# COMP331/COMP557: Optimisation

Martin Gairing

Computer Science Department

University of Liverpool

1st Semester 2018/19

Material adapted from a course by Martin Skutella at TU Berlin

# My Background

## FH Esslingen

- 1995-2000: Diplom (Electrical Engineering)

## Clemson University

- 2000-2001: MSc (Computer Science)

## University of Paderborn

- 2002-2007: PhD + Postdoc

## International Computer Science Institute Berkeley

- 2007- 2009: Postdoc

## Liverpool University

- Since 2009: Lecturer/Senior Lecturer

# Administrative Details

Lectures:

- Mondays, 11:00 - 12:00
- Tuesdays, 10:00 - 11:00
- Thursdays, 12:00 -13:00

Tutorials:

- Flávia Alves (F.Alves@liverpool.ac.uk)
- starting from Friday 28 September

Assessment:

- 25 % continuous assessment
- 75 % final exam

# The webpage for this module

- https://cgi.csc.liv.ac.uk/~gairing/COMP557/
- lecture notes
- resources
- announcements

# Course Aims

- To provide a foundation for modelling various continuous and discrete optimisation problems.
- To provide the tools and paradigms for the design and analysis of algorithms for continuous and discrete optimisation problems. Apply these tools to real-world problems.
- To review the links and interconnections between optimisation and computational complexity theory.
- To provide an in-depth, systematic and critical understanding of selected significant topics at the intersection of optimisation, algorithms and (to a lesser extent) complexity theory, together with the related research issues.

# Learning Outcomes

Upon completion of the module you should have:

- A critical awareness of current problems and research issues in the field of optimisation.
- The ability to formulate optimisation models for the purpose of modelling particular applications.
- The ability to use appropriate algorithmic paradigms and techniques in context of a particular optimisation model.
- The ability to read, understand and communicate research literature in the field of optimisation.
- The ability to recognise potential research opportunities and research directions.

# Outline


1. Introduction
2. Linear Programming Basics
3. The Geometry of Linear Programming
4. The Simplex Method
5. Duality
6. Applications of Linear Programming

# Chapter 1: Introduction

# A Motivating (and Refreshing) Example

Small brewery produces ale and beer.

- Production limited by scarce resources: corn, hops, barley malt.
- Recipes for ale and beer require different proportions of resources.

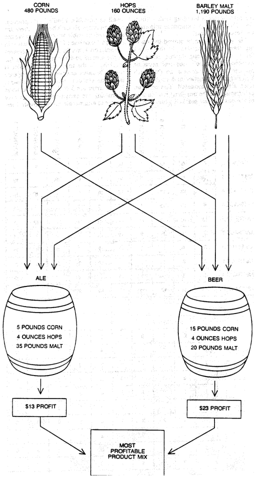
| Beverage | Corn (lb) | Hops (oz) | Malt (lb) | Profit (£) |
|---|---|---|---|---|
| Ale (barrel) | 5 | 4 | 35 | 13 |
| Beer (barrel) | 15 | 4 | 20 | 23 |
| Quantity | 480 | 160 | 1190 | |

- Devote all resources to ale: 34 barrels of ale $\Longrightarrow$ £442
- Devote all resources to beer: 32 barrels of beer $\Longrightarrow$ £736
- 7.5 barrels of ale, 29.5 barrels of beer $\Longrightarrow$ £776
- 12 barrels of ale, 28 barrels of beer $\Longrightarrow$ £800

Is this best possible?

# A Motivating (and Refreshing) Example

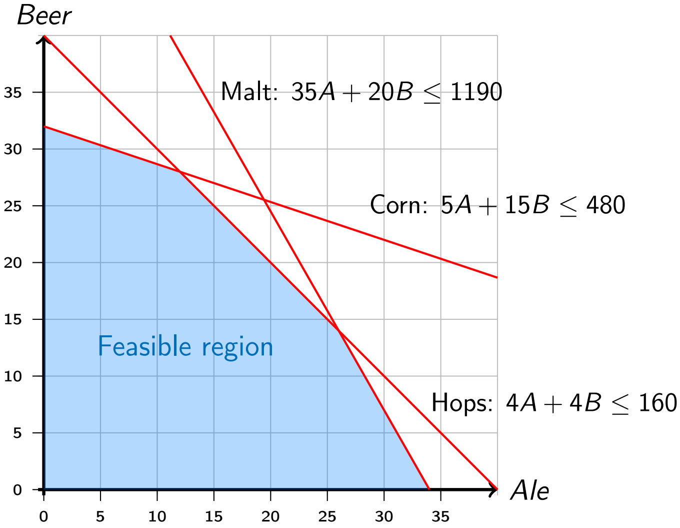
| Beverage | Corn (lb) | Hops (oz) | Malt (lb) | Profit (£) |
|---|---|---|---|---|
| Ale (barrel) | 5 | 4 | 35 | 13 |
| Beer (barrel) | 15 | 4 | 20 | 23 |
| Quantity | 480 | 160 | 1190 | |



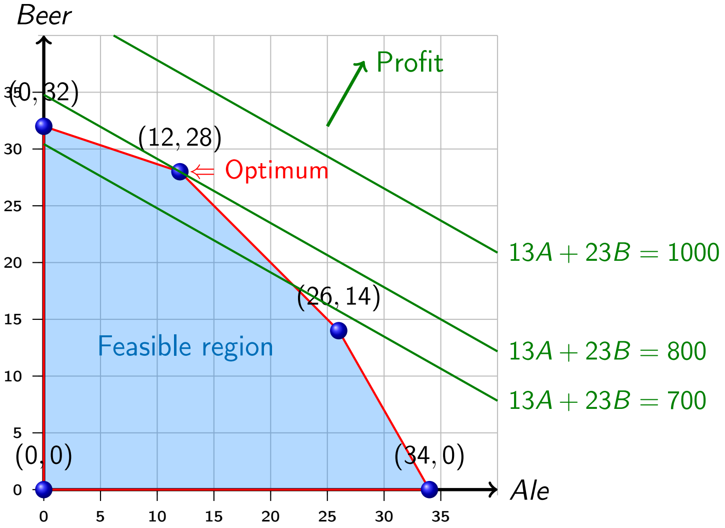

- Mathematical Formulation:

$$
\begin{array}{llcll}
\max & 13A & + & 23B & \text{Profit} \\
\text{s.t.} & 5A & + & 15B \leq 480 & \text{Corn} \\
 & 4A & + & 4B \leq 160 & \text{Hops} \\
 & 35A & + & 20B \leq 1190 & \text{Malt} \\
 & & & A, B \geq 0 &
\end{array}
$$

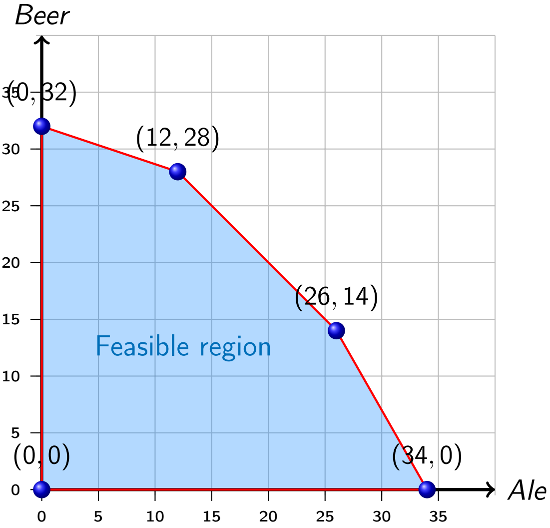
# A Motivating (and Refreshing) Example



# A Motivating (and Refreshing) Example



# A Motivating (and Refreshing) Example




Observation: Regardless of objective function coefficients, an optimal solution occurs at an extreme point (vertex).

# Terminology and Notation

Numbers:

- $\mathbb{R}$ ... set of real numbers
- $\mathbb{R}_{\geq 0}$ or $\mathbb{R}_{+}$ ... set of non-negative real numbers
- $\mathbb{R}^n$ ... n-dimensional real vector space
- $\mathbb{Z}$, $\mathbb{Z}_{\geq 0}$, $\mathbb{Z}^n$ ... set of integers, non-negative integers, n-dimensional ...

Sets:

- $S = \{s_1, s_2, \cdots, s_k\}$ ... a set of $k$ elements
- $S = \{x \mid P(x)\}$ ... set of elements $x$ for which condition $P$ is true
  - Example: $\mathbb{Z}_{\geq 0} = \{i \mid i \in \mathbb{Z} \text{ and } i \geq 0\}$
- $|S|$ ... size (number of elements) of a finite set $S$
- $2^S$ ... set of all subsets of $S$
  - e.g.: $2^{\{a,b,c\}} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}$
- $\mu : S \mapsto T$ ... $\mu$ is a mapping (or function) from set $S$ to set $T$

# Terminology and Notation – Linear Algebra

- matrix of dimension $m \times n$:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} = \begin{pmatrix} | & | & & | \\ A_1 & A_2 & \dots & A_n \\ | & | & & | \end{pmatrix} = \begin{pmatrix} — & a_1^T & — \\ & \vdots & \\ — & a_m^T & — \end{pmatrix}$$

- and its transpose: $A^T = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{pmatrix}$
- Column vector $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ ; row vector $x^T$ (the transpose of $x$)
- Inner product of $x, y \in \mathbb{R}^n$: $\quad x^T y = y^T x = \sum_{i=1}^n x_i y_i$
- Matrix equation $Ax = b$
  is equivalent to $a_i^T x = b_i$ for all $i \in \{1, \dots, m\}$
  ($b$ is an m-vector, $b_i$ is its $i$'th component)

# Terminology and Notation – Linear Algebra

- $\det(A)$ ... determinant of a matrix
  - e.g.: $\det\begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}$
- $e_i$ ... unit vector (dimension from context)
  - 1 in $i$'th component, 0 else
  - e.g. (dimension 3): $e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ $e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ $e_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
- $I = \begin{pmatrix} | & | & & | \\ e_1 & e_2 & \cdots & e_n \\ | & | & & | \end{pmatrix}$ ... identity matrix (dimension from context, here $n$)
  - 1 on main diagonal, 0 else
  - e.g. (dimension 3): $I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- $\text{rank}(A)$ = size of the largest set of linearly independent columns
  = size of the largest set of linearly independent rows
- $A^{-1}$ ... matrix inverse of square matrix $A$
  - $A^{-1}A = A\,A^{-1} = I$
  - $A^{-1}$ exists if and only if $\det(A) \neq 0$

# 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!

# Convex Optimization Problems

## Definition 1.1.

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

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

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

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

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

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

# Local and Global Optimality

**Definition 1.2.**

Let $X \subseteq \mathbb{R}^n$ and $f : X \mapsto \mathbb{R}$.
$x' \in X$ 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 1.3.**

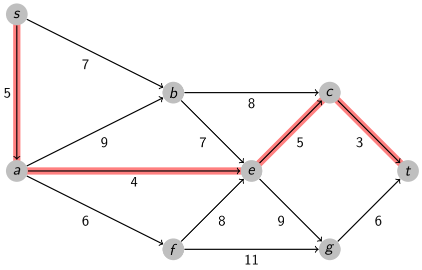
For a convex optimization problem, every local optimum is a (global) optimum.

# 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.

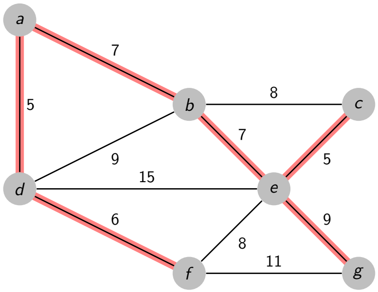
That is, $X = \{P \subseteq A \mid P \text{ is } s\text{-}t\text{-path in } D\}$ and $f : X \to \mathbb{R}$ is given by

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

**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

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

## 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,

$$0 \leq x(a) \leq u(a) \qquad \text{for all } a \in A,$$

$$\sum_{a \in \delta^+(v)} x(a) - \sum_{a \in \delta^-(v)} x(a) = b(v) \qquad \text{for all } v \in V,$$

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):

$$
\begin{aligned}
\text{minimize} \quad & \sum_{a \in A} c(a) \cdot x(a) && && (1.1)\\
\text{subject to} \quad & \sum_{a \in \delta^+(v)} x(a) - \sum_{a \in \delta^-(v)} x(a) = b(v) && \text{for all } v \in V, && (1.2)\\
& x(a) \leq u(a) && \text{for all } a \in A, && (1.3)\\
& x(a) \geq 0 && \text{for all } a \in A. && (1.4)
\end{aligned}
$$

- Objective function given by (1.1). Set of feasible solutions:

$$X = \{x \in \mathbb{R}^A \mid x \text{ satisfies (1.2), (1.3), and (1.4)}\} \ .$$

- Notice that (1.1) is a linear function of $x$ and (1.2) – (1.4) 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):

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

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 $e \in M$.

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

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

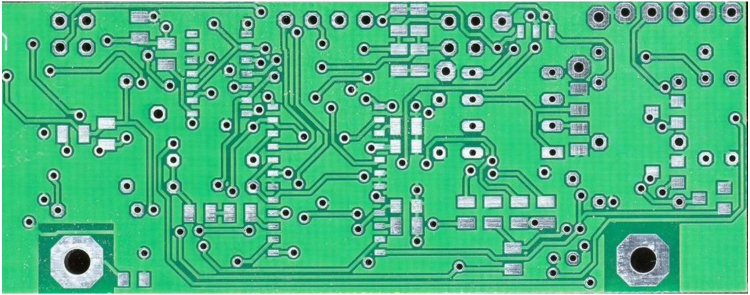
# Example: Traveling Salesperson Problem (TSP)

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.)

Application: Drilling holes in printed circuit boards.



Formulation as an integer linear program? (maybe later!)

# Example: Weighted Vertex Cover Problem

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

Task: find $U \subseteq V$ of minimum total weight such that every edge $e \in E$ has at least one endpoint in $U$.

Formulation as an integer linear program (IP):

Variables: $x_v \in \{0, 1\}$ for $v \in V$ with $x_v = 1$ if and only if $v \in U$.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{v \in V} w(v) \cdot x_v \\
\text{subject to} \quad & x_v + x_{v'} \geq 1 && \text{for all } e = \{v, v'\} \in E, \\
& x_v \in \{0, 1\} && \text{for all } v \in V.
\end{aligned}
$$

# Markowitz' Portfolio Optimisation Problem

Given: $n$ different securities (stocks, bonds, etc.) with random returns, target return $R$, for each security $i \in [n]$:

- expected return $\mu_i$, variance $\sigma_i$

For each pair of securities $i, j$:

- covariance $\rho_{ij}$,

Task: Find a portfolio $x_1, \ldots, x_n$ that minimises *"risk"* (aka *variance*) and has expected return $\geq R$.

Formulation as a quadratic programme (QP):

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j} \rho_{ij}\sigma_i\sigma_j x_i x_j \\
\text{subject to} \quad & \sum_i x_i = 1 \\
& \sum_i \mu_i x_i \geq R \\
& x_i \geq 0, \qquad \text{for all } i.
\end{aligned}
$$

# 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?

# 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.

# Literature on Combinatorial Optimization (not complete)


- R. K. Ahuja, T. L. Magnanti, J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, 1993.
- W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver, *Combinatorial Optimization*, Wiley, 1998.
- L. R. Ford, D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.
- M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, 1979.
- B. Korte, J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, Springer, 2002.
- C. H. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, reprint 1998.
- A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer, 2003.