Introduction to Mathematical Programming
(Linear Programming)
Algorithmics, 186.814, VU 4.0

Daniel Obszelka

Algorithms and Complexity Group
Institute of Logic and Computation
TU Wien

WS 2019/20

Topics of this Part

- Model-based Algorithms
- Linear Programming
  - Introduction
  - Geometry
  - Algorithms
  - Duality Theory
- (Mixed) Integer Linear Programming
  - Formulations
  - Relaxations/Bounds
  - Branch-and-Bound

Model-based Algorithms

Algorithm Development

Traditional

1. identify the problem
2. learn about the problem's properties
3. design an algorithm
4. implement the algorithm
5. compute a solution

Model-based

1. identify the problem
2. learn about the problem's properties
3. design a model
4. feed the model to a solver
5. obtain a solution from the solver

Variants

- Linear Programming / (Mixed) Integer Linear Programming
  - based on mathematical equalities and inequalities
  - based on linear programming algorithms and branch-and-bound
  - covered in the remainder of this block
  - $\Rightarrow$ advanced topics: VU Mathematical Programming
- Constraint Programming
  - based on various types of constraints
  - solved via constraint propagation and branch-and-bound
  - $\Rightarrow$ VU Modeling and Solving Constrained Optimization Problems
- (MAX-)SAT
  - only binary variables
  - efficient w.r.t. feasibility problems
  - limited applicability for optimization

These approaches guarantee to provide provably optimal solutions (given sufficient time).

Example: The Maximum Flow Problem

Definition (Recap)

Given:

- directed graph $G = (V, A)$, source $s \in V$, target $t \in V$
- $\deg^-(s) = \deg^+(t) = 0$
- capacities $c \colon A \to \mathbb{N}_{\geq 0}$
- flow conservation for all $v \in V \setminus \{s, t\}$

The goal is to assign flow to the arcs s.t. the above constraints are satisfied and the flow leaving the source is maximized.

Maximum Flow Problem: Problem-specific Algorithm

Ford-Fulkerson Algorithm

- not so easy to come up with
- non-trivial implementation details: selecting augmenting path

Maximum Flow Problem: Linear Program

$$
\begin{aligned}
\max \quad & \sum_{(s,i)\in A} f_{si} && && (1)\\
\text{s.t.} \quad & \sum_{(i,v)\in A} f_{iv} = \sum_{(v,j)\in A} f_{vj}, && \forall v \in V \setminus \{s,t\}, && (2)\\
& 0 \le f_a \le c_a, && \forall a \in A. && (3)
\end{aligned}
$$

- essentially the same as the formal description
- can be solved directly

Maximum Flow Problem: Adjustments

What if also a demand ($d\colon A \to \mathbb{N}_{\geq 0}$) has to be satisfied per arc?

Ford-Fulkerson Algorithm

- can be adapted
- requires non-trivial adjustments of the algorithm or input transformation

Linear Program

- simply switch
  $0 \leq f_a \leq c_a, \qquad \forall a \in A,$
  to
  $d_a \leq f_a \leq c_a, \qquad \forall a \in A,$
- The implementation of the model can be adjusted by changing a single line of code!

Linear Programming

Literature

1. D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, 1997.
2. 6.251J/15.081J Introduction to Mathematical Programming - Fall 2009, MIT OpenCourseWare (http://ocw.mit.edu).
3. G. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, 1999.
4. A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, 1986.

Linear Programming

Introduction

What is Linear Optimization / Linear Programming?

Definition (Linear Programming)

Linear programming (LP) is the problem of minimizing a linear cost function subject to linear equality and inequality constraints.

Example

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

General Linear Programming Problem

In the general case we are given a

- cost vector $\boldsymbol{c} = (c_1, \ldots, c_n)$

and seek to minimize a

- linear cost function $\boldsymbol{c}'\boldsymbol{x} = \sum_{i=1}^{n} c_i x_i$

over all n-dimensional vectors $\boldsymbol{x} = (x_1, \ldots, x_n)$, subject to a set of

- linear equality and inequality constraints.

General Linear Programming Problem

$$
\begin{aligned}
\min \quad & \boldsymbol{c}'\boldsymbol{x} & \\
\text{s.t.} \quad & \boldsymbol{a_i}'\boldsymbol{x} \geq b_i, & \forall i \in M_1, \\
& \boldsymbol{a_i}'\boldsymbol{x} \leq b_i, & \forall i \in M_2, \\
& \boldsymbol{a_i}'\boldsymbol{x} = b_i, & \forall i \in M_3, \\
& x_j \geq 0, & \forall j \in N_1, \\
& x_j \leq 0, & \forall j \in N_2.
\end{aligned}
$$

- decision variables $x_1, \ldots, x_n$
- objective (cost) function $\boldsymbol{c}'\boldsymbol{x}$
- free (unrestricted) variables x_j, $j \notin N_1 \cup N_2$
- M_1, M_2, M_3 finite index sets for the constraints
- N_1, N_2 subsets of $\{1, \ldots, n\}$ for variables constraints

Further Notations and Definitions

- vector $\boldsymbol{x}$ satisfying all constraints is called a feasible solution or feasible vector
- set of feasible solutions is called feasible set or feasible region
- a feasible solution $\boldsymbol{x}^*$ that minimizes the objective function ($\boldsymbol{c}'\boldsymbol{x}^* \leq \boldsymbol{c}'\boldsymbol{x}$ for all feasible solutions $\boldsymbol{x}$) is called an optimal feasible solution or optimal solution;
  value $\boldsymbol{c}'\boldsymbol{x}^*$ is then called optimal cost

Transformation Rules

1. $\max \boldsymbol{c}'\boldsymbol{x} \Leftrightarrow \min -\boldsymbol{c}'\boldsymbol{x}$
2. $\boldsymbol{a_i}'\boldsymbol{x} = b_i \Leftrightarrow \begin{array}{l} \boldsymbol{a_i}'\boldsymbol{x} \leq b_i \\ \boldsymbol{a_i}'\boldsymbol{x} \geq b_i \end{array}$
3. $\boldsymbol{a_i}'\boldsymbol{x} \leq b_i \Leftrightarrow -\boldsymbol{a_i}'\boldsymbol{x} \geq -b_i$
4. $x_j \geq 0$ and $x_j \leq 0$ are special cases of $\boldsymbol{a_i}'\boldsymbol{x} \geq b_i$
   (j-th unit vector $\boldsymbol{a_i}$, $b_i = 0$)

$\Rightarrow$ feasible set of a linear programming problem can be expressed exclusively in terms of inequality constraints of the form $\boldsymbol{a_i}'\boldsymbol{x} \geq b_i$.

Suppose there are

- m constraints (indexed by $i = 1, \ldots, m$)
- $\boldsymbol{A}$ is an $m \times n$ matrix with row vectors $\boldsymbol{a_i}'$, $i = 1, \ldots, m$
- $\boldsymbol{b} = (b_1, \ldots, b_m)$

Definition (Linear Program in General Form (Compact))

$$\begin{aligned} \min \quad & \boldsymbol{c}'\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \in \mathbb{R}^n. \end{aligned}$$

Definition (Feasible Set)

$$P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$$

Where do Linear Optimization Problems arise?

Real world problems

- may have a (straightforward) description as LP or
- may be approximated as LPs
- modeling is an important task!
- (mixed) integer linear programs
  - ⇒ final part of this introduction
  - ⇒ *VU Mathematical Programming* (summer term)

- transportation and logistics
- telecommunications
- scheduling
- manufacturing
- engineering
- portfolio optimization
- ...

Exemplary Transportation Problem (Specification)

Given

- m factories
- n warehouses
- s_i supply of i-th factory, $i = 1, \ldots, m$
- d_j demand of j-th warehouse, $j = 1, \ldots, n$
- c_{ij} transportation costs per unit from factory i to warehouse j
- $\sum_{i=1}^{m} s_i \geq \sum_{j=1}^{n} d_j$

Objective

- find a minimum cost transportation plan such that
  - all demands are satisfied and
  - all supplies are respected

Exemplary Transportation Problem (Formulation)

Decision variables

- x_{ij}... number of units to send from factory i to warehouse j

Formulation (Linear Program)

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{i=1}^{m} x_{ij} = d_j, & j = 1, \ldots, n, \\
& \sum_{j=1}^{n} x_{ij} \leq s_i, & i = 1, \ldots, m, \\
& x_{ij} \geq 0. &
\end{aligned}
$$

Exemplary (Nurse) Scheduling Problem (Data)

Given

- planning horizon of seven days ($J = \{1, \ldots, 7\}$)
- nurses work five days in a row
- d_j: number of nurses required on day $j \in J$

Objective

- identify the minimum number of nurses to be hired

Exemplary Scheduling Problem (Formulation)

Decision variables

- $x_j \ldots$ number of nurses starting to work on day $j \in J$

Formulation (Linear Program)

$$
\begin{array}{lllllllll}
\min & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & & \\
\text{s.t.} & x_1 & & & +x_4 & +x_5 & +x_6 & +x_7 & \geq d_1 & \\
 & x_1 & +x_2 & & & +x_5 & +x_6 & +x_7 & \geq d_2 & \\
 & x_1 & +x_2 & +x_3 & & & +x_6 & +x_7 & \geq d_3 & \\
 & x_1 & +x_2 & +x_3 & +x_4 & & & +x_7 & \geq d_4 & \\
 & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & & & \geq d_5 & \\
 & & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & & \geq d_6 & \\
 & & & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & \geq d_7 & \\
 & & & & & & & x_j & \geq 0, & \forall j \in J.
\end{array}
$$

Guidelines

How to formulate

1. define your decision variables
2. write constraints and objective function

Good LP formulations often have

- a small number of variables and constraints
- sparse constraint matrix

Reminder: Models must be linear, e.g., you are not allowed to multiply two variables!

Standard Form Problems

Definition (Standard Form)

A linear programming problem of the form

$$\begin{aligned} \min \quad & \boldsymbol{c}'\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned}$$

is said to be in standard form.

Interpretation

- $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{A}_i$ is ith column of $\boldsymbol{A}$
- then $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ can be written as $\sum_{i=1}^{n} \boldsymbol{A}_i x_i = \boldsymbol{b}$
- $\boldsymbol{A}_i$ are resource vectors that should synthesize $\boldsymbol{b}$ by using a nonnegative amount x_i of each resource
- c_i is the unit cost of each resource

Reduction to Standard Form

- obviously, standard form is a special case of general form
- we will see that general form can be transformed into standard form

Note

- usually, the general form $\boldsymbol{Ax} \geq \boldsymbol{b}$ is used to develop the theory of linear programming
- standard form $\boldsymbol{Ax} = \boldsymbol{b}$, $\boldsymbol{x} \geq \boldsymbol{0}$, is more convenient for use in algorithms (e.g., simplex algorithms)

Reduction Rules

Elimination of free variables

Assume x_j is an an unrestricted variable in a general form problem

- $x_j \Leftrightarrow x_j^+ - x_j^-$ with $x_j^+, x_j^- \geq 0$

Elimination of inequality constraints

Given an inequality $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$

- introduce a slack variable $s_i \geq 0$
- $\sum_{j=1}^{n} a_{ij} x_j + s_i = b_i$

Given an inequality $\sum_{j=1}^{n} a_{ij} x_j \geq b_i$

- introduce a surplus variable $s_i \geq 0$
- $\sum_{j=1}^{n} a_{ij} x_j - s_i = b_i$

Reduction to Standard Form (Example)

The problem

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

is equivalent to the standard form problem

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

Graphical Representation of LPs

Motivation

- provide useful geometric insights into the nature of linear programming problems
- LPs with two variables can (easily) be solved geometrically

HowTo (Minimization Problem)

1. determine feasible region
2. move line corresponding to objective function in direction of decreasing objective value
3. stop if no further moving is possible without leaving the feasible region
4. optimal solutions are then given by all feasible points of the "objective function line"

Graphical Representation of LPs (Example 1)

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

- line $z = \boldsymbol{c}'\boldsymbol{x} = -x_1 - x_2$ is perpendicular to $\boldsymbol{c}$
- $z = -2$ is the optimal cost

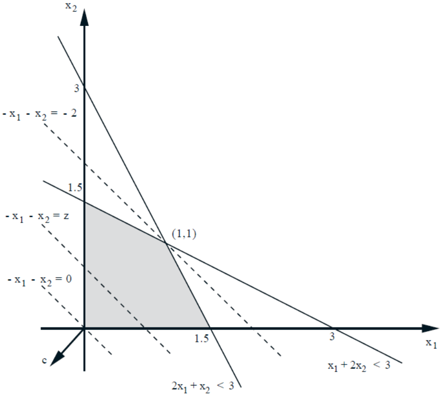


Figure from Bertsimas, Tsitsiklis, 1997

Graphical Representation of LPs (Example 2)

$$-x_1 + x_2 \leq 1 \quad \text{and} \quad x_1, x_2 \geq 0$$

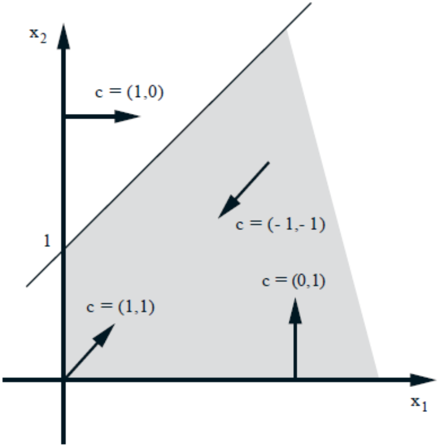


Figure from Bertsimas, Tsitsiklis, 1997

Optimal Solutions of a Linear Program

For a linear program

$$\min \boldsymbol{c}'\boldsymbol{x}$$
$$\text{s.t. } \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b},$$

with feasible set $P = \{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ the following possibilities exist

- $P = \emptyset \Rightarrow$ no feasible solution exists
- optimal cost is $-\infty$, i.e., $\inf\{\boldsymbol{c}'\boldsymbol{x} \mid \boldsymbol{x} \in P\}$ does not exist, and hence no feasible solution is optimal (LP is unbounded)
- a unique optimal solution exists
- multiple optimal solutions exist; set of optimal solutions may be bounded or unbounded

Oil Refinery (Modeling)

- create different end products from raw oil
  - heavy oil (H)
  - medium oil (M)
  - light oil (L)
- different crack processes produce different quantities
  - crack process 1: 2H, 2M, 1L   costs: 3 EUR
  - crack process 2: 1H, 2M, 4L   costs: 5 EUR
- required quantities
  - 3H, 5M, 4L
- Objective: minimize costs

Oil Refinery (Variables)

Variables

- x_1: level of production of crack process 1
- x_2: level of production of crack process 2

Interpretation of $x_1 = 2.5$

- process 1 is applied to 2.5 units raw oil
- costs $2.5 \cdot 3$ EUR
- result $2.5 \cdot 2$ H, $2.5 \cdot 2$ M and $2.5 \cdot 1$ L

Each non-negative vector $(x_1, x_2) \in \mathbb{R}^2$ describes a certain level of production of both crack processes.

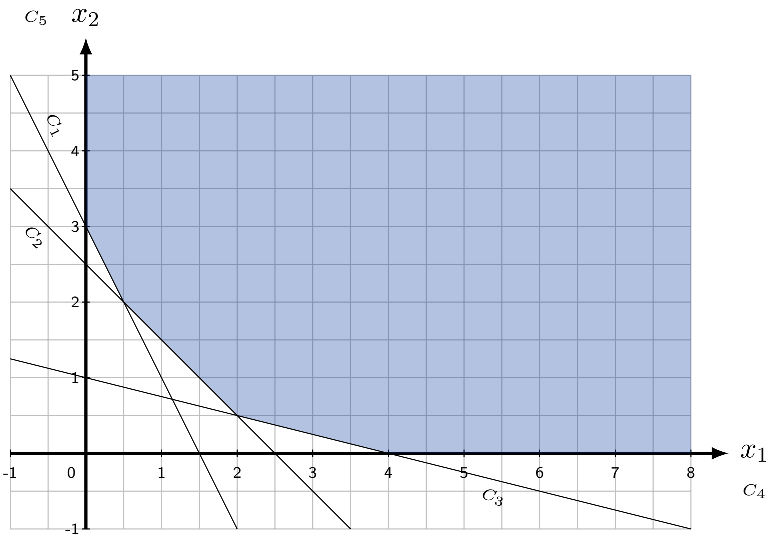
Oil Refinery (Constraints)

- at least 3 units H
  $2x_1 + x_2 \geq 3$
- at least 5 units M
  $2x_1 + 2x_2 \geq 5$
- at least 4 units L
  $x_1 + 4x_2 \geq 4$
- total costs
  $z = 3x_1 + 5x_2$

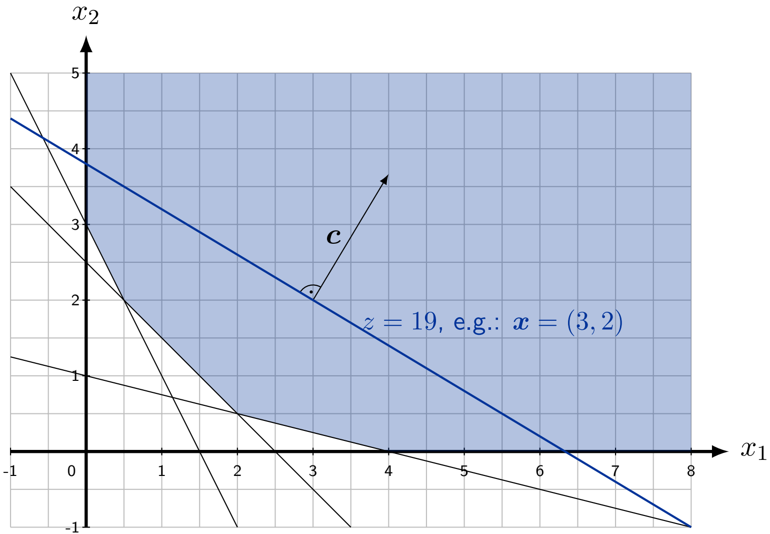
Oil Refinery (Linear Program)

$$
\begin{array}{rrcrcrl}
\min & 3x_1 & + & 5x_2 & & & \\
\textsf{s.t.} & 2x_1 & + & x_2 & \geq & 3 & (C_1) \\
 & 2x_1 & + & 2x_2 & \geq & 5 & (C_2) \\
 & x_1 & + & 4x_2 & \geq & 4 & (C_3) \\
 & x_1 & & & \geq & 0 & (C_4) \\
 & & & x_2 & \geq & 0. & (C_5)
\end{array}
$$

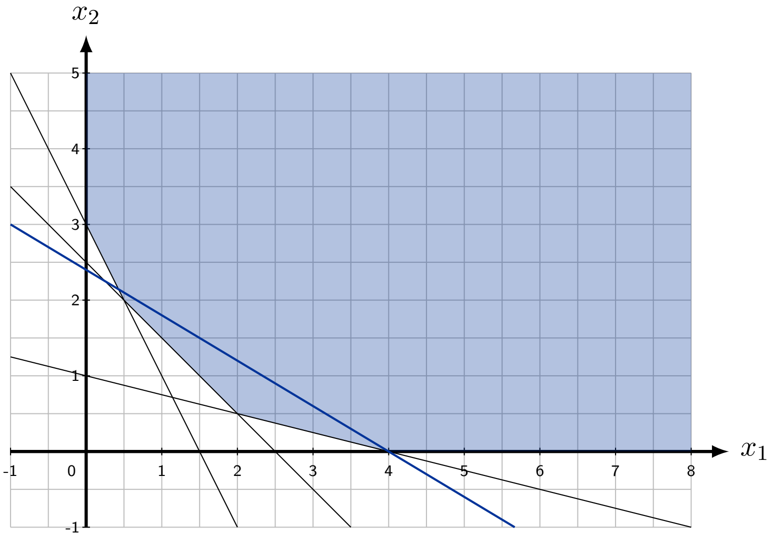
Oil Refinery (Graphical Solution)



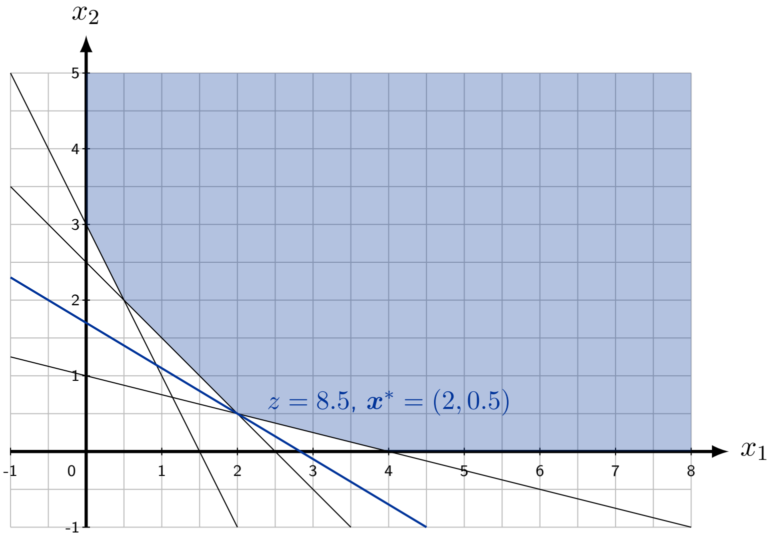
Oil Refinery (Graphical Solution)



Oil Refinery (Graphical Solution)



Oil Refinery (Graphical Solution)



Linear Programming

Geometry

Linear Combination

Definition

Let $x_1, x_2, \ldots, x_k \in \mathbb{R}$ and

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} \in \mathbb{R}^k.$$

Then

$$y = \sum_{i=1}^{k} \lambda_i x_i$$

is called *linear combination* of $x_1, \ldots, x_k$.

(and analogously for $x_1, \ldots, x_k \in \mathbb{R}^n$)

Specific Linear Combinations

Definition

If

$$\boldsymbol{y} = \sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i$$

and

$$\left.\begin{array}{l} \boldsymbol{\lambda} \geq \mathbf{0} \\ \sum_{i=1}^{k} \lambda_i = 1 \\ \sum_{i=1}^{k} \lambda_i = 1,\ \boldsymbol{\lambda} \geq \mathbf{0} \end{array}\right\} \text{ then } y \text{ is called } \left\{\begin{array}{l} conic \\ affine \\ convex \end{array}\right\}$$

combination of $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k$.

Convex Hull and Convex Set

Definition

For $\emptyset \neq S \subseteq \mathbb{R}^n$ $\operatorname{conv}(S)$ is the *convex hull* of S.

Definition

A set $S \subseteq \mathbb{R}^n$ is *convex* if for any $\boldsymbol{x}, \boldsymbol{y} \in S$ and any $\lambda \in [0, 1]$, we have $\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in S$.

Hyperplanes, Halfspaces, and Polyhedra

Definition (Polyhedron)

A polyhedron P is a set that can be described in the form
$P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$ where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$.
(cf. feasible set of LPs)

Definition

A set $S \subseteq \mathbb{R}^n$ is bounded if there exists a constant K such that the absolute value of every component of every element of S is less than or equal to K.

Definition (Polytope)

A bounded polyhedron is called polytope.

Hyperplanes, Halfspaces, and Polyhedra

Definition

Let $\boldsymbol{a}$ be a nonzero vector in $\mathbb{R}^n$ and let b be a scalar

(a) the set $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{x} \geq b\}$ is called a halfspace

(b) the set $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{x} = b\}$ is called a hyperplane

- a hyperplane is the boundary of a corresponding halfspace
- halfspaces are polyhedra
- each polyhedron is equal to the intersection of a finite number of halfspaces
- a polyhedron P is a convex set, i.e., if $\boldsymbol{x}, \boldsymbol{y} \in P$, then $\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in P$ for all $\lambda \in [0, 1]$
  $\Rightarrow$ the feasible set of each LP is a convex set

Hyperplanes, Halfspaces, and Polyhedra

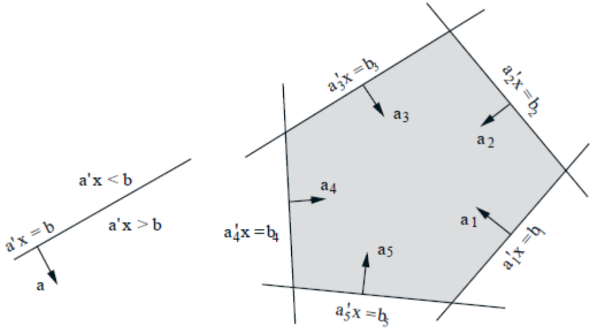


Figure from Bertsimas, Tsitsiklis, 1997

Why is convexity important?

Considering a cost function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a minimization objective:

Definition (Local Optimum)

Vector $\boldsymbol{x}$ is a *local optimum* of f if $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$ for all $\boldsymbol{y}$ in the vicinity of $\boldsymbol{x}$.

Definition (Global Optimum)

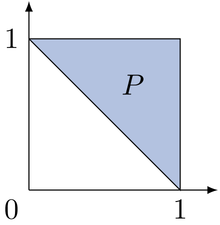
Vector $\boldsymbol{x}$ is a *global optimum* of f if $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$ for all $\boldsymbol{y}$.

Important Property

- A convex function cannot have local minima that fail to be global minima.

Polytopes and Convex Hull

Polytopes can be represented as $P = \operatorname{conv}(V)$ and vice versa.



Polytope

$$P = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2 \mid x \leq 1,\ y \leq 1,\ x + y \geq 1 \right\}$$

can be represented as

$$\operatorname{conv}(\{(0,1),(1,0),(1,1)\}).$$

Extreme Points

Definition (Extreme Point)

Let P be a polyhedron. A vector $\boldsymbol{x} \in P$ is an extreme point of P if we cannot find two vectors $\boldsymbol{y}, \boldsymbol{z} \in P$, both different from $\boldsymbol{x}$, and a scalar $\lambda \in [0, 1]$ such that $\boldsymbol{x} = \lambda \boldsymbol{y} + (1 - \lambda)\boldsymbol{z}$.

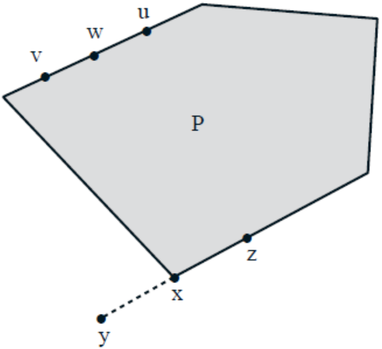


Figure from Bertsimas, Tsitsiklis, 1997

- $\boldsymbol{x}$ is an extreme point of P
- $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$, $\boldsymbol{y}$, $\boldsymbol{z}$ are not extreme points of P

Vertices

Definition (Vertex)

Let P be a polyhedron. A vector $\boldsymbol{x} \in P$ is a vertex of P if there exists some $\boldsymbol{c}$ such that $\boldsymbol{c}'\boldsymbol{x} < \boldsymbol{c}'\boldsymbol{y}$ for all $\boldsymbol{y}$ satisfying $\boldsymbol{y} \in P$ and $\boldsymbol{y} \neq \boldsymbol{x}$.

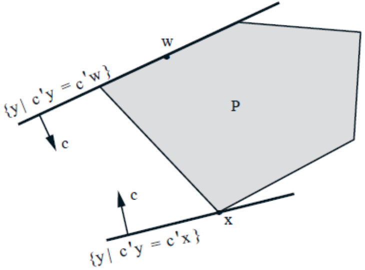


Figure from Bertsimas, Tsitsiklis, 1997

Figure: $\boldsymbol{x}$ is a vertex of P, $\boldsymbol{w}$ is not a vertex of P

Active Constraints

Consider a polyhedron $P \subseteq \mathbb{R}^n$ defined by linear equality and inequality constraints

$$
\begin{aligned}
\boldsymbol{a_i}'\boldsymbol{x} &\geq b_i, \quad i \in M_1, \\
\boldsymbol{a_i}'\boldsymbol{x} &\leq b_i, \quad i \in M_2, \\
\boldsymbol{a_i}'\boldsymbol{x} &= b_i, \quad i \in M_3,
\end{aligned}
$$

where M_1, M_2, and M_3 are finite index sets, $\boldsymbol{a_i} \in \mathbb{R}^n$, and $b_i \in \mathbb{R}$.

Definition

If a vector $\boldsymbol{x}^*$ satisfies $\boldsymbol{a_i}'\boldsymbol{x}^* = b_i$ for some $i \in M_1$, M_2, or M_3, we say that the corresponding constraint is active or binding at $\boldsymbol{x}^*$.

Basic Solutions

Definition

Consider a polyhedron P defined by linear equality and inequality constraints, and let $\boldsymbol{x}^*$ be an element of $\mathbb{R}^n$.

(a) Vector $\boldsymbol{x}^*$ is a basic solution if

   (i) all equality constraints are active and

   (ii) out of the constraints that are active at $\boldsymbol{x}^*$, there are n of them that are linearly independent.

(b) If $\boldsymbol{x}^*$ is a basic solution that satisfies all constraints, we say that it is a basic feasible solution (BFS).

Active Constraints and Basic Feasible Solutions

$$P = \{(x_1, x_2, x_3) \mid x_1 + x_2 + x_3 = 1,\ x_1, x_2, x_3 \geq 0\}$$

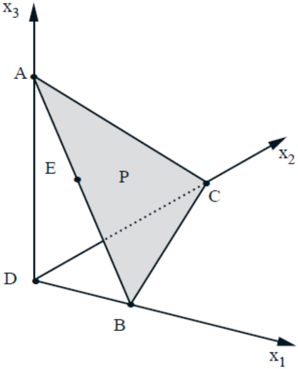


Figure from Bertsimas, Tsitsiklis, 1997

- 3 constraints active at A, B, C, and D
- 2 constraints active at E, i.e., $x_1 + x_2 + x_3 = 1$ and $x_2 = 0$
- A, B, and C are basic feasible solutions

Active Constraints and Basic Feasible Solutions

$$P = \{(x_1, x_2, x_3) \mid x_1 + x_2 + x_3 \leq 1, x_1 + x_2 + x_3 \geq 1,\ x_1, x_2, x_3 \geq 0\}$$

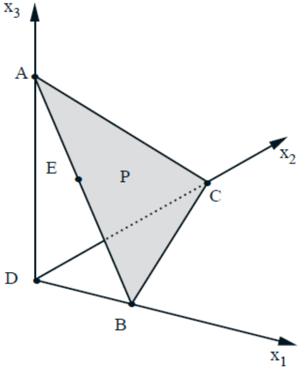


Figure from Bertsimas, Tsitsiklis, 1997

- D is now a basic solution

$\Rightarrow$ property is representation dependent!

- the basic feasible solutions stay the same: A, B, and C

Extreme Point, Vertex, Basic Feasible Solution

Theorem

Let P be a nonempty polyhedron and let $\boldsymbol{x}^* \in P$. Then the following are equivalent:

(a) $\boldsymbol{x}^*$ *is a vertex.*

(b) $\boldsymbol{x}^*$ *is an extreme point.*

(c) $\boldsymbol{x}^*$ *is a basic feasible solution.*

Corollary

Given a finite number of linear inequality constraints, there can only be a finite number of basic feasible solutions.

Existence of Extreme Points

Definition

A polyhedron $P \subseteq \mathbb{R}^n$ *contains a line* if there exists a vector $\boldsymbol{x} \in P$ and a nonzero vector $\boldsymbol{d} \in \mathbb{R}^n$ s.t. $\boldsymbol{x} + \lambda \boldsymbol{d} \in P$ for all scalars λ.

Theorem

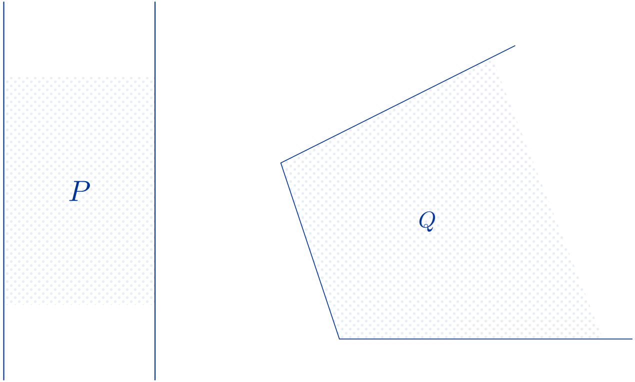
Suppose that the polyhedron $P = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{a}_i' \boldsymbol{x} \geq b_i,\ i = 1, \ldots, m\}$ *is nonempty. Then the following are equivalent:*

(a) P *has at least one extreme point.*

(b) P *does not contain a line.*

(c) *There exist* n *vectors out of* $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$ *which are linearly independent.*

Existence of Extreme Points



Polyhedron P contains a line, Q does not. Therefore, only Q has extreme points.

Existence of Extreme Points

- A bounded polyhedron does not contain a line.
- The positive orthant $\{\boldsymbol{x} \mid \boldsymbol{x} \geq \mathbf{0}\}$ does not contain a line.
- Since a polyhedron in standard form is contained in the positive orthant, it does not contain a line either.

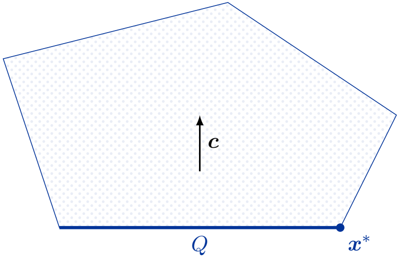
Corollary

Every nonempty bounded polyhedron and every nonempty polyhedron in standard form has at least one basic feasible solution.

Optimality of Extreme Points

Theorem

Consider the linear programming problem of minimizing $\mathbf{c}'\boldsymbol{x}$ over a polyhedron P. Suppose that P has at least one extreme point and that there exists an optimal solution. Then there exists an optimal solution which is an extreme point of P.



- $P = \{x \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$
- optimal cost: v
- $Q = \{x \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}, \mathbf{c}'\boldsymbol{x} = v\}$

Optimality of Extreme Points

Theorem (stronger variant)

Consider the linear programming problem of minimizing $\boldsymbol{c}'\boldsymbol{x}$ over a polyhedron P. Suppose that P has at least one extreme point. Then either the optimal cost is equal to $-\infty$, or there exists an extreme point which is optimal.

Corollary

Consider the linear programming problem of minimizing $\boldsymbol{c}'\boldsymbol{x}$ over a nonempty polyhedron. Then either the optimal cost is equal to $-\infty$ or there exists an optimal solution.

Summary

Conclusions regarding the solutions of LPs

(a) If the feasible set is nonempty and bounded, there exists an optimal solution. Furthermore, there exists an optimal solution which is an extreme point.

(b) If the feasible set is unbounded, there are three possibilities:
   (i) There exists an optimal solution which is an extreme point.
   (ii) There exists an optimal solution, but no optimal solution is an extreme point. (Only possible if the feasible set has no extreme points; This never happens in standard form problems)
   (iii) The optimal cost is $-\infty$.

(c) If the optimal cost is finite and the feasible set contains at least one extreme point, then there are only finitely many extreme points.

(d) The number of extreme points may increase exponentially with the number of variables and constraints.

Linear Programming

Algorithms

Overview

Well-known algorithms for the solution of LPs are:

- Simplex Algorithms,
- the Ellipsoid Method, and
- Interior Point Methods.

Simplex Algorithms

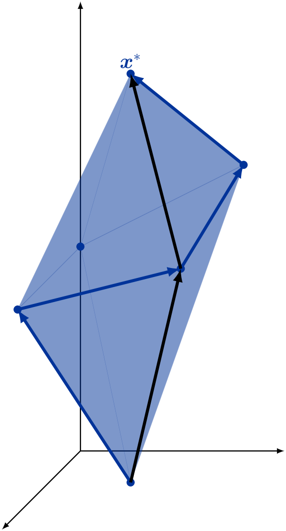
- Dantzig 1947
- for all known variants there exists an example with exponential time complexity (e.g., Klee-Minty cube)
- performs well in practice

Basic Idea (assuming minimization):

- start at a basic feasible solution
  - does not exist $\Rightarrow$ infeasible
- traverse line segments in cost reducing direction to the next (adjacent) basic feasible solution
  - if there exists a half-line extending towards $-\infty \Rightarrow$ unbounded
- once this is no longer possible, the current basic feasible solution is optimal

(Simplex algorithms traverse the surface of the polyhedron)

Simplex Algorithms



- several variants exist:
  - different rules for choosing direction
  - different rules to avoid cycling
  - ...

Ellipsoid Method

- Khachiyan 1979 (preliminary version: Shor 1970)
- first algorithm that showed that LPs can be solved in polynomial time
- no efficient implementation available $\Rightarrow$ not used in practice

Basic Idea:

- tests whether a polyhedron is empty
- this can be used to solve LPs since any linear programming problem can be reduced to a linear feasibility problem

Interior Point Methods

- Karmarkar 1984 (preliminary version: von Neumann 1947)
- polynomial runtime
- several variants are efficient in practice

⇒ valuable alternative for problems that are difficult for simplex algorithms

Variants:

- (Ellipsoid Method)
- affine scaling algorithm
- potential reduction algorithm
- path following algorithms (also known as barrier algorithms, very efficient in practice)

(Interior point methods traverse the interior of the polyhedron)

Linear Programming

Duality Theory

Motivation

Consider the primal problem

$$\begin{aligned} \min\ & \boldsymbol{c}'\boldsymbol{x} \\ \text{s.t. } & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

with an optimal solution $\boldsymbol{x}^*$ (assumed to exist).

Relaxed problem:

$$\begin{aligned} g(\boldsymbol{p}) = \min\ & \boldsymbol{c}'\boldsymbol{x} + \boldsymbol{p}'(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{aligned}$$

where $\boldsymbol{p}$ is a price vector (with dimension of $\boldsymbol{b}$) and $\boldsymbol{p}'(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})$ a penalty.

Motivation

Since the relaxed problem allows for more options, we should have $g(\boldsymbol{p}) \leq \boldsymbol{c}'\boldsymbol{x}^*$. Indeed,

$$g(\boldsymbol{p}) = \min_{\boldsymbol{x} \geq 0} \left[\boldsymbol{c}'\boldsymbol{x} + \boldsymbol{p}'(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})\right] \leq \boldsymbol{c}'\boldsymbol{x}^* + \boldsymbol{p}'(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^*) = \boldsymbol{c}'\boldsymbol{x}^*$$

since $\boldsymbol{x}^*$ is feasible and satisfies $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}$. Hence, each $\boldsymbol{p}$ yields a lower bound $g(\boldsymbol{p})$ for the optimal cost $\boldsymbol{c}'\boldsymbol{x}^*$.

Dual problem

"search for the tightest lower bound of this type"

$$\max \; g(\boldsymbol{p})$$

Motivation

By definition of $g(\boldsymbol{p})$ we have

$$g(\boldsymbol{p}) = \min_{\boldsymbol{x} \geq 0} \left[\boldsymbol{c}'\boldsymbol{x} + \boldsymbol{p}'(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) \right] = \boldsymbol{p}'\boldsymbol{b} + \min_{\boldsymbol{x} \geq 0} (\boldsymbol{c}' - \boldsymbol{p}'\boldsymbol{A})\boldsymbol{x}.$$

Note that

$$\min_{\boldsymbol{x} \geq 0} (\boldsymbol{c}' - \boldsymbol{p}'\boldsymbol{A})\boldsymbol{x} = \begin{cases} 0 & \text{if } \boldsymbol{c}' - \boldsymbol{p}'\boldsymbol{A} \geq 0 \\ -\infty & \text{otherwise.} \end{cases}$$

Thus, when maximizing $g(\boldsymbol{p})$, we only need to consider values of $\boldsymbol{p}$ for which $g(\boldsymbol{p})$ is not $-\infty$.

The Dual Problem

Dual (Linear Programming) Problem

The dual problem is the same as the linear programming problem

$$\begin{aligned} \max\ & \boldsymbol{p}'\boldsymbol{b} \\ \text{s.t. } & \boldsymbol{p}'\boldsymbol{A} \leq \boldsymbol{c}' \end{aligned}$$

Main result in duality theory: The optimal cost in the dual problem is equal to the optimal cost in the primal problem.

The Dual Problem

Construction of the dual of a minimization problem

- we have a vector of parameters (dual variables) $\boldsymbol{p}$
- for every $\boldsymbol{p}$ we have a method for obtaining a lower bound on the optimal primal cost
- dual problem is a maximization problem that looks for the tightest lower bound
- for some vectors $\boldsymbol{p}$ we get $-\infty$ (no useful information)

$\Rightarrow$ we only need to maximize over those $\boldsymbol{p}$ leading to nontrivial lower bounds

The Dual Problem

Let $\boldsymbol{A}$ be a matrix with rows $\boldsymbol{a}_i'$ and columns $\boldsymbol{A}_j$. Then the following defines a pair of primal and dual problems:

$$
\begin{aligned}
\min\ & \boldsymbol{c}'\boldsymbol{x} & & \\
\text{s.t. } & \boldsymbol{a}_i'\boldsymbol{x} \geq b_i, & & i \in M_1, \\
& \boldsymbol{a}_i'\boldsymbol{x} \leq b_i, & & i \in M_2, \\
& \boldsymbol{a}_i'\boldsymbol{x} = b_i, & & i \in M_3, \\
& x_j \geq 0, & & j \in N_1, \\
& x_j \leq 0, & & j \in N_2, \\
& x_j \text{ free}, & & j \in N_3,
\end{aligned}
\qquad
\begin{aligned}
\max\ & \boldsymbol{p}'\boldsymbol{b} & & \\
\text{s.t. } & p_i \geq 0, & & i \in M_1, \\
& p_i \leq 0, & & i \in M_2, \\
& p_i \text{ free}, & & i \in M_3, \\
& \boldsymbol{p}'\boldsymbol{A}_j \leq c_j, & & j \in N_1, \\
& \boldsymbol{p}'\boldsymbol{A}_j \geq c_j, & & j \in N_2, \\
& \boldsymbol{p}'\boldsymbol{A}_j = c_j, & & j \in N_3.
\end{aligned}
$$

Example

Primal

$$\begin{aligned} \min \quad & x_1 + 2x_2 + 3x_3 \\ \text{s.t.} \quad & -x_1 + 3x_2 = 5 \\ & 2x_1 - x_2 + 3x_3 \geq 6 \\ & x_3 \leq 4 \\ & x_1 \geq 0 \\ & x_2 \leq 0 \\ & x_3 \text{ free}, \end{aligned}$$

Dual

$$\begin{aligned} \max \quad & 5p_1 + 6p_2 + 4p_3 \\ \text{s.t.} \quad & p_1 \text{ free} \\ & p_2 \geq 0 \\ & p_3 \leq 0 \\ & -p_1 + 2p_2 \leq 1 \\ & 3p_1 - p_2 \geq 2 \\ & 3p_2 + p_3 = 3. \end{aligned}$$

Relations between primal and dual variables and constraints

PRIMAL	minimize	DUAL	maximize
constraints	$\geq b_i$	variables	≥ 0
	$\leq b_i$		≤ 0
	$= b_i$		free
variables	≥ 0	constraints	$\leq c_j$
	≤ 0		$\geq c_j$
	free		$= c_j$

Theorem

If we transform the dual into an equivalent minimization problem and then form its dual, we obtain a problem equivalent to the original problem.

Or more compact: "The dual of the dual is the primal".

Duality Theorems

Theorem (Weak duality)

If $\boldsymbol{x}$ is a feasible solution to the primal problem an $\boldsymbol{p}$ is a feasible solution to the dual problem, then

$$\boldsymbol{p}'\boldsymbol{b} \leq \boldsymbol{c}'\boldsymbol{x}.$$

Hence, the cost of any dual solution is a lower bound for the optimal cost.

Corollary

(a) *If the optimal cost in the primal is $-\infty$ (unbounded), then the dual problem must be infeasible.*

(b) *If the optimal cost in the dual is $+\infty$ (unbounded), then the primal problem must be infeasible.*

Duality Theorems

Corollary

Let $\boldsymbol{x}$ and $\boldsymbol{p}$ be feasible solutions to the primal and the dual, respectively, and suppose that $\boldsymbol{p}'\boldsymbol{b} = \boldsymbol{c}'\boldsymbol{x}$. Then $\boldsymbol{x}$ and $\boldsymbol{p}$ are optimal solutions to the primal and the dual, respectively.

Theorem (Strong duality)

If a linear programming problem has an optimal solution, so does its dual, and the respective optimal costs are equal.

Theorem (Complementary Slackness)

Let $\boldsymbol{x}$ and $\boldsymbol{p}$ be feasible solutions to the primal and the dual problem, respectively. They are optimal if and only if:

$$p_i(\boldsymbol{a}_i'\boldsymbol{x} - b_i) = 0 \quad \forall i \text{ and}$$
$$(c_j - \boldsymbol{p}'\boldsymbol{A}_j)x_j = 0 \quad \forall j.$$

Summary

- for each LP, we can define a dual LP
- the dual of the dual is the primal
- strong duality theorem!
- allows to develop the dual simplex algorithm (important for warm start when solving LPs)
- (more advanced) duality theory provides powerful tools for further insights and understanding of LPs and polyhedra