Mathematical Tools for Engineering and Management

Lecture 1

20 Oct 2010

- ▷ Introduction
- ▷ Organisational issues
  - Lectures
  - Exercises
  - Computers
  - Book
  - Exam
- ▷ Topics
  - Why mathematics?
  - Models and Data Sets
  - Two Examples

- Responsible for the Mathematics Section of GPE:
  Prof. Dr. Dr. h.c. mult. Martin Grötschel
  Zuse Institute Berlin, Chair for Combinatorial Optimization at TU Berlin (Mathematics Department)

- Lectures:
  Dr. Axel Werner
  Zuse Institute Berlin, Room 3102
  http://www.zib.de/werner
  e-mail: werner@zib.de or awerner@math.tu-berlin.de
- Exercises:
  Dipl. Math. Olaf Maurer
  TU Berlin, Room MA 519
  e-mail: maurer@math.tu-berlin.de
- **Lecture Website:**
  http://www.math.tu-berlin.de/Vorlesungen/WS10/gpe

▷ Exercise Sessions:

- Every Wednesday, 12–14 (after the lecture)

▷ Exercise sheets:

- Graded homework every other week, to be solved over two weeks, in groups of (maximally) 2

▷ Presentation:

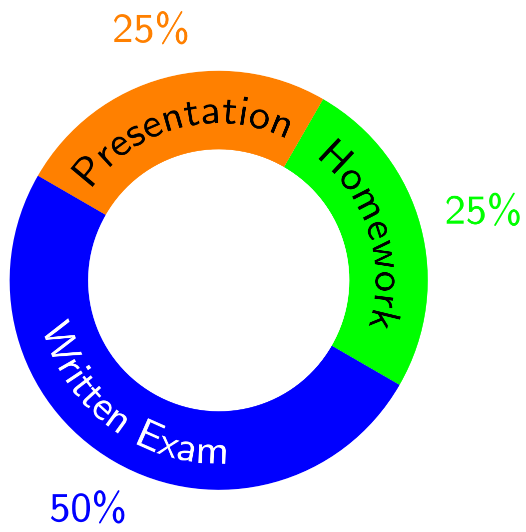
- More extensive exercise, to be solved and presented in groups (somewhen in the middle of semester)

▷ Exam:

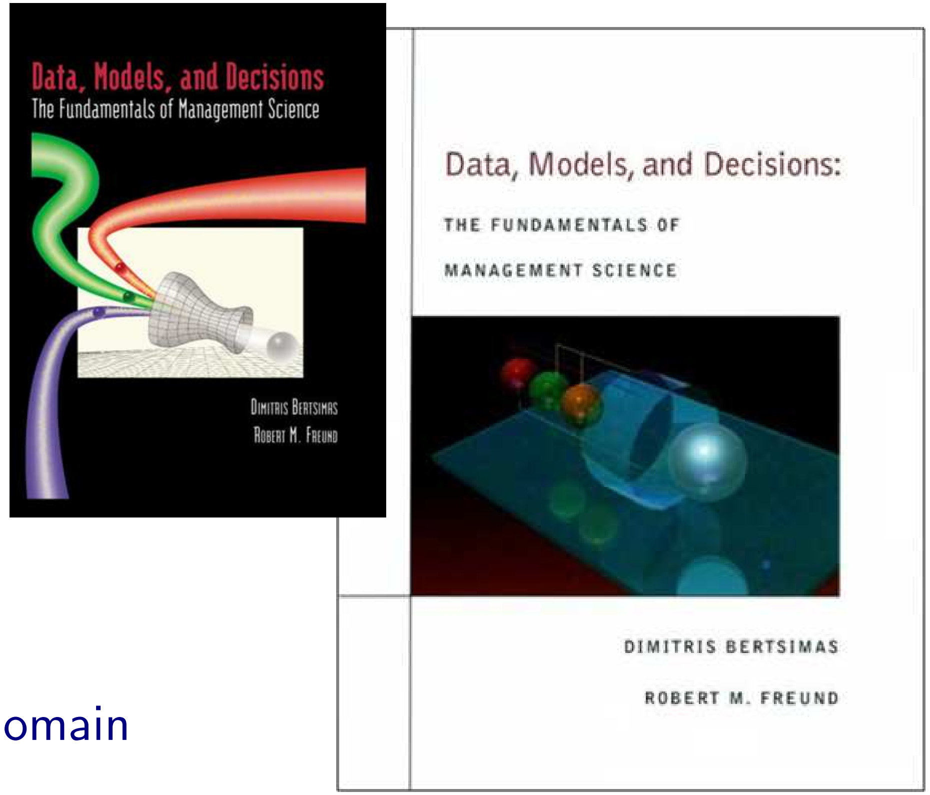
- Written exam in the last two weeks of semester

▷ Grade:

- Computed from results in homeworks, presentation and exam



- Dimitris Bertsimas & Robert M. Freund, *Data, Models, and Decisions – The Fundamentals of Management Science*, 1st ed South-Western College Publishing 2000, 2nd ed Dynamic Ideas 2004
  - supplementary, not necessary
  - copies at the GPE library



- Computer software:
  - AIMMS
    `http://www.aimms.com`
  - ZIB Optimization Suite
    `http://zibopt.zib.de`

  ➡ free academic licences! (download from domain `.tu-berlin.de`)

  - More commercial solvers (most notably CPLEX, Gurobi) and modelling languages (AMPL, GAMS)

- Models, Data and Algorithms
- Linear Optimization
- Mathematical Background: Polyhedra, Simplex-Algorithm
- (Mixed) Integer Programming
- Mathematical Background: Cuts, Branch & Bound
- Combinatorial Optimization
- Mathematical Background: Graphs, Algorithms
- Complexity Theory
- Nonlinear Optimization
- Scheduling
- Lot Sizing
- Multicriteria Optimization
- Exam

▷ **Mathematical Tools**

- How can mathematics support technical/management decisions?

Lectures

▷ **Mathematical Modelling**

- How can a technical/management problem be modelled as a mathematical problem?

Lectures
Exercises

▷ **Mathematical Solutions**

- How can I solve my mathematical problem?

Lectures
Exercises

▷ **Mathematical Background**

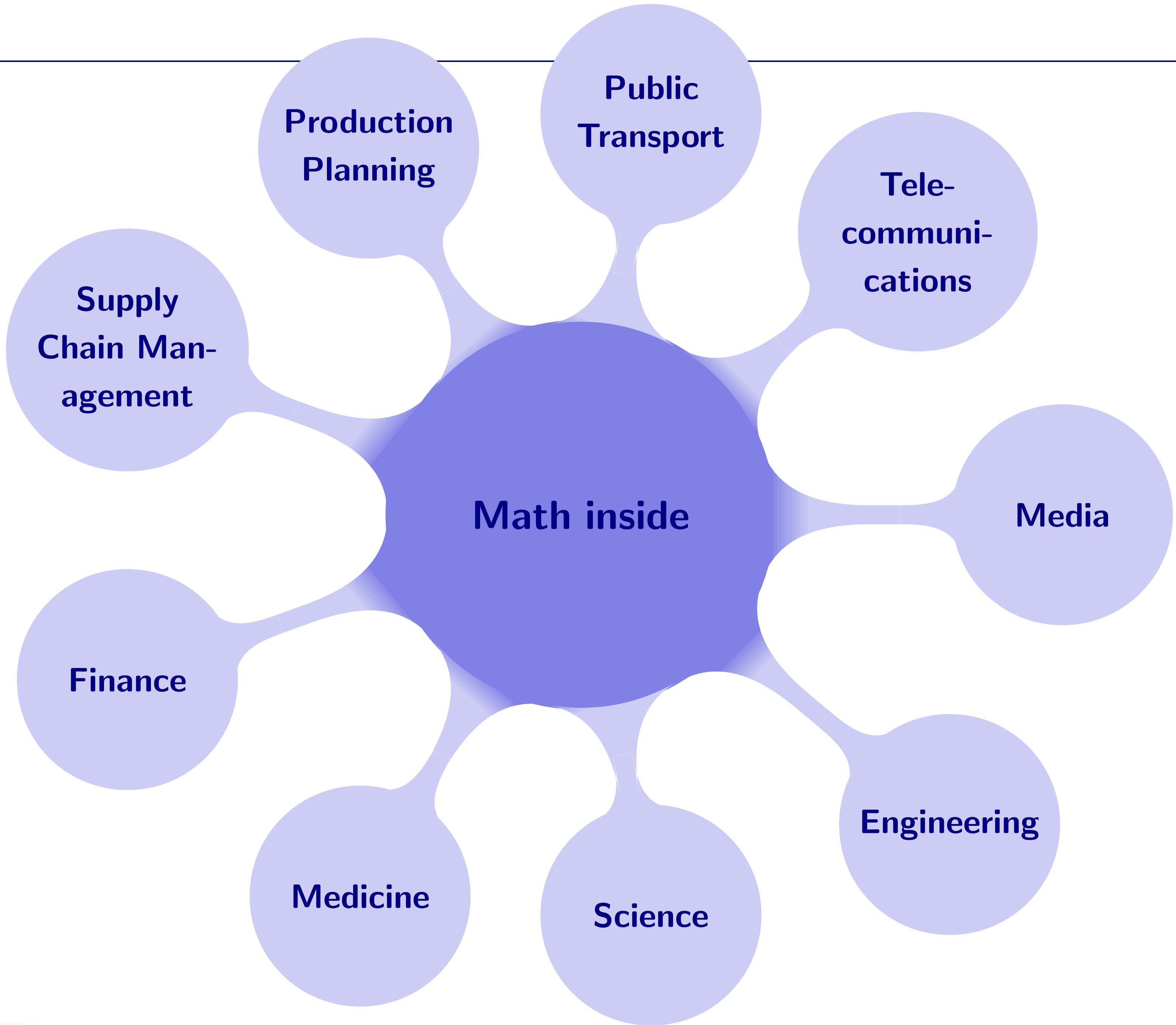
- Why does it work?

Lectures

▷ **Practise**

- How do I use computers and my own wits to solve problems?

Exercises


Math inside
Production Planning
Public Transport
Tele-communi-cations
Supply Chain Man-agement
Media
Finance
Medicine
Science
Engineering

▷ **Mathematics...**

- ...is inevitable

  (for handling complex systems and decisions)

- ...is incorruptible

  (but only if you know what you're doing!)

- ...helps understanding

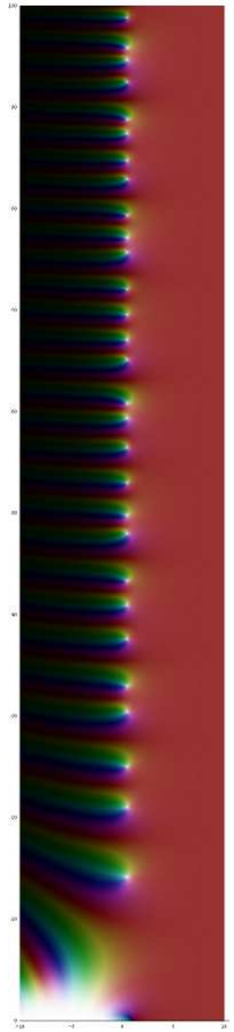
  (complicated matters)

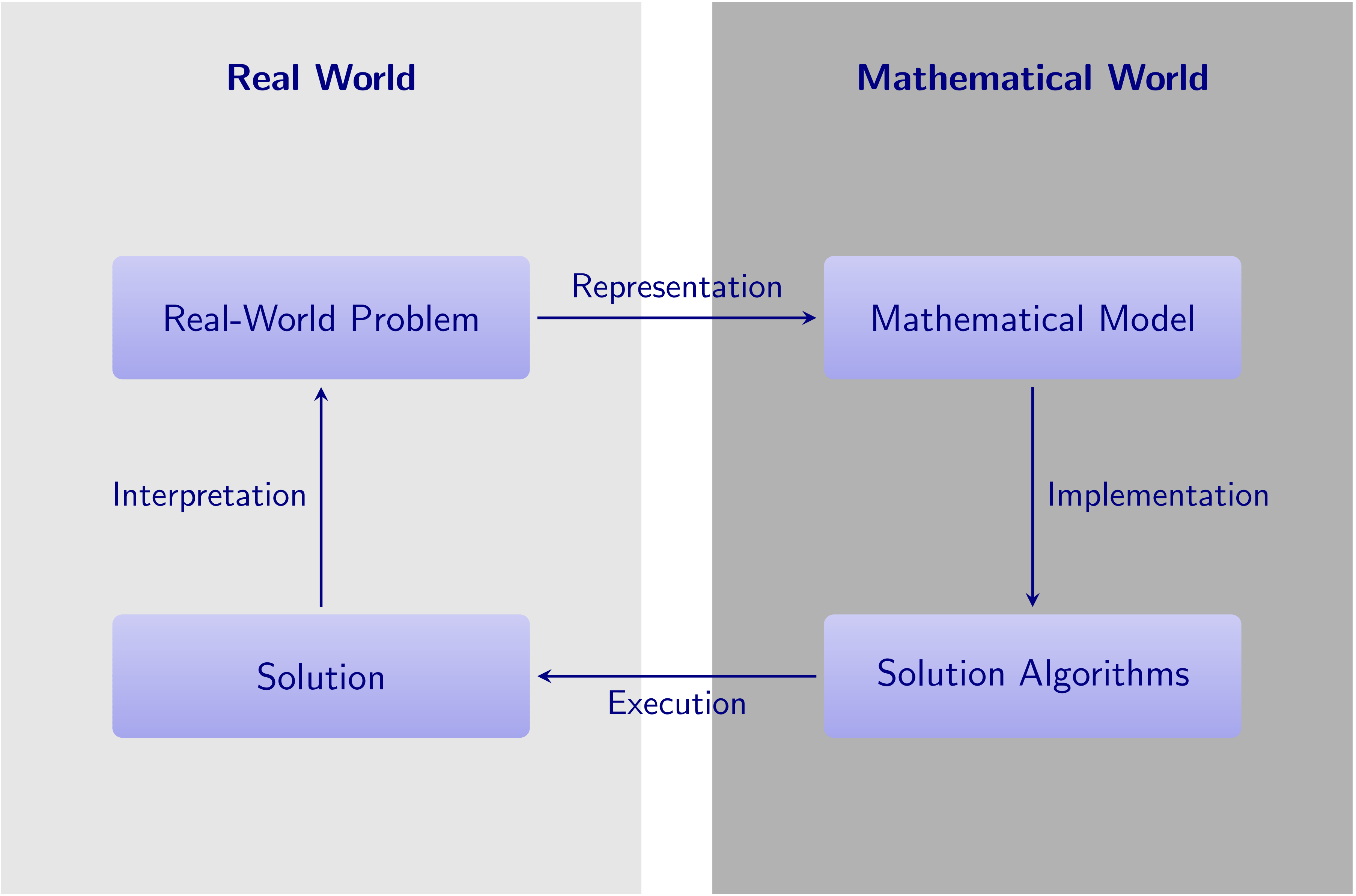
- ...reveals similarities

  (between seemingly unrelated questions)

- ...is beautiful and fascinating

  (which might be a personal viewpoint...)

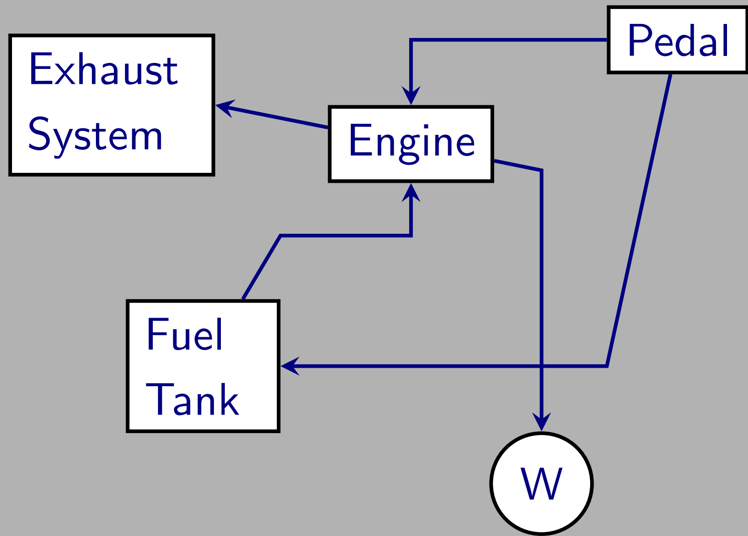



Real World
Mathematical World
Real-World Problem
Representation
Mathematical Model
Interpretation
Implementation
Solution
Execution
Solution Algorithms

Concrete Models

Used for experiments leaving out some functionality

Abstract Models



Objects representing real-world elements, properties etc

Concrete Models

Used for experiments leaving out some functionality

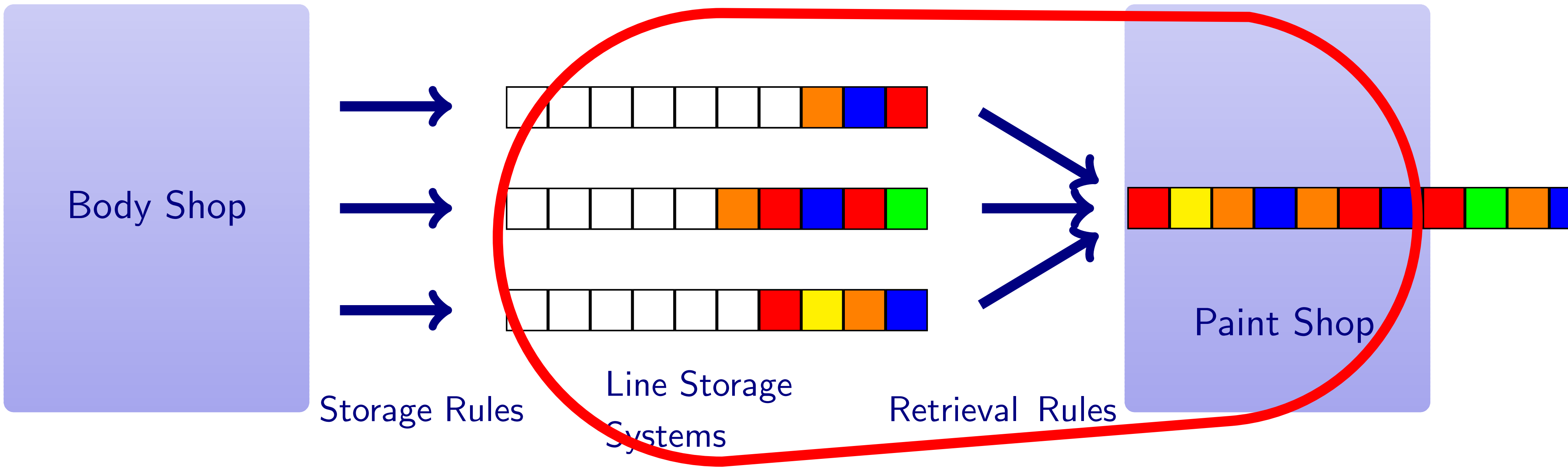
Mathematical Models

$$
\begin{array}{lll}
\max & \displaystyle\sum_{(m,i,j)\in I'} \left(\frac{\tilde{w}_i^{(\pi)} - \tilde{w}_j^{(\pi)}}{l_m}\right) x_{mij} & \\
\text{s.t.} & P_{\text{min}} x_{mij} \;\leq\; p_{mij} \;\leq\; P_{\text{max}} x_{mij} & \forall\ (m,i,j) \in I' \\
 & \displaystyle\sum_{j\in S}\sum_{m\in M} (x_{mij} + x_{mji}) \;\leq\; 1 & \forall\ i \in S \\
 & \gamma_{ij} p_{mij} + \xi_m SG_\infty^{(m,i,j)} (1 - x_{mij}) \geq & \\
 & \xi_m \left(\nu + \displaystyle\sum_{\substack{(n,s,r)\in I' \\ s \| (m,i,j)}} \gamma_{si} p_{nsr}\right) & \forall\ (m,i,j) \in I' \\
 & x_{mij} + x_{nsr} \leq 1 & \forall\ (m,i,j),(n,s,r) \in I' : \\
 & & (m,i,j) \nparallel (n,s,r) \\
 & x_{mij} \in \{0,1\} & \forall\ (m,i,j) \in I' \\
 & p_{mij} \in [0, P_{\text{max}}] & \forall\ (m,i,j) \in I'
\end{array}
$$

Symbols representing real-world objects, decisions etc

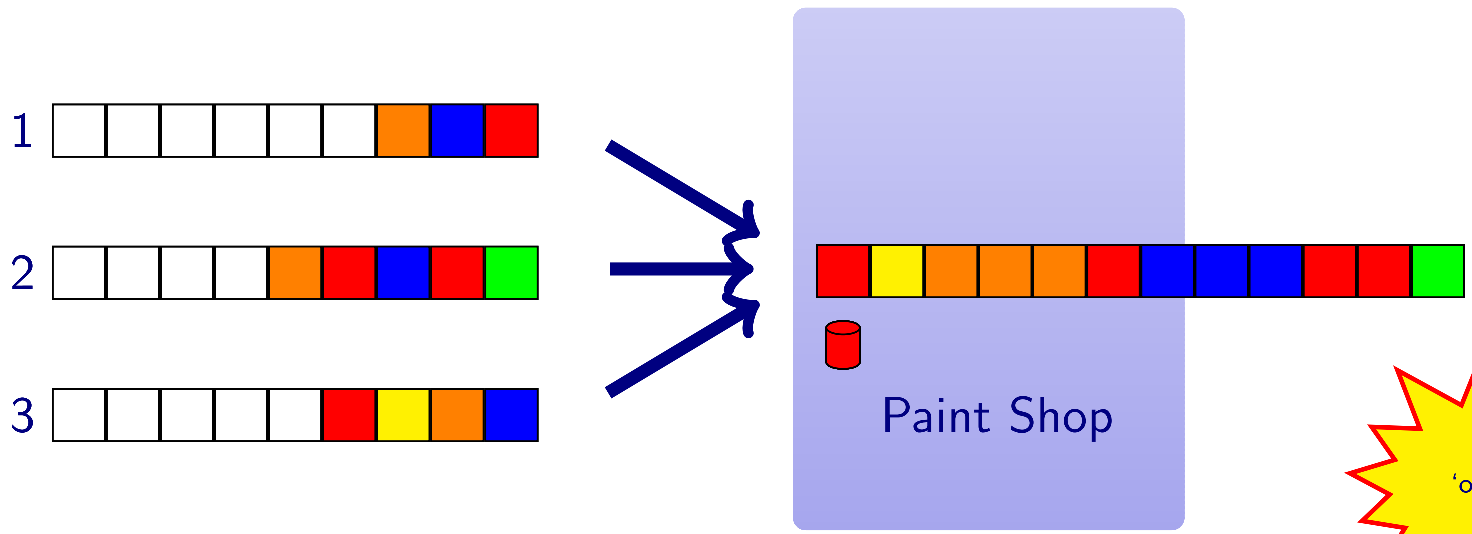
- Automobile assembly line:

- Body to Paint Shop:



- Problem to solve: Design good retrieval rules!

- ▷ What is a 'Retrieval Rule'?
  - ➡ Instructions to produce a sequence of storage line numbers from incoming colour requests
- ▷ What does 'good' mean?
  - ➡ Minimum number of colour changes



- ▷ Example: Retrieval Sequence: 2, 1, 2, 2, 1, 3, 2, 2, 1, 3, 3, 3
  - ➡ Number of colour changes: 6 (optimal!)

Abstract Model: Given a list of colour requests for each of n storage lines, find a sequence of storage line numbers (1 to n), such that the resulting colour list has minimal possible colour changes.

▷ **NOTE:**

- Solving the model means presenting an **Algorithm** that leads to an optimal solution for every possible input!

  ➡ Algorithms are independent of the actual input colour lists!

- It is well possible that there is no such algorithm!

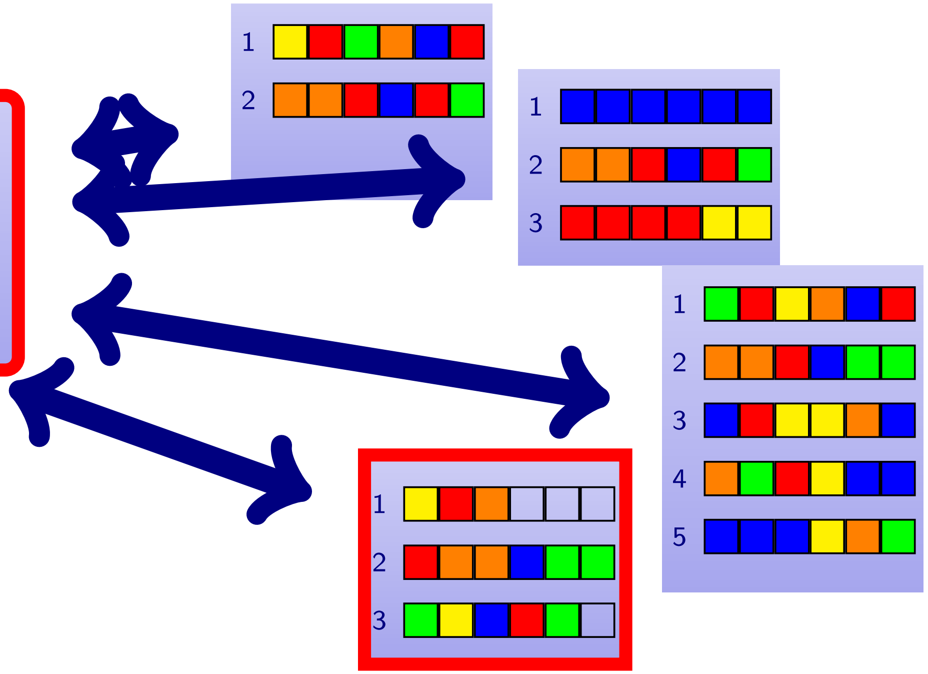
  ➡ Instead solve a (hopefully) easier problem: Give an **approximation algorithm** to find a 'good' resulting colour list.

Precise mathematical model: Given n sequences of colours $c_{i,j}$ $(1 \leq i \leq n,\ j \in \mathbb{N})$ find a mapping $f : \mathbb{N} \to \{0, \ldots, n\}$ such that $\#\{k \mid c_{f(k),j(k)} \neq c_{f(k+1),j(k+1)}\}$ is minimised, where $j(k) := \begin{cases} 1 & \text{if } k{=}1 \text{ or } f(k'){\neq}f(k)\ \forall k'{<}k \\ j(k'){+}1 & \text{otherwise, where } k' \text{ is maximal with } k'{<}k,\ f(k'){=}f(k) \end{cases}$

Models, Data and Instances

Given n sequences of colours $c_{i,j}$ ($1 \leq i \leq n$, $j \in \mathbb{N}$) find a mapping $f : \mathbb{N} \to \{0, \dots, n\}$ such that $\#\{k \mid c_{f(k),j(k)} \neq c_{f(k+1),j(k+1)}\}$ is minimized, where $j(k) :=$

$$\begin{cases} 1 & \text{if } k=1 \text{ or } f(k') \neq f(k)\ \forall k' < k \\ j(k')+1 & \text{otherwise, where } k' \text{ is maximal with } k' < k,\ f(k') = f(k) \end{cases}$$



Model

Data Sets

- An instance is a mathematical model, together with one associated data set
- Worst case: An instance where a given algorithm performs worst
- Average case analysis: Average of the algorithm performance over all instances
- The Objective of an optimization problem is the quantity to be minimized/maximized

- A Model is the mathematical representation of a real-world problem, independent of any Data Set (possible input)
- An Instance is a model, together with one associated data set
- An Algorithm produces a Solution (output) for every instance
- If the model is an optimization problem, an algorithm must produce an optimal solution for every instance
- A solution is optimal if it has the best Objective Value of all possible solutions to the instance that comply with the restrictions of the model
- An Algorithm might solve only special instances of a given model optimally, while giving only suboptimal solutions to the others.
- An Approximation algorithm guarantees that for all (specified) instances the produced solution has objective value not worse than a given factor times the optimum

- Models, Data and Algorithms
- Linear Optimization
- Mathematical Background: Polyhedra, Simplex-Algorithm
- (Mixed) Integer Programming
- Mathematical Background: Cuts, Branch & Bound
- Combinatorial Optimization
- Mathematical Background: Graphs, Algorithms
- Complexity Theory
- Nonlinear Optimization
- Scheduling
- Lot Sizing
- Multicriteria Optimization
- Exam