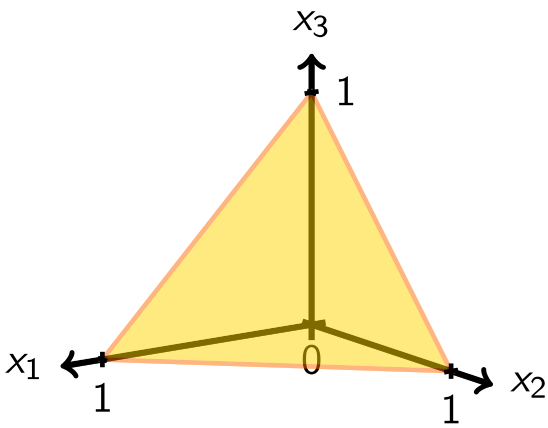
These are indeed all cases that can occur in general (see also later).

(Notice that the set of optimal solutions is a face of the polyhedron given by the set of feasible solutions.)

Visualizing LPs in Standard Form

Example:
Let $A = (1, 1, 1) \in \mathbb{R}^{1\times 3}$, $b = (1) \in \mathbb{R}^1$ and consider the set of feasible solutions

$$P = \{x \in \mathbb{R}^3 \mid A \cdot x = b,\ x \geq 0\} \ .$$



More general:

- if $A \in \mathbb{R}^{m\times n}$ with $m \leq n$ and the rows of A are linearly independent, then
$$\{x \in \mathbb{R}^n \mid A \cdot x = b\}$$
is an $(n - m)$-dimensional affine subspace of $\mathbb{R}^n$.
- set of feasible solutions lies in this affine subspace and is only constrained by non-negativity constraints $x \geq 0$.

Reminder: Extreme Points and Vertices of Polyhedra

Definition. (cp. Notation 3.6, Remark 3.7)

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

a. $x \in P$ is an extreme point of P if

$$x \neq \lambda \cdot y + (1 - \lambda) \cdot z \qquad \text{for all } y, z \in P \setminus \{x\},\ 0 \leq \lambda \leq 1,$$

i. e., x is not a convex combination of two other points in P.

b. $x \in P$ is a vertex of P if there is some $c \in \mathbb{R}^n$ such that

$$c^T \cdot x < c^T \cdot y \qquad \text{for all } y \in P \setminus \{x\},$$

i. e., x is the unique optimal solution to the LP $\min\{c^T \cdot z \mid z \in P\}$.

Active and Binding Constraints

In the following, let $P \subseteq \mathbb{R}^n$ be a polyhedron defined by

$$\begin{aligned} {a_i}^T \cdot x &\geq b_i && \text{for } i \in M_1, \\ {a_i}^T \cdot x &= b_i && \text{for } i \in M_2, \end{aligned}$$

with $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$, for all i.

Definition 6.2.

If $x^* \in \mathbb{R}^n$ satisfies ${a_i}^T \cdot x^* = b_i$ for some i, then the corresponding constraint is active (or binding) at x^*.

Basic Facts from Linear Algebra

Theorem 6.3.

Let $x^* \in \mathbb{R}^n$ and $I = \{i \mid {a_i}^T \cdot x^* = b_i\}$. The following are equivalent:

- i there are n vectors in $\{a_i \mid i \in I\}$ which are linearly independent;
- ii the vectors in $\{a_i \mid i \in I\}$ span $\mathbb{R}^n$;
- iii x^* is the unique solution to the system of equations ${a_i}^T \cdot x = b_i$, $i \in I$.

Vertices, Extreme Points, and Basic Feasible Solutions

Definition 6.4.

- a $x^* \in \mathbb{R}^n$ is a basic solution of P if
  - all equality constraints are active and
  - there are n linearly independent constraints that are active.
- b A basic solution satisfying all constraints is a basic feasible solution.

Theorem 6.5 (cp. Remark 3.7).

For $x^* \in P$, the following are equivalent:

- i x^* is a vertex of P;
- ii x^* is an extreme point of P;
- iii x^* is a basic feasible solution of P.

Proof: . . . $\square$

Reminder: Number of Vertices

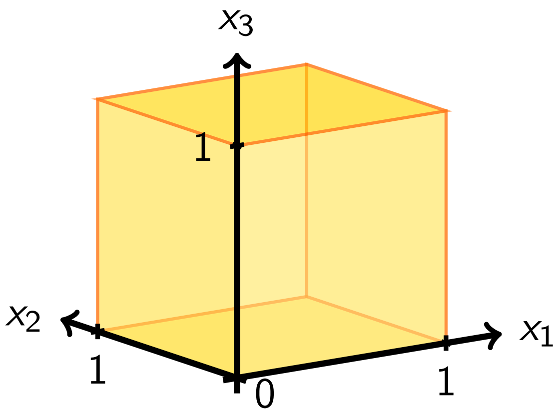
Corollary.

- a) A polyhedron has a finite number of vertices and basic solutions.
- b) For a polyhedron in $\mathbb{R}^n$ given by linear equations and m linear inequalities, this number is at most $\binom{m}{n}$.

Example:

$P := \{x \in \mathbb{R}^n \mid 0 \leq x_i \leq 1,\ i = 1, \ldots, n\}$
(n-dimensional unit cube)

- number of constraints: $m = 2n$
- number of vertices: 2^n



Adjacent Basic Solutions and Edges

Definition 6.6 (cp. Notation 3.6).

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

- a) Two distinct basic solutions are adjacent if there are $n - 1$ linearly independent constraints that are active at both of them.
- b) If both solutions are feasible, the line segment that joins them is an edge of P.

Polyhedra in Standard Form

Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $P = \{x \in \mathbb{R}^n \mid A \cdot x = b,\ x \geq 0\}$ a polyhedron in standard form representation.

Observation.

One can assume without loss of generality that $\mathrm{rank}(A) = m$.

Theorem 6.7.

$x \in \mathbb{R}^n$ is a basic solution of P if and only if $A \cdot x = b$ and there are indices $B(1), \dots, B(m) \in \{1, \dots, n\}$ such that

- columns $A_{B(1)}, \dots, A_{B(m)}$ of matrix A are linearly independent and
- $x_i = 0$ for all $i \notin \{B(1), \dots, B(m)\}$.

Proof: ... □

- $x_{B(1)}, \dots, x_{B(m)}$ are basic variables, the remaining variables non-basic.
- The vector of basic variables is denoted by $x_B := (x_{B(1)}, \dots, x_{B(m)})^T$.
- $A_{B(1)}, \dots, A_{B(m)}$ are basic columns of A and form a basis of $\mathbb{R}^m$.
- The matrix $B := (A_{B(1)}, \dots, A_{B(m)}) \in \mathbb{R}^{m \times m}$ is called basis matrix.

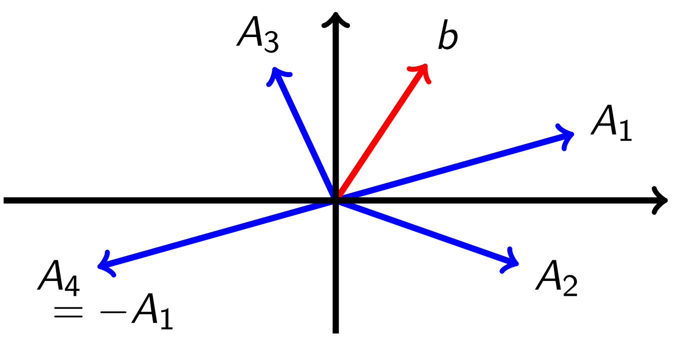
Basic Columns and Basic Solutions

Observation 6.8.

Let $x \in \mathbb{R}^n$ be a basic solution, then:

- $B \cdot x_B = b$ and thus $x_B = B^{-1} \cdot b$;
- x is a basic feasible solution if and only if $x_B = B^{-1} \cdot b \geq 0$.

Example: $m = 2$



- A_1, A_3 or A_2, A_3 form bases with corresp. basic feasible solutions.
- A_1, A_4 do not form a basis.
- A_1, A_2 and A_2, A_4 and A_3, A_4 form bases with infeasible basic solution.

Bases and Basic Solutions

Corollary 6.9.

- Every basis $A_{B(1)}, \ldots, A_{B(m)}$ determines a unique basic solution.
- Thus, different basic solutions correspond to different bases.
- But: two different bases might yield the same basic solution.

Example: If $b = 0$, then $x = 0$ is the only basic solution.

Adjacent Bases

Definition 6.10.

Two bases $A_{B(1)}, \ldots, A_{B(m)}$ and $A_{B'(1)}, \ldots, A_{B'(m)}$ are adjacent if they share all but one column.

Observation 6.11.

a. Two adjacent basic solutions can always be obtained from two adjacent bases.
b. If two adjacent bases lead to distinct basic solutions, then the latter are adjacent.

Degeneracy

Definition 6.12 (cp. Def. 3.21 of simple polytope).

A basic solution x of a polyhedron P is degenerate if more than n constraints are active at x.

Observation 6.13.

Let $P = \{x \in \mathbb{R}^n \mid A \cdot x = b,\ x \geq 0\}$ be a polyhedron in standard form with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

a. A basic solution $x \in P$ is degenerate if and only if more than $n - m$ components of x are zero.
b. For a non-degenerate basic solution $x \in P$, there is a unique basis.

Three Different Reasons for Degeneracy

i. redundant variables

   Example:
$$\begin{aligned} x_1 + x_2 \quad &= 1 \\ x_3 &= 0 \\ x_1,\ x_2,\ x_3 &\geq 0 \end{aligned} \quad \longleftrightarrow \quad A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

ii. redundant constraints

   Example:
$$\begin{aligned} x_1 + 2\,x_2 &\leq 3 \\ 2\,x_1 + x_2 &\leq 3 \\ x_1 + x_2 &\leq 2 \\ x_1,\ x_2 &\geq 0 \end{aligned}$$

iii. geometric reasons (non-simple polyhedra)

   Example: Octahedron

Observation 6.14 (cp. Proof of Lemma 3.26).

Perturbing the right hand side vector b may remove degeneracy.

Existence of Extreme Points

Definition 6.15 (cp. Proof of Theorem 2.11).

A polyhedron $P \subseteq \mathbb{R}^n$ contains a line if there is $x \in P$ and a direction $d \in \mathbb{R}^n \setminus \{0\}$ such that

$$x + \lambda \cdot d \in P \qquad \text{for all } \lambda \in \mathbb{R}.$$

Theorem 6.16.

Let $P = \{x \in \mathbb{R}^n \mid A \cdot x \geq b\} \neq \emptyset$ with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. The following are equivalent:

i. There exists an extreme point $x \in P$.
ii. P does not contain a line.
iii. A contains n linearly independent rows.

Proof: . . . □

Existence of Extreme Points (cont.)

Corollary 6.17.

a. A non-empty polytope contains an extreme point.
b. A non-empty polyhedron in standard form contains an extreme point.

Proof of b:

$$\begin{aligned} A \cdot x &= b \\ x &\geq 0 \end{aligned} \qquad \longleftrightarrow \qquad \begin{pmatrix} A \\ -A \\ I \end{pmatrix} \cdot x \geq \begin{pmatrix} b \\ -b \\ 0 \end{pmatrix}$$

□

Example:

$$P = \left\{ \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \mathbb{R}^3 \;\middle|\; \begin{array}{rcrl} x_1 & + & x_2 & \geq 1 \\ x_1 & + & 2\,x_2 & \geq 0 \end{array} \right\}$$

contains a line since $\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \lambda \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \in P \quad \text{for all } \lambda \in \mathbb{R}.$

Optimality of Extreme Points

Theorem 6.18.

Let $P \subseteq \mathbb{R}^n$ a polyhedron and $c \in \mathbb{R}^n$. If P has an extreme point and $\min\{c^T \cdot x \mid x \in P\}$ is bounded, there is an extreme point that is optimal.

Proof: ... $\square$

Corollary 6.19.

Every linear programming problem is either infeasible or unbounded or there exists an optimal solution.

Proof: Every linear program is equivalent to an LP in standard form. The claim thus follows from Corollary 6.17 and Theorem 6.18. $\square$

Chapter 7:
The Simplex Method

(cp. Bertsimas & Tsitsiklis, Chapter 3)

Linear Program in Standard Form

Throughout this chapter, we consider the following standard form problem:

$$\begin{aligned} \text{minimize} \quad & c^T \cdot x \\ \text{subject to} \quad & A \cdot x = b \\ & x \geq 0 \end{aligned}$$

with $A \in \mathbb{R}^{m \times n}$, $\operatorname{rank}(A) = m$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$.

Basic Directions

Observation 7.1.

Let $B = (A_{B(1)}, \ldots, A_{B(m)})$ be a basis matrix. The values of the basic variables $x_{B(1)}, \ldots, x_{B(m)}$ in the system $A \cdot x = b$ are uniquely determined by the values of the non-basic variables.

Proof:

$$\begin{aligned} A \cdot x = b \quad & \Longleftrightarrow \quad B \cdot x_B + \sum_{j \neq B(1), \ldots, B(m)} A_j \cdot x_j = b \\ & \Longleftrightarrow \quad x_B = B^{-1} \cdot b - \sum_{j \neq B(1), \ldots, B(m)} B^{-1} \cdot A_j \cdot x_j \end{aligned}$$

$\square$

Definition 7.2.

For fixed $j \neq B(1), \ldots, B(m)$, let $d \in \mathbb{R}^n$ be given by

$$d_j := 1, \quad d_B := -B^{-1} \cdot A_j, \quad \text{and} \quad d_{j'} := 0 \quad \text{for } j' \neq j, B(1), \ldots, B(m).$$

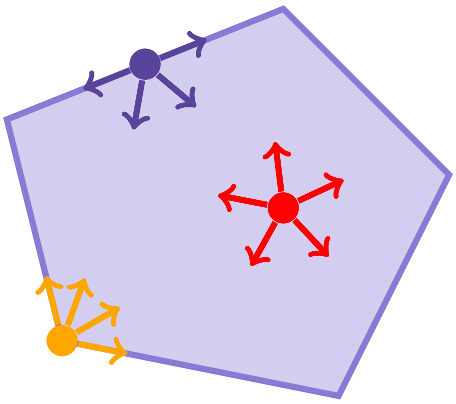
Then $A \cdot (x + \theta \cdot d) = b$, for all $\theta \in \mathbb{R}$, and d is the *jth basic direction*.

Feasible Directions

Definition 7.3.

Let $P \subseteq \mathbb{R}^n$ a polyhedron. For $x \in P$ the vector $d \in \mathbb{R}^n \setminus \{0\}$ is a feasible direction at x if there is a $\theta > 0$ with $x + \theta \cdot d \in P$.

Example: Some feasible directions at several points of a polyhedron.



Feasible Directions

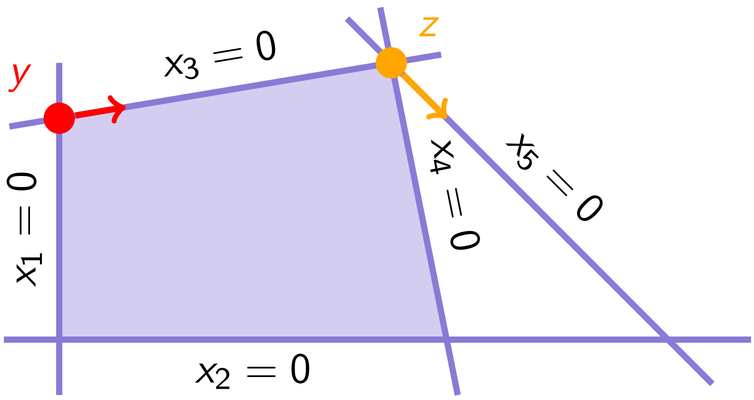
Consider a basic feasible solution x.

Question: Is the jth basic directions d a feasible direction?

Case 1: If x is a non-degenerate feasible solution, then $x_B > 0$ and $x + \theta \cdot d \geq 0$ for $\theta > 0$ small enough. $\longrightarrow$ answer is yes!

Case 2: If x is degenerate, the answer might be no!
E. g., if $x_{B(i)} = 0$ and $d_{B(i)} < 0$, then $x + \theta \cdot d \not\geq 0$, for all $\theta > 0$.

Example: $n = 5$, $m = 3$, $n - m = 2$



- 1st basic direction at y (basic variables x_2, x_4, x_5)
- 3rd basic direction at z (basic variables x_1, x_2, x_4)

Reduced Cost Coefficients

Consider a basic solution x.

Question:

How does the cost change when moving along the jth basic direction d?

$$c^T \cdot (x + \theta \cdot d) = c^T \cdot x + \theta \cdot c^T \cdot d = c^T \cdot x + \theta \cdot \underbrace{(c_j - c_B{}^T \cdot B^{-1} \cdot A_j)}_{\bar{c}_j :=}$$

Definition 7.4.

For a given basic solution x, the reduced cost of variable x_j, $j = 1, \dots, n$, is

$$\bar{c}_j := c_j - c_B{}^T \cdot B^{-1} \cdot A_j \ .$$

Observation 7.5.

The reduced cost of a basic variable $x_{B(i)}$ is zero.

Proof: $\bar{c}_{B(i)} = c_{B(i)} - c_B{}^T \cdot \underbrace{B^{-1} \cdot A_{B(i)}}_{= e_i} = c_{B(i)} - c_{B(i)} = 0$ □

Optimality Criterion

Theorem 7.6.

Let x be a basic feasible solution and $\bar{c}$ the vector of reduced costs.

a) If $\bar{c} \geq 0$, then x is an optimal solution.

b) If x is an optimal solution and non-degenerate, then $\bar{c} \geq 0$.

Proof: . . . □

Definition 7.7.

A basis matrix B is optimal if

i) $B^{-1} \cdot b \geq 0$ and

ii) $\bar{c}^T = c^T - c_B{}^T \cdot B^{-1} \cdot A \geq 0$.

Observation 7.8.

If B is an optimal basis, the associated basic solution x is feasible and optimal.

Developement of the Simplex Method

Assumption (for now): only *non-degenerate* basic feasible solutions

Let x be a basic feasible solution with $\bar{c}_j < 0$ for some $j \neq B(1), \dots, B(m)$.

Let d be the jth basic direction:

$$0 > \bar{c}_j = c^T \cdot d$$

It is desirable to go to $y := x + \theta^* \cdot d$ with $\theta^* := \max\{\theta \mid x + \theta \cdot d \in P\}$.

Question: How to determine θ^*?

By construction of d, it holds that $A \cdot (x + \theta \cdot d) = b$ for all $\theta \in \mathbb{R}$, i. e.,

$$x + \theta \cdot d \in P \quad \Longleftrightarrow \quad x + \theta \cdot d \geq 0 \ .$$

Case 1: $d \geq 0 \quad \Longrightarrow \quad x + \theta \cdot d \geq 0$ for all $\theta \geq 0 \quad \Longrightarrow \quad \theta^* = \infty$

Thus, the LP is unbounded.

Case 2: $d_k < 0$ for some $k \quad \Longrightarrow \quad \left(x_k + \theta\, d_k \geq 0 \quad \Longleftrightarrow \quad \theta \leq \dfrac{-x_k}{d_k} \right)$

Thus, $\theta^* = \min\limits_{k:\, d_k < 0} \dfrac{-x_k}{d_k} = \min\limits_{\substack{i=1,\dots,m \\ d_{B(i)} < 0}} \dfrac{-x_{B(i)}}{d_{B(i)}} > 0$.

Developement of the Simplex Method (cont.)

Assumption (for now): only *non-degenerate* basic feasible solutions

Let x be a basic feasible solution with $\bar{c}_j < 0$ for some $j \neq B(1), \dots, B(m)$.

Let d be the jth basic direction:

$$0 > \bar{c}_j = c^T \cdot d$$

It is desirable to go to $y := x + \theta^* \cdot d$ with $\theta^* := \max\{\theta \mid x + \theta \cdot d \in P\}$.

$$\theta^* = \min_{k:\, d_k < 0} \frac{-x_k}{d_k} = \min_{\substack{i=1,\dots,m \\ d_{B(i)} < 0}} \frac{-x_{B(i)}}{d_{B(i)}}$$

Let $\ell \in \{1, \dots, m\}$ with $\theta^* = \dfrac{-x_{B(\ell)}}{d_{B(\ell)}}$, then $y_j = \theta^*$ and $y_{B(\ell)} = 0$.

$\Longrightarrow \quad x_j$ replaces $x_{B(\ell)}$ as a basic variable and we get a new basis matrix

$$\bar{B} = \left(A_{B(1)}, \dots, A_{B(\ell-1)}, A_j, A_{B(\ell+1)}, \dots, A_{B(m)} \right) = \left(A_{\bar{B}(1)}, \dots, A_{\bar{B}(m)} \right)$$

with $\quad \bar{B}(i) = \begin{cases} B(i) & \text{if } i \neq \ell, \\ j & \text{if } i = \ell. \end{cases}$

Core of the Simplex Method

Theorem 7.9.

Let x be a non-degenerate basic feasible solution, $j \neq B(1), \dots, B(m)$ with $\bar{c}_j < 0$, d the jth basic direction, and $\theta^* := \max\{\theta \mid x + \theta \cdot d \in P\} < \infty$.

a $\theta^* = \min\limits_{\substack{i=1,\dots,m \\ d_{B(i)}<0}} \frac{-x_{B(i)}}{d_{B(i)}} = \frac{-x_{B(\ell)}}{d_{B(\ell)}}$ for some $\ell \in \{1, \dots, m\}$.

Let $\bar{B}(i) := B(i)$ for $i \neq \ell$ and $\bar{B}(\ell) := j$.

b $A_{\bar{B}(1)}, \dots, A_{\bar{B}(m)}$ are linearly independent and $\bar{B}$ is a basis matrix.

c $y := x + \theta^* \cdot d$ is a basic feasible solution associated with $\bar{B}$ and $c^T \cdot y < c^T \cdot x$.

Proof: ... □

An Iteration of the Simplex Method

Given: basis $B = (A_{B(1)} \dots A_{B(m)})$, corresponding basic feasible solution x

1. Let $\bar{c}^T := c^T - {c_B}^T \cdot B^{-1} \cdot A$. If $\bar{c} \geq 0$, then STOP; else choose j with $\bar{c}_j < 0$.
2. Let $u := B^{-1} \cdot A_j$. If $u \leq 0$, then STOP (optimal cost is $-\infty$).
3. Let $\theta^* := \min\limits_{i:u_i>0} \frac{x_{B(i)}}{u_i} = \frac{x_{B(\ell)}}{u_\ell}$ for some $\ell \in \{1, \dots, m\}$.
4. Form new basis by replacing $A_{B(\ell)}$ with A_j; corresponding basic feasible solution y is given by

$$y_j := \theta^* \qquad \text{and} \qquad y_{B(i)} = x_{B(i)} - \theta^* u_i \quad \text{for } i \neq \ell.$$

Remark: We say that the nonbasic variable x_j enters the basis and the basic variable $x_{B(\ell)}$ leaves the basis.

Correctness of the Simplex Method

Theorem 7.10.

If every basic feasible solution is non-degenerate, the simplex method terminates after finitely many iterations in one of the following two states:

- i we have an optimal basis B and an associated basic feasible solution x which is optimal;
- ii we have a vector d satisfying $A \cdot d = 0$, $d \geq 0$, and $c^T \cdot d < 0$; the optimal cost is $-\infty$.

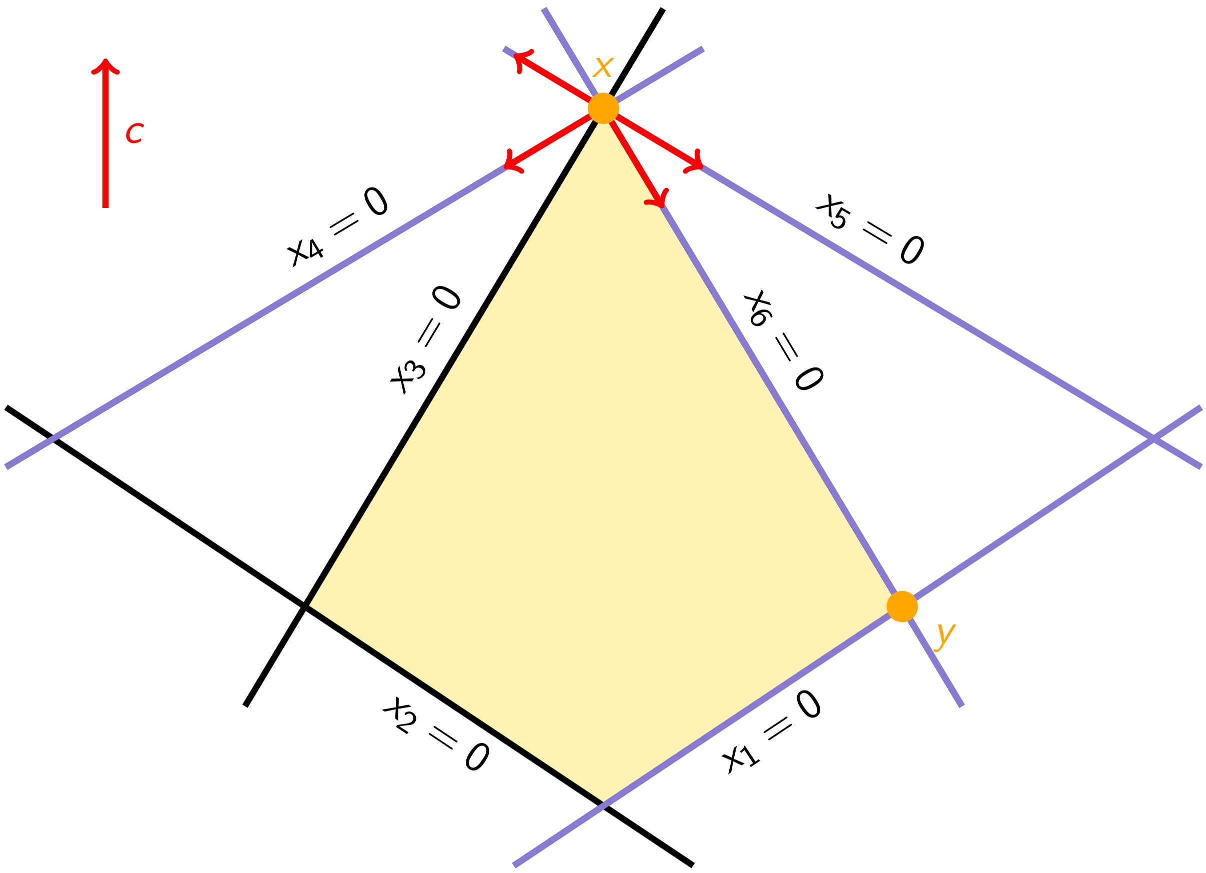
Proof sketch: The simplex method makes progress in every iteration. Since there are only finitely many different basic feasible solutions, it stops after a finite number of iterations. □

Simplex Method for Degenerate Problems

- An iteration of the simplex method can also be applied if x is a degenerate basic feasible solution.
- In this case it might happen that $\theta^* := \min\limits_{i:u_i>0} \dfrac{x_{B(i)}}{u_i} = \dfrac{x_{B(\ell)}}{u_\ell} = 0$ if some basic variable $x_{B(\ell)}$ is zero and $d_{B(\ell)} < 0$.
- Thus, $y = x + \theta^* \cdot d = x$ and the current basic feasible solution does not change.
- But replacing $A_{B(\ell)}$ with A_j still yields a new basis with associated basic feasible solution $y = x$.

Remark: Even if θ^* is positive, more than one of the original basic variables may become zero at the new point $x + \theta^* \cdot d$. Since only one of them leaves the basis, the new basic feasible solution y is degenerate.

Example



Pivot Selection

Question: How to choose j with $\bar{c}_j < 0$ and ℓ with $\dfrac{x_{B(\ell)}}{u_\ell} = \min\limits_{i:u_i>0} \dfrac{x_{B(i)}}{u_i}$ if several possible choices exist?

Attention: Choice of j is critical for overall behavior of simplex method.

Three popular choices are:

- smallest subscript rule: choose smallest j with $\bar{c}_j < 0$.
  (very simple; no need to compute entire vector $\bar{c}$; usually leads to many iterations)
- steepest descent rule: choose j such that $\bar{c}_j < 0$ is minimal.
  (relatively simple; commonly used for mid-size problems; does not necessarily yield the best neighboring solution)
- best improvement rule: choose j such that $\theta^* \bar{c}_j$ is minimal.
  (computationally expensive; used for large problems; usually leads to very few iterations)

Revised Simplex Method

Observation 7.11.

To execute one iteration of the simplex method efficiently, it suffices to know $B(1), \dots, B(m)$, the inverse B^{-1} of the basis matrix and the input data A, b, and c. It is then easy to compute:

$$x_B = B^{-1} \cdot b \qquad \bar{c}^T = c^T - {c_B}^T \cdot B^{-1} \cdot A$$

$$u = B^{-1} \cdot A_j \qquad \theta^* = \min_{i : u_i > 0} \frac{x_{B(i)}}{u_i} = \frac{x_{B(\ell)}}{u_\ell}$$

The new basis matrix is then

$$\bar{B} = \left(A_{B(1)}, \dots, A_{B(\ell-1)}, A_j, A_{B(\ell+1)}, \dots, A_{B(m)}\right)$$

Critical question: How to obtain $\bar{B}^{-1}$ efficiently?

Computing the Inverse of the Basis Matrix

- Notice that $B^{-1} \cdot \bar{B} = (e_1, \dots, e_{\ell-1}, u, e_{\ell+1}, \dots, e_m)$.
- Thus, $\bar{B}^{-1}$ can be obtained from B^{-1} as follows:
  - multiply ℓth row of B^{-1} with $1/u_\ell$;
  - for $i \neq \ell$, subtract u_i times resulting ℓth row from ith row.
- These are exactly the elementary row operations needed to turn $B^{-1} \cdot \bar{B}$ into the identity matrix!
- Elementary row operations are the same as multiplying the matrix with corresponding elementary matrices from the left hand side.
- Equivalently:

Obtaining $\bar{B}^{-1}$ from B^{-1}

Apply elementary row operations to the matrix $(B^{-1} \mid u)$ to make the last column equal to the unit vector e_ℓ. The first m columns of the resulting matrix form the inverse $\bar{B}^{-1}$ of the new basis matrix $\bar{B}$.

An Iteration of the Revised Simplex Method

Given: $B = \left(A_{B(1)}, \dots, A_{B(m)}\right)$, corresp. basic feasible sol. x, and B^{-1}.

1. Let $p^T := {c_B}^T \cdot B^{-1}$ and $\bar{c}_j := c_j - p^T \cdot A_j$, $j \neq B(1), \dots, B(m)$;

   if $\bar{c} \geq 0$, then STOP; else choose j with $\bar{c}_j < 0$.

2. Let $u := B^{-1} \cdot A_j$. If $u \leq 0$, then STOP (optimal cost is $-\infty$).

3. Let $\theta^* := \min\limits_{i:u_i>0} \dfrac{x_{B(i)}}{u_i} = \dfrac{x_{B(\ell)}}{u_\ell}$ for some $\ell \in \{1, \dots, m\}$.

4. Form new basis by replacing $A_{B(\ell)}$ with A_j; corresponding basic feasible solution y is given by

   $$y_j := \theta^* \qquad \text{and} \qquad y_{B(i)} = x_{B(i)} - \theta^* \, u_i \quad \text{for } i \neq \ell.$$

5. Apply elementary row operations to the matrix $(B^{-1} \mid u)$ to make the last column equal to the unit vector e_ℓ.

   The first m columns of the resulting matrix yield $\bar{B}^{-1}$.

Full Tableau Implementation

Main idea

Instead of maintaining and updating the matrix B^{-1}, we maintain and update the $m \times (n+1)$-matrix

$$B^{-1} \cdot (b \mid A) = (B^{-1} \cdot b \mid B^{-1} \cdot A)$$

which is called simplex tableau.

- The zeroth column $B^{-1} \cdot b$ contains x_B.
- For $i = 1, \dots, n$, the ith column of the tableau is $B^{-1} \cdot A_i$.
- The column $u = B^{-1} \cdot A_j$ corresponding to the variable x_j that is about to enter the basis is the pivot column.
- If the ℓth basic variable $x_{B(\ell)}$ exits the basis, the ℓth row of the tableau is the pivot row.
- The element $u_\ell > 0$ is the pivot element.

Full Tableau Implementation (cont.)

Notice: The simplex tableau $B^{-1} \cdot (b \mid A)$ represents the linear equation

$$B^{-1} \cdot b = B^{-1} \cdot A \cdot x$$

which is equivalent to $A \cdot x = b$.

Updating the simplex tableau

- At the end of an iteration, the simplex tableau $B^{-1} \cdot (b \mid A)$ has to be updated to $\bar{B}^{-1} \cdot (b \mid A)$.
- $\bar{B}^{-1}$ can be obtained from B^{-1} by elementary row operations, i. e., $\bar{B}^{-1} = Q \cdot B^{-1}$ where Q is a product of elementary matrices.
- Thus, $\bar{B}^{-1} \cdot (b \mid A) = Q \cdot B^{-1} \cdot (b \mid A)$ and new tableau $\bar{B}^{-1} \cdot (b \mid A)$ can be obtained by applying the same elementary row operations.

Zeroth Row of the Simplex Tableau

In order to keep track of the objective function value and the reduced costs, we consider the following augmented simplex tableau:

$-c_B{}^T B^{-1} b$	$c^T - c_B{}^T B^{-1} A$
$B^{-1} b$	$B^{-1} A$

or in more detail

$$\begin{array}{c|ccc}
-c_B{}^T x_B & \bar{c}_1 & \cdots & \bar{c}_n \\
\hline
x_{B(1)} & | & & | \\
\vdots & B^{-1}A_1 & \cdots & B^{-1}A_n \\
x_{B(m)} & | & & |
\end{array}$$

Update after one iteration

The zeroth row is updated by adding a multiple of the pivot row to the zeroth row to set the reduced cost of the entering variable to zero.

An Iteration of the Full Tableau Implementation

Given: Simplex tableau corresp. to feasible basis $B = \big(A_{B(1)}, \dots, A_{B(m)}\big)$.

1. If $\bar{c} \geq 0$ (zeroth row), then STOP; else choose pivot column j with $\bar{c}_j < 0$.
2. If $u = B^{-1}A_j \leq 0$ (jth column), STOP (optimal cost is $-\infty$).
3. Let $\theta^* := \min\limits_{i:u_i>0} \frac{x_{B(i)}}{u_i} = \frac{x_{B(\ell)}}{u_\ell}$ for some $\ell \in \{1, \dots, m\}$ (cp. columns 0 and j).
4. Form new basis by replacing $A_{B(\ell)}$ with A_j.
5. Apply elementary row operations to the simplex tableau so that u_ℓ (pivot element) becomes one and all other entries of the pivot column become zero.

Full Tableau Implementation: An Example

A simple linear programming problem:

$$
\begin{array}{lrcrcrcr}
\min & -10\,x_1 & - & 12\,x_2 & - & 12\,x_3 & & \\
\text{s.t.} & x_1 & + & 2\,x_2 & + & 2\,x_3 & \leq & 20 \\
 & 2\,x_1 & + & x_2 & + & 2\,x_3 & \leq & 20 \\
 & 2\,x_1 & + & 2\,x_2 & + & x_3 & \leq & 20 \\
 & & & & & x_1, x_2, x_3 & \geq & 0
\end{array}
$$

Set of Feasible Solutions

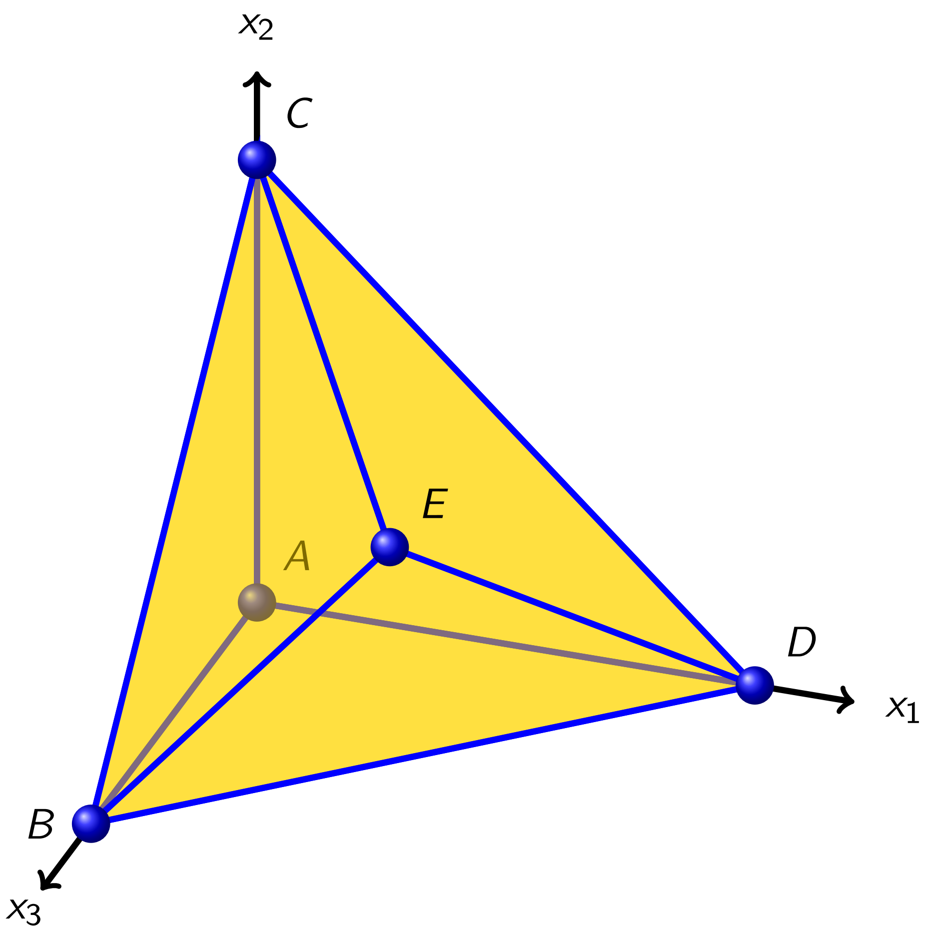
$A = (0, 0, 0)^T$

$B = (0, 0, 10)^T$

$C = (0, 10, 0)^T$

$D = (10, 0, 0)^T$

$E = (4, 4, 4)^T$



Introducing Slack Variables

$$
\begin{array}{lrcrcrcr}
\min & -10\,x_1 & - & 12\,x_2 & - & 12\,x_3 & & \\
\text{s.t.} & x_1 & + & 2\,x_2 & + & 2\,x_3 & \leq & 20 \\
 & 2\,x_1 & + & x_2 & + & 2\,x_3 & \leq & 20 \\
 & 2\,x_1 & + & 2\,x_2 & + & x_3 & \leq & 20 \\
 & & & & & x_1, x_2, x_3 & \geq & 0
\end{array}
$$

LP in standard form

$$
\begin{array}{lrcrcrcccccccr}
\min & -10\,x_1 & - & 12\,x_2 & - & 12\,x_3 & & & & & & & & \\
\text{s.t.} & x_1 & + & 2\,x_2 & + & 2\,x_3 & + & x_4 & & & & & = & 20 \\
 & 2\,x_1 & + & x_2 & + & 2\,x_3 & & & + & x_5 & & & = & 20 \\
 & 2\,x_1 & + & 2\,x_2 & + & x_3 & & & & & + & x_6 & = & 20 \\
 & & & & & & & & & & & x_1, \ldots, x_6 & \geq & 0
\end{array}
$$

Observation

The right hand side of the system is non-negative. Therefore the point $(0, 0, 0, 20, 20, 20)^T$ is a basic feasible solution and we can start the simplex method with basis $B(1) = 4, B(2) = 5, B(3) = 6$.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6
	0	-10	-12	-12	0	0	0
$x_4 =$	20	1	2	2	1	0	0
$x_5 =$	20	2	1	2	0	1	0
$x_6 =$	20	2	2	1	0	0	1

- Determine pivot column (e. g., take smallest subscript rule).

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6	$\frac{x_{B(i)}}{u_i}$
	0	-10	-12	-12	0	0	0	
$x_4 =$	20	1	2	2	1	0	0	20
$x_5 =$	20	2	1	2	0	1	0	10
$x_6 =$	20	2	2	1	0	0	1	10

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6	$\frac{x_{B(i)}}{u_i}$
	0	-10	-12	-12	0	0	0	
$x_4 =$	20	1	2	2	1	0	0	20
$x_5 =$	20	2	1	2	0	1	0	10
$x_6 =$	20	2	2	1	0	0	1	10

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.
- Choose $i = 2$ with $B(i) = 5$. $\implies$ x_5 leaves the basis.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6
	0	-10	-12	-12	0	0	0
$x_4 =$	20	1	2	2	1	0	0
$x_5 =$	20	2	1	2	0	1	0
$x_6 =$	20	2	2	1	0	0	1

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.
- Choose $i = 2$ with $B(i) = 5$. $\implies$ x_5 leaves the basis.
- Perform basis change: Eliminate other entries in the pivot column.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	-7	-2	0	5	0
$x_4 =$	20	1	2	2	1	0	0
$x_5 =$	20	2	1	2	0	1	0
$x_6 =$	20	2	2	1	0	0	1

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.
- Choose $i = 2$ with $B(i) = 5$. $\Longrightarrow$ x_5 leaves the basis.
- Perform basis change: Eliminate other entries in the pivot column.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	-7	-2	0	5	0
$x_4 =$	20	1	2	2	1	0	0
$x_5 =$	20	2	1	2	0	1	0
$x_6 =$	20	2	2	1	0	0	1

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.
- Choose $i = 2$ with $B(i) = 5$. $\Longrightarrow$ x_5 leaves the basis.
- Perform basis change: Eliminate other entries in the pivot column.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	-7	-2	0	5	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_5 =$	20	2	1	2	0	1	0
$x_6 =$	20	2	2	1	0	0	1

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.
- Choose $i = 2$ with $B(i) = 5$. $\Longrightarrow$ x_5 leaves the basis.
- Perform basis change: Eliminate other entries in the pivot column.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	-7	-2	0	5	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_5 =$	20	2	1	2	0	1	0
$x_6 =$	20	2	2	1	0	0	1

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.
- Choose $i = 2$ with $B(i) = 5$. $\Longrightarrow$ x_5 leaves the basis.
- Perform basis change: Eliminate other entries in the pivot column.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	−7	−2	0	5	0
$x_4 =$	10	0	1.5	1	1	−0.5	0
$x_5 =$	20	2	1	2	0	1	0
$x_6 =$	0	0	1	−1	0	−1	1

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.
- Choose $i = 2$ with $B(i) = 5$. $\implies$ x_5 leaves the basis.
- Perform basis change: Eliminate other entries in the pivot column.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	−7	−2	0	5	0
$x_4 =$	10	0	1.5	1	1	−0.5	0
$x_5 =$	20	2	1	2	0	1	0
$x_6 =$	0	0	1	−1	0	−1	1

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.
- Choose $i = 2$ with $B(i) = 5$. $\implies$ x_5 leaves the basis.
- Perform basis change: Eliminate other entries in the pivot column.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	-7	-2	0	5	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	10	1	0.5	1	0	0.5	0
$x_6 =$	0	0	1	-1	0	-1	1

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.
- Choose $i = 2$ with $B(i) = 5$. $\implies$ x_5 leaves the basis.
- Perform basis change: Eliminate other entries in the pivot column.

Setting Up the Simplex Tableau

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	-7	-2	0	5	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	10	1	0.5	1	0	0.5	0
$x_6 =$	0	0	1	-1	0	-1	1

- Determine pivot column (e. g., take smallest subscript rule).
- $\bar{c}_1 < 0$ and x_1 enters the basis.
- Find pivot row with $u_i > 0$ minimizing $\frac{x_{B(i)}}{u_i}$.
- Rows 2 and 3 both attain the minimum.
- Choose $i = 2$ with $B(i) = 5$. $\implies$ x_5 leaves the basis.
- Perform basis change: Eliminate other entries in the pivot column.
- Obtain new basic feasible solution $(10, 0, 0, 10, 0, 0)^T$ with cost -100.

Geometric Interpretation in the Original Polyhedron

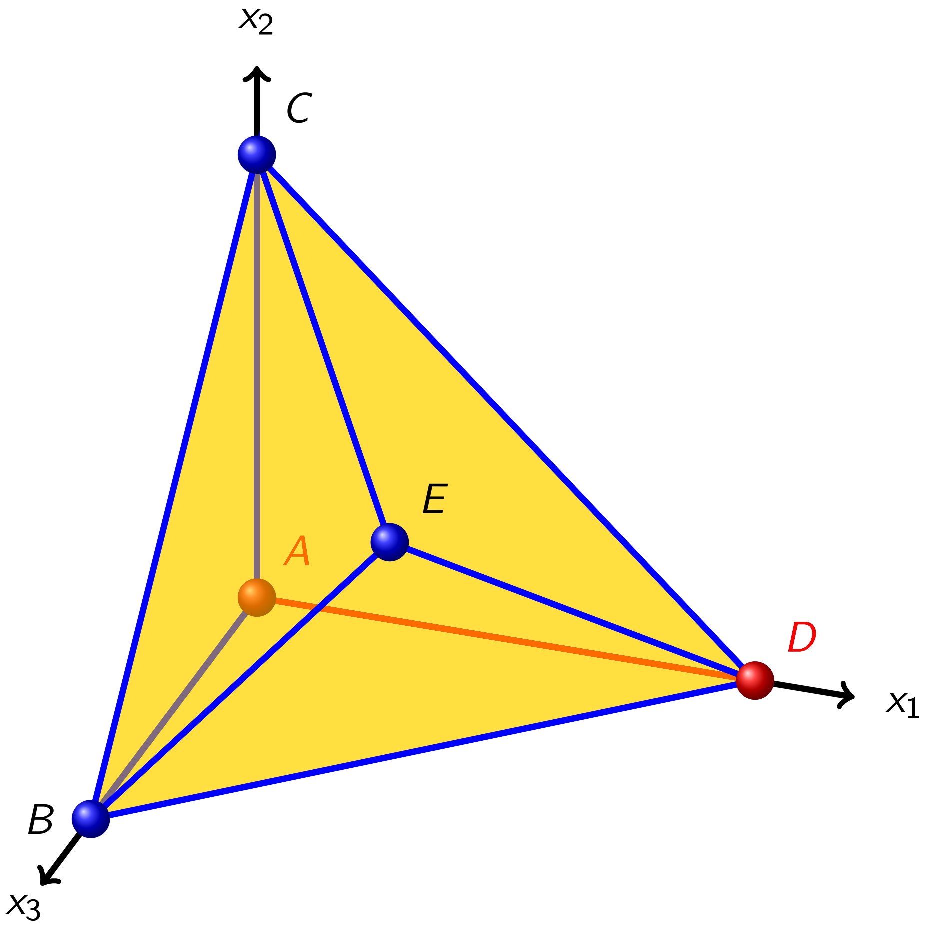
$A = (0, 0, 0)^T$

$B = (0, 0, 10)^T$

$C = (0, 10, 0)^T$

$D = (10, 0, 0)^T$

$E = (4, 4, 4)^T$



Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	-7	-2	0	5	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	10	1	0.5	1	0	0.5	0
$x_6 =$	0	0	1	-1	0	-1	1

- $\bar{c}_2, \bar{c}_3 < 0 \implies$ two possible choices for pivot column.

Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6	$\frac{x_{B(i)}}{u_i}$
	100	0	-7	-2	0	5	0	
$x_4 =$	10	0	1.5	1	1	-0.5	0	10
$x_1 =$	10	1	0.5	1	0	0.5	0	10
$x_6 =$	0	0	1	-1	0	-1	1	–

- $\bar{c}_2, \bar{c}_3 < 0 \implies$ two possible choices for pivot column.
- Choose x_3 to enter the new basis.
- $u_3 < 0 \implies$ third row cannot be chosen as pivot row.
- Choose x_4 to leave basis.

Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	-7	-2	0	5	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	10	1	0.5	1	0	0.5	0
$x_6 =$	0	0	1	-1	0	-1	1

- $\bar{c}_2, \bar{c}_3 < 0 \implies$ two possible choices for pivot column.
- Choose x_3 to enter the new basis.
- $u_3 < 0 \implies$ third row cannot be chosen as pivot row.
- Choose x_4 to leave basis.

Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6
	120	0	-4	0	2	4	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	10	1	0.5	1	0	0.5	0
$x_6 =$	0	0	1	-1	0	-1	1

- $\bar{c}_2, \bar{c}_3 < 0 \implies$ two possible choices for pivot column.
- Choose x_3 to enter the new basis.
- $u_3 < 0 \implies$ third row cannot be chosen as pivot row.
- Choose x_4 to leave basis.

Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6
	120	0	-4	0	2	4	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	10	1	0.5	1	0	0.5	0
$x_6 =$	0	0	1	-1	0	-1	1

- $\bar{c}_2, \bar{c}_3 < 0 \implies$ two possible choices for pivot column.
- Choose x_3 to enter the new basis.
- $u_3 < 0 \implies$ third row cannot be chosen as pivot row.
- Choose x_4 to leave basis.

Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6
	120	0	-4	0	2	4	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	0	1	-1	0	-1	1	0
$x_6 =$	0	0	1	-1	0	-1	1

- $\bar{c}_2, \bar{c}_3 < 0 \implies$ two possible choices for pivot column.
- Choose x_3 to enter the new basis.
- $u_3 < 0 \implies$ third row cannot be chosen as pivot row.
- Choose x_4 to leave basis.

Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6
	120	0	-4	0	2	4	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	0	1	-1	0	-1	1	0
$x_6 =$	0	0	1	-1	0	-1	1

- $\bar{c}_2, \bar{c}_3 < 0 \implies$ two possible choices for pivot column.
- Choose x_3 to enter the new basis.
- $u_3 < 0 \implies$ third row cannot be chosen as pivot row.
- Choose x_4 to leave basis.

Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6
	120	0	-4	0	2	4	0
$x_4 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	0	1	-1	0	-1	1	0
$x_6 =$	10	0	2.5	0	1	-1.5	1

- $\bar{c}_2, \bar{c}_3 < 0 \implies$ two possible choices for pivot column.
- Choose x_3 to enter the new basis.
- $u_3 < 0 \implies$ third row cannot be chosen as pivot row.
- Choose x_4 to leave basis.

Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6
	120	0	-4	0	2	4	0
$x_3 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	0	1	-1	0	-1	1	0
$x_6 =$	10	0	2.5	0	1	-1.5	1

- $\bar{c}_2, \bar{c}_3 < 0 \implies$ two possible choices for pivot column.
- Choose x_3 to enter the new basis.
- $u_3 < 0 \implies$ third row cannot be chosen as pivot row.
- Choose x_4 to leave basis.
- New basic feasible solution $(0, 0, 10, 0, 0, 10)^T$ with cost -120, corresponding to point B in the original polyhedron.

Geometric Interpretation in the Original Polyhedron

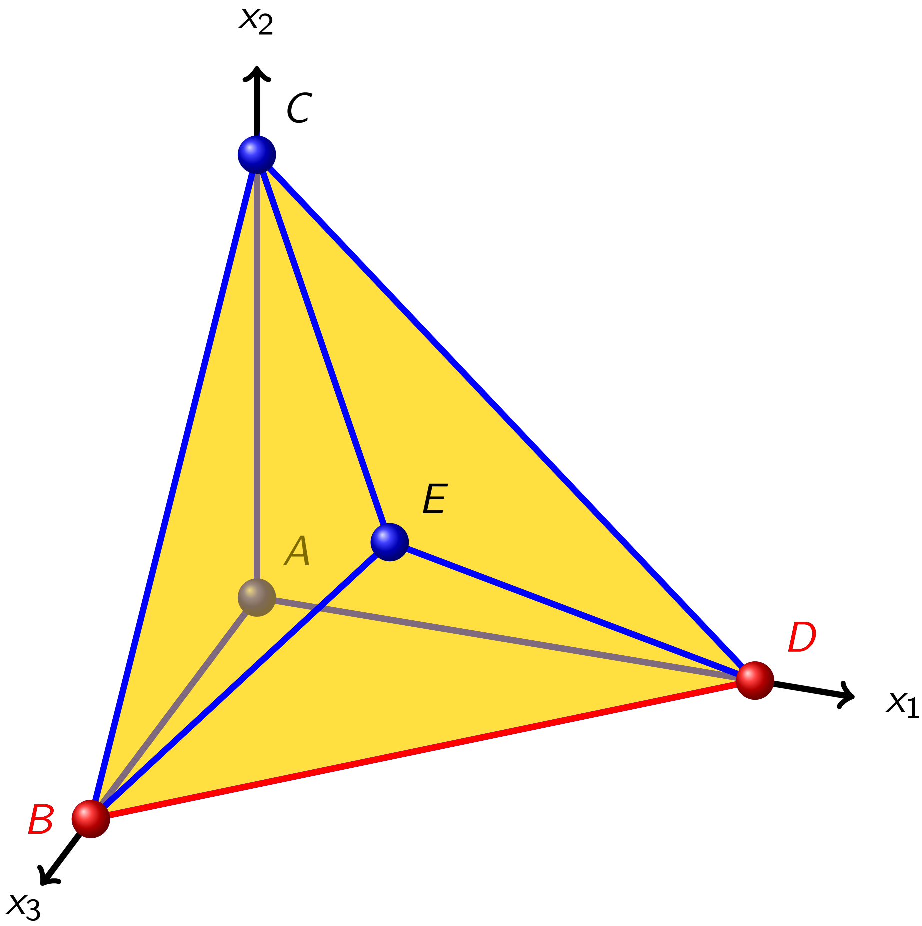
$A = (0,0,0)^T$

$B = (0,0,10)^T$

$C = (0,10,0)^T$

$D = (10,0,0)^T$

$E = (4,4,4)^T$



Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6	$\frac{x_{B(i)}}{u_i}$
	120	0	-4	0	2	4	0	
$x_3 =$	10	0	1.5	1	1	-0.5	0	$\frac{20}{3}$
$x_1 =$	0	1	-1	0	-1	1	0	—
$x_6 =$	10	0	2.5	0	1	-1.5	1	4

Thus $(4,4,4,0,0,0)$ is an optimal solution with cost -136, corresponding to point $E = (4,4,4)$ in the original polyhedron.

Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6	$\frac{x_{B(i)}}{u_i}$
	120	0	-4	0	2	4	0	
$x_3 =$	10	0	1.5	1	1	-0.5	0	$\frac{20}{3}$
$x_1 =$	0	1	-1	0	-1	1	0	$-$
$x_6 =$	10	0	2.5	0	1	-1.5	1	$4 < \frac{20}{3}$

Thus $(4, 4, 4, 0, 0, 0)$ is an optimal solution with cost -136, corresponding to point $E = (4, 4, 4)$ in the original polyhedron.

Next Iterations

		x_1	x_2	x_3	x_4	x_5	x_6
	120	0	-4	0	2	4	0
$x_3 =$	10	0	1.5	1	1	-0.5	0
$x_1 =$	0	1	-1	0	-1	1	0
$x_6 =$	10	0	2.5	0	1	-1.5	1

x_2 enters the basis, x_6 leaves it. We get

		x_1	x_2	x_3	x_4	x_5	x_6
	136	0	0	0	3.6	1.6	1.6
$x_3 =$	4	0	0	1	0.4	0.4	-0.6
$x_1 =$	4	1	0	0	-0.6	0.4	0.4
$x_2 =$	4	0	1	0	0.4	-0.6	0.4

and the reduced costs are all non-negative.

Thus $(4, 4, 4, 0, 0, 0)$ is an optimal solution with cost -136, corresponding to point $E = (4, 4, 4)$ in the original polyhedron.

All Iterations from Geometric Point of View

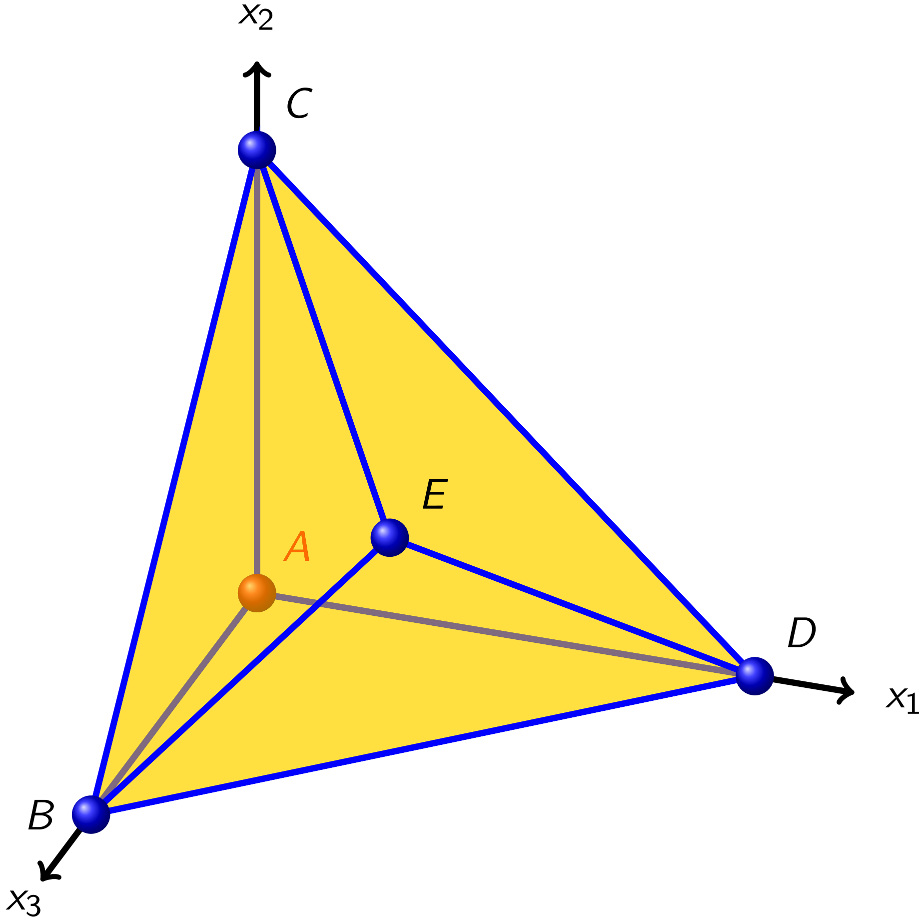
$A = (0,0,0)^T$

$B = (0,0,10)^T$

$C = (0,10,0)^T$

$D = (10,0,0)^T$

$E = (4,4,4)^T$



All Iterations from Geometric Point of View

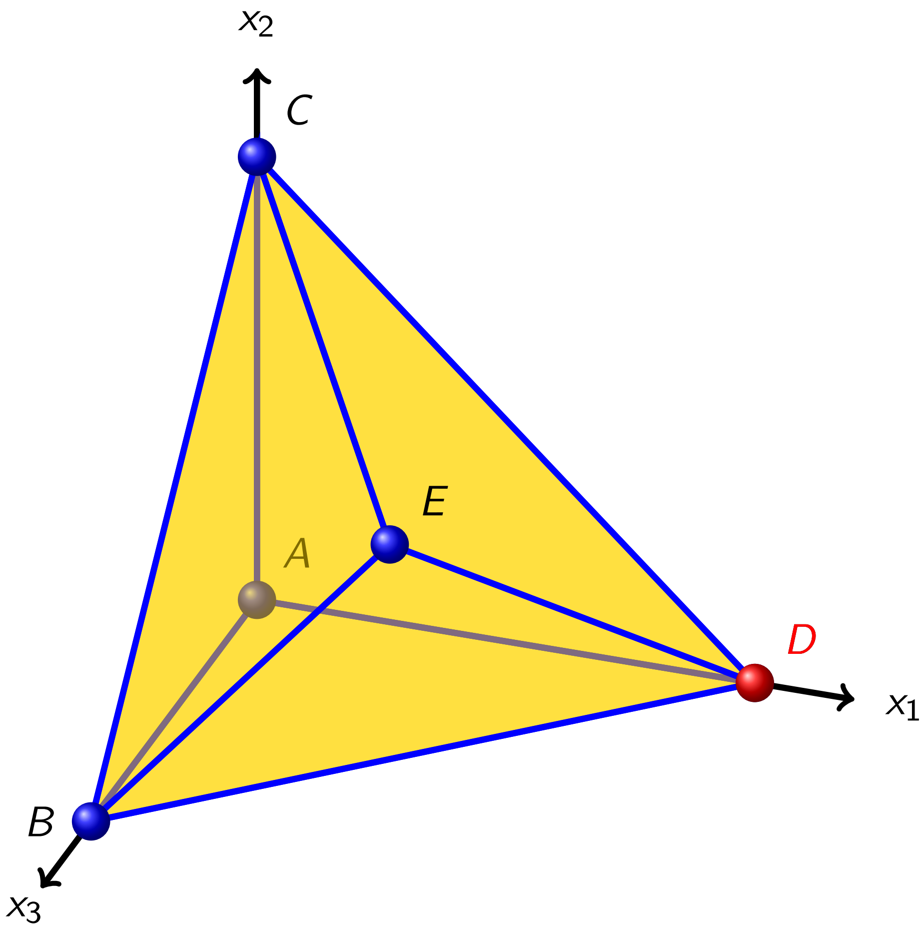
$A = (0,0,0)^T$

$B = (0,0,10)^T$

$C = (0,10,0)^T$

$D = (10,0,0)^T$

$E = (4,4,4)^T$



All Iterations from Geometric Point of View

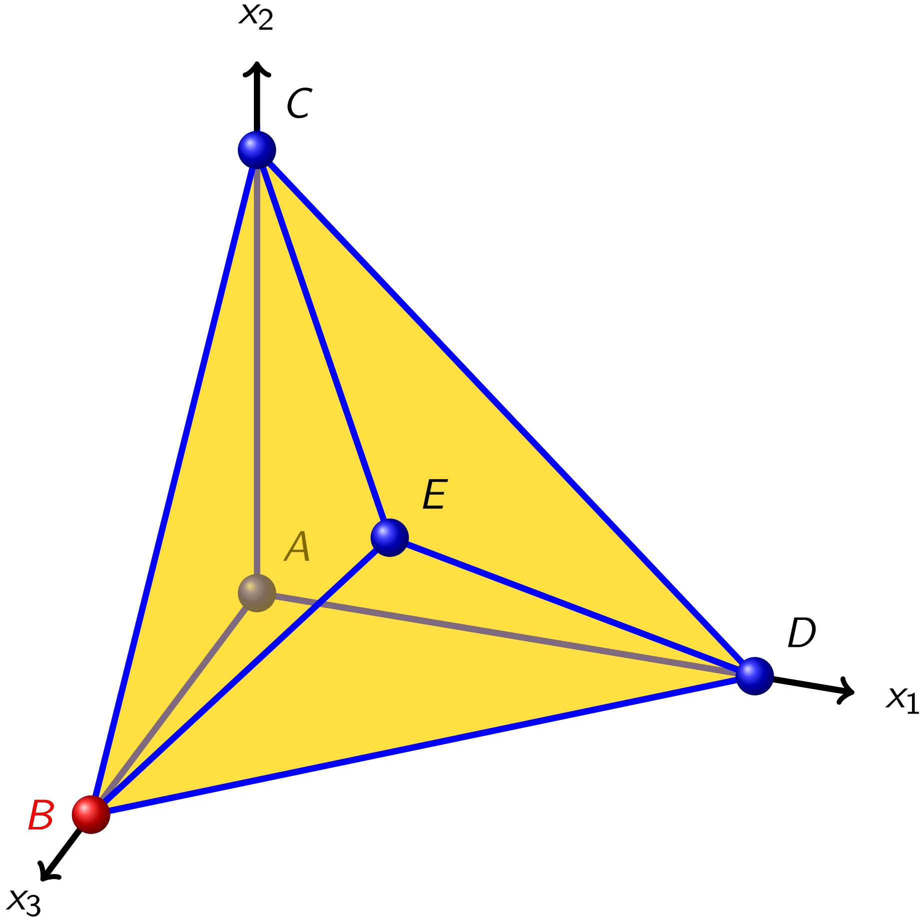
$A = (0,0,0)^T$

$B = (0,0,10)^T$

$C = (0,10,0)^T$

$D = (10,0,0)^T$

$E = (4,4,4)^T$



All Iterations from Geometric Point of View

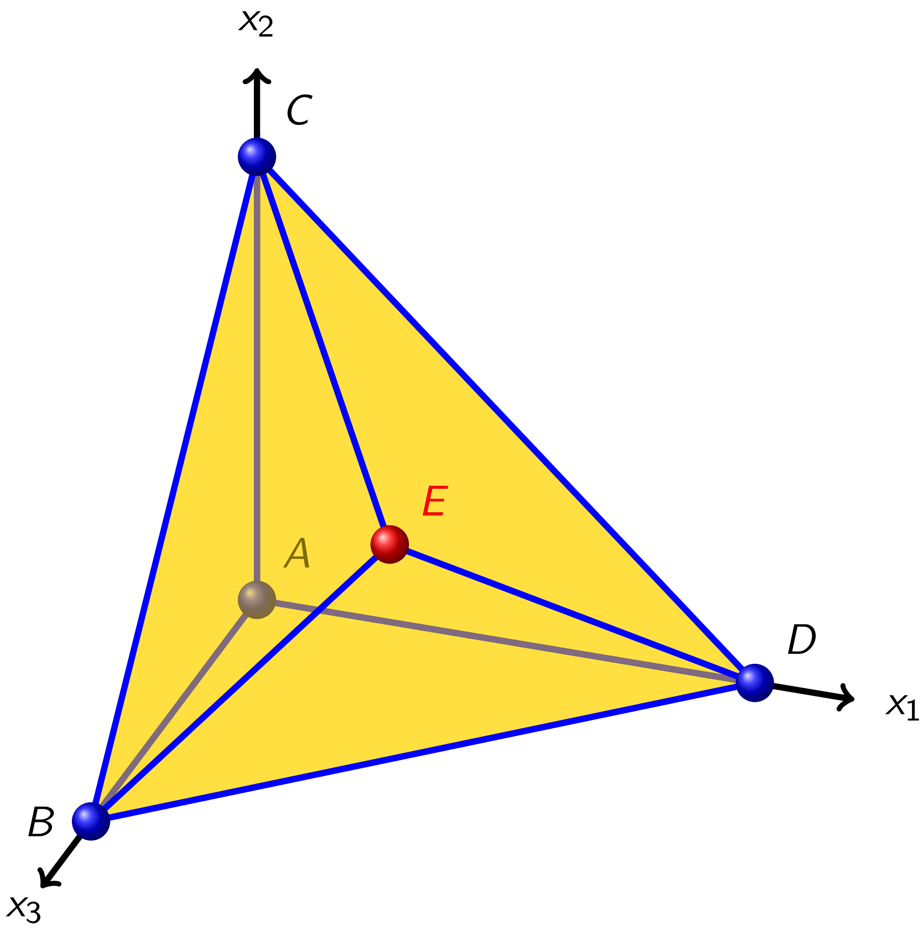
$A = (0,0,0)^T$

$B = (0,0,10)^T$

$C = (0,10,0)^T$

$D = (10,0,0)^T$

$E = (4,4,4)^T$



Comparison of Full Tableau and Revised Simplex Methods

The following table gives the computational cost of one iteration of the simplex method for the two variants introduced above.

	full tableau	revised simplex
memory	$O(mn)$	$O(m^2)$
worst-case time	$O(mn)$	$O(mn)$
best-case time	$O(mn)$	$O(m^2)$

Conclusion

- For implementation purposes, the revised simplex method is clearly preferable due to its smaller memory requirement and smaller average running time.
- The full tableau method is convenient for solving small LP instances by hand since all necessary information is readily available.

Practical Performance Enhancements

Numerical stability

The most critical issue when implementing the (revised) simplex method is numerical stability. In order to deal with this, a number of additional ideas from numerical linear algebra are needed.

- Every update of B^{-1} introduces roundoff or truncation errors which accumulate and might eventually lead to highly inaccurate results. Solution: Compute the matrix B^{-1} from scratch once in a while.
- Instead of computing B^{-1} explicitly, it can be stored as a product of matrices $Q_k \cdot Q_{k-1} \cdot \ldots \cdot Q_1$ where each matrix Q_i can be specified in terms of m coefficients. Then $\overline{B}^{-1} = Q_{k+1} \cdot B^{-1} = Q_{k+1} \cdot \ldots \cdot Q_1$. This might also save space.
- Instead of computing B^{-1} explicitly, compute and store an LR-decomposition.

Cycling

Problem: If an LP is degenerate, the simplex method might end up in an infinite loop (cycling).

Example:

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	$-3/4$	20	$-1/2$	6	0	0	0
$x_5 =$	0	$1/4$	-8	-1	9	1	0	0
$x_6 =$	0	$1/2$	-12	$-1/2$	3	0	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Pivoting rules

- Column selection: let nonbasic variable with most negative reduced cost $\bar{c}_j$ enter the basis, i. e., steepest descent rule.
- Row selection: among basic variables that are eligible to exit the basis, select the one with smallest subscript.

Iteration 1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\frac{x_{B(i)}}{u_i}$
	3	$-3/4$	20	$-1/2$	6	0	0	0	
$x_5 =$	0	$1/4$	-8	-1	9	1	0	0	0
$x_6 =$	0	$1/2$	-12	$-1/2$	3	0	1	0	0
$x_7 =$	1	0	0	1	0	0	0	1	–

Basis change: x_1 enters the basis, x_5 leaves the basis.

Bases visited

$(5, 6, 7)$

Iteration 1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	$-3/4$	20	$-1/2$	6	0	0	0
$x_5 =$	0	$1/4$	-8	-1	9	1	0	0
$x_6 =$	0	$1/2$	-12	$-1/2$	3	0	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Basis change: x_1 enters the basis, x_5 leaves the basis.

Bases visited

$(5, 6, 7)$

Iteration 1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	0	-4	$-7/2$	33	3	0	0
$x_5 =$	0	$1/4$	-8	-1	9	1	0	0
$x_6 =$	0	$1/2$	-12	$-1/2$	3	0	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Basis change: x_1 enters the basis, x_5 leaves the basis.

Bases visited

$(5, 6, 7)$

Iteration 1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	0	-4	$-7/2$	33	3	0	0
$x_5 =$	0	$1/4$	-8	-1	9	1	0	0
$x_6 =$	0	$1/2$	-12	$-1/2$	3	0	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Basis change: x_1 enters the basis, x_5 leaves the basis.

Bases visited

$(5, 6, 7)$

Iteration 1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	0	-4	$-7/2$	33	3	0	0
$x_5 =$	0	$1/4$	-8	-1	9	1	0	0
$x_6 =$	0	0	4	$3/2$	-15	-2	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Basis change: x_1 enters the basis, x_5 leaves the basis.

Bases visited

$(5, 6, 7)$

Iteration 1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	0	-4	$-7/2$	33	3	0	0
$x_5 =$	0	1/4	-8	-1	9	1	0	0
$x_6 =$	0	0	4	3/2	-15	-2	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Basis change: x_1 enters the basis, x_5 leaves the basis.

Bases visited

$(5, 6, 7)$

Iteration 1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	0	-4	$-7/2$	33	3	0	0
$x_5 =$	0	1/4	-8	-1	9	1	0	0
$x_6 =$	0	0	4	3/2	-15	-2	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Basis change: x_1 enters the basis, x_5 leaves the basis.

Bases visited

$(5, 6, 7)$

Iteration 1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	0	-4	$-7/2$	33	3	0	0
$x_1 =$	0	1	-32	-4	36	4	0	0
$x_6 =$	0	0	4	$3/2$	-15	-2	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Basis change: x_1 enters the basis, x_5 leaves the basis.

Bases visited

$(5,6,7)$

Iteration 2

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\frac{x_{B(i)}}{u_i}$
	3	0	-4	$-7/2$	33	3	0	0	
$x_1 =$	0	1	-32	-4	36	4	0	0	–
$x_6 =$	0	0	4	$3/2$	-15	-2	1	0	0
$x_7 =$	1	0	0	1	0	0	0	1	–

Basis change: x_2 enters the basis, x_6 leaves the basis.

Bases visited

$(5,6,7) \rightarrow (1,6,7)$

Iteration 3

$$
\begin{array}{r|c|ccccccc|c}
\cline{2-9}
 & & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & \frac{x_{B(i)}}{u_i} \\
\cline{2-9}
 & 3 & 0 & 0 & -2 & 18 & 1 & 1 & 0 & \\
\cline{2-9}
x_1 = & 0 & 1 & 0 & \boxed{8} & 156 & -12 & 4 & 0 & 0 \\
x_2 = & 0 & 0 & 1 & 3/8 & -15/4 & -1/2 & 1/4 & 0 & 0 \\
x_7 = & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
\cline{2-9}
\end{array}
$$

Basis change: x_3 enters the basis, x_1 leaves the basis.

Bases visited

$(5,6,7) \rightarrow (1,6,7) \rightarrow (1,2,7)$

Iteration 4

$$
\begin{array}{r|c|ccccccc|c}
\cline{2-9}
 & & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & \frac{x_{B(i)}}{u_i} \\
\cline{2-9}
 & 3 & 1/4 & 0 & 0 & -3 & -2 & 3 & 0 & \\
\cline{2-9}
x_3 = & 0 & 1/8 & 0 & 1 & -21/2 & -3/2 & 1 & 0 & - \\
x_2 = & 0 & -3/64 & 1 & 0 & \boxed{3/16} & 1/16 & -1/8 & 0 & 0 \\
x_7 = & 1 & -1/8 & 0 & 0 & 21/2 & 3/2 & -1 & 1 & 2/21 \\
\cline{2-9}
\end{array}
$$

Basis change: x_4 enters the basis, x_2 leaves the basis.

Bases visited

$(5,6,7) \rightarrow (1,6,7) \rightarrow (1,2,7) \rightarrow (3,2,7)$

Iteration 5

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\frac{x_{B(i)}}{u_i}$
	3	$-1/2$	16	0	0	-1	1	0	
$x_3 =$	0	$-5/2$	56	1	0	2	-6	0	0
$x_4 =$	0	$-1/4$	$16/3$	0	1	$1/3$	$-2/3$	0	0
$x_7 =$	1	$5/2$	-56	0	0	-2	6	1	–

Basis change: x_5 enters the basis, x_3 leaves the basis.

Bases visited

$(5,6,7) \to (1,6,7) \to (1,2,7) \to (3,2,7) \to (3,4,7)$

Observation

After 4 pivoting iterations our basic feasible solution still has not changed.

Iteration 6

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\frac{x_{B(i)}}{u_i}$
	3	$-7/4$	44	$1/2$	0	0	-2	0	
$x_5 =$	0	$-5/4$	28	$1/2$	0	1	-3	0	–
$x_4 =$	0	$1/6$	-4	$-1/6$	1	0	$1/3$	0	0
$x_7 =$	1	0	0	1	0	0	0	1	–

Basis change: x_6 enters the basis, x_4 leaves the basis.

Bases visited

$(5,6,7) \to (1,6,7) \to (1,2,7) \to (3,2,7) \to (3,4,7)$
$\to (5,4,7)$

Back at the Beginning

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	$-3/4$	20	$-1/2$	6	0	0	0
$x_5 =$	0	$1/4$	-8	-1	9	1	0	0
$x_6 =$	0	$1/2$	-12	$-1/2$	3	0	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Bases visited

$(5,6,7) \rightarrow (1,6,7) \rightarrow (1,2,7) \rightarrow (3,2,7) \rightarrow (3,4,7)$
$\rightarrow (5,4,7) \rightarrow (5,6,7)$

This is the same basis that we started with.

Conclusion

Continuing with the pivoting rules we agreed on at the beginning, the simplex method will never terminate in this example.

Anticycling

We discuss two pivoting rules that are guaranteed to avoid cycling:

- lexicographic rule
- Bland's rule

Lexicographic Order

Definition 7.12.

- A vector $u \in \mathbb{R}^n$ is lexicographically positive (negative) if $u \neq 0$ and the first nonzero entry of u is positive (negative). Symbolically, we write $u \stackrel{L}{>} 0$ (resp. $u \stackrel{L}{<} 0$).
- A vector $u \in \mathbb{R}^n$ is lexicographically larger (smaller) than a vector $v \in \mathbb{R}^n$ if $u \neq v$ and $u - v \stackrel{L}{>} 0$ (resp. $u - v \stackrel{L}{<} 0$).
  We write $u \stackrel{L}{>} v$ (resp. $u \stackrel{L}{<} v$).

Example:

$$(0, 2, 3, 0)^T \stackrel{L}{>} (0, 2, 1, 4)^T$$

$$(0, 4, 5, 0)^T \stackrel{L}{<} (1, 2, 1, 2)^T$$

Lexicographic Pivoting Rule

Lexicographic pivoting rule in the full tableau implementation:

Lexicographic pivoting rule

1. Choose an arbitrary column A_j with $\bar{c}_j < 0$ to enter the basis. Let $u := B^{-1}A_j$ be the jth column of the tableau.
2. For each i with $u_i > 0$, divide the ith row of the tableau by u_i and choose the lexicographically smallest row ℓ. Then the ℓth basic variable $x_{B(\ell)}$ exits the basis.

Remark

The lexicographic pivoting rule always leads to a unique choice for the exiting variable. Otherwise two rows of $B^{-1}A$ would have to be linearly dependent which contradicts our assumption on the matrix A.

Lexicographic Pivoting Rule (cont.)

Theorem 7.13.

Suppose that the simplex algorithm starts with lexicographically positive rows $1, \ldots, m$ in the simplex tableau. Suppose that the lexicographic pivoting rule is followed. Then:

a. Rows $1, \ldots, m$ of the simplex tableau remain lexicographically positive throughout the algorithm.

b. The zeroth row strictly increases lexicographically at each iteration.

c. The simplex algorithm terminates after a finite number of iterations.

Proof: . . . □

Remarks on Lexicographic Pivoting Rule

- The lexicographic pivoting rule was derived by considering s small perturbation of the right hand side vector b leading to a non-degenerate problem (see exercises).
- The lexicographic pivoting rule can also be used in conjunction with the revised simplex method, provided that B^{-1} is computed explicitly (this is not the case in sophisticated implementations).
- The assumption in the theorem on the lexicographically positive rows in the tableau can be made without loss of generality: Rearrange the columns of A such that the basic columns (forming the identity matrix in the tableau) come first. Since the zeroth column is nonnegative for a basic feasible solution, all rows are lexicographically positive.

Bland's Rule

Smallest subscript pivoting rule (Bland's rule)

1. Choose the column A_j with $\bar{c}_j < 0$ and j minimal to enter the basis.
2. Among all basic variables x_i that could exit the basis, select the one with smallest i.

Theorem (without proof)

The simplex algorithm with Bland's rule terminates after a finite number of iterations.

Remark
Bland's rule is compatible with an implementation of the revised simplex method in which the reduced costs of the nonbasic variables are computed one at a time, in the natural order, until a negative one is discovered.

Finding an Initial Basic Feasible Solution

So far we always assumed that the simplex algorithm starts with a basic feasible solution. We now discuss how such a solution can be obtained.

- Introducing artificial variables
- The two-phase simplex method
- The big-M method

Introducing Artificial Variables

Example:

$$
\begin{array}{lrcrcrcrcl}
\min & x_1 & + & x_2 & + & x_3 & & & & \\
\text{s.t.} & x_1 & + & 2\,x_2 & + & 3\,x_3 & & & = & 3 \\
 & -x_1 & + & 2\,x_2 & + & 6\,x_3 & & & = & 2 \\
 & & & 4\,x_2 & + & 9\,x_3 & & & = & 5 \\
 & & & & & 3\,x_3 & + & x_4 & = & 1 \\
 & & & & & & x_1, \ldots, & x_4 & \geq & 0
\end{array}
$$

Auxiliary problem with artificial variables:

$$
\begin{array}{lrrrrrrrrcl}
\min & & & & & x_5 & +\,x_6 & +\,x_7 & +\,x_8 & & \\
\text{s.t.} & x_1 & +2\,x_2 & +3\,x_3 & & x_5 & & & & = & 3 \\
 & -x_1 & +2\,x_2 & +6\,x_3 & & & +\,x_6 & & & = & 2 \\
 & & 4\,x_2 & +9\,x_3 & & & & +\,x_7 & & = & 5 \\
 & & & 3\,x_3 & +x_4 & & & & +\,x_8 & = & 1 \\
 & & & & & \multicolumn{4}{r}{x_1, \ldots, x_4, x_5, \ldots, x_8} & \geq & 0
\end{array}
$$

Auxiliary Problem

Auxiliary problem with artificial variables:

$$
\begin{array}{lrrrrrrrrcl}
\min & & & & & x_5 & +x_6 & +x_7 & +x_8 & & \\
\text{s.t.} & x_1 & +2\,x_2 & +3\,x_3 & & x_5 & & & & = & 3 \\
 & -x_1 & +2\,x_2 & +6\,x_3 & & & +x_6 & & & = & 2 \\
 & & 4\,x_2 & +9\,x_3 & & & & +x_7 & & = & 5 \\
 & & & 3\,x_3 & +x_4 & & & & +x_8 & = & 1 \\
 & & & & & \multicolumn{4}{r}{x_1, \ldots, x_4, x_5, \ldots, x_8} & \geq & 0
\end{array}
$$

Observation

$x = (0, 0, 0, 0, 3, 2, 5, 1)$ is a basic feasible solution for this problem with basic variables (x_5, x_6, x_7, x_8). We can form the initial tableau.

Initial Tableau

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	0	0	0	0	0	1	1	1	1
$x_5 =$	3	1	2	3	0	1	0	0	0
$x_6 =$	2	-1	2	6	0	0	1	0	0
$x_7 =$	5	0	4	9	0	0	0	1	0
$x_8 =$	1	0	0	3	1	0	0	0	1

Calculate reduced costs by eliminating the nonzero-entries for the basis-variables.

Now we can proceed as seen before...

Initial Tableau

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	0	0	0	0	0	1	1	1	1
$x_5 =$	3	1	2	3	0	1	0	0	0
$x_6 =$	2	-1	2	6	0	0	1	0	0
$x_7 =$	5	0	4	9	0	0	0	1	0
$x_8 =$	1	0	0	3	1	0	0	0	1

Calculate reduced costs by eliminating the nonzero-entries for the basis-variables.

Now we can proceed as seen before...

Initial Tableau

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-3	-1	-2	-3	0	0	1	1	1
$x_5 =$	3	1	2	3	0	1	0	0	0
$x_6 =$	2	-1	2	6	0	0	1	0	0
$x_7 =$	5	0	4	9	0	0	0	1	0
$x_8 =$	1	0	0	3	1	0	0	0	1

Calculate reduced costs by eliminating the nonzero-entries for the basis-variables.

Now we can proceed as seen before...

Initial Tableau

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-5	0	-4	-9	0	0	0	1	1
$x_5 =$	3	1	2	3	0	1	0	0	0
$x_6 =$	2	-1	2	6	0	0	1	0	0
$x_7 =$	5	0	4	9	0	0	0	1	0
$x_8 =$	1	0	0	3	1	0	0	0	1

Calculate reduced costs by eliminating the nonzero-entries for the basis-variables.

Now we can proceed as seen before...

Initial Tableau

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-10	0	-8	-18	0	0	0	0	1
$x_5 =$	3	1	2	3	0	1	0	0	0
$x_6 =$	2	-1	2	6	0	0	1	0	0
$x_7 =$	5	0	4	9	0	0	0	1	0
$x_8 =$	1	0	0	3	1	0	0	0	1

Calculate reduced costs by eliminating the nonzero-entries for the basis-variables.

Now we can proceed as seen before...

Initial Tableau

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-11	0	-8	-21	-1	0	0	0	0
$x_5 =$	3	1	2	3	0	1	0	0	0
$x_6 =$	2	-1	2	6	0	0	1	0	0
$x_7 =$	5	0	4	9	0	0	0	1	0
$x_8 =$	1	0	0	3	1	0	0	0	1

Calculate reduced costs by eliminating the nonzero-entries for the basis-variables.

Now we can proceed as seen before...

Minimizing the Auxiliary Problem

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-11	0	-8	-21	-1	0	0	0	0
$x_5 =$	3	1	2	3	0	1	0	0	0
$x_6 =$	2	-1	2	6	0	0	1	0	0
$x_7 =$	5	0	4	9	0	0	0	1	0
$x_8 =$	1	0	0	3	1	0	0	0	1

Basis change: x_4 enters the basis, x_8 exits.

Minimizing the Auxiliary Problem

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-10	0	-8	-18	0	0	0	0	1
$x_5 =$	3	1	2	3	0	1	0	0	0
$x_6 =$	2	-1	2	6	0	0	1	0	0
$x_7 =$	5	0	4	9	0	0	0	1	0
$x_4 =$	1	0	0	3	1	0	0	0	1

Basis change: x_3 enters the basis, x_4 exits.

Minimizing the Auxiliary Problem

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-4	0	-8	0	6	0	0	0	7
$x_5 =$	2	1	2	0	-1	1	0	0	-1
$x_6 =$	0	-1	2	0	-2	0	1	0	-2
$x_7 =$	2	0	4	0	-3	0	0	1	-3
$x_3 =$	$1/3$	0	0	1	$1/3$	0	0	0	$1/3$

Basis change: x_2 enters the basis, x_6 exits.

Minimizing the Auxiliary Problem

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	-4	-4	0	0	-2	0	4	0	-1
$x_5 =$	2	2	0	0	1	1	-1	0	1
$x_2 =$	0	$-1/2$	1	0	-1	0	$1/2$	0	-1
$x_7 =$	2	2	0	0	1	0	-2	1	1
$x_3 =$	$1/3$	0	0	1	$1/3$	0	0	0	$1/3$

Basis change: x_1 enters the basis, x_5 exits.

Minimizing the Auxiliary Problem

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	0	0	0	0	0	2	2	0	1
$x_1 =$	1	1	0	0	$1/2$	$1/2$	$-1/2$	0	$1/2$
$x_2 =$	$1/2$	0	1	0	$-3/4$	$1/4$	$1/4$	0	$-3/4$
$x_7 =$	0	0	0	0	0	-1	-1	1	0
$x_3 =$	$1/3$	0	0	1	$1/3$	0	0	0	$1/3$

Basic feasible solution for auxiliary problem with (auxiliary) cost value 0

$\Rightarrow$ Also feasible for the original problem - but not (yet) basic.

Obtaining a Basis for the Original Problem

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
	0	0	0	0	0	2	2	0	1
$x_1 =$	1	1	0	0	$1/2$	$1/2$	$-1/2$	0	$1/2$
$x_2 =$	$1/2$	0	1	0	$-3/4$	$1/4$	$1/4$	0	$-3/4$
$x_7 =$	0	0	0	0	0	-1	-1	1	0
$x_3 =$	$1/3$	0	0	1	$1/3$	0	0	0	$1/3$

Observation

Restricting the tableau to the original variables, we get a zero-row.
Thus the original equations are linearily dependent.
$\rightarrow$ We can remove the third row.

Obtaining a Basis for the Original Problem

		x_1	x_2	x_3	x_4
	$-11/6$	0	0	0	$-1/12$
$x_1 =$	1	1	0	0	$1/2$
$x_2 =$	$1/2$	0	1	0	$-3/4$
$x_3 =$	$1/3$	0	0	1	$1/3$

We finally obtain a basic feasible solution for the original problem.

After computing the reduced costs for this basis (as seen in the beginning), the simplex method can start with its typical iterations.

Omitting Artificial Variables

Auxiliary problem

$$\begin{array}{lrrrrrrrrcl}
\min & & & & & x_5 & +x_6 & +x_7 & +x_8 & & \\
\text{s.t.} & x_1 & +2\,x_2 & +3\,x_3 & & x_5 & & & & = & 3 \\
 & -x_1 & +2\,x_2 & +6\,x_3 & & & +x_6 & & & = & 2 \\
 & & 4\,x_2 & +9\,x_3 & & & & +x_7 & & = & 5 \\
 & & & 3\,x_3 & +x_4 & & & & +x_8 & = & 1 \\
 & & & & & & & \multicolumn{2}{r}{x_1,\dots,x_8} & \geq & 0
\end{array}$$

Artificial variable x_8 could have been omitted by setting x_4 to 1 in the initial basis. This is possible as x_4 does only appear in one constraint.

Generally, this can be done, e. g., with all slack variables that have nonnegative right hand sides.

Phase I of the Simplex Method

Given: LP in standard form: $\min\{c^T \cdot x \mid A \cdot x = b,\ x \geq 0\}$

1. Transform problem such that $b \geq 0$ (multiply constraints by -1).
2. Introduce artificial variables $y_1, \dots, y_m$ and solve auxiliary problem
$$\min \sum_{i=1}^{m} y_i \quad \text{s.t. } A \cdot x + I_m \cdot y = b,\ x, y \geq 0 \ .$$
3. If optimal cost is positive, then STOP (original LP is infeasible).
4. If no artificial variable is in final basis, eliminate artificial variables and columns and STOP (feasible basis for original LP has been found).
5. If ℓth basic variable is artificial, find $j \in \{1, \dots, n\}$ with ℓth entry in $B^{-1} \cdot A_j$ nonzero. Use this entry as pivot element and replace ℓth basic variable with x_j.
6. If no such $j \in \{1, \dots, n\}$ exists, eliminate ℓth row (constraint).

The Two-phase Simplex Method

Two-phase simplex method

1. Given an LP in standard from, first run phase I.
2. If phase I yields a basic feasible solution for the original LP, enter "phase II" (see above).

Possible outcomes of the two-phase simplex method

i. Problem is infeasible (detected in phase I).
ii. Problem is feasible but rows of A are linearly dependent (detected and corrected at the end of phase I by eliminating redundant constraints.)
iii. Optimal cost is $-\infty$ (detected in phase II).
iv. Problem has optimal basic feasible solution (found in phase II).

Remark: (ii) is not an outcome but only an intermediate result leading to outcome (iii) or (iv).

Big-M Method

Alternative idea: Combine the two phases into one by introducing sufficiently large penalty costs for artificial variables.

This way, the LP

$$\begin{array}{lrcl} \min & \sum_{i=1}^{n} c_i x_i & & \\ \text{s.t.} & A \cdot x & = & b \\ & x & \geq & 0 \end{array}$$

becomes:

$$\begin{array}{lrcrcl} \min & \sum_{i=1}^{n} c_i x_i & + & M \cdot \sum_{j=1}^{m} y_j & & \\ \text{s.t.} & A \cdot x & + & I_m \cdot y & = & b \\ & & & x, y & \geq & 0 \end{array}$$

Remark: If M is sufficiently large and the original program has a feasible solution, all artificial variables will be driven to zero by the simplex method.

How to Choose M?

Observation

Initially, M only occurs in the zeroth row. As the zeroth row never becomes pivot row, this property is maintained while the simplex method is running.

All we need to have is an order on all values that can appear as reduced cost coefficients.

Order on cost coefficients

$$a\,M + b < c\,M + d \quad :\Longleftrightarrow \quad (a < c) \;\vee\; (a = c \wedge b < d)$$

In particular, $-a\,M + b < 0 < a\,M + b$ for any positive a and arbitrary b, and we can decide whether a cost coefficient is negative or not.

$\rightarrow$ There is no need to give M a fixed numerical value.

Example

Example:

$$\begin{array}{lrcrcrcrcl}
\min & x_1 & + & x_2 & + & x_3 & & & & \\
\text{s.t.} & x_1 & + & 2\,x_2 & + & 3\,x_3 & & & = & 3 \\
& -x_1 & + & 2\,x_2 & + & 6\,x_3 & & & = & 2 \\
& & & 4\,x_2 & + & 9\,x_3 & & & = & 5 \\
& & & & & 3\,x_3 & + & x_4 & = & 1 \\
& & & & & & x_1, \ldots, & x_4 & \geq & 0
\end{array}$$

Introducing Artificial Variables and M

Auxiliary problem:

$$\begin{array}{lrrrrrrrcl}
\min & x_1 & +x_2 & +x_3 & & +\,M\,x_5 & +\,M\,x_6 & +\,M\,x_7 & & \\
\text{s.t.} & x_1 & +2\,x_2 & +3\,x_3 & & x_5 & & & = & 3 \\
& -x_1 & +2\,x_2 & +6\,x_3 & & & +\,x_6 & & = & 2 \\
& & 4\,x_2 & +9\,x_3 & & & & +\,x_7 & = & 5 \\
& & & 3\,x_3 & +x_4 & & & & = & 1 \\
& & & & & & & x_1, \ldots, x_4, x_5, x_6, x_7 & \geq & 0
\end{array}$$

Note that this time the unnecessary artificial variable x_8 has been omitted.

We start off with $(x_5, x_6, x_7, x_4) = (3, 2, 5, 1)$.

Forming the Initial Tableau

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	1	1	1	0	M	M	M
3	1	2	3	0	1	0	0
2	-1	2	6	0	0	1	0
5	0	4	9	0	0	0	1
1	0	0	3	1	0	0	0

Compute reduced costs by eliminating the nonzero entries for the basic variables.

Forming the Initial Tableau

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	1	1	1	0	M	M	M
3	1	2	3	0	1	0	0
2	-1	2	6	0	0	1	0
5	0	4	9	0	0	0	1
1	0	0	3	1	0	0	0

Compute reduced costs by eliminating the nonzero entries for the basic variables.

Forming the Initial Tableau

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$-3M$	$-M+1$	$-2M+1$	$-3M+1$	0	0	M	M
3	1	2	3	0	1	0	0
2	-1	2	6	0	0	1	0
5	0	4	9	0	0	0	1
1	0	0	3	1	0	0	0

Compute reduced costs by eliminating the nonzero entries for the basic variables.

Forming the Initial Tableau

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$-5M$	1	$-4M+1$	$-9M+1$	0	0	0	M
3	1	2	3	0	1	0	0
2	-1	2	6	0	0	1	0
5	0	4	9	0	0	0	1
1	0	0	3	1	0	0	0

Compute reduced costs by eliminating the nonzero entries for the basic variables.

Forming the Initial Tableau

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$-10M$	1	$-8M+1$	$-18M+1$	0	0	0	0
3	1	2	3	0	1	0	0
2	-1	2	6	0	0	1	0
5	0	4	9	0	0	0	1
1	0	0	3	1	0	0	0

Compute reduced costs by eliminating the nonzero entries for the basic variables.

Forming the Initial Tableau

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$-10M$	1	$-8M+1$	$-18M+1$	0	0	0	0
3	1	2	3	0	1	0	0
2	-1	2	6	0	0	1	0
5	0	4	9	0	0	0	1
1	0	0	3	1	0	0	0

Compute reduced costs by eliminating the nonzero entries for the basic variables.

First Iteration

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$-10M$	1	$-8M+1$	$-18M+1$	0	0	0	0
3	1	2	3	0	1	0	0
2	-1	2	6	0	0	1	0
5	0	4	9	0	0	0	1
1	0	0	3	1	0	0	0

Reduced costs for x_2 and x_3 are negative.

Basis change: x_3 enters the basis, x_4 leaves.

Second Iteration

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$-4M-1/3$	1	$-8M+1$	0	$6M-1/3$	0	0	0
2	1	2	0	-1	1	0	0
0	-1	2	0	-2	0	1	0
2	0	4	0	-3	0	0	1
$1/3$	0	0	1	$1/3$	0	0	0

Basis change: x_2 enters the basis, x_6 leaves.

Third Iteration

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$-4M-1/3$	$-4M+3/2$	0	0	$-2M+2/3$	0	$4M-1/2$	0
2	2	0	0	1	1	-1	0
0	$-1/2$	1	0	-1	0	$1/2$	0
2	2	0	0	1	0	-2	1
$1/3$	0	0	1	$1/3$	0	0	0

Basis change: x_1 enters the basis, x_5 leaves.

Fourth Iteration

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$-11/6$	0	0	0	$-1/12$	$2M-3/4$	$2M+1/4$	0
1	1	0	0	$1/2$	$1/2$	$-1/2$	0
$1/2$	0	1	0	$-3/4$	$1/4$	$1/4$	0
0	0	0	0	0	-1	-1	1
$1/3$	0	0	1	$1/3$	0	0	0

Note that all artificial variables have already been driven to 0.

Basis change: x_4 enters the basis, x_3 leaves.

Fifth Iteration

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$-7/4$	0	0	$1/4$	0	$2M-3/4$	$2M+1/4$	0
$1/2$	1	0	$-3/2$	0	$1/2$	$-1/2$	0
$5/4$	0	1	$9/4$	0	$1/4$	$1/4$	0
0	0	0	0	0	-1	-1	1
1	0	0	3	1	0	0	0

We now have an optimal solution to the auxiliary problem, as all costs are nonnegative (M presumed large enough).

By elimiating the third row as in the previous example, we get a basic feasible and also optimal solution to the original problem.

Computational Efficiency of the Simplex Method

Observation

The computational efficiency of the simplex method is determined by

i the computational effort of each iteration;

ii the number of iterations.

Question: How many iterations are needed in the worst case?

Idea for negative answer (lower bound)

Describe

- a polyhedron with an exponential number of vertices;
- a path that visits all vertices and always moves from a vertex to an adjacent one that has lower costs.

Computational Efficiency of the Simplex Method

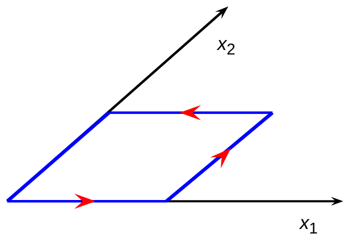
Unit cube

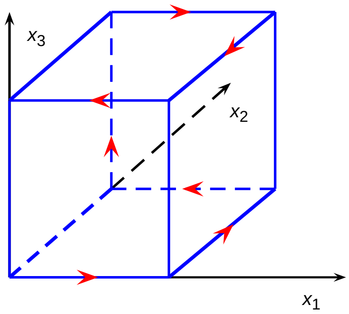
Consider the unit cube in $\mathbb{R}^n$, defined by the constraints

$$0 \leq x_i \leq 1, \quad i = 1, \dots, n$$

The unit cube has

- 2^n vertices;
- a *spanning path*, i. e., a path traveling the edges of the cube visiting each vertex exactly once.





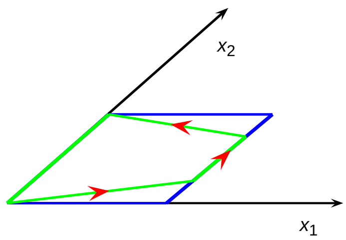
Computational Efficiency of the Simplex Method (cont.)

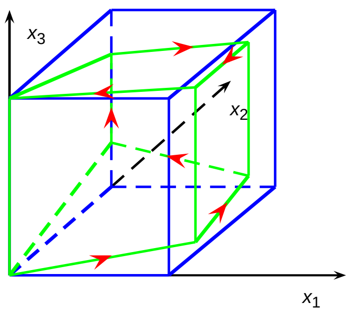
Klee-Minty cube

Consider a perturbation of the unit cube in $\mathbb{R}^n$, defined by the constraints

$$\begin{aligned} 0 \leq x_1 \leq 1&, \\ \epsilon x_{i-1} \leq x_i \leq 1 - \epsilon x_{i-1}&, \quad i = 2, \dots, n \end{aligned}$$

for some $\epsilon \in (0, 1/2)$.





Computational Efficiency of the Simplex Method (cont.)

Klee-Minty cube

$$0 \le x_1 \le 1,$$
$$\epsilon x_{i-1} \le x_i \le 1 - \epsilon x_{i-1}, \quad i = 2, \ldots, n$$

Theorem 7.14.

Consider the linear programming problem of minimizing $-x_n$ subject to the constraints above. Then,

a. the feasible set has 2^n vertices;
b. the vertices can be ordered so that each one is adjacent to and has lower cost than the previous one;
c. there exists a pivoting rule under which the simplex method requires $2^n - 1$ changes of basis before it terminates.

Remark.

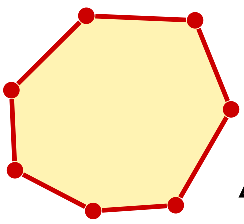
Such 'bad' instances exist for (almost) all popular pivoting rules.

Reminder: Diameter of Polyhedra

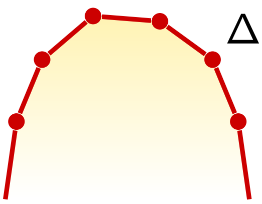
Definition 7.15.

- The distance $d(x, y)$ between two vertices x, y is the minimum number of edges required to reach y starting from x.
- The diameter $D(P)$ of polyhedron P is the maximum $d(x, y)$ over all pairs of vertices x, y.
- $\Delta(n, m)$ is the maximum $D(P)$ over all polytopes in $\mathbb{R}^n$ that are represented in terms of m inequality constraints.
- $\Delta_u(n, m)$ is the maximum $D(P)$ over all polyhedra in $\mathbb{R}^n$ that are represented in terms of m inequality constraints.

Examples.



$$\Delta(2, 7) = \left\lfloor \tfrac{7}{2} \right\rfloor = 3$$
$$\Delta(2, m) = \left\lfloor \tfrac{m}{2} \right\rfloor$$



$$\Delta_u(2, 7) = 7 - 2 = 5$$
$$\Delta_u(2, m) = m - 2$$

Reminder: Hirsch Conjecture

Observation: The diameter of the feasible set in a linear programming problem is a lower bound on the number of steps required by the simplex method, no matter which pivoting rule is being used.

Polynomial Hirsch Conjecture

$$\Delta(n, m) \leq poly(m, n)$$

Remarks

- Known lower bounds: $\Delta_u(n, m) \geq m - n + \left\lfloor \frac{n}{5} \right\rfloor$
- Known upper bounds:

$$\Delta(n, m) \leq \Delta_u(n, m) < m^{1+\log_2 n} = (2n)^{\log_2 m}$$

- The Strong Hirsch Conjecture

$$\Delta(n, m) \leq m - n$$

  was disproven in 2010 by Paco Santos for $n = 43,\ m = 86$.

Average Case Behavior of the Simplex Method

- Despite the exponential lower bounds on the worst case behavior of the simplex method (Klee-Minty cubes etc.), the simplex method usually behaves well in practice.
- The number of iterations is "typically" $O(m)$.
- There have been several attempts to explain this phenomenon from a more theoretical point of view.
- These results say that "on average" the number of iterations is $O(\cdot)$ (usually polynomial).
- One main difficulty is to come up with a meaningful and, at the same time, manageable definition of the term "on average".

Chapter 8: Duality Theory

(cp. Bertsimas & Tsitsiklis, Chapter 4)

Motivation

For $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$, consider the linear program

$$\min \; c^T \cdot x \qquad \text{s.t.} \quad A \cdot x \geq b, \; x \geq 0$$

Question: How to derive lower bounds on the optimal solution value?

Idea: For $p \in \mathbb{R}^m$ with $p \geq 0$: $\quad A \cdot x \geq b \quad \Longrightarrow \quad (p^T \cdot A) \cdot x \geq p^T \cdot b$

Thus, if $c^T \geq p^T \cdot A$, then

$$c^T \cdot x \geq (p^T \cdot A) \cdot x \geq p^T \cdot b \qquad \text{for all feasible solutions } x.$$

Find the best (largest) lower bound in this way:

$$\begin{array}{rl} \max & p^T \cdot b \\ \text{s.t.} & p^T \cdot A \leq c^T \\ & p \geq 0 \end{array} \qquad \longleftrightarrow \qquad \begin{array}{rl} \max & b^T \cdot p \\ \text{s.t.} & A^T \cdot p \leq c \\ & p \geq 0 \end{array}$$

This LP is the dual linear program of our initial LP.

Primal and Dual Linear Program

Consider the general linear program:

$$
\begin{array}{lll}
\min & c^T \cdot x & \\
\text{s.t.} & a_i{}^T \cdot x \geq b_i & \text{for } i \in M_1 \\
& a_i{}^T \cdot x \leq b_i & \text{for } i \in M_2 \\
& a_i{}^T \cdot x = b_i & \text{for } i \in M_3 \\
& x_j \geq 0 & \text{for } j \in N_1 \\
& x_j \leq 0 & \text{for } j \in N_2 \\
& x_j \text{ free} & \text{for } j \in N_3
\end{array}
$$

Obtain a lower bound:

$$
\begin{array}{lll}
\max & p^T \cdot b & \\
\text{s.t.} & p_i \geq 0 & \text{for } i \in M_1 \\
& p_i \leq 0 & \text{for } i \in M_2 \\
& p_i \text{ free} & \text{for } i \in M_3 \\
& p^T \cdot A_j \leq c_j & \text{for } j \in N_1 \\
& p^T \cdot A_j \geq c_j & \text{for } j \in N_2 \\
& p^T \cdot A_j = c_j & \text{for } j \in N_3
\end{array}
$$

The linear program on the right hand side is the dual linear program of the primal linear program on the left hand side.

Primal and Dual Variables and Constraints

primal LP (minimize)		dual LP (maximize)	
	$\geq b_i$	≥ 0	
constraints	$\leq b_i$	≤ 0	variables
	$= b_i$	free	
	≥ 0	$\leq c_i$	
variables	≤ 0	$\geq c_i$	constraints
	free	$= c_i$	

Examples of Primal and Dual LPs

primal LP	dual LP
$\min \; c^T \cdot x$ s.t. $A \cdot x \geq b$	$\max \; p^T \cdot b$ s.t. $p^T \cdot A = c^T$, $p \geq 0$
$\min \; c^T \cdot x$ s.t. $A \cdot x = b$, $x \geq 0$	$\max \; p^T \cdot b$ s.t. $p^T \cdot A \leq c^T$

Basic Properties of the Dual Linear Program

Theorem 8.1.

The dual of the dual LP is the primal LP.

Proof: . . . □

Theorem 8.2.

Let Π_1 and Π_2 be two LPs where Π_2 has been obtained from Π_1 by (several) transformations of the following type:

i. replace a free variable by the difference of two non-negative variables;
ii. introduce a slack variable in order to replace an inequality constraint by an equation;
iii. if some row of a feasible equality system is a linear combination of the other rows, eliminate this row.

Then the dual of Π_1 is equivalent to the dual of Π_2.

Proof: . . . □

Weak Duality Theorem

Theorem 8.3.

If x is a feasible solution to the primal LP (minimization problem) and p a feasible solution to the dual LP (maximization problem), then

$$c^T \cdot x \geq p^T \cdot b \enspace .$$

Proof: ... □

Corollary 8.4.

Consider a primal-dual pair of linear programs as above.

a. If the primal LP is unbounded (i. e., optimal cost $= -\infty$), then the dual LP is infeasible.
b. If the dual LP is unbounded (i. e., optimal cost $= \infty$), then the primal LP is infeasible.
c. If x and p are feasible solutions to the primal and dual LP, resp., and if $c^T \cdot x = p^T \cdot b$, then x and p are optimal solutions.

Strong Duality Theorem

Theorem 8.5.

If an LP has an optimal solution, so does its dual and the optimal costs are equal.

Proof: ... □

Different Possibilities for Primal and Dual LP

primal \ dual	finite optimum	unbounded	infeasible
finite optimum	possible	impossible	impossible
unbounded	impossible	impossible	possible
infeasible	impossible	possible	possible

Example of infeasible primal and dual LP:

$$
\begin{array}{llcl}
\min & x_1 + 2\,x_2 \\
\text{s.t.} & x_1 + x_2 = 1 \\
& 2\,x_1 + 2\,x_2 = 3
\end{array}
\qquad\qquad
\begin{array}{ll}
\max & p_1 + 3\,p_2 \\
\text{s.t.} & p_1 + 2\,p_2 = 1 \\
& p_1 + 2\,p_2 = 2
\end{array}
$$

Complementary Slackness

Consider the following pair of primal and dual LPs:

$$
\begin{array}{ll}
\min & c^T \cdot x \\
\text{s.t.} & A \cdot x \geq b
\end{array}
\qquad\qquad
\begin{array}{ll}
\max & p^T \cdot b \\
\text{s.t.} & p^T \cdot A = c^T \\
& p \geq 0
\end{array}
$$

If x and p are feasible solutions, then $c^T \cdot x = p^T \cdot A \cdot x \geq p^T \cdot b$.Thus,

$$c^T \cdot x = p^T \cdot b \quad \Longleftrightarrow \quad \text{for all } i\text{: } p_i = 0 \text{ if } {a_i}^T \cdot x > b_i.$$

Theorem 8.6.

Consider an arbitrary pair of primal and dual LPs. Let x and p be feasible solutions to the primal and dual LP, respectively. Then x and p are both optimal if and only if

$$
\begin{aligned}
u_i &:= p_i\,({a_i}^T \cdot x - b_i) = 0 \quad \text{for all } i, \\
v_j &:= (c_j - p^T \cdot A_j)\,x_j = 0 \quad \text{for all } j.
\end{aligned}
$$

Proof: . . . □

Geometric View

Consider pair of primal and dual LPs with $A \in \mathbb{R}^{m \times n}$ and $\text{rank}(A) = n$:

$$\begin{array}{rl} \min & c^T \cdot x \\ \text{s.t.} & A \cdot x \geq b \end{array} \qquad\qquad \begin{array}{rl} \max & p^T \cdot b \\ \text{s.t.} & p^T \cdot A = c^T \\ & p \geq 0 \end{array}$$

Let $I \subseteq \{1, \ldots, m\}$ with $|I| = n$ and a_i, $i \in I$, linearly independent.

$\Longrightarrow \quad {a_i}^T \cdot x = b_i$, $i \in I$, has unique solution x^I (basic solution)

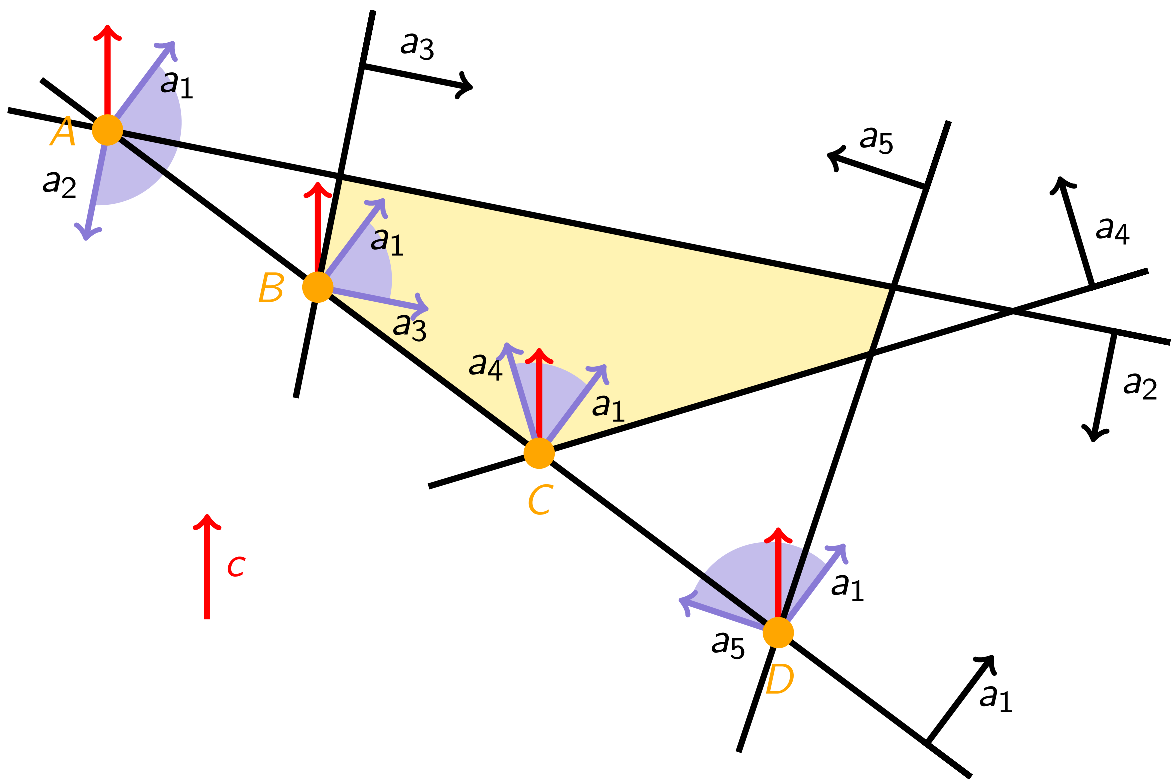
Let $p \in \mathbb{R}^m$ (dual vector). Then x, p are optimal solutions if

- i ${a_i}^T \cdot x \geq b_i$ for all i (primal feasibility)
- ii $p_i = 0$ for all $i \notin I$ (complementary slackness)
- iii $\sum_{i=1}^m p_i \cdot a_i = c$ (dual feasibility)
- iv $p \geq 0$ (dual feasibility)

(ii) and (iii) imply $\sum_{i \in I} p_i \cdot a_i = c$ which has a unique solution p^I.

The a_i, $i \in I$, form basis for dual LP and p^I is corresponding basic solution.

Geometric View (cont.)

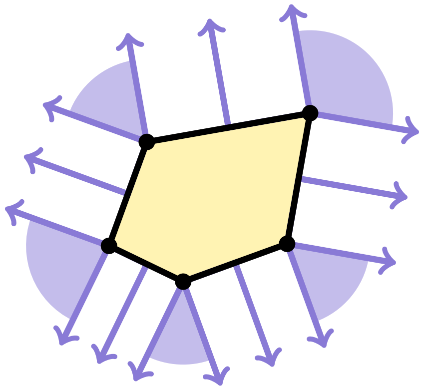


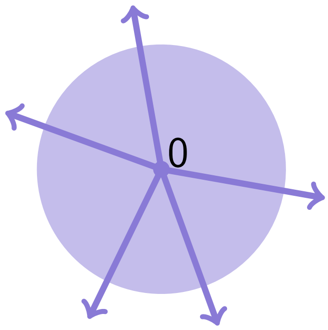
Geometric View: Outer Normal Cone

Definition 8.7.

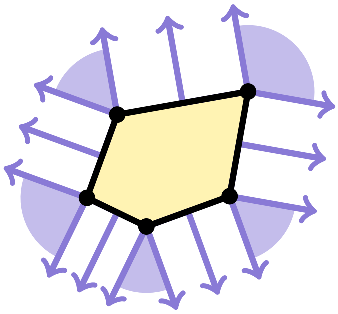
Let x^* be a boundary point of the polyhedron $P = \{x \mid A \cdot x \leq b\}$ and let I denote the (non-empty) set of indices of rows a_i^T of A with $a_i^T x^* = b_i$. The conic hull $\mathrm{pos}\{a_i \mid i \in I\}$ is the outer normal cone of P in x^*.

Example.





Properties of Outer Normal Cones



Observation 8.8.

Consider a polyhedron $P \subseteq \mathbb{R}^n$.

i For some $c \in \mathbb{R}^n \setminus \{0\}$ a point $x \in P$ maximizes $c^T x$ over P if and only if c is in the outer normal cone of P in x.

ii If x is a boundary point of P, the polyhedral cone consisting of the feasible directions at x is polar to the outer normal cone of P in x.

iii If $P \subseteq \mathbb{R}^n$ is a polytope, every point in $\mathbb{R}^n$ is in the outer normal cone of some vertex. The interiors of these cones do not intersect.

Dual Variables as Marginal Costs

Consider the primal dual pair:

$$\begin{aligned} \min \quad & c^T \cdot x \\ \text{s.t.} \quad & A \cdot x = b \\ & x \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \max \quad & p^T \cdot b \\ \text{s.t.} \quad & p^T \cdot A \leq c^T \end{aligned}$$

Let x^* be optimal basic feasible solution to primal LP with basis B, i. e., $x^*_B = B^{-1} \cdot b$ and assume that $x^*_B > 0$ (i. e., x^* non-degenerate).

Replace b by $b + d$. For small d, the basis B remains feasible and optimal:

$$B^{-1} \cdot (b + d) = B^{-1} \cdot b + B^{-1} \cdot d \geq 0 \qquad \text{(feasibility)}$$
$$\bar{c}^T = c^T - c_B{}^T \cdot B^{-1} \cdot A \geq 0 \qquad \text{(optimality)}$$

Optimal cost of perturbed problem is

$$c_B{}^T \cdot B^{-1} \cdot (b + d) = c_B{}^T \cdot x^*_B + \underbrace{\left(c_B{}^T \cdot B^{-1}\right)}_{=p^T} \cdot d$$

Thus, p_i is the marginal cost per unit increase of b_i.

Dual Variables as Shadow Prices

Diet problem:

- a_{ij} := amount of nutrient i in one unit of food j
- b_i := requirement of nutrient i in some ideal diet
- c_j := cost of one unit of food j on the food market

LP duality: Let x_j := number of units of food j in the diet:

$$\begin{aligned} \min \quad & c^T \cdot x \\ \text{s.t.} \quad & A \cdot x = b \\ & x \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \max \quad & p^T \cdot b \\ \text{s.t.} \quad & p^T \cdot A \leq c^T \end{aligned}$$

Dual interpretation:

- p_i is “fair” price per unit of nutrient i
- $p^T \cdot A_j$ is value of one unit of food j on the nutrient market
- food j used in ideal diet ($x^*_j > 0$) is consistently priced at the two markets (by complementary slackness)
- ideal diet has the same cost on both markets (by strong duality)

Dual Basic Solutions

Consider LP in standard form with $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = m$, and dual LP:

$$\begin{array}{llll} \min & c^T \cdot x & \qquad \max & p^T \cdot b \\ \text{s.t.} & A \cdot x = b & \qquad \text{s.t.} & p^T \cdot A \leq c^T \\ & x \geq 0 & & \end{array}$$

Observation 8.9.

A basis B yields

- a primal basic solution given by $x_B := B^{-1} \cdot b$ and
- a dual basic solution $p^T := {c_B}^T \cdot B^{-1}$.

Moreover,

a) the values of the primal and the dual basic solutions are equal:

$$ {c_B}^T \cdot x_B = {c_B}^T \cdot B^{-1} \cdot b = p^T \cdot b \ ; $$

b) p is feasible if and only if $\bar{c} \geq 0$;

c) reduced cost $\bar{c}_i = 0$ corresponds to active dual constraint;

d) p is degenerate if and only if $\bar{c}_i = 0$ for some non-basic variable x_i.

Dual Simplex Method

- Let B be a basis whose corresponding dual basic solution p is feasible.
- If also the primal basic solution x is feasible, then x, p are optimal.
- Assume that $x_{B(\ell)} < 0$ and consider the ℓth row of the simplex tableau

$$ (x_{B(\ell)}, v_1, \dots, v_n) \qquad \text{(pivot row)} $$

I) Let $j \in \{1, \dots, n\}$ with $v_j < 0$ and

$$ \frac{\bar{c}_j}{|v_j|} = \min_{i: v_i < 0} \frac{\bar{c}_i}{|v_i|} $$

Performing an iteration of the simplex method with pivot element v_j yields new basis B' and corresponding dual basic solution p' with

$$ {c_{B'}}^T \cdot B'^{-1} \cdot A \leq c^T \quad \text{and} \quad p'^T \cdot b \geq p^T \cdot b \quad (\text{with} > \text{if } \bar{c}_j > 0). $$

II) If $v_i \geq 0$ for all $i \in \{1, \dots, n\}$, then the dual LP is unbounded and the primal LP is infeasible.

Remarks on the Dual Simplex Method

- Dual simplex method terminates if lexicographic pivoting rule is used:
  - Choose any row ℓ with $x_{B(\ell)} < 0$ to be the pivot row.
  - Among all columns j with $v_j < 0$ choose the one which is lexicographically minimal when divided by $|v_j|$.
- Dual simplex method is useful if, e. g., dual basic solution is readily available.
- Example: Resolve LP after right-hand-side b has changed.

Chapter 9: The Ellipsoid Method

(cp. Bertsimas & Tsitsiklis, Chapter 8;
book of Grötschel, Lovász & Schrijver)

Complexity of Linear Programming

- As discussed in Chapter 7, so far no variant of the simplex method has been shown to have a polynomial running time.
- Therefore, the complexity of Linear Programming remained unresolved for a long time.
- Only in 1979, the Soviet mathematician Leonid Khachiyan proved that the so-called ellipsoid method earlier developed for nonlinear optimization can be modified in order to solve LPs in polynomial time.
- In November 1979, the New York Times featured Khachiyan and his algorithm in a front-page story.
- We give a sketch of the ellipsoid method and its analysis.

An Approach to Difficult Problems

Mathematicians disagree as to the ultimate practical value of Leonid Khachiyan's new technique, but concur that in any case it is an important theoretical accomplishment.

Mr. Khachiyan's method is believed to offer an approach for the linear programming of computers to solve so-called "traveling salesman" problems. Such problems are among the most intractable in mathematics. They involve, for instance, finding the shortest route by which a salesman could visit a number of cities without his path touching the same city twice.

Each time a new city is added to the route, the problem becomes very much more complex. Very large numbers of variables must be calculated from large numbers of equations using a system of linear programming. At a certain point, the compexity becomes so great that a computer would require billions of years to find a solution.

In the past, "traveling salesmen" problems, including the efficient scheduling of airline crews or hospital nursing staffs, have been solved on computers using the "simplex method" invented by George B. Dantzig of Stanford University.

As a rule, the simplex method works well, but it offers no guarantee that after a certain number of computer steps it will always find an answer. Mr. Khachiyan's approach offers a way of telling right from the start whether or not a problem will be soluble in a given number of steps.

Two mathematicians conducting research at Stanford already have applied the Khachiyan method to develop a program for a pocket calculator, which has solved problems that would not have been possible with a pocket calculator using the simplex method.

Mathematically, the Khachiyan approach uses equations to create imaginary ellipsoids that encapsulate the answer, unlike the simplex method, in which the answer is represented by the intersections of the sides of polyhedrons. As the ellipsoids are made smaller and smaller, the answer is known with greater precision. MALCOLM W. BROWNE

New York Times, Nov. 27, 1979

Geometric Basics: Positive Definite Matrices

Definition 9.1.

A symmetric matrix $D \in \mathbb{R}^{n \times n}$ is positive definite if

$$x^T \cdot D \cdot x > 0 \qquad \text{for all } x \in \mathbb{R}^n \setminus \{0\}.$$

Lemma 9.2.

For a symmetric matrix $D \in \mathbb{R}^{n \times n}$, the following statements are equivalent:

- i D is positive definite.
- ii D^{-1} exists and is positive definite.
- iii D has only real and positive eigenvalues.
- iv $D = B^T \cdot B$ for a non-singular matrix $B \in \mathbb{R}^{n \times n}$.

Proof: Basic linear algebra... $\square$

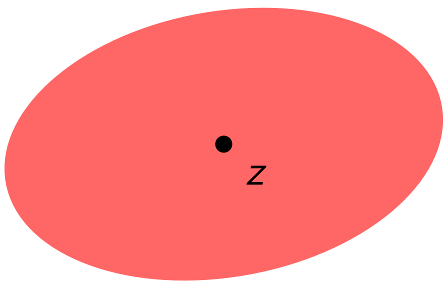
Geometric Basics: Ellipsoids

Definition 9.3.

A set $E \subseteq \mathbb{R}^n$ of the form

$$E = E(z, D) := \{x \in \mathbb{R}^n \mid (x - z)^T \cdot D^{-1} \cdot (x - z) \leq 1\}$$

where $z \in \mathbb{R}^n$ and $D \in \mathbb{R}^{n \times n}$ is positive definite is called an ellipsoid with center z.



Example: For $r > 0$, the ellipsoid

$$E(z, r^2 \cdot I) = \{x \in \mathbb{R}^n \mid (x-z)^T \cdot (x-z) \leq r^2\} = \{x \in \mathbb{R}^n \mid \|x-z\|_2 \leq r\}$$

is the ball of radius r centered at z.

Geometric Basics: Affine Transformations and Volume

Definition 9.4.

If $D \in \mathbb{R}^{n\times n}$ is non-singular and $b \in \mathbb{R}^n$, then the mapping $S : \mathbb{R}^n \to \mathbb{R}^n$ defined by $S(x) := D \cdot x + b$ is called an affine transformation.

- For $L \subseteq \mathbb{R}^n$ let $S(L) := \{y \in \mathbb{R}^n \mid y = D \cdot x + b \text{ for some } x \in L\}$.
- Define the volume of $L \subseteq \mathbb{R}^n$ as $\mathrm{Vol}(L) := \int_{x\in L} dx$.

Lemma 9.5.

For the affine transformation S with $S(x) = D \cdot x + b$, the volume of $S(L)$ is $\mathrm{Vol}(S(L)) = |\det(D)| \cdot \mathrm{Vol}(L)$.

Proof:

$$\mathrm{Vol}(S(L)) = \int_{y\in S(L)} dy = \int_{x\in L} |\det(J(x))|\, dx$$

where $J(x)$ is the Jacobian matrix associated with S, i. e.,

$$(J(x))_{ij} = \frac{\partial S_i(x)}{\partial x_j} = D_{ij} \ .$$

Thus, $J(x) = D$. $\square$

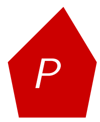
Ellipsoid Method — Rough Idea

The ellipsoid method solves the following problem:

Given: $A \in \mathbb{Z}^{m\times n}$ and $b \in \mathbb{Z}^m$, polyhedron $P := \{x \in \mathbb{Q}^n \mid A \cdot x \geq b\}$.

Task: Find a point $x \in P$ or determine that P is empty.

Example:



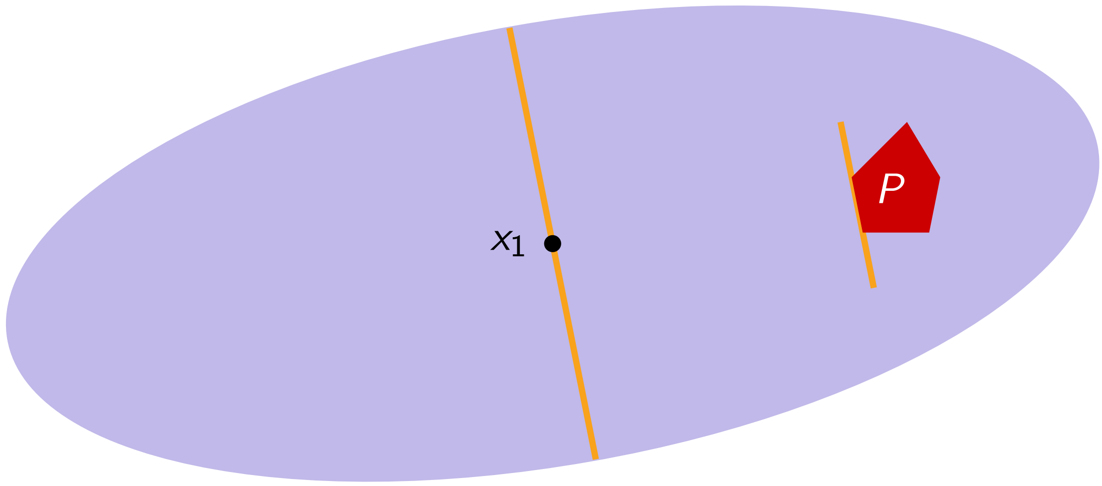
Ellipsoid Method — Rough Idea

The ellipsoid method solves the following problem:

Given: $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$, polyhedron $P := \{x \in \mathbb{Q}^n \mid A \cdot x \geq b\}$.

Task: Find a point $x \in P$ or determine that P is empty.

Example:



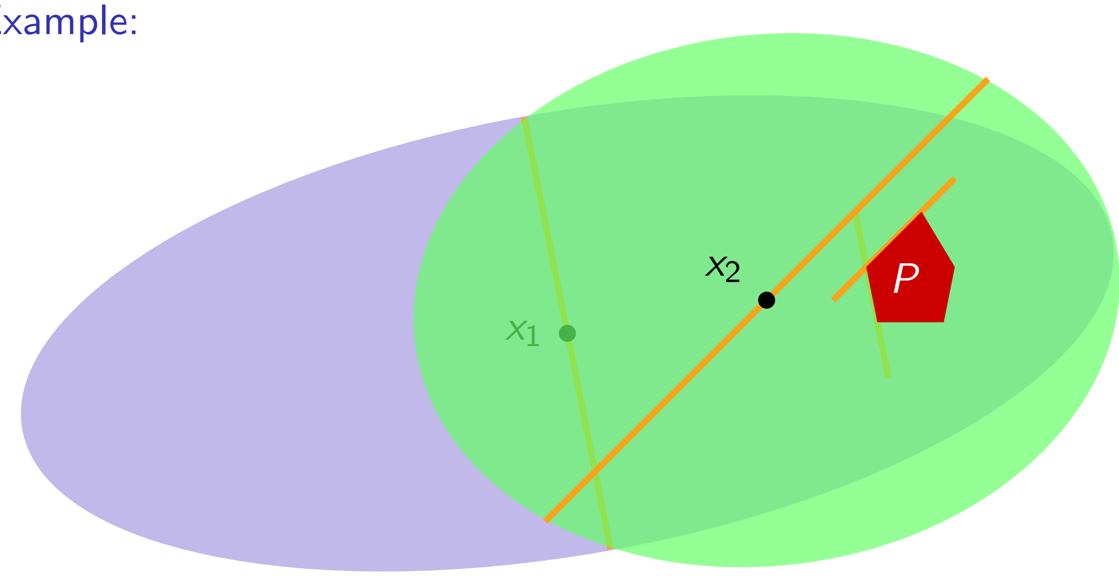
Ellipsoid Method — Rough Idea

The ellipsoid method solves the following problem:

Given: $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$, polyhedron $P := \{x \in \mathbb{Q}^n \mid A \cdot x \geq b\}$.

Task: Find a point $x \in P$ or determine that P is empty.

Example:



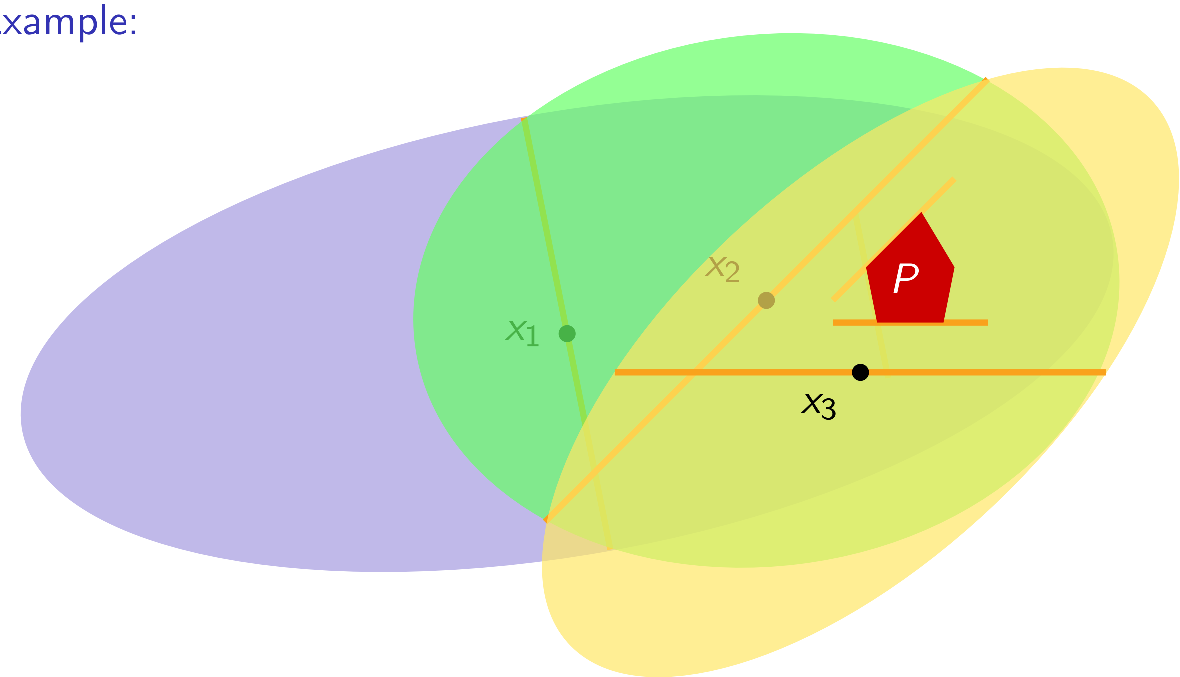
Ellipsoid Method — Rough Idea

The ellipsoid method solves the following problem:

Given: $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$, polyhedron $P := \{x \in \mathbb{Q}^n \mid A \cdot x \geq b\}$.

Task: Find a point $x \in P$ or determine that P is empty.

Example:



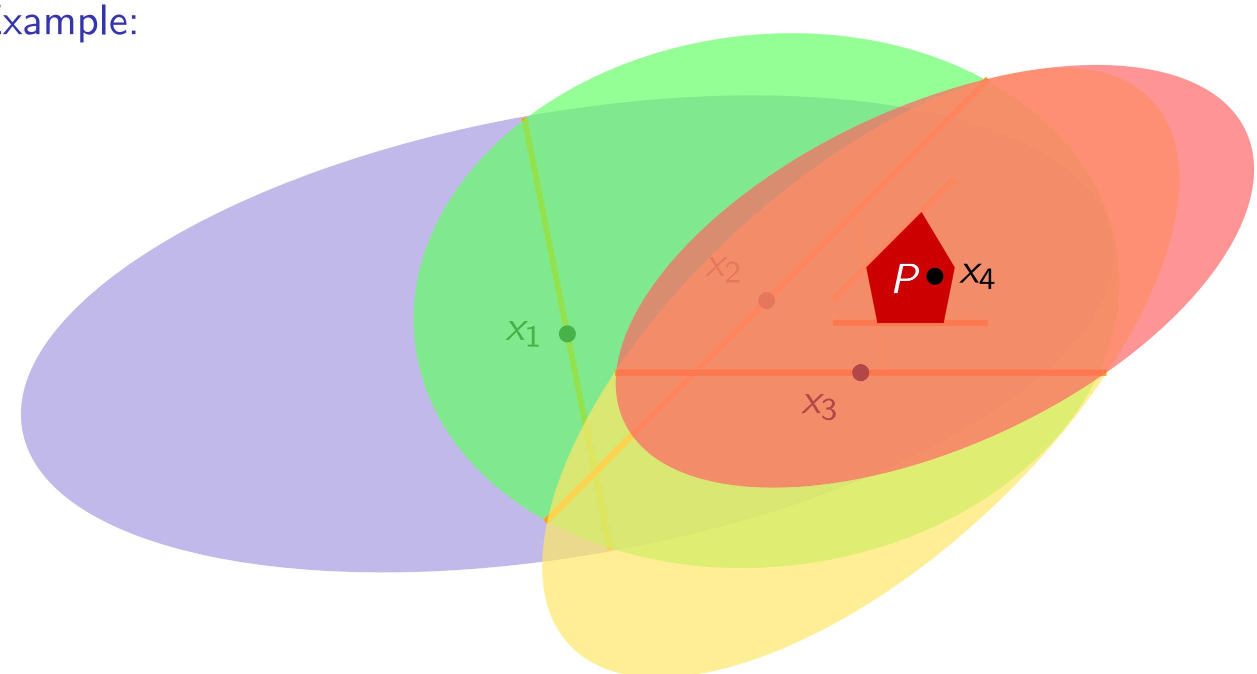
Ellipsoid Method — Rough Idea

The ellipsoid method solves the following problem:

Given: $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$, polyhedron $P := \{x \in \mathbb{Q}^n \mid A \cdot x \geq b\}$.

Task: Find a point $x \in P$ or determine that P is empty.

Example:



How to Find the Next Ellipsoid?

Theorem 9.6.

Let $E = E(z, D)$ be an ellipsoid in $\mathbb{R}^n$ and $a \in \mathbb{R}^n \setminus \{0\}$. Consider the halfspace $H := \{x \in \mathbb{R}^n \mid a^T \cdot x \geq a^T \cdot z\}$ and set

$$\bar{z} := z + \frac{1}{n+1} \cdot \frac{D \cdot a}{\sqrt{a^T \cdot D \cdot a}} ,$$
$$\bar{D} := \frac{n^2}{n^2 - 1} \cdot \left(D - \frac{2}{n+1} \cdot \frac{D \cdot a \cdot a^T \cdot D}{a^T \cdot D \cdot A} \right) .$$

The matrix $\bar{D}$ is symmetric and positive definite. Thus, $\bar{E} := E(\bar{z}, \bar{D})$ is an ellipsoid. Moreover:

i $E \cap H \subseteq \bar{E}$

ii $\mathrm{Vol}(\bar{E}) < e^{-\frac{1}{2(n+1)}} \cdot \mathrm{Vol}(E)$

Proof: See Bertsimas & Tsitsiklis, Section 8.2. □

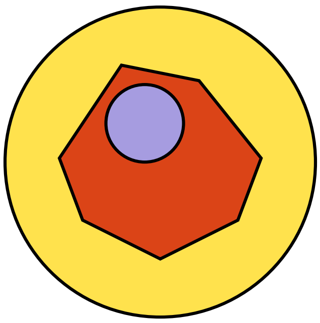
Simplifying Assumptions

Definition 9.7.

A polyhedron $P \subseteq \mathbb{R}^n$ is full-dimensional if it has non-zero volume.

To state the ellipsoid method we first make some simplifying assumptions:

- Polyhedron P is bounded (i. e., a polytope) and either empty or full-dimensional, i. e.,
  - $P \subseteq E(x_0, r^2 \cdot I) =: E_0$ with $r > 0$; $V := \mathrm{Vol}(E_0)$.
  - $\mathrm{Vol}(P) > v$ for some $v > 0$ (or P is empty).
  - Assume that E_0, V, and v are known a priori.

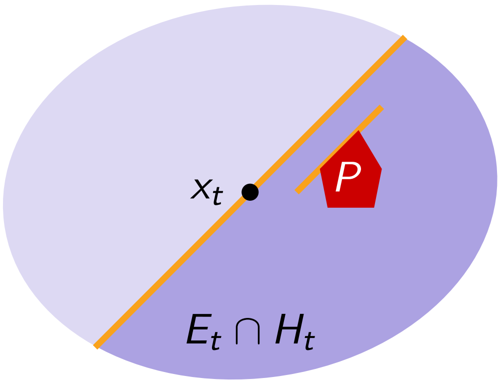


- Calculations (including square roots) can be made in infinite precision.

We discuss these assumptions later in greater detail. . .

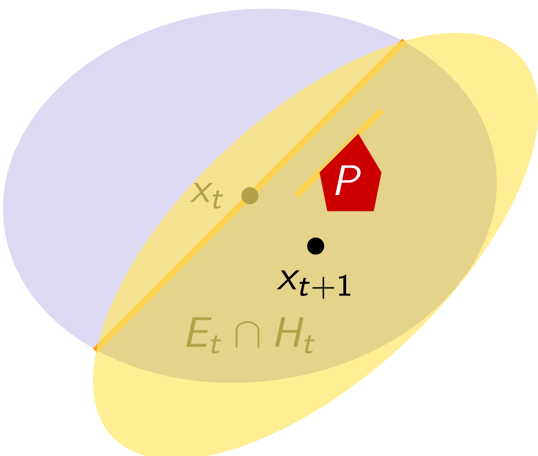
Ellipsoid Method

- i set $t^* := \lceil 2(n+1)\log(V/v)\rceil$; $E_0 := E(x_0, r^2 \cdot I)$; $D_0 := r^2 \cdot I$; $t := 0$;
- ii if $t = t^*$ then stop and output "P is empty";
- iii if $x_t \in P$ then stop and output x_t;
- iv find violated constraint in $A \cdot x_t \geq b$, i. e., ${a_i}^T \cdot x_t < b_i$ for some i;
- v set $H_t := \{x \in \mathbb{R}^n \mid {a_i}^T \cdot x \geq {a_i}^T \cdot x_t\}$; (halfspace containing P)
- vi find ellipsoid $E_{t+1} \supseteq E_t \cap H_t$ by applying Theorem 9.6;
- vii set $t := t + 1$ and go to step ii;



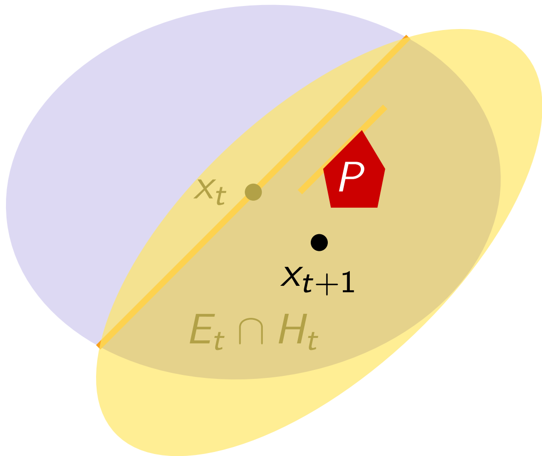
Ellipsoid Method

- i set $t^* := \lceil 2(n+1)\log(V/v)\rceil$; $E_0 := E(x_0, r^2 \cdot I)$; $D_0 := r^2 \cdot I$; $t := 0$;
- ii if $t = t^*$ then stop and output "P is empty";
- iii if $x_t \in P$ then stop and output x_t;
- iv find violated constraint in $A \cdot x_t \geq b$, i. e., ${a_i}^T \cdot x_t < b_i$ for some i;
- v set $H_t := \{x \in \mathbb{R}^n \mid {a_i}^T \cdot x \geq {a_i}^T \cdot x_t\}$; (halfspace containing P)
- vi find ellipsoid $E_{t+1} \supseteq E_t \cap H_t$ by applying Theorem 9.6;
- vii set $t := t + 1$ and go to step ii;



Ellipsoid Method

1. set $t^* := \lceil 2(n+1)\log(V/v) \rceil$; $E_0 := E(x_0, r^2 \cdot I)$; $D_0 := r^2 \cdot I$; $t := 0$;
2. if $t = t^*$ then stop and output "P is empty";
3. if $x_t \in P$ then stop and output x_t;
4. find violated constraint in $A \cdot x_t \geq b$, i. e., ${a_i}^T \cdot x_t < b_i$ for some i;
5. set $H_t := \{x \in \mathbb{R}^n \mid {a_i}^T \cdot x \geq {a_i}^T \cdot x_t\}$; (halfspace containing P)
6. find ellipsoid $E_{t+1} \supseteq E_t \cap H_t$ by applying Theorem 9.6;
7. set $t := t + 1$ and go to step ii;



Correctness of the Ellipsoid Method

Theorem 9.8.
The ellipsoid method returns a point in P or decides correctly that $P = \emptyset$.

Proof: If $x_t \in P$ for some $t < t^*$, then the algorithm returns x_t.

Otherwise: By induction $P \subseteq E_k$ for $k = 0, 1, \ldots, t^*$.

By Theorem 9.6 we get $\dfrac{\mathrm{Vol}(E_{t+1})}{\mathrm{Vol}(E_t)} < e^{-\frac{1}{2(n+1)}}$ for all t.

Thus, $\dfrac{\mathrm{Vol}(E_{t^*})}{\mathrm{Vol}(E_0)} < e^{-\frac{t^*}{2(n+1)}}$.

$$\Longrightarrow \quad \mathrm{Vol}(E_{t^*}) < V \cdot e^{-\frac{\lceil 2(n+1)\log(V/v)\rceil}{2(n+1)}} \leq V \cdot e^{-\log(V/v)} = v \ .$$

Since v is a lower bound on the volume of non-empty P, the algorithm correctly decides that $P = \emptyset$. □

What if P is Unbounded?

Lemma 9.9.

Let $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{R}^m$, and let U be the largest absolute value of entries in A and b. Every extreme point of polyhedron $P = \{x \in \mathbb{R}^n \mid A \cdot x \geq b\}$ satisfies

$$-(nU)^n \leq x_j \leq (nU)^n \qquad \text{for } j = 1, \ldots, n. \tag{9.1}$$

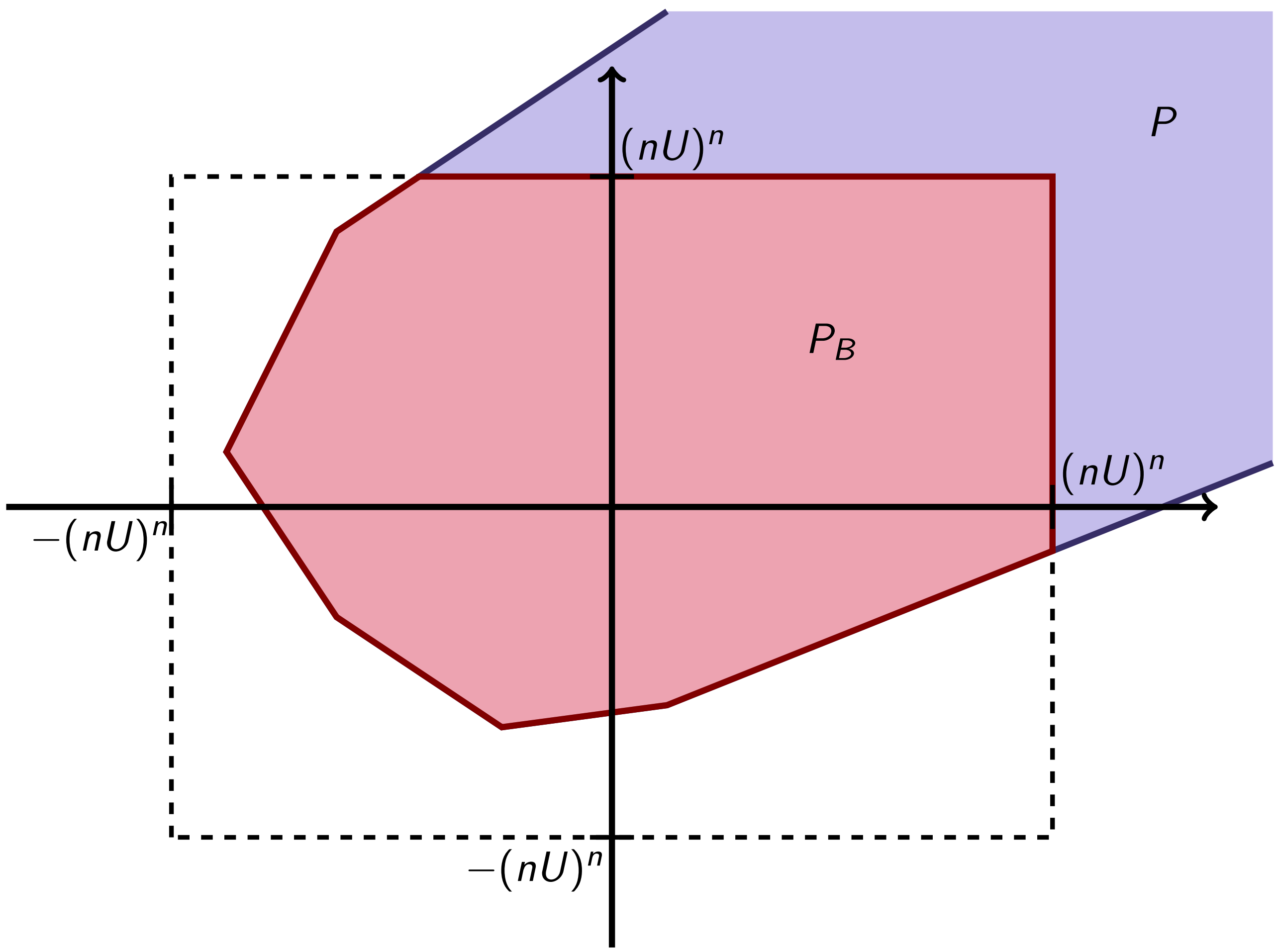
Proof: ... □

Remarks.

- Define the polytope $P_B := \{x \in P \mid -(nU)^n \leq x_j \leq (nU)^n \text{ for all } j\}$.
- Under the assumption that $\text{rank}(A) = n$, we get:

$$P \neq \emptyset \quad \Longleftrightarrow \quad P \text{ contains an extreme point} \quad \Longleftrightarrow \quad P_B \neq \emptyset$$

- Thus, we can look for a point in $P_B \subseteq P$ instead of P.
- Start with ellipsoid $E_0 := E(0, n(nU)^{2n} \cdot I) \supseteq P_B$.
- Note that $\text{Vol}(E_0) \leq V := (2\sqrt{n}(nU)^n)^n = (2\sqrt{n})^n \cdot (nU)^{n^2}$.



What if P is not Full-Dimensional?

Lemma 9.10.

Let $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, and let U be the largest absolute value of entries in A and b. Consider polyhedron $P = \{x \in \mathbb{R}^n \mid A \cdot x \geq b\}$, define

$$\varepsilon := \frac{1}{2(n+1)} \cdot \big((n+1)U\big)^{-(n+1)}$$

and a new polyhedron

$$P_\varepsilon := \{x \in \mathbb{R}^n \mid A \cdot x \geq b - \varepsilon \cdot \mathbf{1}\} \ .$$

Then it holds that

- a $P = \emptyset \quad \Longrightarrow \quad P_\varepsilon = \emptyset$
- b $P \neq \emptyset \quad \Longrightarrow \quad P_\varepsilon$ is full-dimensional
- c Given a point in P_ε, a point in P can be obtained in polynomial time.

Proof: . . . □

Bounding the Number of Iterations

Lemma 9.11.

If the polyhedron $P = \{x \in \mathbb{R}^n \mid A \cdot x \geq b\}$ is full-dimensional and bounded with U as above, then

$$\mathrm{Vol}(P) > n^{-n}(nU)^{-n^2(n+1)} \ .$$

Proof idea:

One can show that P has $n+1$ extreme points $v^0, \ldots, v^n$ such that

$$\mathrm{Vol}\big(\mathrm{conv}(v^0, \ldots, v^n)\big) > n^{-n}(nU)^{-n^2(n+1)} \ .$$

□

Theorem 9.12.

The number of iterations of the ellipsoid method can be bounded by $O(n^6 \log(nU))$. □

Required Numeric Precision

Major problems:

- Bound number of arithmetic operations per iteration.
- How to take square roots?
- Only finite precision possible!

Theorem 9.13.

Using only $O(n^3 \log U)$ binary digits of precision, the ellipsoid method still correctly decides whether P is empty in $O(n^6 \log(nU))$ iterations. Thus, the Linear Inequalities problem can be solved in polynomial time. $\square$

Solving LPs in Polynomial Time

Consider a pair of primal and dual LPs:

$$\begin{array}{rrr} \min & c^T \cdot x \\ \text{s.t.} & A \cdot x \geq b \\ & x \geq 0 \end{array} \qquad\qquad \begin{array}{rr} \max & p^T \cdot b \\ \text{s.t.} & p^T \cdot A \leq c^T \\ & p \geq 0 \end{array}$$

Solve the primal and dual LP by finding a point (x, p) in the polyhedron given by

$$\{(x,p) \mid c^T \cdot x = p^T \cdot b,\ A \cdot x \geq b,\ p^T \cdot A \leq c^T,\ x, p \geq 0\} \ .$$

Theorem 9.14.

Linear programs can be solved in polynomial time. $\square$

LPs with Exponentially Many Constraints

- The number of iterations of the ellipsoid method only depends on the dimension n and U, but not on the number of constraints m.
- Thus there is hope to solve LPs with exponentially many constraints (that are implicitly given) in polynomial time.

Example: Consider the following LP relaxation of the TSP (subtour LP):

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e \cdot x_e & & \\
\text{s.t.} \quad & \sum_{e \in \delta(v)} x_e = 2 & & \text{for all nodes } v \in V, \\
& \sum_{e \in \delta(X)} x_e \geq 2 & & \text{for all subsets } \emptyset \neq X \subsetneq V, \qquad (9.2) \\
& 0 \leq x_e \leq 1 & & \text{for all edges } e.
\end{aligned}$$

Notice that there are $2^{n-1} - 1$ subtour elimination constraints (9.2).

LPs with Exponentially Many Constraints (cont.)

- Describe polyhedron $P = \{x \in \mathbb{R}^n \mid A \cdot x \geq b\}$ by specifying n and an integer vector h of primary data of dimension $O(n^k)$ with k constant. Let $U_0 := \max_i |h_i|$.
- There is a mapping which, given n and h, defines $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$. Let

$$U := \max\{|a_{ij}|, |b_i| \mid i = 1, \ldots, m, \; j = 1, \ldots, n\} \; .$$

- We assume that there are constants C and ℓ such that

$$\log U \leq C \cdot n^\ell \cdot \log^\ell U_0 \; ,$$

  that is, U can be encoded polynomially in the input size.
- The number of iterations of the ellipsoid method is

$$O(n^6 \log(nU)) = O(n^6 \log n + n^{6+\ell} \log^\ell U_0)$$

  and thus polynomial in the input size of the primary problem data.

Separation Problem

In every iteration of the ellipsoid method, we have to solve the following problem:

Definition 9.15.

Given a polyhedron $P \subseteq \mathbb{R}^n$ and $x \in \mathbb{R}^n$, the separation problem is to

i either decide that $x \in P$, or

ii find $d \in \mathbb{R}^n$ with $d^T \cdot x < d^T \cdot y$ for all $y \in P$.

Example: The subtour elimination constraints (9.2) can be separated in polynomial time by finding a minimum capacity cut.

Optimization is as Difficult as Separation

The following Theorem by Grötschel, Lovász, and Schrijver is a consequence of the ellipsoid method.

Theorem 9.16.

i Given a family of polyhedra, if we can solve the separation problem in time polynomial in n and $\log U$, then we can also solve LPs over those polyhedra in time polynomial in n and $\log U$.

ii The converse is also true under some technical conditions. □

Example: The subtour LP for the TSP can be solved in polynomial time since the subtour elimination constraints (9.2) can be separated efficiently.

Path-Based LP Formulation for Flow Problems

Let $\mathcal{P}$ be the set of all s-t-dipaths in digraph D.

$$\begin{aligned} \max \quad & \sum_{P \in \mathcal{P}} y_P & & \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}: a \in P} y_P \leq u(a) & & \text{for all } a \in A \\ & y_P \geq 0 & & \text{for all } P \in \mathcal{P} \end{aligned}$$

Dual LP:

$$\begin{aligned} \min \quad & \sum_{a \in A} u(a) \cdot z_a & & \\ \text{s.t.} \quad & \sum_{a \in P} z_a \geq 1 & & \text{for all } P \in \mathcal{P} \qquad (9.3) \\ & z_a \geq 0 & & \text{for all } a \in A \end{aligned}$$

Since constraints (9.3) can be separated efficiently by a shortest path computation, the dual LP can be solved efficiently. Using complementary slackness conditions, also the primal LP can be solved in polynomial time.

Chapter 10: Representation of Polyhedra

(cp. Bertsimas & Tsitsiklis, Chapters 4.6–4.9)

Polyhedral Cones

Definition 10.1.

A polyhedron of the form $P = \{x \in \mathbb{R}^n \mid A \cdot x \geq 0\}$ is called polyhedral cone.

Remark: $0 \in P$ is the only possible vertex of a polyhedral cone P.
If $0 \in P$ is a vertex, then P is called pointed.

Theorem 10.2.

Let $C \subseteq \mathbb{R}^n$ be the polyhedral cone defined by the constraints ${a_i}^T \cdot x \geq 0$, $i = 1, \ldots, m$. Then, the following are equivalent:

i. The zero vector is an extreme point of C.
ii. The cone C does not contain a line.
iii. There exist n vectors out of the family $a_1, \ldots, a_m$, which are linearly independent.

Proof: Follows from Theorem 6.16. $\square$

Reminder: Recession Cones

Definition 10.3.

Consider a non-empty polyhedron $P = \{x \in \mathbb{R}^n \mid A \cdot x \geq b\}$ and $y \in P$. The recession cone of P (at y) is the set

$$\{d \in \mathbb{R}^n \mid y + \lambda \cdot d \in P \text{ for all } \lambda \geq 0\} \enspace .$$

The non-zero elements of the recession cone are the rays of P.

Remarks:

- Notice that
$$\begin{aligned}&\{d \in \mathbb{R}^n \mid y + \lambda \cdot d \in P \text{ for all } \lambda \geq 0\}\\ &\quad = \{d \in \mathbb{R}^n \mid A \cdot (y + \lambda \cdot d) \geq b \text{ for all } \lambda \geq 0\}\\ &\quad = \{d \in \mathbb{R}^n \mid A \cdot d \geq 0\} \enspace .\end{aligned}$$
- The definition of the recession cone of P is independent of $y \in P$.
- The recession cone of P is a polyhedral cone.
- The recession cone of $P = \{x \in \mathbb{R}^n \mid A \cdot x = b, \, x \geq 0\}$ is
$$\{d \in \mathbb{R}^n \mid A \cdot d = 0, \, d \geq 0\} \enspace .$$

Extreme Rays

Observation.

A non-empty polyhedron P has a vertex if and only if its recession cone is pointed. In this case we also say that P is pointed.

Definition 10.4.

a. A non-zero element x of a polyhedral cone $C \subseteq \mathbb{R}^n$ is an extreme ray if there are $n-1$ linearly independent constraints that are active at x.
b. An extreme ray of the recession cone of a polyhedron P is also called an extreme ray of P.

Remark:

- A non-zero element x of a polyhedral cone C is an extreme ray if and only if $\text{pos}(\{x\})$ is a 1-face of C.
- Up to multiplication with positive factors, there are only finitely many extreme rays of a polyhedron.

Characterization of Unbounded LPs

Theorem 10.5.

Let $C := \{x \in \mathbb{R}^n \mid {a_i}^T \cdot x \geq 0,\ i = 1, \ldots, m\}$ a pointed polyhedral cone and $c \in \mathbb{R}^n$. The minimal cost $c^T \cdot x$ subject to $x \in C$ is equal to $-\infty$ if and only if there is an extreme ray d of C with $c^T \cdot d < 0$.

Proof: . . . □

The result also holds if C is a polyhedron with at least one extreme point.

Theorem 10.6.

Let $P \subseteq \mathbb{R}^n$ be a polyhedron with at least one extreme point and $c \in \mathbb{R}^n$. The minimal cost $c^T \cdot x$ subject to $x \in P$ is equal to $-\infty$ if and only if there is an extreme ray d of P with $c^T \cdot d < 0$.

Proof: . . . □

Remark: If the simplex method observes that an LP is unbounded, the corresponding jth basic direction is an extreme ray d with $c^T \cdot d < 0$.

Resolution Theorem

Theorem 10.7.

Let $P := \{x \in \mathbb{R}^n \mid A \cdot x \geq b\} \neq \emptyset$ pointed. Let $x^1, \ldots, x^k$ be the extreme points and $w^1, \ldots, w^r$ a complete set of extreme rays of P. Then,

$$P = \left\{ \sum_{i=1}^{k} \lambda_i \cdot x^i + \sum_{j=1}^{r} \theta_j \cdot w^j \mid \lambda_i, \theta_j \geq 0, \sum_{i=1}^{k} \lambda_i = 1 \right\} .$$

Proof: . . . □

Corollary 10.8 (cp. Lemma 3.11).

A non-empty polytope is equal to the convex hull of its extreme points. □

Corollary 10.9.

Every element of a pointed polyhedral cone is a non-negative linear combination of its extreme rays. □

Converse to the Resolution Theorem

Definition 10.10.

A set $Q \subseteq \mathbb{R}^n$ is finitely generated if there are $x^1, \ldots, x^k, w^1, \ldots, w^r \in \mathbb{R}^n$ such that

$$Q = \left\{ \sum_{i=1}^{k} \lambda_i \cdot x^i + \sum_{j=1}^{r} \theta_j \cdot w^j \mid \lambda_i, \theta_j \geq 0, \sum_{i=1}^{k} \lambda_i = 1 \right\} .$$

Remark: The Resolution Theorem states that a polyhedron with at least one extreme point is finitely generated (also true for general polyhedra).

Theorem 10.11.

A finitely generated set is a polyhedron.

Proof: . . . □

Representation of Polyhedra

We have thus generalized Theorem 3.3 (Minkowski, Weyl).

Conclusion: There are two ways of representing a polyhedron:

i in terms of a finite set of linear constraints (outer representation);

ii as a finitely generated set, in terms of its extreme points and rays (inner representation).

Remarks:

- Passing from one type of description to the other is, in general, a complicated computational task.
- One description can be small while the other one is huge. Examples:
  - An n-dimensional cube is given by $2n$ linear constraints and has 2^n extreme points.
  - A representation of the convex hull of the $2n$ points

$$e_1, -e_1, e_2, -e_2, \ldots, e_n, -e_n$$

in terms of linear constraints needs at least 2^n linear inequalities.

Chapter 11: Large-Scale Linear Programming

(cp. Bertsimas & Tsitsiklis, Chapter 6)

Delayed Column Generation

Let $A \in \mathbb{R}^{m \times n}$ with $\operatorname{rank}(A) = m$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and $m \ll n$.

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

Suppose that the number of columns n is huge such that A cannot be generated and stored in your computer's memory.

Remember: Revised simplex method only requires m basic columns and the column which shall enter the basis.

Pricing problem: How to find column that should enter basis (i. e., $\bar{c}_j < 0$)?

Solution: Sometimes one can find j with $\bar{c}_j = \min_i \bar{c}_i$ efficiently.

Conclusion:

- Only work with few columns at a time (basic columns and some "promising" non-basic columns).
- Generate new relevant columns by solving pricing problem.

Example: Min-Cost Multi-Commodity Flows

Given: Digraph $D = (V, A)$, capacities $u : A \to \mathbb{R}_{\geq 0}$, costs $c : A \to \mathbb{R}_{\geq 0}$; k source-sink pairs $(s_i, t_i) \in V \times V$ with demands $d_i \in \mathbb{R}_{\geq 0}$, $i = 1, \dots, k$.

Task: Send d_i units of flow from s_i to t_i for all i without violating arc capacities; minimize total cost.

Path-based LP formulation: Let $\mathcal{P}_i$ be the set of all s_i-t_i-dipath in D, $\mathcal{P} := \bigcup_{i=1}^k \mathcal{P}_i$. Cost of path $P \in \mathcal{P}$ is $c_P := \sum_{a \in P} c(a)$.

$$\begin{aligned} \min \quad & \sum_{P \in \mathcal{P}} c_P \, x_P \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P} : a \in P} x_P + s_a = u(a) && \text{for all } a \in A \\ & \sum_{P \in \mathcal{P}_i} x_P = d_i && \text{for all } i = 1, \dots, k \\ & x_P, s_a \geq 0 && \text{for all } P \in \mathcal{P},\ a \in A \end{aligned}$$

Notice: The number of variables is exponential in the size of D.

Pricing Problem and Dual Separation Problem

Consider the dual LP:

$$\begin{aligned}
\max \quad & \sum_{a \in A} u(a) \cdot y_a + \sum_{i=1}^{k} d_i \cdot z_i \\
\text{s.t.} \quad & z_i + \sum_{a \in P} y_a \le c_P && \text{for all } P \in \mathcal{P}_i,\ i = 1, \dots, k \\
& y_a \le 0 && \text{for all } a \in A
\end{aligned}$$

Notice: The reduced cost of a primal variable is negative if and only if the corresponding dual constraint is violated ($\longrightarrow$ dual separation problem).

Easy for slack variable s_a: Check whether $y_a > 0$.

For path variable x_P, $P \in \mathcal{P}_i$:

$$\begin{aligned}
& z_i + \sum_{a \in P} y_a > c_P = \sum_{a \in P} c(a) \\
\iff \quad & \sum_{a \in P} (c(a) - y_a) < z_i
\end{aligned}$$

Conclusion: Solve pricing problem by computing shortest s_i-t_i-paths w.r.t. non-negative arc weights $c(a) - y_a$, for $i = 1, \dots, k$.

Cutting Plane Methods

Delayed column generation viewed in terms of the dual LP:

$$\max \quad p^T \cdot b \qquad \text{s.t.} \quad p^T \cdot A_i \le c_i \qquad \text{for all } i = 1, \dots, n.$$

If n is huge, instead of dealing with all n constraints, restrict to subset $I \subset \{1, \dots, n\}$ and consider relaxed problem

$$\max \quad p^T \cdot b \qquad \text{s.t.} \quad p^T \cdot A_i \le c_i \qquad \text{for all } i \in I.$$

Let p^* be an optimal basic feasible solution:

- If p^* is feasible for original LP, it is also optimal there.
- Otherwise, find a violated constraint and add it to relaxed problem.

Remark: Notice the similarity to the ellipsoid method where, in every iteration, the separation problem needs to be solved.

Example: Solving the Subtour LP

For a given TSP instance, consider the subtour LP:

$$\begin{array}{lll} \min & \displaystyle\sum_{e \in E} c_e \cdot x_e & \\ \text{s.t.} & \displaystyle\sum_{e \in \delta(v)} x_e = 2 & \text{for all nodes } v \in V, \\ & \displaystyle\sum_{e \in \delta(X)} x_e \geq 2 & \text{for all subsets } \emptyset \neq X \subsetneq V, \qquad (*) \\ & 0 \leq x_e \leq 1 & \text{for all edges } e. \end{array}$$

Notice that there are $2^{n-1} - 1$ subtour elimination constraints (*).

The corresponding separation problem is a min-cut problem that can be solved efficiently by network flow methods.

Conclusion: Subtour LP is typically being solved by cutting plane methods.

Dantzig-Wolfe Decomposition

Consider a linear program of the form

$$\begin{array}{lcccccc} \min & c_1{}^T \cdot x_1 & + & c_2{}^T \cdot x_2 & & \\ \text{s.t.} & D_1 \cdot x_1 & + & D_2 \cdot x_2 & = & b_0 \\ & F_1 \cdot x_1 & & & = & b_1 \\ & & & F_2 \cdot x_2 & = & b_2 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

with $c_1 \in \mathbb{R}^{n_1}$, $c_2 \in \mathbb{R}^{n_2}$, $b_0 \in \mathbb{R}^{m_0}$, $b_1 \in \mathbb{R}^{m_1}$, $b_2 \in \mathbb{R}^{m_2}$.

Reformulation of the problem: For $i = 1, 2$, let $P_i := \{x_i \geq 0 \mid F_i \cdot x_i = b_i\}$.

$$\begin{array}{lcccccc} \min & c_1{}^T \cdot x_1 & + & c_2{}^T \cdot x_2 & & \\ \text{s.t.} & D_1 \cdot x_1 & + & D_2 \cdot x_2 & = & b_0 \\ & x_1 \in P_1 & , & x_2 \in P_2 & & \end{array}$$

- Let x_i^j, $j \in J_i$, be the extreme points of P_i.
- Let w_i^k, $k \in K_i$, be a complete set of extreme rays of P_i.

Dantzig-Wolfe Decomposition: The Master Problem

For $i = 1, 2$, any $x_i \in P_i$ can be written as

$$x_i = \sum_{j \in J_i} \lambda_i^j \cdot x_i^j + \sum_{k \in K_i} \theta_i^k \cdot w_i^k$$

with $\lambda_i^j, \theta_i^k \geq 0$ and $\sum_{j \in J_i} \lambda_i^j = 1$.

The reformulation thus leads to the following master problem:

$$\begin{aligned}
\min \sum_{j \in J_1} \lambda_1^j \, (c_1^T x_1^j) + \sum_{k \in K_1} \theta_1^k \, (c_1^T w_1^k) + \sum_{j \in J_2} \lambda_2^j \, (c_2^T x_2^j) + \sum_{k \in K_2} \theta_2^k \, (c_2^T w_2^k) \\
\text{s.t.} \sum_{j \in J_1} \lambda_1^j \begin{pmatrix} D_1 x_1^j \\ 1 \\ 0 \end{pmatrix} + \sum_{k \in K_1} \theta_1^k \begin{pmatrix} D_1 w_1^k \\ 0 \\ 0 \end{pmatrix} + \sum_{j \in J_2} \lambda_2^j \begin{pmatrix} D_2 x_2^j \\ 0 \\ 1 \end{pmatrix} + \sum_{k \in K_2} \theta_2^k \begin{pmatrix} D_2 w_2^k \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} b_0 \\ 1 \\ 1 \end{pmatrix} \\
\lambda_1, \lambda_2, \theta_1, \theta_2 \geq 0
\end{aligned}$$

The master problem has only $m_0 + 2$ constraints but a huge number of variables. $\longrightarrow$ Employ delayed column generation!

Dantzig-Wolfe Decomposition: Pricing Problem

Let B be a feasible basis to the master problem and $p^T := c_B{}^T \cdot B^{-1}$ the associated dual solution: $p^T = (q^T, r_1, r_2)$ with $q \in \mathbb{R}^{m_0}$, $r_1, r_2 \in \mathbb{R}$.

Compute the reduced cost coefficient of a variable λ_1^j:

$$c_1{}^T \cdot x_1^j - (q^T, r_1, r_2) \cdot \begin{pmatrix} D_1 \cdot x_1^j \\ 1 \\ 0 \end{pmatrix} = (c_1{}^T - q^T \cdot D_1) \cdot x_1^j - r_1$$

Compute the reduced cost coefficient of a variable θ_1^k:

$$c_1{}^T \cdot w_1^k - (q^T, r_1, r_2) \cdot \begin{pmatrix} D_1 \cdot w_1^k \\ 0 \\ 0 \end{pmatrix} = (c_1{}^T - q^T \cdot D_1) \cdot w_1^k$$

To solve the pricing problem for variables λ_i^j and θ_i^k, we consider the LP:

$$\begin{aligned} \min \quad & (c_i{}^T - q^T \cdot D_i) \cdot x_i \\ \text{s.t.} \quad & x_i \in P_i \end{aligned}$$

This is called the ith subproblem.

Dantzig-Wolfe Decomposition: Pricing Problem (cont.)

Consider ith subproblem: $\min \; \left(c_i{}^T - q^T \cdot D_i\right) \cdot x_i \quad \text{s.t.} \quad x_i \in P_i$

Case 1: ith subproblem is unbounded:

$\Longrightarrow$ simplex algorithm yields extreme ray w_i^k with $\left(c_i{}^T - q^T \cdot D_i\right) \cdot w_i^k < 0$

$\Longrightarrow$ reduced cost of θ_i^k is negative

$\longrightarrow$ generate column $\begin{pmatrix} D_i w_i^k \\ 0 \\ 0 \end{pmatrix}$ and let it enter the basis in master problem.

Case 2: ith subproblem has finite optimal cost $< r_i$:

$\Longrightarrow$ simplex algorithm yields extreme point x_i^j with $\left(c_i{}^T - q^T \cdot D_i\right) \cdot x_i^j < r_i$

$\Longrightarrow$ reduced cost of λ_i^j is negative

$\longrightarrow$ generate column $\begin{pmatrix} D_i x_i^j \\ \vdots \end{pmatrix}$ and let it enter the basis in master problem.

Case 3: ith subproblem has finite optimal cost $\geq r_i$:

$$\Longrightarrow \quad \begin{array}{ll} \left(c_i{}^T - q^T \cdot D_i\right) \cdot x_i^j \geq r_i & \text{for all } j \in J_i \text{ and} \\ \left(c_i{}^T - q^T \cdot D_i\right) \cdot w_i^k \geq 0 & \text{for all } k \in K_i. \end{array}$$

$\Longrightarrow$ Variables λ_i^j and θ_i^k have reduced cost ≥ 0, for all $j \in J_i$, $k \in K_i$.

Dantzig-Wolfe Decomposition: Summary

- The given problem is transformed into an equivalent problem with few constraints but many variables.
- The pricing problem can be solved by solving smaller LPs over the polyhedra P_i.

Economic interpretation: Organization with two divisions and common objective $D_1 \cdot x_1 + D_2 \cdot x_2 = b_0$.

- Central planner assigns values q for each unit of contribution towards common objective.
- Division i wants to minimize $c_i{}^T \cdot x_i$ s.t. its own constraint $x_i \in P_i$.
- Since x_i contributes $D_i \cdot x_i$ towards common objective, the overall objective for division i is: $\min \; \left(c_i{}^T - q^T \cdot D_i\right) \cdot x$.
- The divisions propose solutions to the central planner who combines them with previous solutions and comes up with new values q.

Dantzig-Wolfe Decomposition: Generalization

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{t} {c_i}^T \cdot x_i \\
\text{s.t.} \quad & \sum_{i=1}^{t} D_i \cdot x_i = b_0 \\
& F_i \cdot x_i = b_i \qquad \text{for } i = 1, \dots, t \\
& x_1, \dots, x_t \geq 0
\end{aligned}
$$

- Proceed as before $\longrightarrow$ t subproblems for pricing.
- Sometimes even useful for $t = 1$.

Dantzig-Wolfe Decomposition: Phase I

How to find an initial basic feasible solution?

- Use phase I of simplex method to find an extreme point x_i^1 of P_i, for $i = 1, \dots, t$.
- W.l.o.g. $\sum_{i=1}^{t} D_i \cdot x_i^1 \leq b_0$. Introduce slack variables $y \in \mathbb{R}^{m_0}$ and solve auxiliary master problem:

$$
\begin{aligned}
\min \quad & \sum_{s=1}^{m_0} y_s \\
\text{s.t.} \quad & \sum_{i=1}^{t} \Big(\sum_{j \in J_i} \lambda_i^j (D_i \cdot x_i^j) + \sum_{k \in K_i} \theta_i^k (D_i \cdot w_i^k) \Big) + y = b_0 \\
& \sum_{j \in J_i} \lambda_i^j = 1 \qquad \text{for } i = 1, \dots, t \\
& \lambda, \theta, y \geq 0
\end{aligned}
$$

Dantzig-Wolfe Decomposition: Example

Arc-based LP formulation of min-cost multi-commodity flow problem:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{t} \Big(\sum_{a \in A} c(a) \cdot x_{i,a} \Big) & \\
\text{s.t.} \quad & \sum_{i=1}^{t} x_{i,a} \le u(a) & \text{for } a \in A \\
& \sum_{a \in \delta^-(v)} x_{i,a} - \sum_{a \in \delta^+(v)} x_{i,a} = \begin{cases} d_i & \text{if } v = t_i \\ -d_i & \text{if } v = s_i \\ 0 & \text{otherwise} \end{cases} & \text{for } i = 1, \dots, t \\
& x \ge 0 &
\end{aligned}
$$

- For $i = 1, \dots, t$, let $P_i := \{x_i \mid x_i \text{ is } s_i\text{-}t_i\text{-flow of value } d_i\}$.
- Extreme points of polyhedron P_i: s_i-t_i-path flows of value d_i (denoted by x_i^P for s_i-t_i-path $P \in \mathcal{P}_i$)

Dantzig-Wolfe Decomposition: Example (cont.)

Master problem:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{t} \Big(\sum_{P \in \mathcal{P}_i} \lambda_i^P \cdot (d_i \, c_P) \Big) & \\
\text{s.t.} \quad & \sum_{i=1}^{t} \Big(\sum_{P \in \mathcal{P}_i : a \in P} \lambda_i^P \cdot d_i \Big) \le u(a) & \text{for } a \in A \\
& \sum_{P \in \mathcal{P}_i} \lambda_i^P = 1 & \text{for } i = 1, \dots, t \\
& \lambda \ge 0 &
\end{aligned}
$$

- Setting $x_P := \lambda_i^P \cdot d_i$ for $P \in \mathcal{P}_i$ yields the path-based LP formulation!
- The ith subproblem (pricing problem for variables λ_i^P, $P \in \mathcal{P}_i$) is a shortest s_i-t_i-path problem.

Chapter 12: Interior Point Methods

(cp. Bertsimas & Tsitsiklis, Chapter 9)

Basic Ideas and Facts

- Approach an optimal LP solution while moving in the interior of the feasible set. $\longrightarrow$ "interior point methods"
- Interior point methods combine the advantages of the simplex method and of the ellipsoid method.
- From a theoretical point of view, they lead to polynomial time algorithms and use interesting geometric ideas.
- In practice, interior point methods are competitive with the simplex method.
- For large, sparse problems, they often outperform the simplex method.

Three Major Types of Interior Point Methods

i Affine scaling algorithm

- simplest interior point algorithm
- iteratively uses ellipsoids contained in the polyhedron
- in each iteration, optimize objective function over the ellipsoid
- very good practical performance
- closest to the simplex algorithm
- moves approximately along edges of the feasible set

ii Potential reduction algorithm

- does not measure progress towards optimality by reduction in objective function value (as simplex and affine scaling)
- instead uses nonlinear potential function balancing two conflicting objectives:
  1. decreasing the objective function value
  2. staying away from the boundary of the feasible set
- polynomial worst-case running time

Three Major Types of Interior Point Methods (Cont.)

iii Path following algorithms

- transform constrained LP to an unconstrained problem with logarithmic barrier function imposing growing penalty towards the boundary
- solve unconstrained problem approximately by Newton's method
- iteratively decrease strength of barrier function such that optimum follows certain path ending at optimal LP solution
- method of choice in large scale implementations
- polynomial worst-case running time

Affine Scaling Algorithm: Basic Idea

Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and consider primal-dual pair of LPs:

$$\begin{array}{rl} \min & c^T \cdot x \\ \text{s.t.} & A \cdot x = b \\ & x \geq 0 \end{array} \qquad\qquad \begin{array}{rl} \max & p^T \cdot b \\ \text{s.t.} & p^T \cdot A \leq c^T \end{array}$$

Definition 12.1.

Let $P := \{x \in \mathbb{R}^n \mid A \cdot x = b,\, x \geq 0\}$. A point $x \in P$ with $x > 0$ is an interior point of P. The set $\{x \in P \mid x > 0\}$ is the interior of P.

Main idea:

- Start with interior point x^0.
- Form ellipsoid S_0 centered at x^0 and contained in the interior of P.
- Optimize $c^T \cdot x$ over all $x \in S_0$ and find x^1.
- Form ellipsoid S_1 centered at x^1 etc.

Affine Scaling Algorithm: An Ellipsoid Contained in P

Lemma 12.2.

Let $\beta \in (0,1)$, $y \in \mathbb{R}^n$ with $y > 0$ and

$$S := \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{{y_i}^2} \leq \beta^2 \right\} \quad .$$

Then, $x > 0$ for every $x \in S$.

Proof: . . . □

- Fix some $y > 0$ with $A \cdot y = b$ and set $Y := \mathrm{diag}(y_1, \ldots, y_n) \in \mathbb{R}^{n \times n}$.
- Then,

$$x \in S \quad \iff \quad \|Y^{-1} \cdot (x - y)\|_2 \leq \beta \ ,$$

- That is, S is an ellipsoid centered at y.
- Let $S_0 := \{x \in S \mid A \cdot x = b\}$ (section of ellipsoid S).
- S_0 is itself an ellipsoid contained in the interior of P.

Affine Scaling Algorithm: Optimizing over the Ellipsoid

We replace the original LP with $\min c^T \cdot x$ s.t. $x \in S_0$.

$$\min \quad c^T \cdot x \qquad \text{s.t.} \quad A \cdot x = b, \quad \|Y^{-1} \cdot (x - y)\|_2 \leq \beta$$

Reformulate by setting $d := x - y$.

$$\min \quad c^T \cdot d \qquad \text{s.t.} \quad A \cdot d = 0, \quad \|Y^{-1} \cdot d\|_2 \leq \beta \tag{12.1}$$

Lemma 12.3.

Assume that rows of A are linearly independent and c is not a linear combination of rows of A. An optimal solution d^* to (12.1) is given by

$$d^* := -\beta \frac{Y^2 \cdot (c - A^T \cdot p)}{\|Y \cdot (c - A^T \cdot p)\|_2} \quad \text{with} \quad p := (A \cdot Y^2 \cdot A^T)^{-1} \cdot A \cdot Y^2 \cdot c \ .$$

Moreover, $x := y + d^* \in P$ and

$$c^T \cdot x = c^T \cdot y - \beta \|Y \cdot (c - A^T \cdot p)\|_2 < c^T \cdot y \ .$$

Proof: See Bertsimas & Tsitsiklis, proof of Lemma 9.2. □

Remark: If $d^* \geq 0$, the LP is unbounded since $c^T \cdot d^* < 0$, $A \cdot d^* = 0$, and $y + \alpha \cdot d^* > 0$ for all $\alpha \geq 0$.

Affine Scaling Algorithm: Interpretation of p

- To interpret p, assume for a moment that y is a non-degenerate basic feasible solution (contradicting $y > 0$!) with basis B and $A = [B, N]$.
- Let $Y := \text{diag}(y_1, \ldots, y_m, 0, \ldots, 0)$ and $Y_0 := \text{diag}(y_1, \ldots, y_m)$.
- Then, $A \cdot Y = [B \cdot Y_0, 0]$ and
$$\begin{aligned} p :=& (A \cdot Y^2 \cdot A^T)^{-1} \cdot A \cdot Y^2 \cdot c \\ =& (B^T)^{-1} \cdot Y_0^{-2} \cdot B^{-1} \cdot B \cdot Y_0^2 \cdot c_B = (B^T)^{-1} \cdot c_B \end{aligned}$$
is the corresponding dual basic solution.
- Thus, the vector p, corresponding to a primal solution y, is called dual estimate, even if y is not basic.
- The vector $r := c - A^T \cdot p$ becomes $r = c - A^T \cdot (B^T)^{-1} \cdot c_B$, i. e., the reduced cost vector.
- Notice that, if y is degenerate, the matrix $A \cdot Y^2 \cdot A^T$ is not invertible and this interpretation breaks down.

Affine Scaling Algorithm: Duality Gap

If $r = c - A^T \cdot p \geq 0$, then p is a dual feasible solution and

$$r^T \cdot y = (c - A^T \cdot p)^T \cdot y = c^T \cdot y - p^T \cdot A \cdot y = c^T \cdot y - p^T \cdot b$$

is the difference in objective function values between y and p, called the duality gap.

If $r^T \cdot y = 0$, complementary slackness cond. hold and y, p are optimal.

Lemma 12.4.

Let y and p be a primal and dual feasible solution, respectively, such that

$$r^T \cdot y = c^T \cdot y - p^T \cdot b < \varepsilon \ .$$

Let y^* and p^* be optimal primal and dual feasible solutions, respectively. Then,

$$c^T \cdot y^* \leq c^T \cdot y < c^T \cdot y^* + \varepsilon \ ,$$
$$b^T \cdot p^* - \varepsilon < b^T \cdot p \leq b^T \cdot p^* \ .$$

Proof:. . . □

Affine Scaling Algorithm

1. Start with feasible $x^0 > 0$; set $k := 0$; (initialization)
2. Let
$$\begin{aligned} X_k &:= \mathrm{diag}(x_1^k, \dots, x_n^k) \\ p^k &:= (A \cdot {X_k}^2 \cdot A^T)^{-1} \cdot A \cdot {X_k}^2 \cdot c \\ r^k &:= c - A^T \cdot p^k \end{aligned}$$
(computation of dual estimates and reduced costs)
3. If $r^k \geq 0$ and ${r^k}^T \cdot x^k < \varepsilon$, then stop;
(optimality check; x^k is primal ε-optimal and p^k is dual ε-optimal)
4. If $-{X_k}^2 \cdot r^k \geq 0$, then stop;
(unboundedness check; optimal cost is $-\infty$)
5. Let
$$x^{k+1} := x^k - \beta \cdot \frac{{X_k}^2 \cdot r^k}{\|X_k \cdot r^k\|_2} \ ;$$
set $k := k + 1$ and go to 2. (update of primal solution)

Affine Scaling Algorithm: Convergence

Assumptions:

- rows of A are linearly independent
- c is not a linear combination of rows of A
- there exists an optimum solution
- there exists a positive feasible solution
- every basic feasible solution to the primal is non-degenerate
- at every basic feasible solution to the primal, the reduced costs of non-basic variables are all non-zero

Theorem 12.5.

Under these assumptions, for $\varepsilon = 0$, the algorithm converges to a pair of primal and dual optimal solutions. $\square$

Affine Scaling Algorithm: Initialization and Performance

Initialization: Consider auxiliary problem ($e = (1, \dots, 1)^T$):

$$\begin{aligned} \min \;\; & c^T \cdot x + M\, x_{n+1} \\ \text{s.t.} \;\; & A \cdot x + (b - A \cdot e)\, x_{n+1} = b \\ & (x, x_{n+1}) \geq 0 \end{aligned}$$

Notice that $(e, 1)$ is a positive feasible solution.

Computational performance:

- Simple algorithm with excellent performance in practice.
- Computational bottleneck in each iteration: calculation of p^k.
- Computing matrix $A \cdot {X_k}^2 \cdot A^T$ takes $O(m^2 n)$ arithmetic operations.
- Solving system of linear equations involving matrix $A \cdot {X_k}^2 \cdot A^T$ takes $O(m^3)$ arithmetic operations.
- In total, $O(m^2 n)$ arithmetic operations per iteration.

Potential Reduction Algorithm: Potential Function

Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and consider primal-dual pair of LPs:

$$\begin{array}{llcll} \min & c^T \cdot x & \qquad & \max & p^T \cdot b \\ \text{s.t.} & A \cdot x = b & & \text{s.t.} & p^T \cdot A + s^T = c^T \\ & x \geq 0 & & & s \geq 0 \end{array}$$

Assume that rows of A are linearly independent and there exists $x^0 > 0$ and (p^0, s^0) with $s^0 > 0$ which are feasible for primal and dual LP, respectively.

Idea: Stay away from the boundary!

Potential function:

$$G(x,s) := q \log(s^T \cdot x) - \sum_{j=1}^{n} \log x_j - \sum_{j=1}^{n} \log s_j$$

where q is a constant larger than n.

Potential Reduction Algorithm: Duality Gap

If x and (p,s) are feasible, then the duality gap is

$$c^T \cdot x - b^T \cdot p = (s^T + p^T \cdot A) \cdot x - x^T \cdot A^T \cdot p = s^T \cdot x$$

Theorem 12.6.
An algorithm that maintains primal and dual feasibility and reduces $G(x,s)$ by at least $\delta > 0$ at each iteration, finds solutions with duality gap $(s^k)^T \cdot x^k \leq \varepsilon$ after

$$K := \left\lceil \frac{G(x^0, s^0) + (q-n)\log(1/\varepsilon) - n \log n}{\delta} \right\rceil$$

iterations.

Proof: . . . $\square$

Potential Reduction Algorithm: Basic Idea

- Start with feasible $x > 0$ and (p, s) with $s > 0$.
- Try to find direction d such that

$$G(x + d, s) < G(x, s) \quad \text{and} \quad A \cdot d = 0, \quad \|X^{-1} \cdot d\|_2 \leq \beta < 1 \ .$$

- Minimizing $G(x + d, s)$ s.t. $A \cdot d = 0$, $\|X^{-1} \cdot d\|_2 \leq \beta$ is difficult due to objective function (non-linear, non-convex).
- Linearize the objective function by taking the first order Taylor series expansion in d:

$$\begin{aligned} \min \quad & \left(\nabla_x G(x, s)\right)^T \cdot d \\ \text{s.t.} \quad & A \cdot d = 0 \\ & \|X^{-1} \cdot d\| \leq \beta \end{aligned}$$

Note: Same as in affine scaling except for different objective funct. $\hat{c}$

$$\hat{c}_i := \frac{\partial G(x, s)}{\partial x_i} = \frac{q \cdot s_i}{s^T \cdot x} - \frac{1}{x_i} \ .$$

Potential Reduction Algorithm: Basic Idea (cont.)

- Applying Lemma 12.3 with $Y := X$ and $c := \hat{c}$, we obtain optimal direction

$$d^* := -\beta \cdot X \cdot \frac{u}{\|u\|_2}$$

with $u := X \cdot \left(\hat{c} - A^T \cdot (A \cdot X^2 \cdot A^T)^{-1} \cdot A \cdot X^2 \cdot \hat{c}\right)$.

- Since $X \cdot \hat{c} = \dfrac{q}{s^T \cdot x} \cdot X \cdot s - e$, we obtain

$$u = \left(I - X \cdot A^T \cdot (A \cdot X^2 \cdot A^T)^{-1} \cdot A \cdot X\right) \cdot \left(\frac{q}{s^T \cdot x} \cdot X \cdot s - e\right) \ .$$

- Moreover, $G(x, s)$ decreases by $\beta \cdot \|u\|_2 + O(\beta^2)$, where the first term comes from Lemma 12.3 and the second is due to omitted higher order terms in the Taylor series expansion of $G(x, s)$.
- Thus, if $\|u\|_2$ is large enough, then $G(x, s)$ decreases by at least δ.
- Otherwise update dual variables.

Potential Reduction Algorithm

1. Let $x^0 > 0$ feasible, (p^0, s^0) with $s^0 > 0$; set $k := 0$; (initialization)
2. If $(s^k)^T \cdot x^k < \varepsilon$, then stop; (optimality test)
3. Let
$$X_k := \operatorname{diag}(x_1^k, \ldots, x_n^k)$$
$$\bar{A}^k := (A \cdot X_k)^T \cdot (A \cdot X_k{}^2 \cdot A^T)^{-1} \cdot A \cdot X_k$$
$$u^k := (I - \bar{A}^k) \cdot \left(\frac{q}{(s^k)^T \cdot x^k} \cdot X_k \cdot s^k - e \right)$$
$$d^k := -\beta \cdot X_k \cdot \frac{u^k}{\|u^k\|_2} \qquad \text{(update direction)}$$
4. If $\|u^k\|_2 \geq \gamma$, then $x^{k+1} := x^k + d^k$, $s^{k+1} := s^k$, $p^{k+1} := p^k$; (primal step)
5. If $\|u^k\|_2 < \gamma$, then $x^{k+1} := x^k$, $s^{k+1} := \frac{(s^k)^T \cdot x^k}{q} \cdot X_k{}^{-1} \cdot (u^k + e)$,
$$p^{k+1} := p^k + (A \cdot X_k{}^2 \cdot A^T)^{-1} \cdot A \cdot X_k \cdot \left(X_k \cdot s^k - \frac{(s^k)^T \cdot x^k}{q} \cdot e \right) \ ;$$
(dual step)
6. Let $k := k + 1$ and go to 2;

Potential Reduction Algorithm: Behavior

For every k, the vectors x^k and (p^k, s^k) are primal and dual feasible solutions, respectively.

Theorem 12.7.

Let $\beta < 1$ and $\gamma < 1$:

a. If $\|u^k\|_2 \geq \gamma$ (primal step), then
$$G(x^{k+1}, s^{k+1}) - G(x^k, s^k) \leq -\beta \cdot \gamma + \frac{\beta^2}{2(1-\beta)} \ .$$
b. If $\|u^k\|_2 < \gamma$ (dual step), then
$$G(x^{k+1}, s^{k+1}) - G(x^k, s^k) \leq -(q - n) + n \log \frac{q}{n} + \frac{\gamma^2}{2(1-\gamma)} \ .$$
c. If $q = n + \sqrt{n}$, $\beta \approx 0.285$ and $\gamma \approx 0.479$, then the potential reduction algorithm reduces $G(x, s)$ by at least $\delta := 0.079$ at each iteration.

Proof: See Bertsimas & Tsitsiklis, proof of Theorem 9.5. □

Potential Reduction Algorithm: Running Time

Initialization: Use auxiliary problem (similar to affine scaling algorithm). For details, see Bertsimas & Tsitsiklis, Sect. 9.3.

Running Time:

- The potential reduction algorithm finds ε-optimal solutions in time

  $$O\big(n^{3.5}\log\varepsilon^{-1} + n^5\log(n\,U)\big)$$

  where $U := \max\{|a_{ij}|, |b_i|, |c_j|\}$ (all integer).

- If ε is taken sufficiently small, an optimal solution can be found by rounding. This results in a polynomial (in n and $\log U$) time algorithm.

Further Interior Point Algorithms

- Primal path following algorithm: see Bertsimas & Tsitsiklis, Sect. 9.4.

- Primal-dual path following algorithm: see Bertsimas & Tsitsiklis, Sect. 9.5.