DM545/DM871
Linear and Integer Programming

Course Introduction

Marco Chiarandini

Department of Mathematics and Computer Science

September 1, 2026

Outline

Outline

Who are we?

47 in total registered in ItsLearning

DM545 (5 ECTS)
33 enrolled

- Math-economics
  (Bachelor, 5th semester)
  Section M1
  6 enrolled
- Others?
  Section H1
  27 enrolled

DM871 (5 ECTS)
13 enrolled

- Section H1, 13 enrolled
  - Computer Science (Master)
  - Mathematics (Bachelor/Master)
  - Applied Mathematics (Bachelor/Master)
  - Data Science (Master)
  - Others?

Aims of the course

Learn about **mathematical optimization**:

- linear programming (continuous/numerical linear optimization)
- integer linear programming (discrete/combinatorial linear optimization)

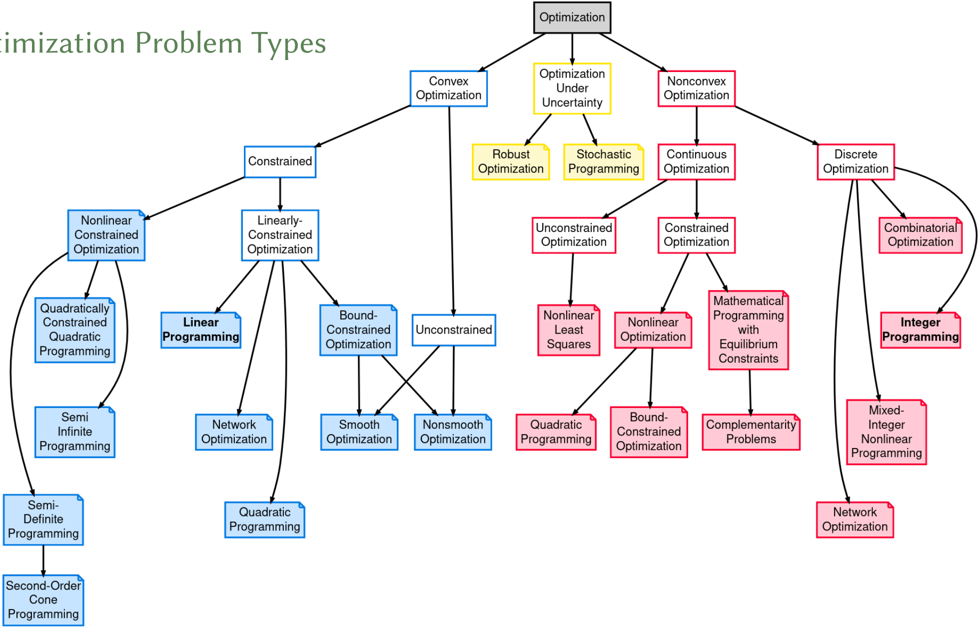
Hence, an alternative course title could be: **Linear Optimization**

Optimization is an important tool in **decision making** and in analyzing physical systems.

In mathematical terms, an **optimization problem** is the problem of finding the **best solution** from the set of all **feasible solutions**.

The first step in the optimization process is constructing an appropriate **mathematical model**

Optimization Problem Types



(NEOS Server, University of Wisconsin)

Contents of the Course (aka Syllabus)

Linear Programming

1 Introduction - Linear Programming, Notation & Modeling
2 Linear Programming, Simplex Method
3 Exception Handling
4 Duality Theory
5 Sensitivity
6 Revised Simplex Method

Integer Linear Programming

7 Modeling Examples, Good Formulations, Relaxations
8 Well Solved Problems
9 Cutting Planes & Branch and Bound
10 Network Optimization Models (Max Flow, Min cost flow, Matching)
11 More on Modeling

Practical Information

Teacher: Marco Chiarandini (imada.sdu.dk/u/marco/)
Instructor: (Marco Chiarandini ?)

Schedule, alternative views:

- mitsdu.sdu.dk, SDU Mobile
- Official course description (læserplanen)
- ItsLearning

Schedule (15 weeks, up to week 51):

- Introductory classes: 28 hours (14 classes), scheduled 32
- Training classes: 14 hours (7 classes), scheduled 28

Communication Means

- Announcements in ItsLearning
- Ask peers and teacher in class
- $\Leftrightarrow$ External Web Page
  (link `https://dm871.github.io`)
- Write to Marco (`marco@imada.sdu.dk`) for confidential and individual issues
- Feedback from classes
- Midway evaluation

$\rightsquigarrow$ **It is good to ask questions!!**

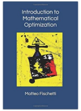
$\rightsquigarrow$ **Let me know if you think we should do things differently!**

Sources — Reading Material

Main references:

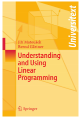
[LN] Lecture Notes (continously updated)

[F] M. Fischetti, **Introduction to Mathematical Optimization**, Independently published, 2019



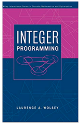
Linear Programming:

[VA] R. Vanderbei. Linear Programming: Foundations and Extensions. Springer US, 2008



Integer Programming:

[Wo] L.A. Wolsey. Integer programming. John Wiley & Sons, New York, USA, 1998



Others

[HL] Frederick S Hillier and Gerald J Lieberman, Introduction to Operations Research, 9th edition, 2010

Course Material

External Web Page is the main reference for list of contents

It contains:

- List of units with topics and references
- Slides
- Exercises and some solutions
- Links
- Tutorials for programming tasks

Assessment

- One **3 h Written Exam**
  January 11, 2027
  all helping tools allowed, but not GenAI
  - style: short open answers about calculations and modeling
  - (language: Danish and/or English)
- (only DM871) Home implementation assignment in the second half of the course

Graded with internal censorship

Training Sessions

- Do the plus exercises in advance
- Prepare the starred exercises in advance to get out the most from the session
- Participate actively in class
- Try the others, unsolved in class, after the session
- Best if carried out in small groups
- (Starred) Exercises are examples of exam questions

Exercise classes are about material presented the same week.

Prerequisites: Linear Algebra

Linear Algebra:
manipulation of matrices and vectors with some theoretical background

Linear Algebra

- Matrices and vectors - Matrix algebra
- Dot (scalar, Euclidean inner) product
- Geometric insights
- Systems of Linear Equations - Row echelon form, Gaussian elimination
- Matrix inversion and determinants
- Rank and linear dependency

Unit 0: sheet0.pdf

Prerequisites: Coding

- gives you the ability to create new and useful artifacts
- allows you to have more control of your world as more and more of it becomes digital
- is just fun.

It can also help you to **understand math**.

- listening to lectures
- watching an instructor work through a derivation
- working through numerical examples by hand
- learn **by doing**, **interacting with Python**.
- Python 3.14+, numpy
- ipython, jupyter, jupyterLab (= interactive python) or **Visual Code** or Spyder3.
- Commercial software: Gurobi, Cplex, Xpress $\approx$ 100 000 Dkk
  Open source: Highs, SCIP
- Python APIs: gurobipy, Pyomo, Python-MIP, PySCIPOpt (Python interfaces to SCIP Optimization Suite)

Unit 0: `sheet0_programming.pdf`

A Note on Generative AI

- Read SDU rules on the topic https://mitsdu.dk/en/aiatsdu. They apply here.
  In short: give credit when using
- Positive aspects:
  better explanations, examples, immediate feedback
- Negative aspects:
  economical costs, impact on environment, sovereignty risk, the social aspect, personal skills
- First learn, then use, and always check
  Comapnies recommend to learn to use but want experts who can verify the outcomes

Use it responsibly, to enhance your learning!
Be aware that the exam form requires you to have achieved the learning goals not GenAI.

Outline

Sets

- A **set** is a collection of objects. eg.: $A = \{1, 2, 3\}$
- $A = \{n \mid n \text{ is a whole number and } 1 \leq n \leq 3\}$
  ('$|$' reads 'such that')
- $B = \{x \mid x \text{ is a student of this course}\}$
- $x \in A$
  x belongs to A
- set of no members: **empty set**, denoted $\emptyset$
- if a set S is a **(proper) subset** of a set T, we write ($S \subset T$) $T \subseteq S$
  $\{1, 2, 5\} \subset \{1, 2, 4, 5, 6, 30\}$
- for two sets A and B, the **union** $A \cup B$ is $\{x \mid x \in A \text{ or } x \in B\}$
- for two sets A and B, the **intersection** $A \cap B$ is $\{x \mid x \in A \text{ and } x \in B\}$
  $A = \{1, 2, 3, 5\}$ and $B = \{2, 4, 5, 7\}$, then $A \cap B = \{2, 5\}$

Numbers

- set of real numbers: $\mathbb{R}$
- set of natural numbers: $\mathbb{N} = \{1, 2, 3, 4, \ldots\}$ (positive integers); $\mathbb{N}_0$ to include zero
- set of all integers: $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$; $\mathbb{Z}_0^+$ only positives and zero
- set of rational numbers: $\mathbb{Q} = \{p/q \mid p, q \in \mathbb{Z}, q \neq 0\}$
- set of complex numbers: $\mathbb{C}$
- absolute value (non-negative):

$$|a| = \begin{cases} a & \text{if } a \geq 0 \\ -a & \text{if } a \leq 0 \end{cases}$$

- the set $\mathbb{R}^2$ is the set of ordered pairs (x, y) of real numbers (eg, coordinates of a point wrt a pair of axes, the **Cartesian plane**)

Matrices and Vectors

- A **matrix** is a rectangular array of numbers or symbols. It can be written as

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- An $n \times 1$ matrix is a **column vector**, or simply a vector:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

- the set $\mathbb{R}^n$ is the set of vectors $[x_1, x_2, \ldots, x_n]^T$ of real numbers (eg, coordinates of a point wrt an n-dimensional space, the **Euclidean Space**)

Graphs

- An **undirected graph G=(V,E)** is a structure made of a pair of sets: V the set of **vertices** and $E \subseteq \{e = \{u, v\} \mid u, v \in V\}$ the set of **edges** (undirected links)
- A **directed graph (digraph)** $D = (V, A)$ is a pair of sets: V the set of **vertices** and $A \subseteq \{uv = (u, v) \mid u, v \in V\}$ the set of **arcs** (directed links)
- Graph drawing (2D embeddings)
- Labelled graphs and weighted graphs
- Walks, Paths, Cycles, Connectedness
- Problems on graphs: eg, determining intrinsic properties such as shortest st-path
- Matrix representations: **adjacency matrix**, **incidence matrix**

- adjacency set (neighborhood set) for $v \in V$: $N(v) = \{u \mid u \in V, uv \in E\}$; vertex degree (number of incident edges), $\delta(v)$; outdegree (number of outgoing edges), $\delta^+(v)$; indegree (number of incoming edges), $\delta^-(v)$;
- subgraph and **induced subgraph** $G' = G[V] \subseteq G$
- cut: set of edges that separates a connected graphs into two connected components $G[X]$, $G[Y]$: $C = \{xy \mid x \in X, y \in Y\}$
- bipartite graphs $G = (U, V, E)$, trees ($=$ connected acycle graphs)
- multi-graphs, hypergraphs, mixed graphs
- Graphs are the basic subject studied by graph theory. The word “graph” was first used in this sense by J.J. Sylvester in 1878 due to a direct relation between mathematics and chemical structure.

Basic Algebra

Elementary Algebra: the study of symbols and the rules for manipulating symbols. It differs from **arithmetic** in the use of abstractions, such as using letters to stand for numbers that are either unknown or allowed to take on many values

- collecting up terms: eg. $2a + 3b - a + 5b = a + 8b$
- multiplication of variables: eg:

$$a(-b) - 3ab + (-2a)(-4b) = -ab - 3ab + 8ab = 4ab$$

- expansion of bracketed terms: eg:

$$\begin{aligned} -(a-2b) &= -a + 2b, \\ (2x-3y)(x+4y) &= 2x^2 - 3xy + 8xy - 12y^2 \\ &= 2x^2 + 5xy - 12y^2 \end{aligned}$$

- $a^r a^s = a^{r+s}, \quad (a^r)^s = a^{rs}, \quad a^{-n} = 1/a^n,$
  $a^{m/n} = (a^{1/n})^m, \quad a^{1/n} = x \iff x^n = a, \quad a^x = n \iff x = \log_a n$

Variables

- In Mathematics and Statistics, a **variable** is an alphabetic character representing a **value**, which is unknown. They are used in **symbolic** calculations.
  Commonly given one-character names.
- in contrast, a **parameter** or **constant** or **given** is a known real number
- in contrast, in **Computer Science**, a **variable** is a storage location paired with an associated identifier, which contains a value that may be known or unknown.
  Commonly given long, explanatory names.

Functions

- a **function** f on a set $\mathcal{X}$ into a set $\mathcal{Y}$ is a rule that assigns a **unique** element $f(x)$ in $\mathcal{Y}$ to each element x in $\mathcal{X}$.

$$y = f(x)$$

y dependent variable

x independent variable

- scalar (element of a field, eg, $\mathbb{R}$, vector of a single component) and vectors

- a **linear function** is a polynomial of degree one or less.
If only one variable:

$$f(x) := ax + b$$

where a and b are constants, often real numbers.
The graph is a nonvertical line (a is the slope and b the intercept).
For any finite number of variables: $f(x_1, \ldots, x_k) = b + a_1x_1 + \ldots + a_kx_k$ and the graph is a hyperplane of dimension k. (actually these are affine functions: linear + translation)

Outline

What is Operations Research?

Operations Research (aka, Management Science, Analytics):
is the discipline that uses a **scientific approach to decision making**.

It seeks to determine how best to design and operate a system,
usually under conditions requiring the allocation of scarce resources,
by means of **mathematics** and **computer science**.

Quantitative methods for planning and analysis.

It encompasses a wide range of problem-solving techniques and methods applied in the pursuit of improved decision-making and efficiency:

- simulation,
- **mathematical optimization**,
- queueing theory and other stochastic-process models,
- Markov decision processes
- econometric methods,
- data envelopment analysis,
- machine learning,
- expert systems

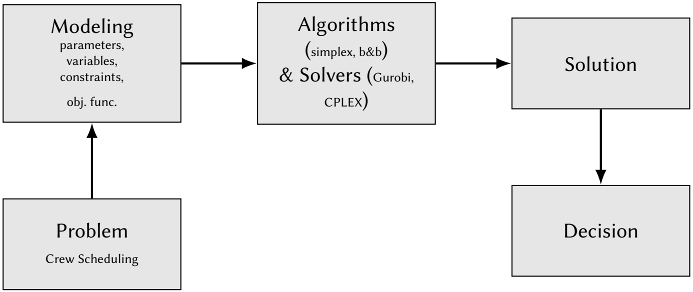
Some Examples ...

- Production Planning and Inventory Control
- Budget Investment
- Blending and Refining
- Manpower Planning
  - Crew Rostering (airline crew, rail crew, nurses)
- Packing Problems
  - Knapsack Problem
- Cutting Problems
  - Cutting Stock Problem
- Routing
  - Vehicle Routing Problem (trucks, planes, trains ...)
- Locational Decisions
  - Facility Location
- Scheduling/Timetabling
  - Examination timetabling/ train timetabling
- .... + many more

Common Characteristics

- Planning decisions must be made
- The problems relate to quantitative issues
  - Cheapest
  - Shortest route
  - Fewest number of people
- Not all plans are feasible - there are constraining rules
  - Limited amount of available resources
- It can be extremely difficult to figure out what to do

OR - The Process?



1. Observe the System
2. Formulate the Problem
3. Formulate Mathematical Model
4. Verify Model
5. Select Alternative
6. Show Results to Company
7. Implementation

Central Idea

Build a mathematical model describing exactly what one wants, and what the “rules of the game” are. However, **what is a mathematical model and how?**

Mathematical Modeling

- Find out exactly what the decision maker needs to know:
  - which investment?
  - which product mix?
  - which job j should a person i do?
- Define **Parameters**, that is the given, known elements of the problem.
- Define **Decision Variables** of suitable type (continuous, integer, binary) corresponding to the needs
- Formulate **Objective Function** computing the benefit/cost
- Formulate mathematical **Constraints** indicating the interplay between the different variables.

Outline

Resource Allocation

In manufacturing industry, **factory planning**: find the best product mix.

Example

A factory makes two products **standard** and **deluxe**. Eg, yogurt, sleeping beds, etc.

A unit of standard gives a profit of 6(k) Dkk.
A unit of deluxe gives a profit of 8(k) Dkk.

The warming|grinding and cooling|polishing times in terms of hours per week for a **unit** of each type of product are given below:

	Standard	Deluxe
(Machine 1) Warming|Grinding	5	10
(Machine 2) Cooling|Polishing	4	4

Warming|Grinding capacity: 60 hours per week
Cooling|Polishing capacity: 40 hours per week

Q: How much of each product, **standard** and **deluxe**, should we produce to maximize the profit?

Mathematical Model

Decision Variables

$x_1 \geq 0$ units of product standard
$x_2 \geq 0$ units of product deluxe

Object Function

$\max\ 6x_1 + 8x_2$ maximize profit

Constraints

$$\begin{array}{rcrcl} 5x_1 & + & 10x_2 & \leq & 60 \quad \text{machine 1 capacity} \\ 4x_1 & + & 4x_2 & \leq & 40 \quad \text{machine 2 capacity} \end{array}$$

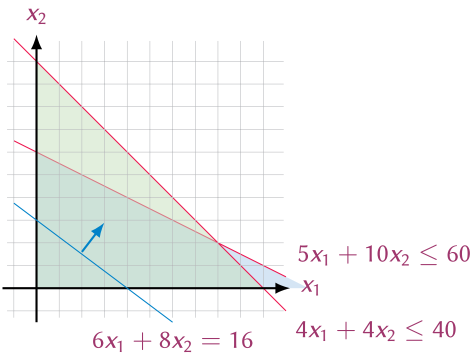
Mathematical Model

Machines/Materials A and B
Products 1 and 2

$$\begin{array}{lrcrcr} \max & 6x_1 & + & 8x_2 & & \\ & 5x_1 & + & 10x_2 & \leq & 60 \\ & 4x_1 & + & 4x_2 & \leq & 40 \\ & & & x_1 & \geq & 0 \\ & & & x_2 & \geq & 0 \end{array}$$

a_{ij}	1	2	b_i
A	5	10	60
B	4	4	40
c_j	6	8	

Graphical Representation:



Resource Allocation - General Model

Managing a production facility

$j = 1, 2, \ldots, n$	products
$i = 1, 2, \ldots, m$	materials
b_i	units of raw material at disposal
a_{ij}	units of raw material i to produce one unit of product j
σ_j	market price of unit of jth product
ρ_i	prevailing market value for material i
$c_j = \sigma_j - \sum_{i=1}^{m} \rho_i a_{ij}$	profit per unit of product j
x_j	amount of product j to produce

$$
\begin{array}{rrcrcrcccrcl}
\max & c_1 x_1 & + & c_2 x_2 & + & c_3 x_3 & + & \ldots & + & c_n x_n & = & z \\
\text{subject to} & a_{11} x_1 & + & a_{12} x_2 & + & a_{13} x_3 & + & \ldots & + & a_{1n} x_n & \leq & b_1 \\
& a_{21} x_1 & + & a_{22} x_2 & + & a_{23} x_3 & + & \ldots & + & a_{2n} x_n & \leq & b_2 \\
& \ldots & & & & & & & & & & \\
& a_{m1} x_1 & + & a_{m2} x_2 & + & a_{m3} x_3 & + & \ldots & + & a_{mn} x_n & \leq & b_m \\
& & & & & & & & & x_1, x_2, \ldots, x_n & \geq & 0
\end{array}
$$

Notation

$$
\begin{array}{rrcrcrcccrcl}
\max & c_1 x_1 & + & c_2 x_2 & + & c_3 x_3 & + & \ldots & + & c_n x_n & = & z \\
\text{s.t.} & a_{11} x_1 & + & a_{12} x_2 & + & a_{13} x_3 & + & \ldots & + & a_{1n} x_n & \leq & b_1 \\
& a_{21} x_1 & + & a_{22} x_2 & + & a_{23} x_3 & + & \ldots & + & a_{2n} x_n & \leq & b_2 \\
& \ldots & & & & & & & & & & \\
& a_{m1} x_1 & + & a_{m2} x_2 & + & a_{m3} x_3 & + & \ldots & + & a_{mn} x_n & \leq & b_m \\
& & & & & & & & & x_1, x_2, \ldots, x_n & \geq & 0
\end{array}
$$

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

In Matrix Form

$$
\begin{array}{rcccccccccccc}
\max & c_1x_1 & + & c_2x_2 & + & c_3x_3 & + & \ldots & + & c_nx_n & = & z \\
\text{s.t.} & a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + & \ldots & + & a_{1n}x_n & \leq & b_1 \\
 & a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & + & \ldots & + & a_{2n}x_n & \leq & b_2 \\
 & \ldots & & & & & & & & & & \\
 & a_{m1}x_1 & + & a_{m2}x_2 & + & a_{m3}x_3 & + & \ldots & + & a_{mn}x_n & \leq & b_m \\
 & & & & & & & & & x_1, x_2, \ldots, x_n & \geq & 0
\end{array}
$$

$$
\boldsymbol{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}
$$

$$
\begin{array}{rrcl}
\max & z & = & \boldsymbol{c}^T\boldsymbol{x} \\
 & A\boldsymbol{x} & \leq & \boldsymbol{b} \\
 & \boldsymbol{x} & \geq & 0
\end{array}
$$

Our Numerical Example

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

$$
\begin{array}{rrcrcr}
\max & 6x_1 & + & 8x_2 & & \\
& 5x_1 & + & 10x_2 & \leq & 60 \\
& 4x_1 & + & 4x_2 & \leq & 40 \\
& & & x_1, x_2 & \geq & 0
\end{array}
$$

$$
\begin{aligned}
\max \quad & \boldsymbol{c}^T \boldsymbol{x} \\
A\boldsymbol{x} & \leq \boldsymbol{b} \\
\boldsymbol{x} & \geq 0
\end{aligned}
$$

$$\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{c} \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m$$

$$\max \quad \begin{bmatrix} 6 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 10 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 60 \\ 40 \end{bmatrix}$$

$$x_1, x_2 \geq 0$$

Outline

1. Course Organization

2. Preliminaries

3. Introduction: Operations Research
    - Resource Allocation
    - Duality

Duality

Resource Valuation problem: Determine the value of the raw materials in hand such that: (i) it would be convenient selling and (ii) an outside company would be willing to buy them.

z_i	value of a unit of raw material i
$\sum_{i=1}^{m} b_i z_i$	total expenses for buying (or opportunity cost, cost of having instead of selling)
ρ_i	prevailing unit market value of material i
σ_j	prevailing unit product price

Goal: for the outside company to minimize the total expenses;
(for the owing company the minimum amount of opportunity cost to accept for selling)

$$\min \sum_{i=1}^{m} b_i z_i \tag{1}$$

$$\sum_{i=1}^{m} z_i a_{ij} \geq \sigma_j, \quad j = 1 \ldots n \tag{2}$$

$$z_i \geq \rho_i, \quad i = 1 \ldots m \tag{3}$$

(2) otherwise not selling and (3) otherwise selling to someone else

Let

$$y_i = z_i - \rho_i$$

markup that the company would make by selling the raw material instead of producing.

$$\begin{aligned} \min \sum_{i=1}^{m} y_i b_i + \cancel{\sum_i \rho_i b_i} \\ \sum_{i=1}^{m} y_i a_{ij} \geq c_j, \quad j = 1 \dots n \\ y_i \geq 0, \quad i = 1 \dots m \end{aligned}$$

Dual Problem

$$\begin{aligned} \max \sum_{j=1}^{n} c_j x_j \\ \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \dots, m \\ x_j \geq 0, \quad j = 1, \dots, n \end{aligned}$$

Primal Problem

Notions of Computer Science

Algorithm: a finite, well-defined sequence of operations to perform a calculation

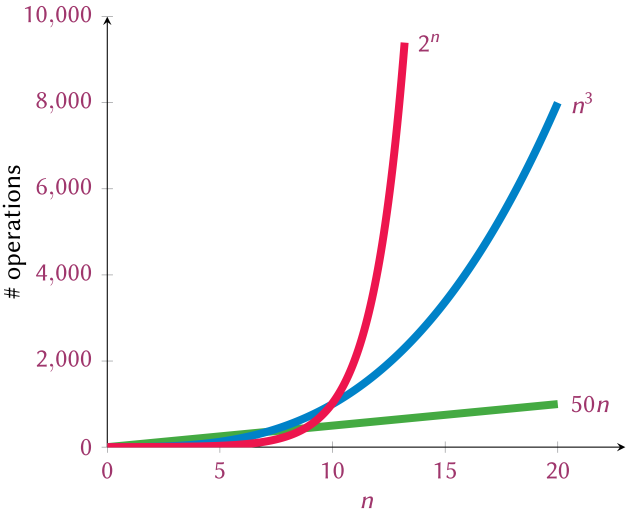
Algorithm: LargestNumber

Input: A non-empty list of numbers L
Output: The largest number in the list L
largest $\leftarrow$ L[0]
foreach each item in the list L **do**
    if the item > largest **then**
        largest $\leftarrow$ the item
return largest

L:							
	2	3	5	1	$\boxed{8}$	1	4

Running time: proportional to number of operations, eg $O(n)$

Growth Functions



NP-hard problems: bad if we have to solve them, good for cryptology