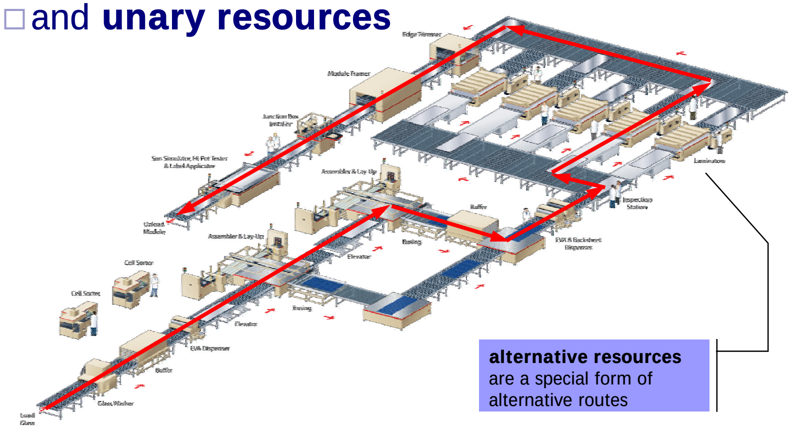
Handling Alternatives in Temporal Networks

Roman Barták
Charles University (Czech Republic)

roman.bartak@mff.cuni.cz
http://ktiml.mff.cuni.cz/~bartak

Motivation

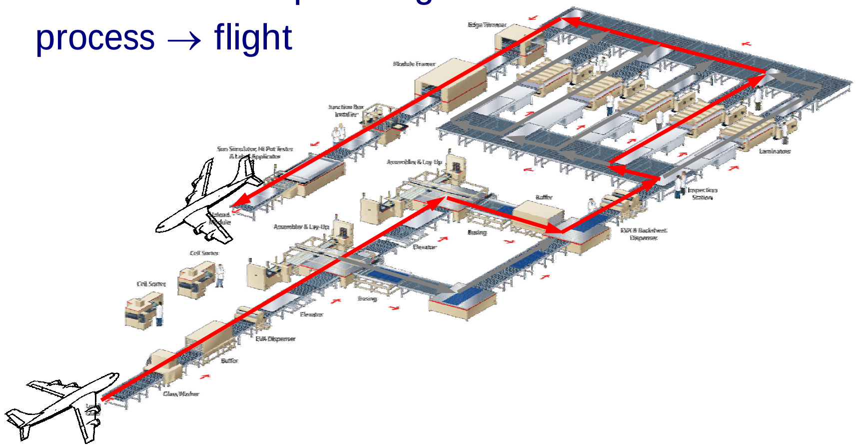
- **Production scheduling**
  - with **alternative product routes**
  - and **unary resources**



alternative resources are a special form of alternative routes

Relation to ATM?

factory → air space
product → airplane
machine → air space segment
process → flight



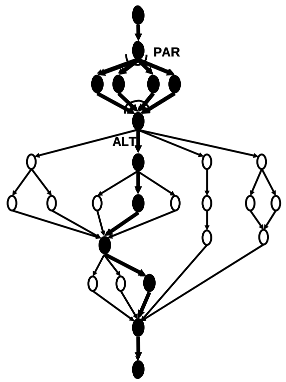
Talk outline

- **Existing model**
  - temporal network with alternatives
  - inference techniques (constraint propagation)
  - search techniques
- **Modelling benefits for ATM**
  - alternative routes with preferred routes and times
  - dynamic features
  - mixed-initiative approach
- **Conclusions**

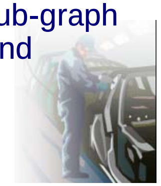
Existing model

Conceptual Model

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

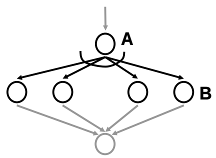


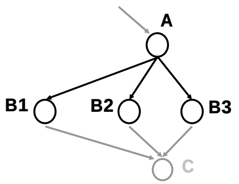
  - The process can split into **parallel branches**, i.e., the nodes on parallel branches are processed in parallel (all must be included).
  - The process can select among **alternative branches**, i.e., nodes of exactly one branch are only processed (only one branch is included).
  - The **problem**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



Logical constraints

- The graph assignment 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 we use the constraint $V_A = \Sigma_{i=1,\dots,k} V_{Bi}$
- The base model can be **strengthen** by adding implied constraints $(V_A = \Sigma_{i=1,\dots,k} V_{Bi} \wedge V_C = \Sigma_{i=1,\dots,k} V_{Bi} \Rightarrow V_A = V_C)$.

Temporal constraints

- We can annotate each arc (X,Y) by a **simple temporal constraint** [a,b] with the meaning $\mathbf{a \leq Y\text{-}X \leq b}$.
  - **(Nested) Temporal Network with Alternatives**
- Base constraint model:
  - each **node** A is annotated by a **temporal variable** T_A with a domain ⟨0,MaxTime⟩, where MaxTime is a constant given by the user.
  - Temporal relation [a,b] between nodes X and Y must hold if both nodes are valid!

$$V_X * V_Y * (T_X + a) \leq T_Y \wedge V_X * V_Y * (T_Y - b) \leq T_X.$$

Notes:

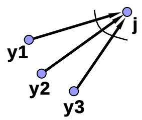
- $V_X = 0 \vee V_Y = 0 \rightarrow 0 \leq T_Y \wedge 0 \leq T_X$
- $V_X = V_Y = 1 \rightarrow (T_X + a) \leq T_Y \wedge (T_Y - b) \leq T_X$.
- The above temporal constraint does not assume the type of branching!

Stronger temporal filtering

Stronger filtering based on two ideas:

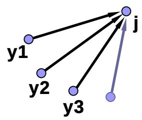
- **always propagate** the temporal constraint (unless any domain becomes empty)
- **assume type of branching** during temporal filtering

- Downstream propagation (upstream is similar)
  - **parallel branching**



- $d(t_j) \leftarrow d(t_j) \cap \cap_{i=1,\dots,k}\{ (d(t_{yi}) + \langle a_{yi,x}, b_{yi,x}\rangle) \}$ if non-empty
- $d(v_j) \leftarrow d(v_j) \cap \{0\}$ if temporal domain becomes empty

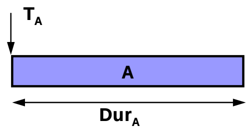
  - **alternative branching** (constructive disjunction)



- $d(t_j) \leftarrow d(t_j) \cap \cup_{i=1,\dots,k}\{ (d(t_{yi}) + \langle a_{yi,x}, b_{yi,x}\rangle) \}$ if non-empty
- $d(v_j) \leftarrow d(v_j) \cap \{0\}$ if temporal domain becomes empty

Resource constraints

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



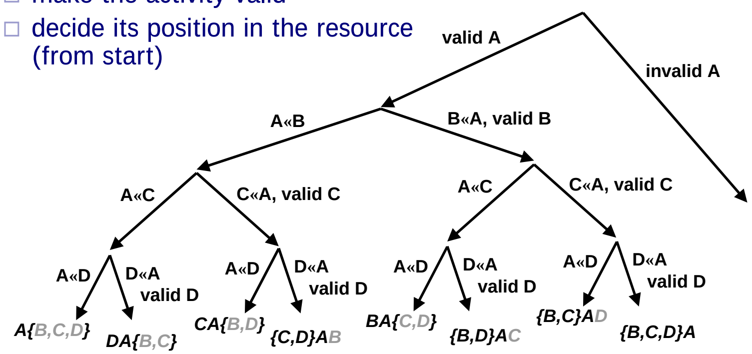
- unary (disjunctive) **resource constraints**
  - two activities allocated to the same resource do not overlap in time

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

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

Branching Strategy

- constraints filter out a lot of infeasibilities, but frequently **some options remain to be explored**
- explored **by search** in a backtracking manner (try some alternative and if it leads to a failure try another one)
  - select some activity (earliest start first)
  - make the activity valid
  - decide its position in the resource (from start)



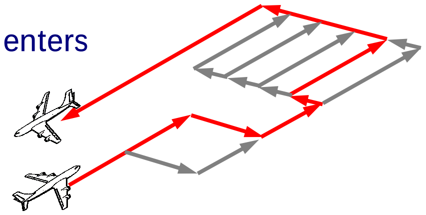
Application to ATM

Base model

- each **aircraft/flight** is modelled using **TNA**
  - **node** = enter to a flight segment
    - typically one enter and one exit point per flight
    - pre-specified segments to enter
  - **temporal relation** = minimal and maximal duration to fly through the segment
    - depends on possible aircraft speeds and other factors
- exclusive use of flying segments is modelled using a **unary resource**
- **The model integrates sequencing decisions with selection among alternative routes.**

Alternative routing

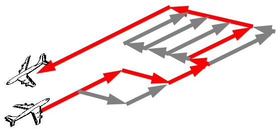
- Structure of TNA can computed in advance from the map of flying segments and required enter/exit points for each flight
- **validity variable**
  - describes whether the flight enters the segment
  - some segments (enter/exit) are pre-selected
  - logical (branching) constraints guarantee feasibility of the route
- **temporal variable**
  - describes when the flight enters the segment
  - temporal constraints ensure „smooth“ flight



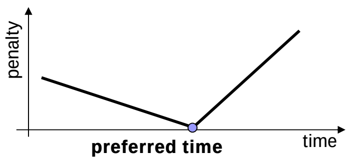
ongoing

Preferences

- **preferred route**
  - each node is annotated by a preference (positive integer)
  - guide for selection of the routes (preferred routes are tried first)

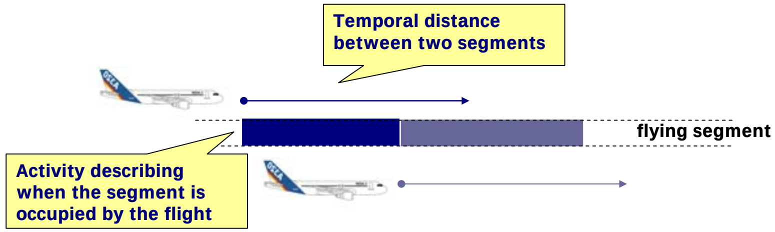


- **preferred time**
  - some nodes are annotated by preferred time and penalty for being late/early
  - optimization of on-time performance



Flying segments

- Entering the flying segment means using it exclusively for some time
  - flying activity consuming unary (disjunctive) resource
- Separation of aircrafts



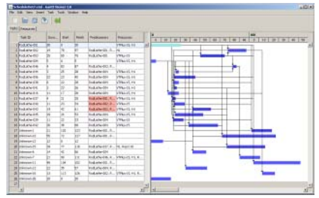
ongoing

Dynamic features

- **On-line demands**
  - new flights are coming during scheduling
    - interruption of scheduling
    - extending the model by new variables and constraints
    - continue in scheduling
- **Unexpected events** (forbidden segment,...)
  - rescheduling
    - remove some decision constraints
    - add constraints describing the event
    - continue in scheduling
- **Rolling horizon**
  - continuous planning
    - use (part) of existing schedule as constraints model and use it in the next iteration

Mixed initiative

co-operated problem solving by humans and computers

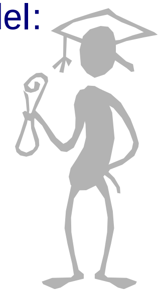


Interactive Gantt chart

- initial schedule displayed as a Gantt chart
- user modifies the schedule (sequencing, timing, resource allocation)
- visualisation of constraint violation
- automatic correction the schedule

Conclusions

- We proposed a **formal model** that **integrates:**
  - **logical reasoning** about alternative routes
  - **temporal reasoning**
  - **resource reasoning**
- The model **exploits** existing constraint satisfaction technology such as **resource constraints**.
- There are two ways of exploiting the model:
  - **constraint propagation**
    - removal of infeasible (conflicting) options
    - possibly incomplete
  - **complete solution**
    - using search techniques
    - possibly long runtimes for optimisation



Handling Alternatives in Temporal Networks

Roman Barták
Charles University (Czech Republic)

roman.bartak@mff.cuni.cz
http://ktiml.mff.cuni.cz/~bartak