ACAT 2010

Roman Barták (Charles University in Prague, Czech Republic)

DATA TRANSFER OPTIMIZATION GOING BEYOND HEURISTICS

What is the talk about?

Constraint Programming (and other optimization techniques)

Combinatorial Optimization Problems
such us data transfer optimization, workflow scheduling etc.

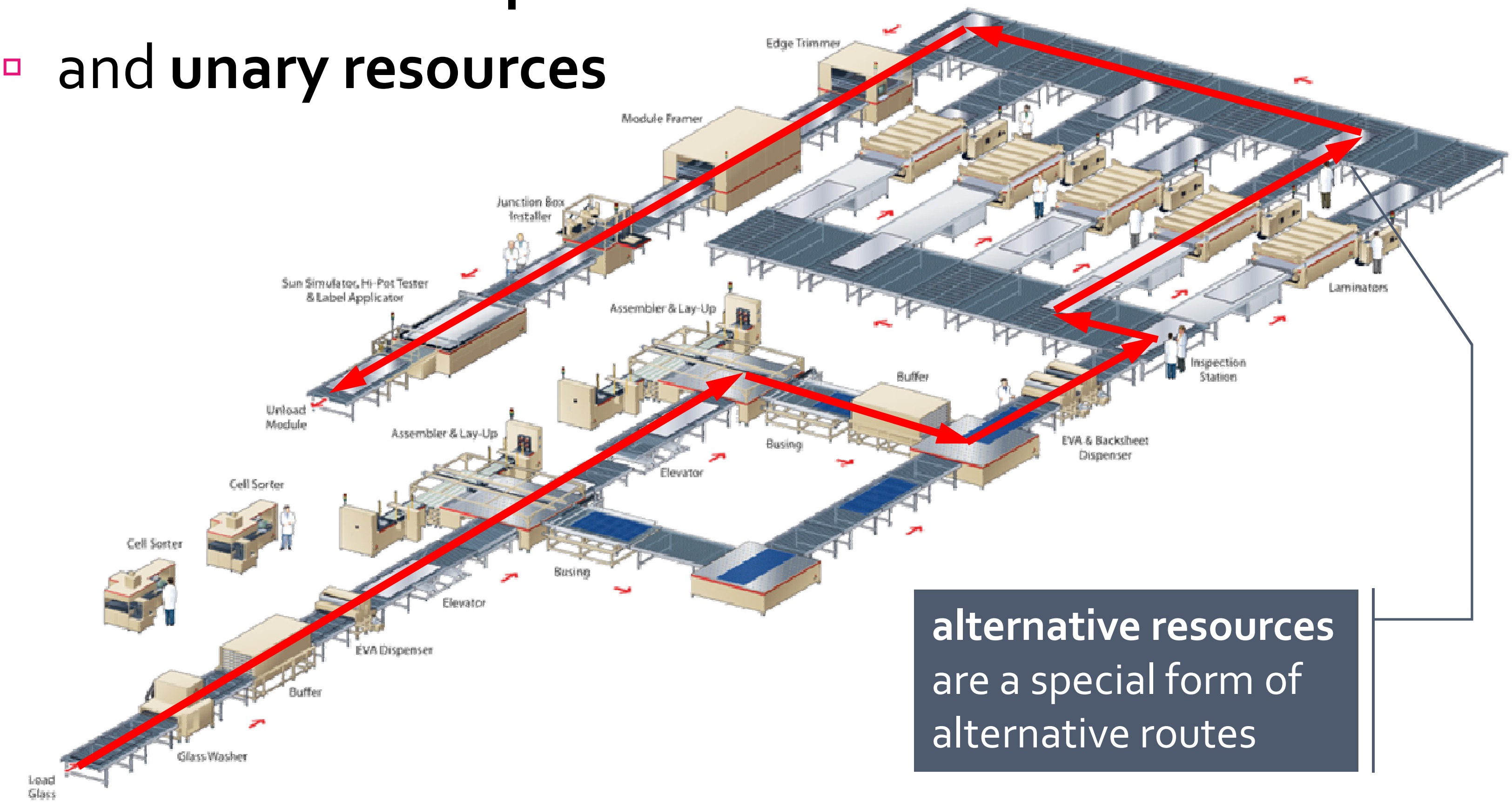
Talk outline

- Motivation
  - planning, scheduling, and constraint satisfaction
- Technology overview
  - constraint programming
- Optimization in Networking Problems
  - path placement problems
- A More Complex View
  - optimizing scientific workflows

Motivation

Production scheduling

- with **alternative product routes**
- and **unary resources**

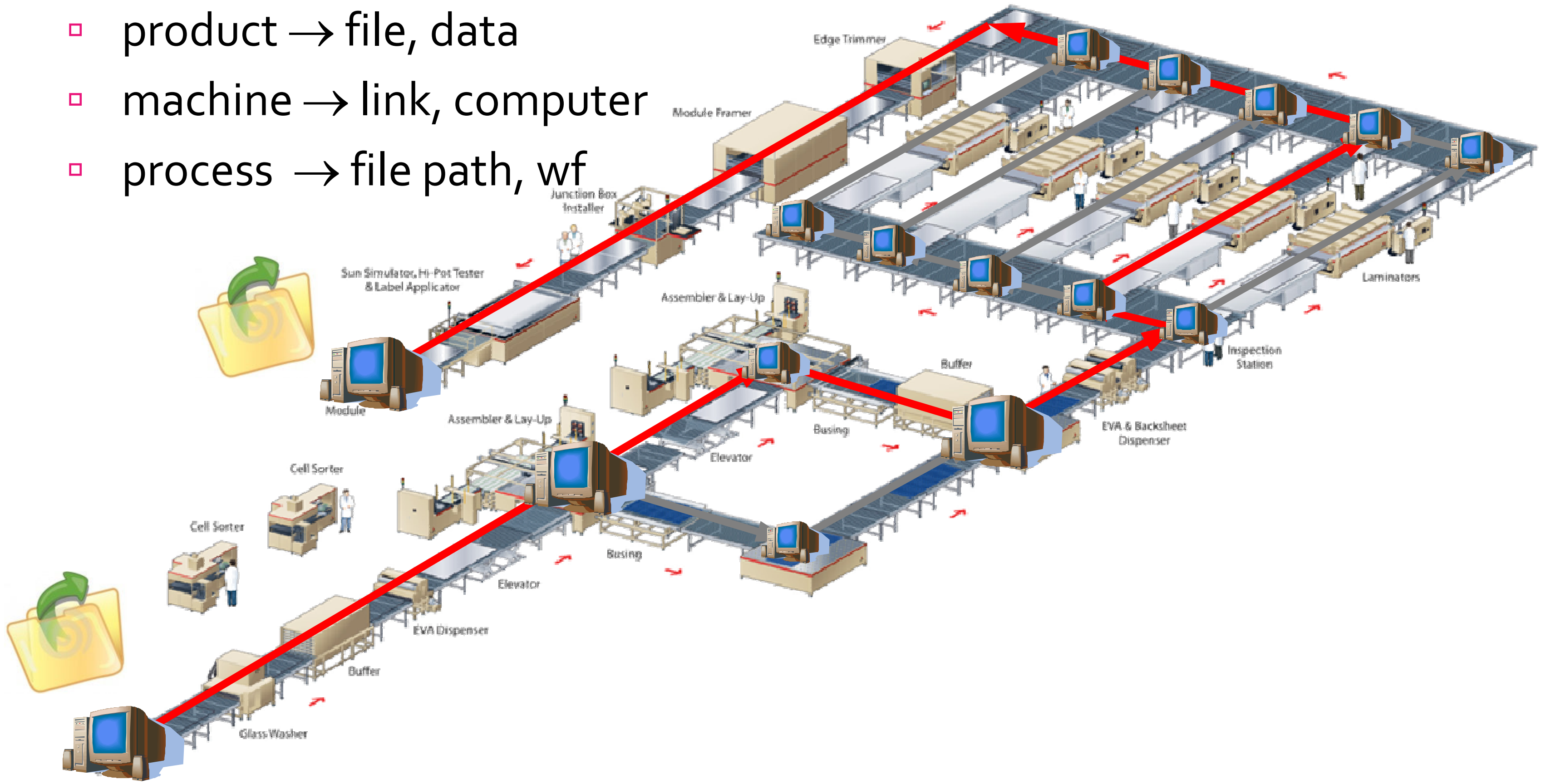


alternative resources are a special form of alternative routes

Motivation

Grids and networking

- factory → network
- product → file, data
- machine → link, computer
- process → file path, wf



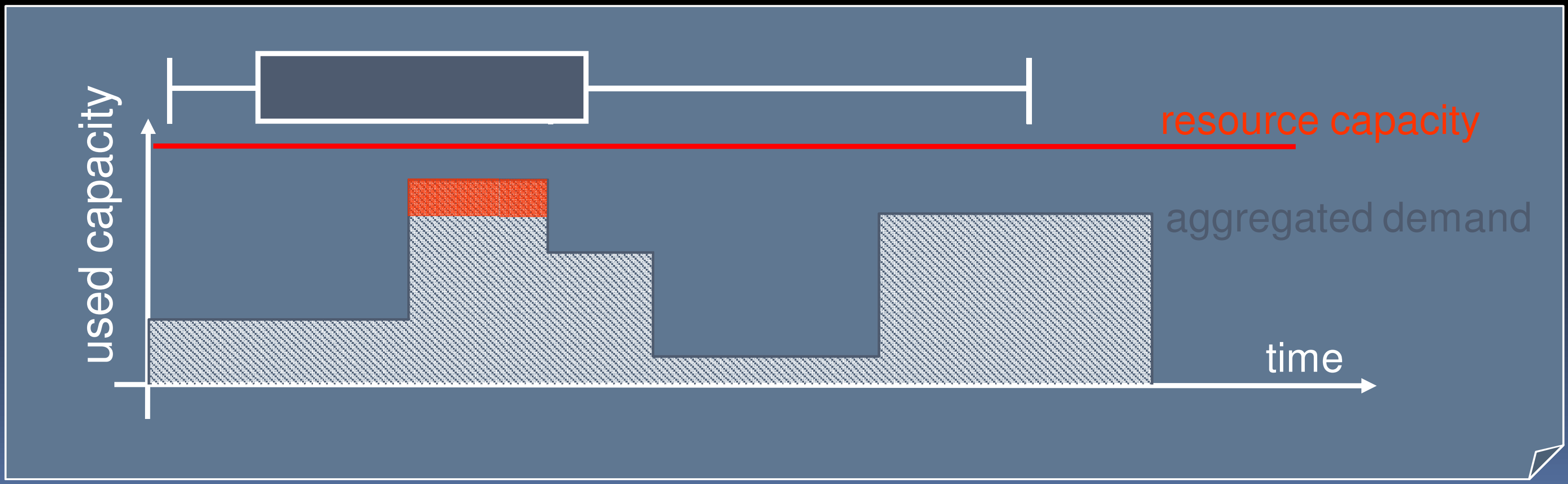
Constraint satisfaction

- Technology for solving combinatorial optimization problems.
- Problem is modeled as a **Constraint Satisfaction Problem** (CSP)
  - a set of (decision) variables
  - each variable has a finite set of values (domain)
  - constraints restrict allowed combinations of values
- Find instantiation of variables satisfying all the constraints.
- Example: Sudoku

9	6	3	1	7	4	2	5	8
1	7	8	3	2	5	6	4	9
2	5	4	6	8	9	7	3	1
8	2	1	4	3	7	5	9	6
4	9	6	8	5	2	3	1	7
7	3	5	9	6	1	8	2	4
5	8	9	7	1	3	4	6	2
3	1	7	2	4	6	9	8	5
6	4	2	5	9	8	1	7	3

CP Technology

- Combination of two approaches
  - **search** (exploring instantiations of variables)
    - incorporates heuristics guiding the solver
  - **inference** (removing values from domains violation any constraint)
    - propagates information through the constraints



CP Application Areas

Planning

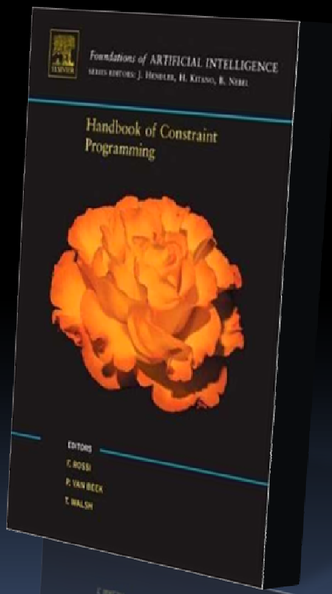
- Autonomous planning of spacecraft operations (Deep Space 1)

Production scheduling

- Saving after applying CSP: US$ 0.2 to 1 million per day

CP in Networking

- Constraint satisfaction can be applied to various networking problems
  - network design
    - how to build the network
  - **path placement** (data transfers)
    - finding a route for each demand satisfying bandwidth constraints)
  - **application placement**
    - workflow optimization



Helmut Simonis: Constraint Application in Networks. Handbook of Constraint Programming, Elsevier, 2006

Path placement

- Given a set of demands for data transfer, **select the demands** and **decide a path** for each demand under the capacity constraints of the network.

$$\max_{\{Z_d, X_{de}\}} \sum_{d \in \mathbf{D}} \texttt{val}(d) Z_d$$

value of satisfied demands is maximized

st.

capacity preserving constraint (Kirchhoff's Law)

$$\forall d \in \mathbf{D}, \forall n \in \mathbf{N}: \quad \sum_{e \in \mathbf{OUT}(n)} X_{de} - \sum_{e \in \mathbf{IN}(n)} X_{de} = \begin{cases} -Z_d & n = \texttt{dest}(d) \\ Z_d & n = \texttt{orig}(d) \\ 0 & \text{otherwise} \end{cases}$$

$$\forall e \in \mathbf{E}: \quad \sum_{d \in \mathbf{D}} \texttt{bw}(d) X_{de} \leq \texttt{cap}(e)$$

capacity of link e is not exceeded

$$Z_d \in \{0, 1\}$$

demand d is satisfied

$$X_{de} \in \{0, 1\}$$

demand d is using link e

Bandwidth on Demand

- Now assume that demands have fixed start and end times. **Select demands and plan routes** without exceeding capacity at any time.

$$\max_{\{Z_d, X_{de}\}} \sum_{d \in \mathbf{D}} \mathtt{val}(d) Z_d$$

st.

$$\forall d \in \mathbf{D}, \forall n \in \mathbf{N} : \quad \sum_{e \in \mathbf{OUT}(n)} X_{de} - \sum_{e \in \mathbf{IN}(n)} X_{de} = \begin{cases} -Z_d & n = \mathtt{dest}(d) \\ Z_d & n = \mathtt{orig}(d) \\ 0 & \text{otherwise} \end{cases}$$

$$\forall t \in \mathbf{T}, \forall e \in \mathbf{E} : \quad \sum_{\substack{d \in \mathbf{D} \\ \mathtt{start}(d) \leq t \\ t < \mathtt{end}(d)}} \mathtt{bw}(d) X_{de} \leq \mathtt{cap}(e)$$

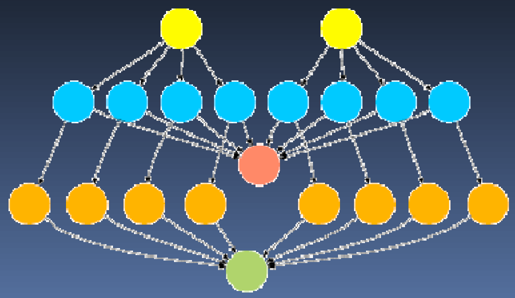
$$Z_d \in \{0, 1\}$$
$$X_{de} \in \{0, 1\}$$

Almost identical model like before!

the capacity is not exceeded at any time t ∈ { start(d) | d ∈ D }

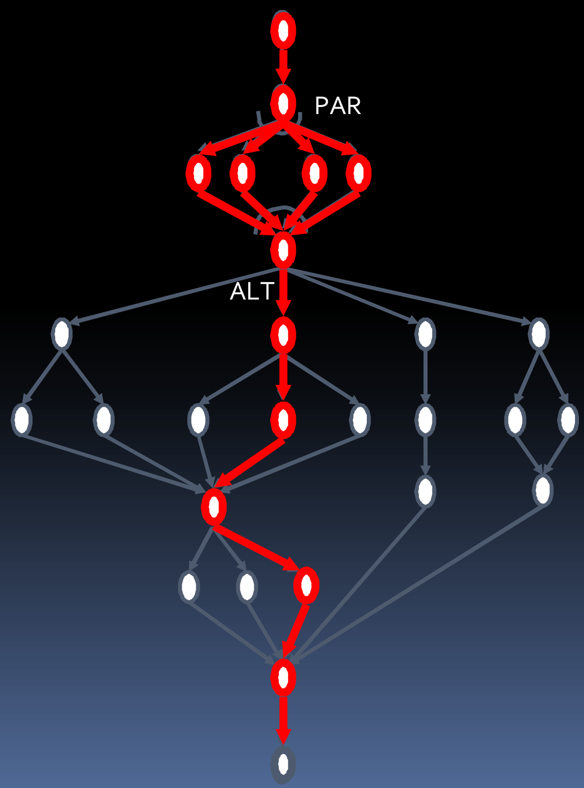
Optimising Workflows

- Transferring data is an interesting problem but why do we need to transfer data?
- **To do some computation with them.**
- Data transfer optimization is just the first step in a more complex process – **optimizing scientific workflows.**
  - where and when the data processing is done?
  - how the necessary data are moved to processing units?



Conceptual Model

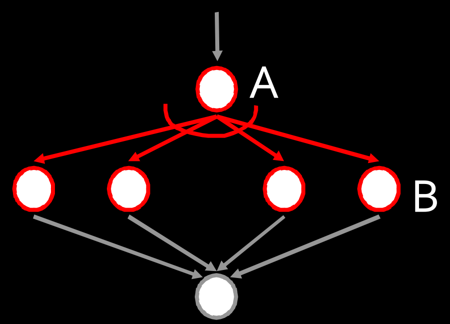
- We model the workflow as a directed acyclic graph called **Temporal network with alternatives** (TNA):
  nodes = operations, arcs = precedence (temporal) relations
  logical dependencies between nodes – **branching relations**.



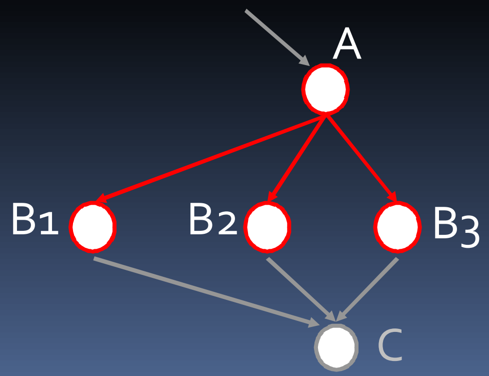
- The process can split into **parallel branches**, i.e., the nodes on parallel branches are processed in parallel (all must be included).
- The process can select among **alternative branches**, i.e., nodes of exactly one branch are only processed (only one branch is included).
- The **problem** is to select a sub-graph satisfying logical, temporal, and resource constraints.

Logical constraints

- The path selection problem can be modeled as a **constraint satisfaction problem**.



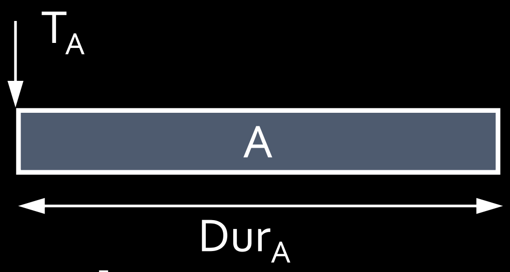
  - each **node** A is annotated by {0,1} variable V_A
  - each arc (A,B) from a **parallel branching** defines the constraint $V_A = V_B$



  - let arc (A,B1),..., (A,Bk) be all arcs from some **alternative branching**, then $V_A = \Sigma_{i=1,\dots,k} V_{Bi}$

Resource constraints

- **standard scheduling model**
  - start time variable: $\mathbf{T_A}$
  - duration variable: $\mathbf{Dur_A}$



- unary (disjunctive) **resource constraints**
  - two operations allocated to the same resource do not overlap in time

$$V_X * V_Y * (T_X + Dur_X) \leq T_Y \vee V_X * V_Y * (T_Y + Dur_Y) \leq T_X$$

  - or, we can use **existing global constraints** modeling unary resource (edge-finding, not-first/not-last, etc. inference techniques) extended to optional operations
    - (in)valid operations: $\mathbf{Val_A = 1 \Leftrightarrow Dur_A > 0}$

Summary

- The heuristics and ad-hoc techniques may lost a global view of the problem (they are **shifting the bottleneck**).
- Optimization techniques provide a **global view** and **high customization** to problem instance.
- They do not provide a universally applicable solution!
  - provide a **dedicated solution** to a specified problem
  - possibly „discover“ a **new type of solution**
  - the solution can **guide the reactive executor**
- The formal abstract model is the first step to real global optimization.
  - can be **easily modified** when the problem changes (new computers are bought, faster connection is established etc.)

How to find the right abstraction level that is close to reality, and appropriate for optimization?

The Vision

- **User** (scientist) **specifies what should be done** and with what data (the workflow)
- The underlying **optimizer** recommends **how to realize this demand efficiently** taking in account other demands (fairness) and infrastructure constraints (speed, capacity).
- The **execution mechanism realizes the plan** and resolves runtime deviations.

ACAT 2010

Roman Barták (Charles University in Prague, Czech Republic)

DATA TRANSFER OPTIMIZATION GOING BEYOND HEURISTICS