Train scheduling for a better society

How SINTEF helps railway companies make better decisions with Gurobi

***Giorgio Sartor**, SINTEF Optimization*

Agenda

Part ONE

1. A world-leading research institute
2. Who we are

Part TWO

1. What is Train Scheduling and why is it so important?
2. Three levels of scheduling

Part THREE

1. The alternative graph formulation
2. An efficient MIP solution method

Part FOUR

1. The train dispatching competition

Part ONE


SINTEF
A WORLD-LEADING RESEARCH INSTITUTE
-INDEPENDENT AND NON-PROFIT
Technology for a better society

Vision: Technology for a better society

Contribute to competitiveness and societal benefit by realizing the UN's Sustainable Development Goals

COLLABORATION IN AN ECOSYSTEM OF INNOVATION

- where new businesses play a bigger part

Part TWO

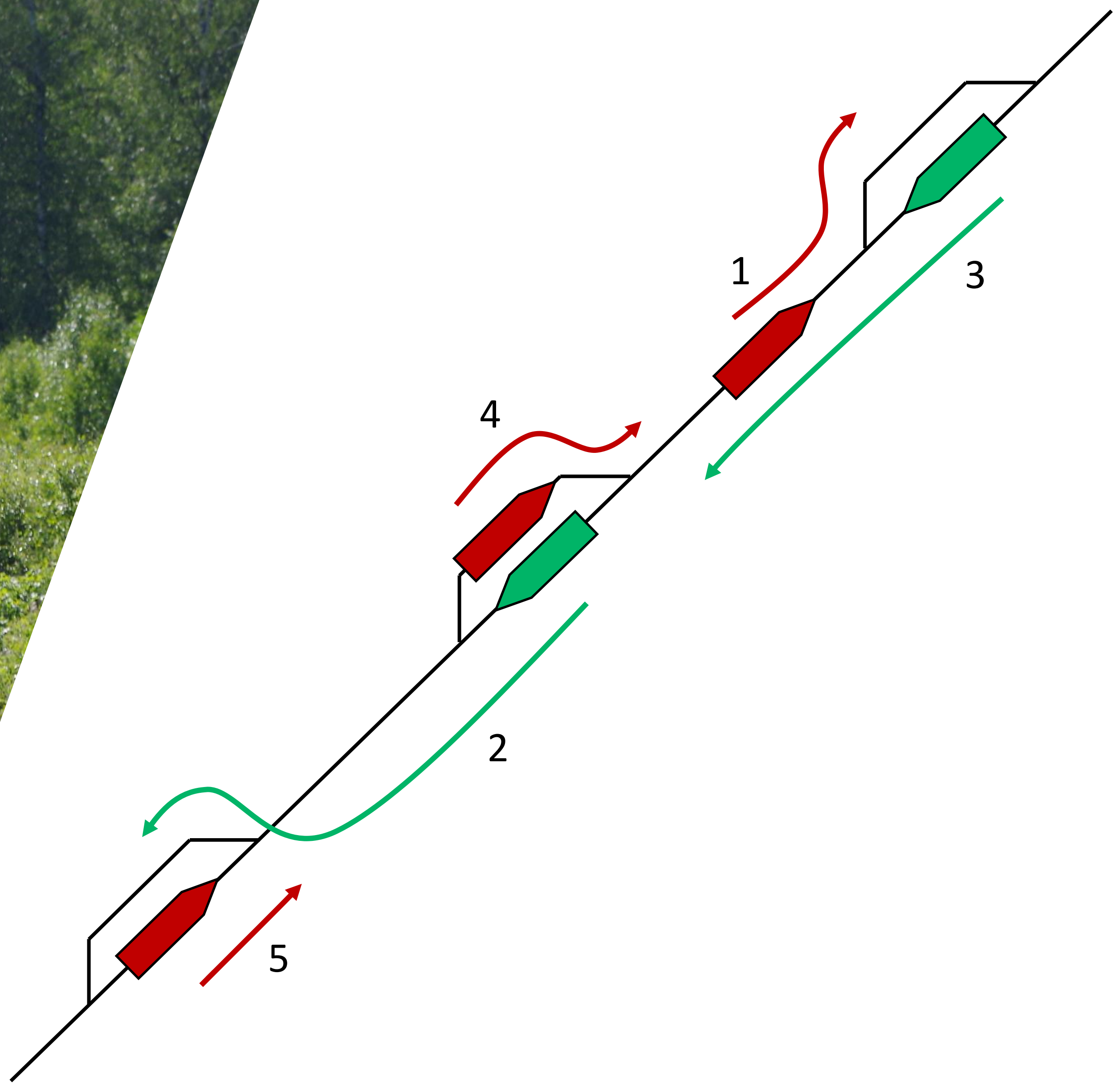
Train scheduling

Definition

Determine arrival and departure time of every train from each section of the railway network while:

- Preventing crashes
- Satisfying safety requirements
- Optimizing an objective function:
  - Minimize delay
  - Minimize passenger travel time
  - Satisfy train operators
  - Etc.

SINTEF



Why is it so important?

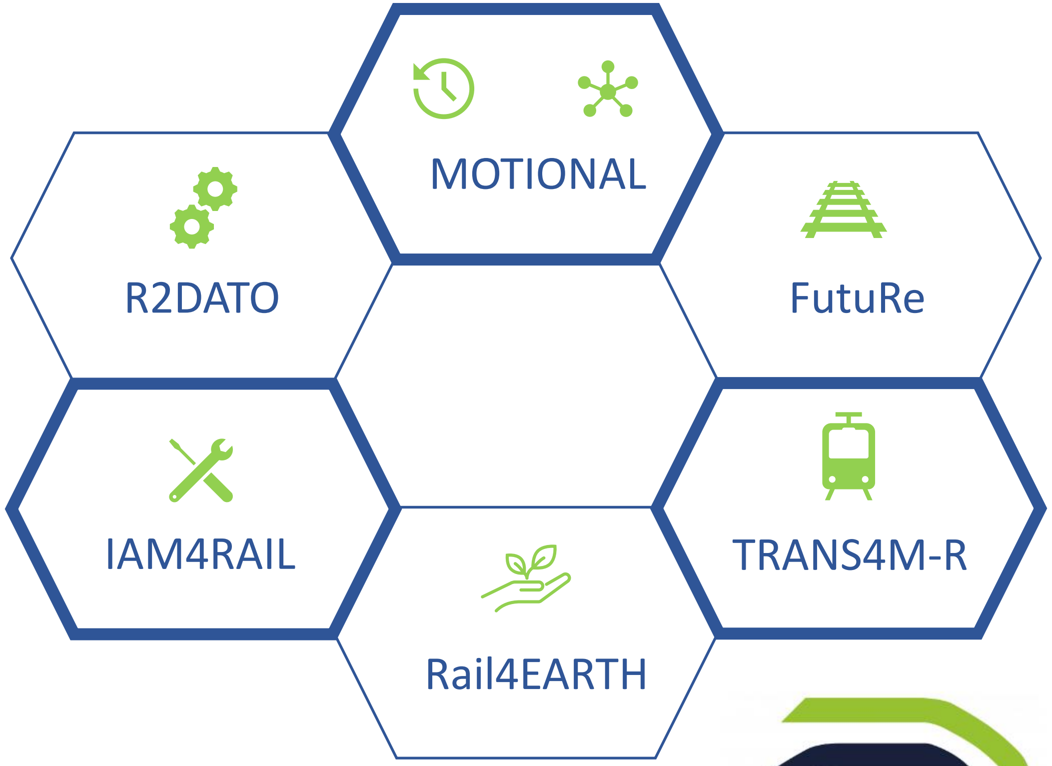
- A recent study [1] estimated that a reduction of one minute of average delay in the Oslo area could provide social savings of up to **$10 million** per year.
- The Norfolk Southern Railroad (NS) freight company determined that "Every mile-per-hour increase in average speed translates to **$200 million** savings in capital and operational expenses annually for NS" [2].

[1] Norløff-Mathisen, C., 2017. På sporet av den tapte tid: samfunnsøkonomisk kostnad av forsinkelse på jernbane. *Master's thesis, Norges handelshøyskole*.

[2] Bollapragada, S., Markley, R., Morgan, H., Telatar, E., Wills, S., Samuels, M., Bieringer, J., Garbiras, M., Orrigo, G., Ehlers, F. and Turnipseed, C., 2018. A novel movement planner system for dispatching trains. Interfaces, 48(1), pp.57-69.

ERJU – Europe's Rail Joint Undertaking

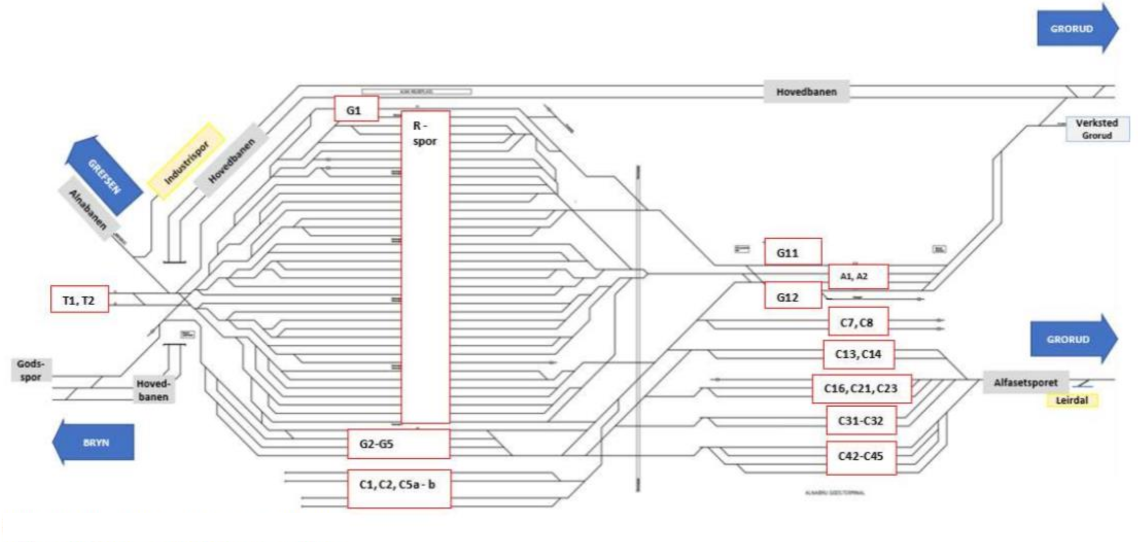
- Most ambitious train innovation project ever:
  - €1.2bn
  - 25 members
  - TRL 6-9
- SINTEF leads the development of timetabling algorithms and contributes to many others:
  - Short-term timetabling in the Genoa node, in collaboration with Hitachi Rail
  - Rolling-stock assignment, in collaboration with Jernbanedirektoratet
  - Enhanced yard planning systems and seamless coordination with last-mile operations, in collaboration with Bane NOR
  - Accurate prediction of freight trains arrivals, in collaboration with Trafikverket

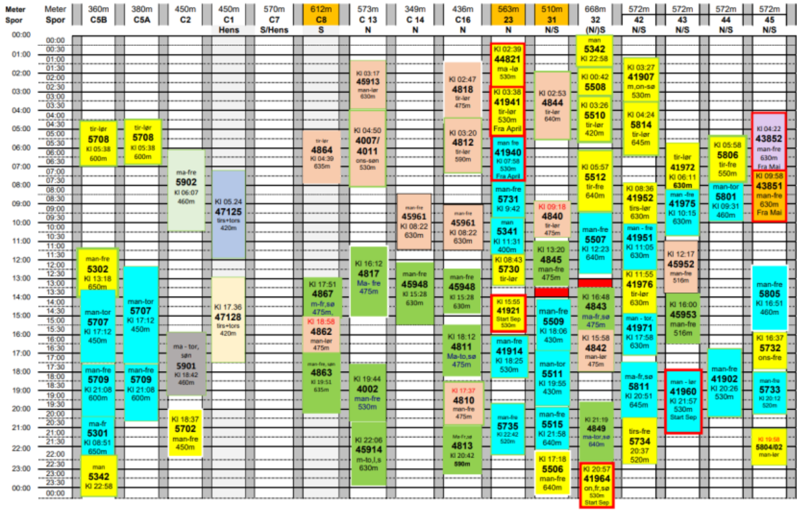


SINTEF

The Alnabru terminal

- Biggest freight terminal in Norway
- Famous for being very close to the city centre, it is also very difficult to expand
- The only reasonable way to increase capacity is through decision intelligence
- Higher efficiency ==> Higher capacity

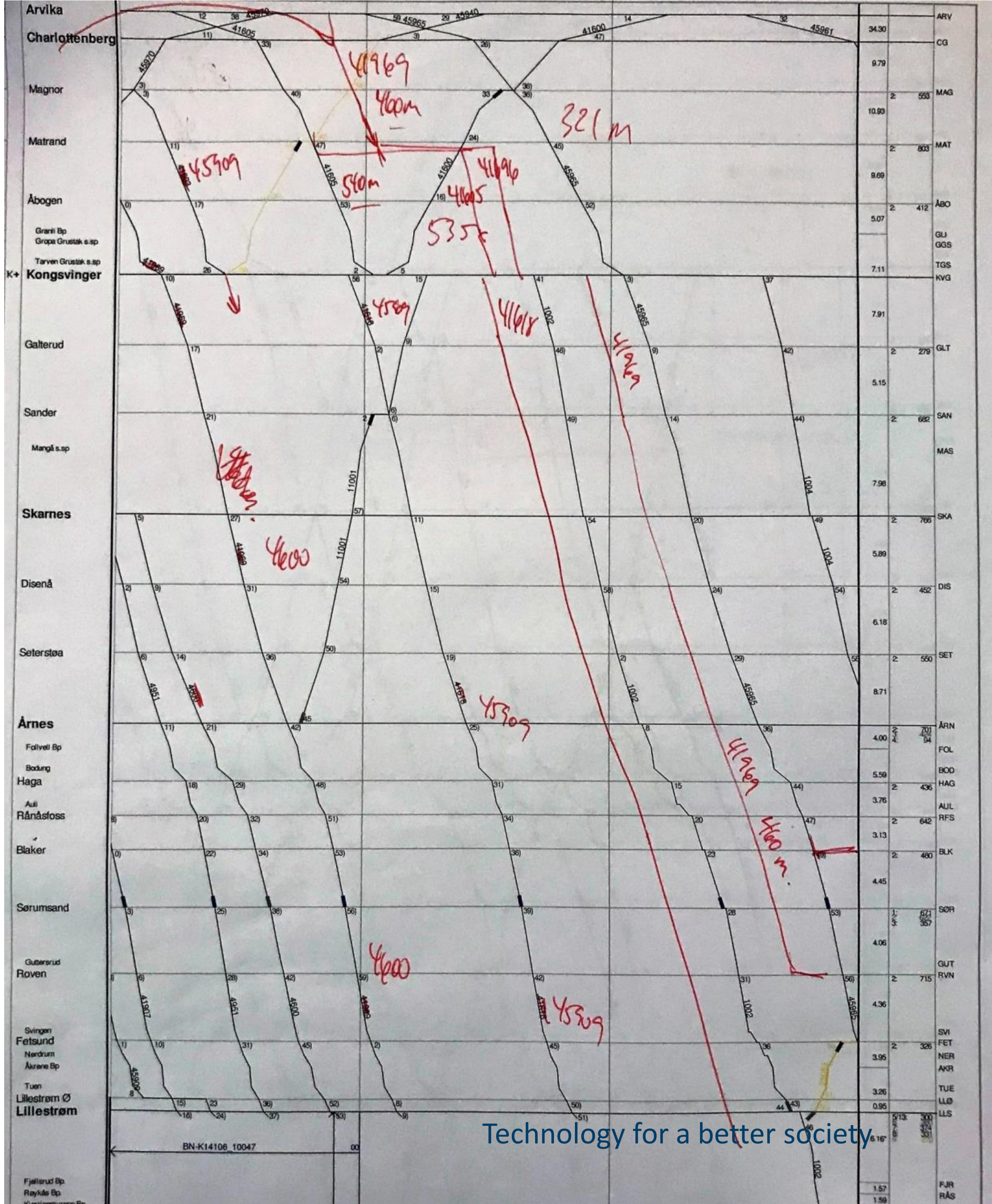




SINTEF

How is it done now?

- Basically…by hand
- Most of national railway managers and private freight companies do not have automatic conflict resolution systems
- Some have semi-automatic systems
- Some have non-automatic digital systems
- Others just have pen and paper



Dispatching

Problem definition

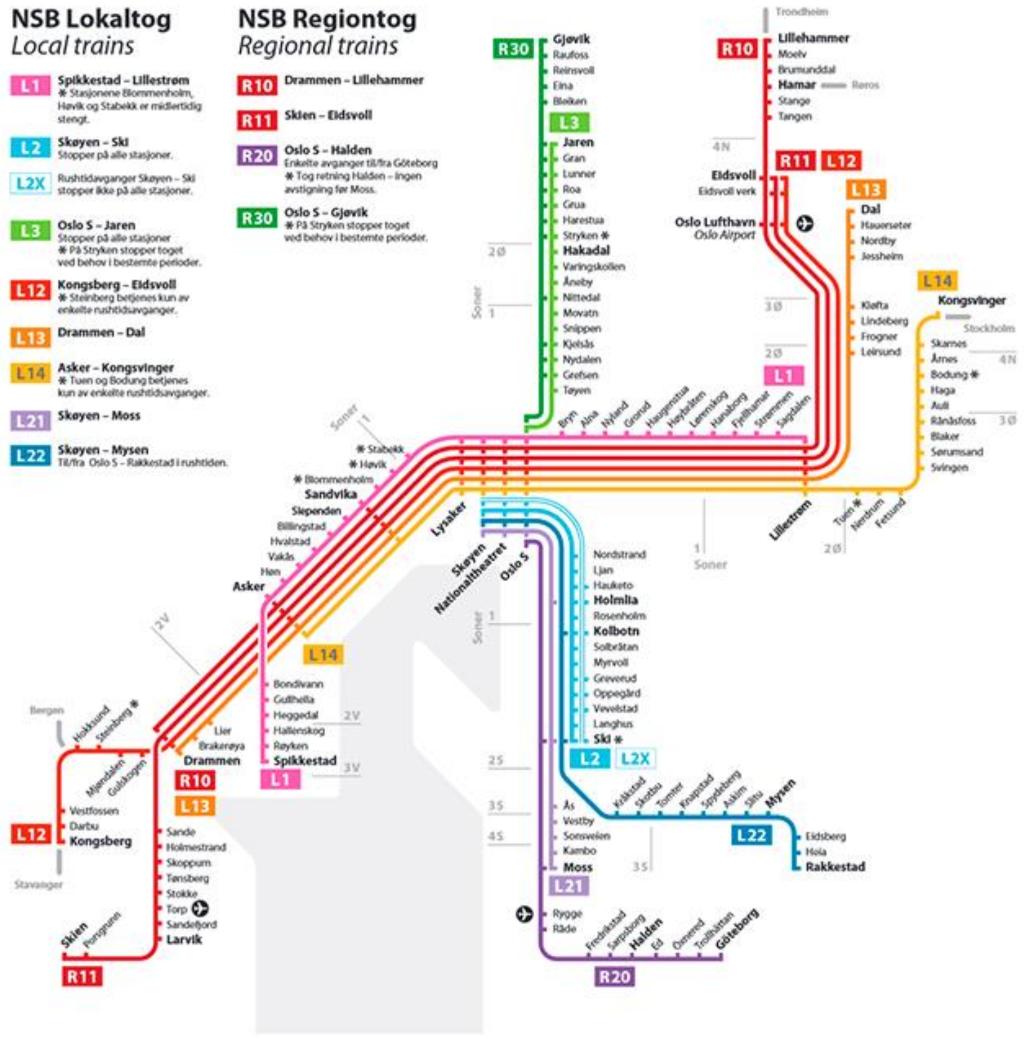
Given that one or more trains are delayed, how to schedule the current and future trains to minimize knock-on delays.

Why is it tricky?

- A solution must be computed in seconds
- Real-time data is not always precise
- Drivers might not follow the plan

Lamorgese, L., Mannino, C. and Piacentini, M., 2016. Optimal train dispatching by Benders'-like reformulation. Transportation Science, 50(3), pp.910-925.

Bach, L., Mannino, C. and Sartor, G., 2019. MILP approaches to practical real-time train scheduling: the Iron Ore Line case. In INOC (pp. 78-82).



Tactical timetabling

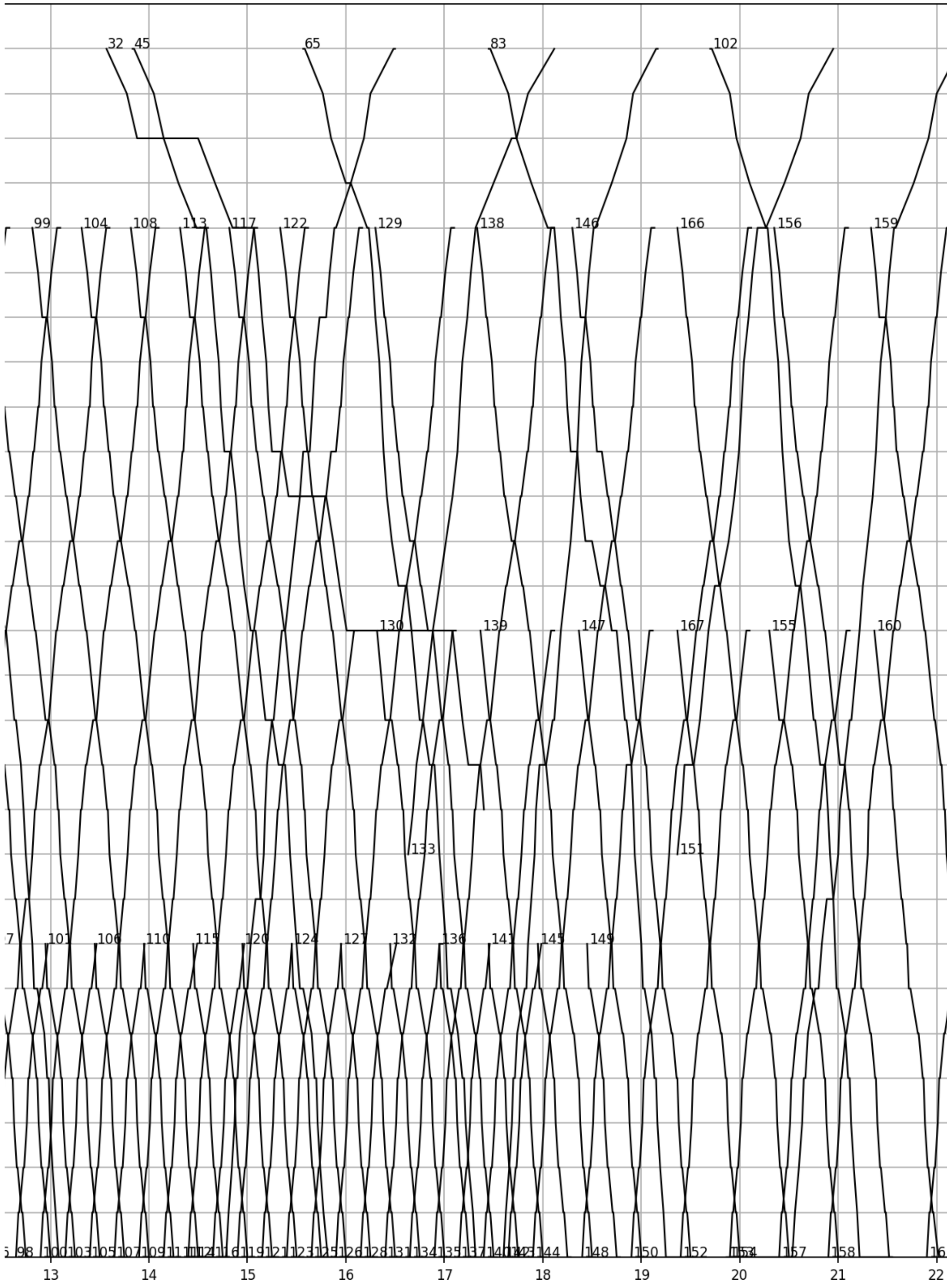
Problem definition

Given a set of desired train services from the train operators, predefined international corridors, and maintenance plans, find a new timetable for the following year.

Why is it tricky?

- Train periodicity must be taken into account
- Different train operators with different objectives have to come to an agreement
- Adaptations for maintenance plans are only temporary

Kloster, O., Luteberget, B., Mannino, C. and Sartor, G., 2023. An Optimization-Based Decision Support Tool for Incremental Train Timetabling. In SN Operations Research Forum (Vol. 4, No. 3, pp. 1-20). Springer.



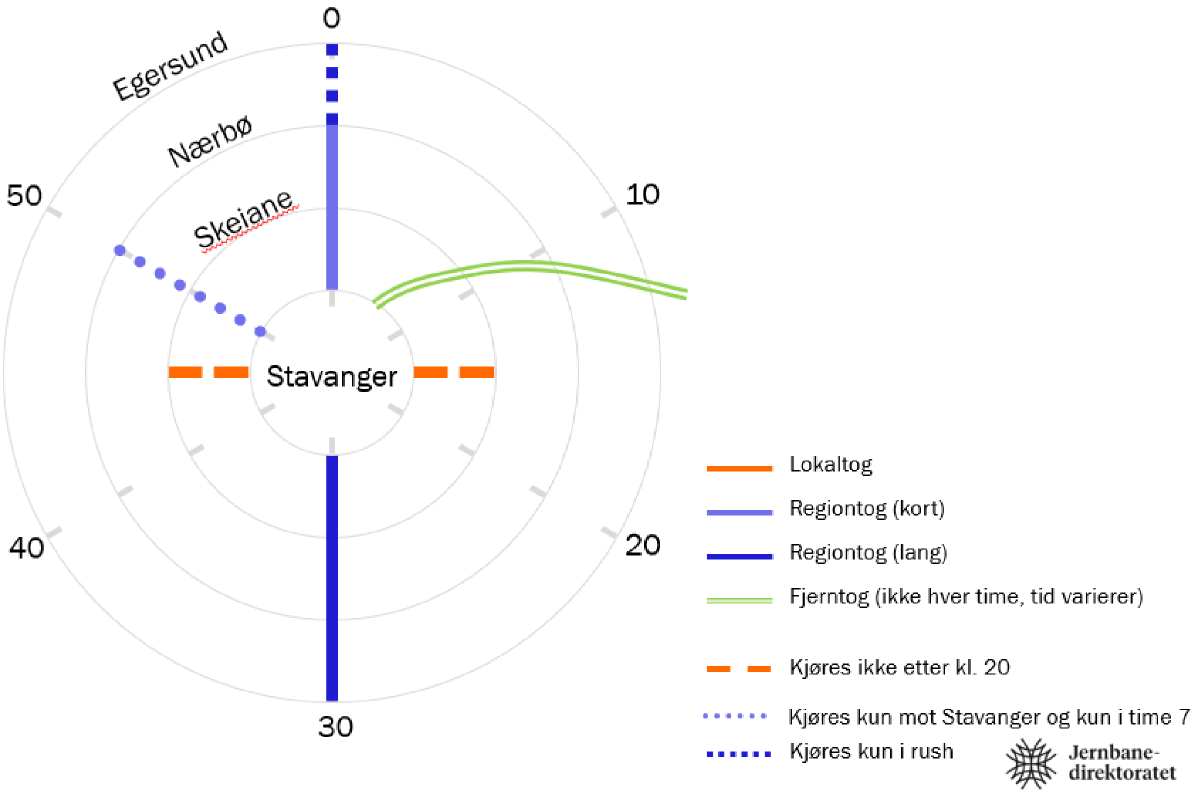
Strategic timetabling

Problem definition

Given a set of possible train services and frequencies, determine a tentative, approximated timetable.

Why is it tricky?

- It is the most complex of the three
- Rolling-stock rotation may be considered
- Handling different future infrastructure upgrades requires a flexible framework

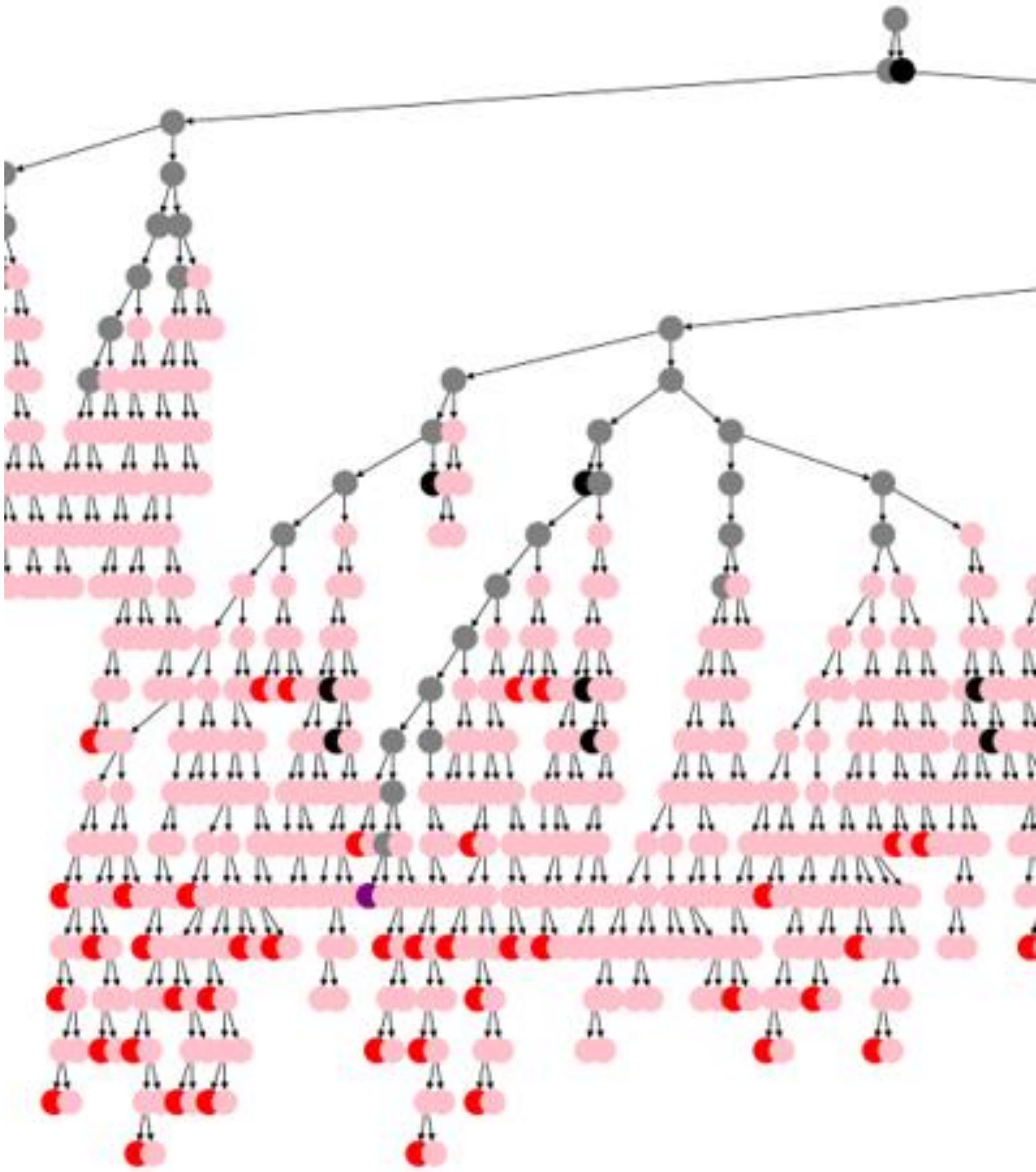


Sartor, G., Mannino, C., Nygreen, T. and Bach, L., 2023. A MILP model for quasi-periodic strategic train timetabling. Omega, 116, p.102798.

Part THREE

Optimization is key

- Single/few railway lines
- Tens of trains
- Hundreds of conflicts
- Thousands of possible train meetings
- Millions of solutions

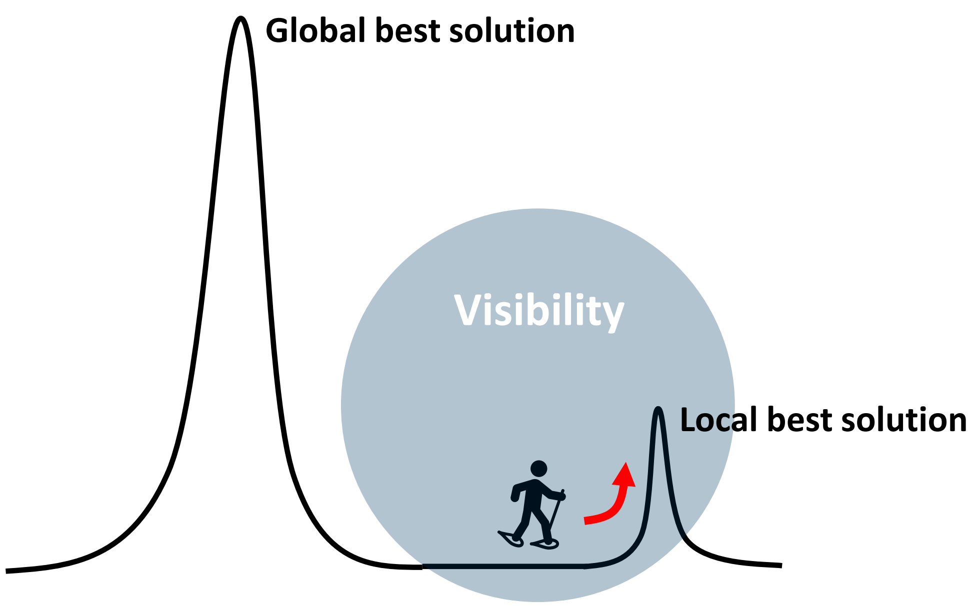


How do we do it then?

Exact methods

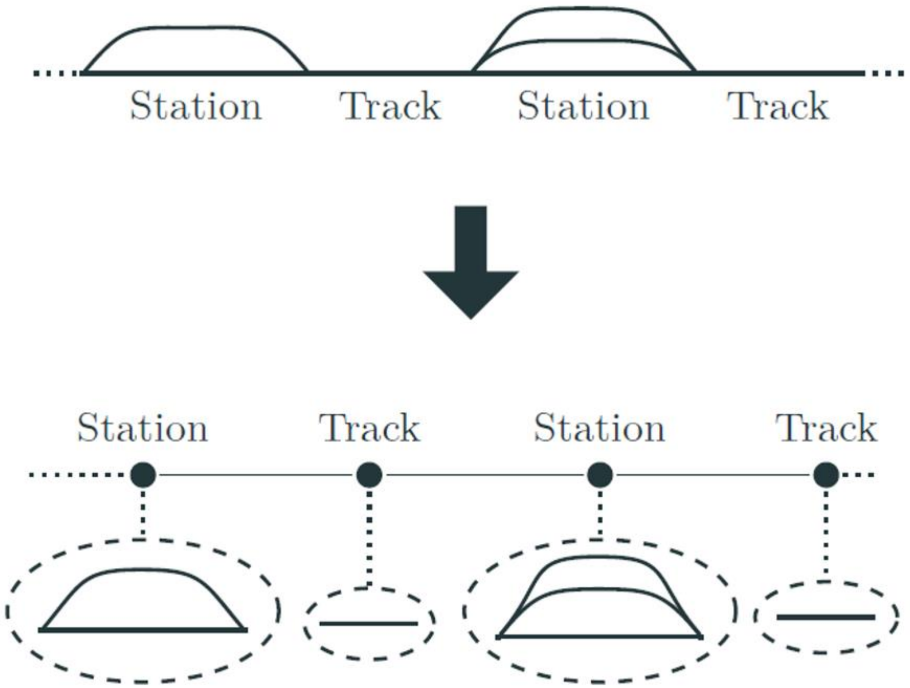
$$
\begin{aligned}
&\min \; c(t) \\
&\text{subject to:} \\
&t_a^r - t^o \geq \Gamma_a, && n_a^r \in N^O \\
&t_a^{r+1} - t_a^r \geq \Lambda_a^r + \Delta_a^r z_a^r, && n_a^r \in N \setminus N^D, r \in R^S \\
&t_a^{r+1} - t_a^r \geq \Lambda_a^r + \bar{\Delta}_a^r z_a^{r-1} + \underline{\Delta}_a^r z_a^{r+1}, && n_a^r \in N \setminus N^D, r \in R^T \\
&t_a^{r+1} - t_a^r \leq \Lambda_a^r + M z_a^r, && n_a^r \in N \setminus N^D, r \in R^S \\
&t_a^{\text{out}} - t_a^r \geq \Lambda_a^r, && n_a^r \in N^D \\
&y_{ba}^r + y_{ab}^r + x_{ab}^r = 1, && \{a,b\} \subseteq A(r), r \in R \\
&t_b^r - t_a^{r+1} \geq -M(1 - y_{ab}^r), && \{a,b\} \subseteq A(r), r \in R \\
&t_a^r - t_b^{r+1} \geq -M(1 - y_{ba}^r), && \{a,b\} \subseteq A(r), r \in R \\
&t_b^{r+1} - t_a^r \geq -M(1 - x_{ab}^r), && \{a,b\} \subseteq A(r), r \in R \\
&t_a^{r+1} - t_b^r \geq -M(1 - x_{ab}^r), && \{a,b\} \subseteq A(r), r \in R \\
&\sum_{\{a,b\} \subseteq Q} x_{ab}^r \leq \binom{|Q|}{2} - 1, && Q \in \mathcal{A}(r), r \in R \\
&y_{ab}^r, y_{ba}^r, x_{ab}^r \in \{0,1\}, && \{a,b\} \subseteq A(r), r \in R \\
&z_a^r \in \{0,1\}, && n_a^r \in N, r \in R^S \\
&t_a^r \in \mathbb{R}, && n_a^r \in N \\
&t_a^{\text{out}} \in \mathbb{R}, && a \in A \\
&t^o \in \mathbb{R}.
\end{aligned}
$$

Heuristics



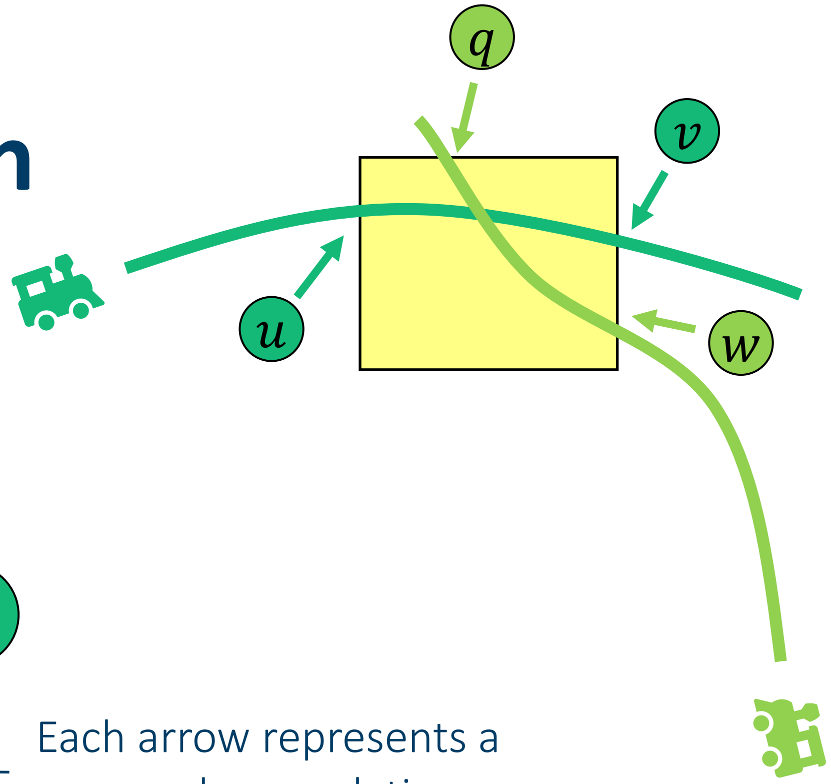
Alternative graph formulation

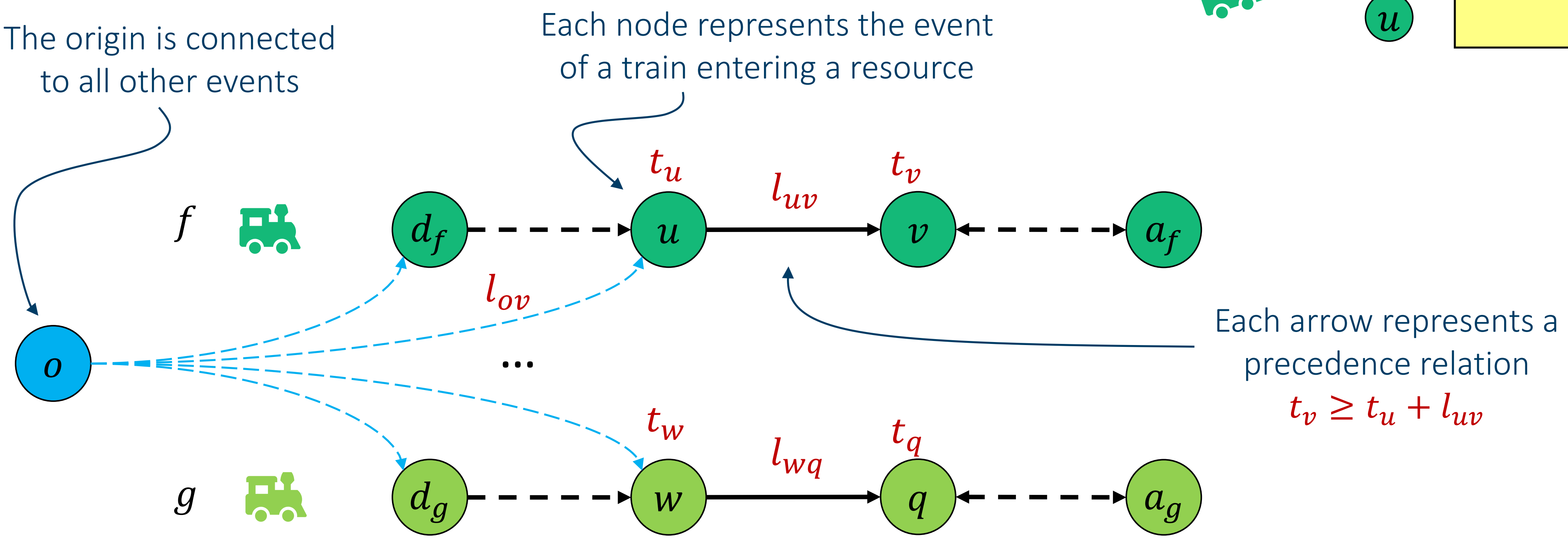
- The network is decomposed in elementary capacitated resources
- The route of a train is a sequence of these resources (alternative routes are possible, not considered in this work)
- It is equivalent to a job-shop scheduling problem with blocking and no-wait constraints *[Mascis and Pacciarelli, 2002]*



Lamorgese L. and Mannino C. *An exact decomposition approach for the real-time train dispatching problem*. Operations Research 63, 1 (2015), 48–64.

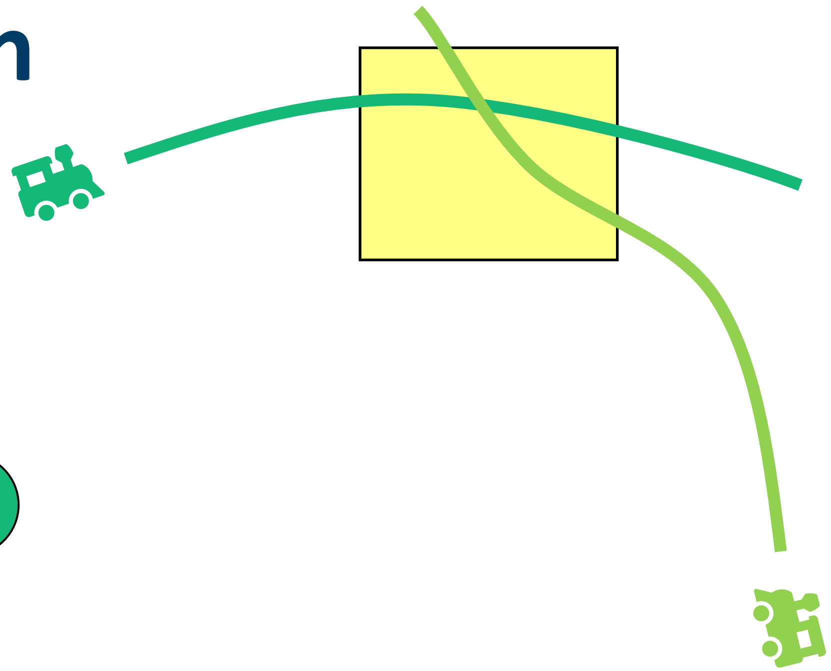
Alternative graph formulation

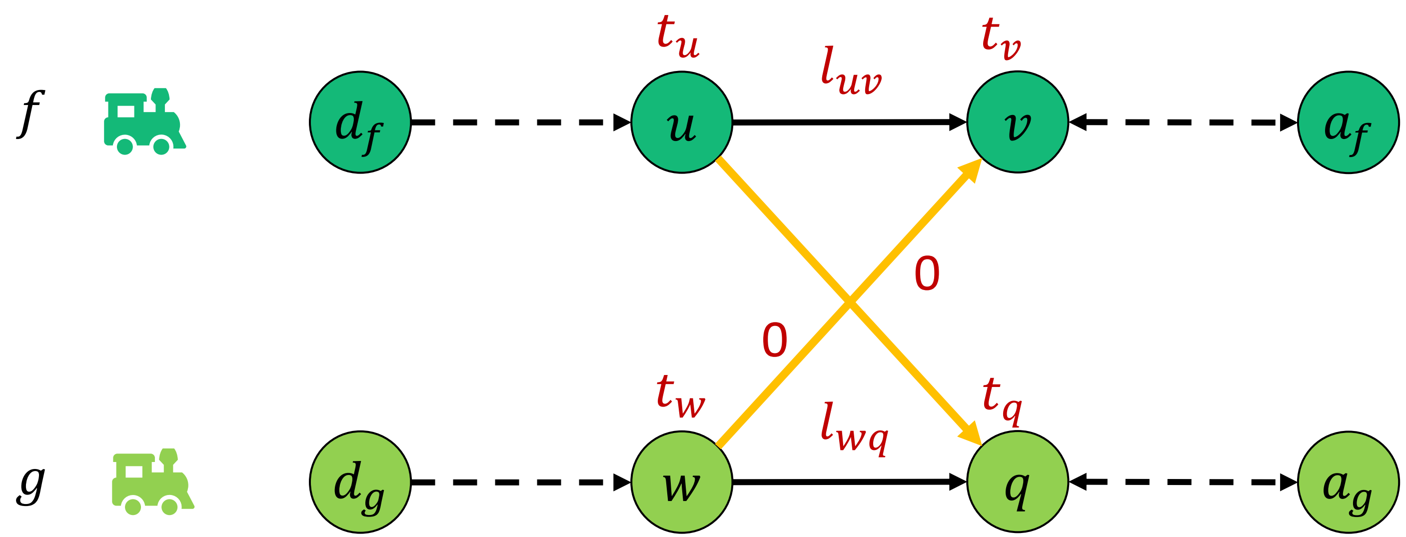




This is the fixed part of the graph, which represent the **speed and time constraints**

Alternative graph formulation



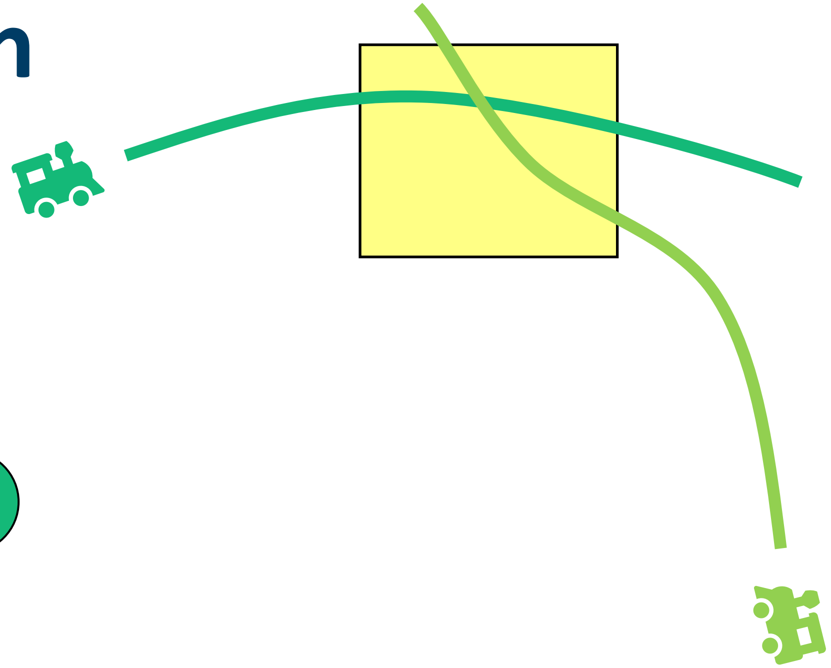


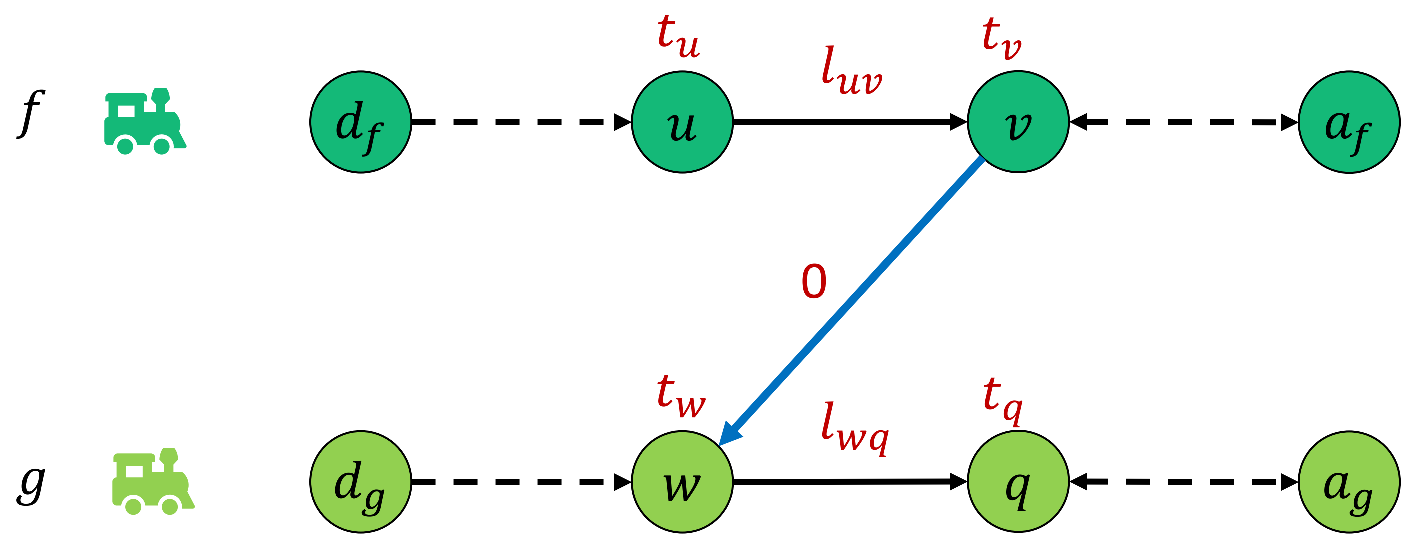
First case: train f and g meet

$t_v \geq t_w$ g enters s before f exits s

$t_q \geq t_u$ f enters s before g exits s

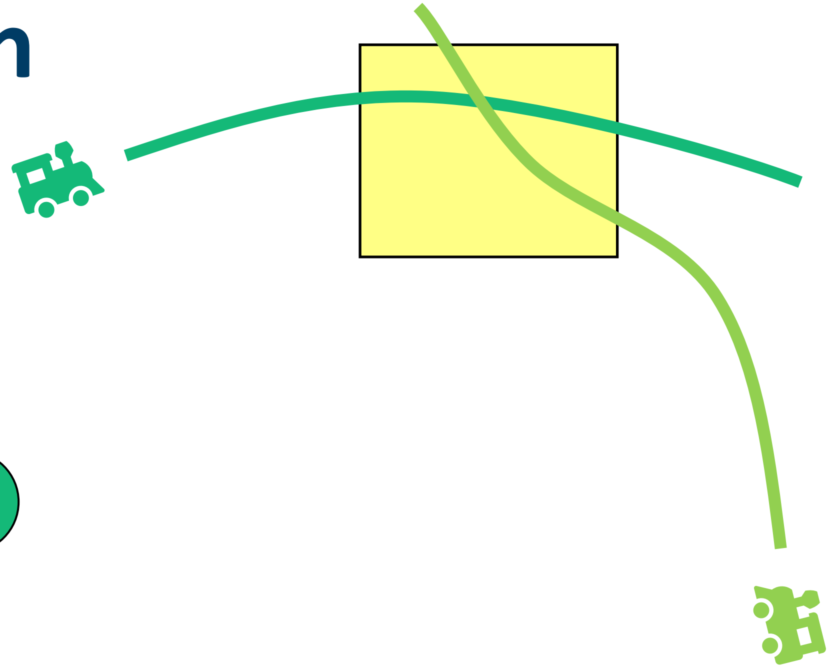
Alternative graph formulation

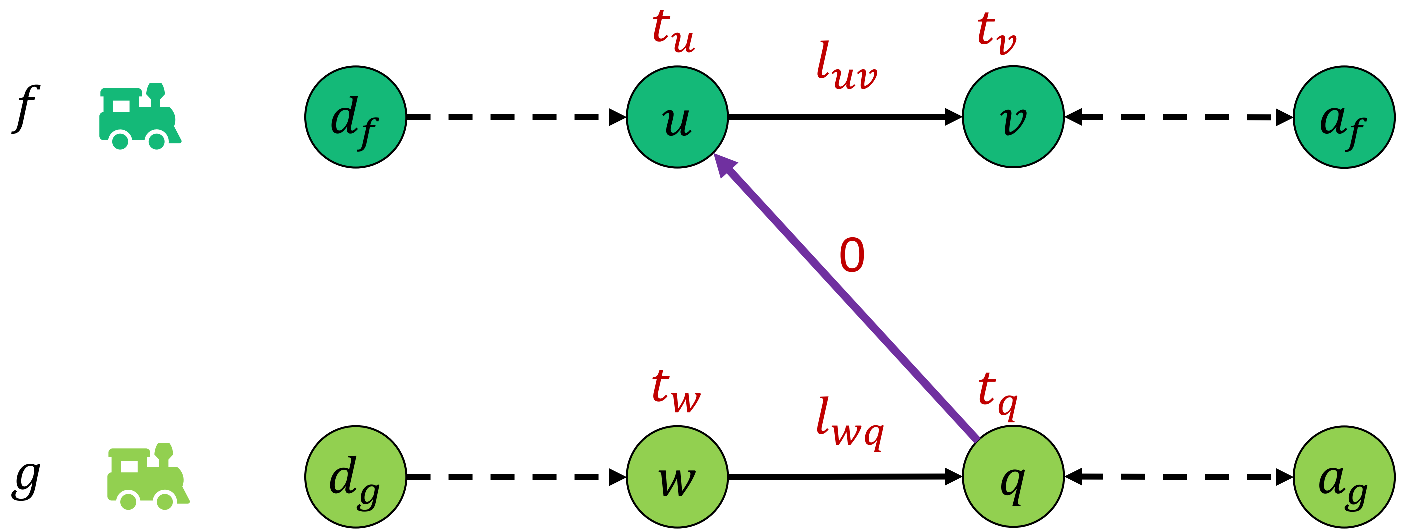




Second case: train f precedes g $t_w \geq t_v$ f exits s before g enters s

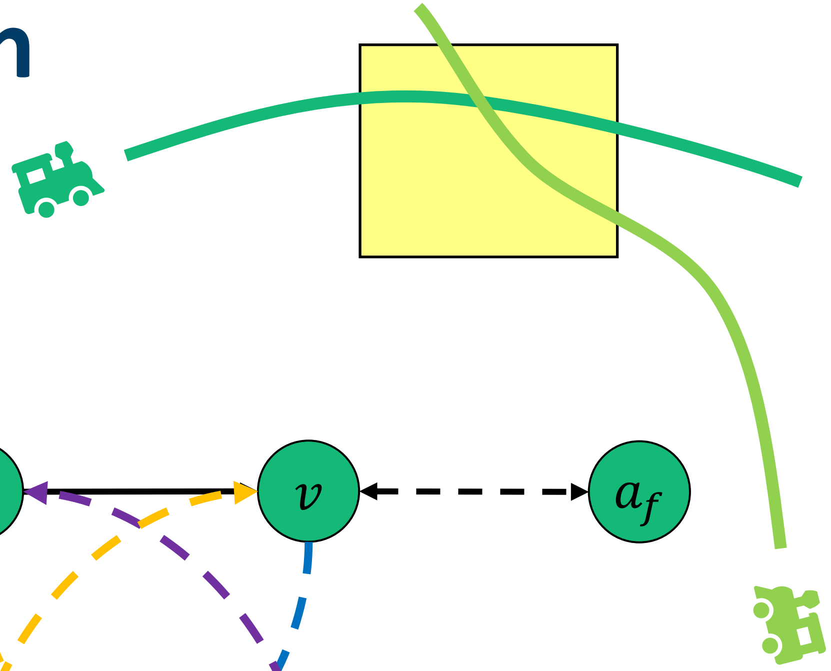
Alternative graph formulation





Third case: train g precedes f $\quad t_u \geq t_q \quad$ g exits s before f enters s

Alternative graph formulation

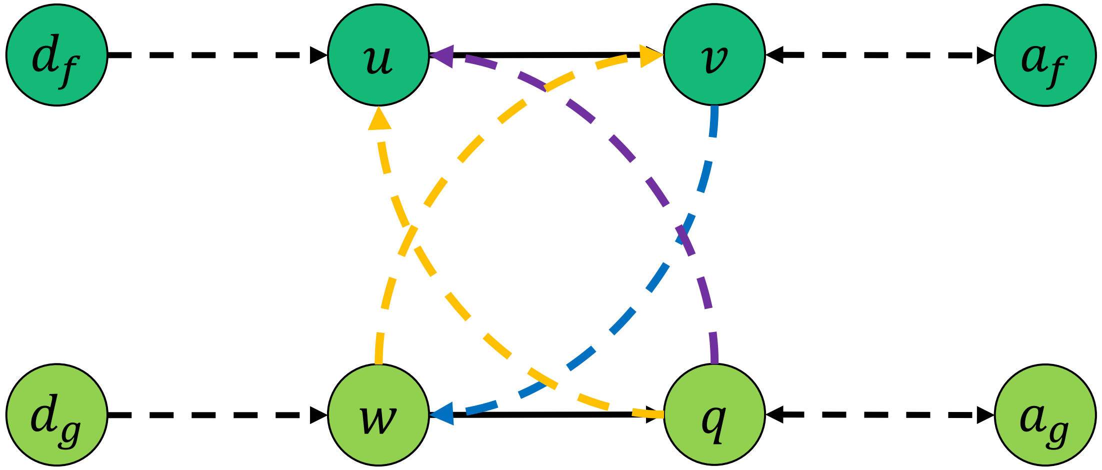


- A feasible solution is a selection of arcs such that:
  - There are no positive cycles
  - The capacity of each resource is satisfied

- For a feasible selection of arcs, the optimal solution lies in the longest path

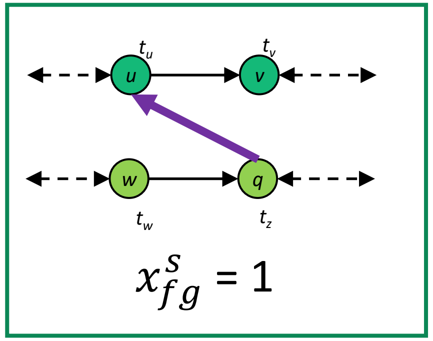
Two formulations:

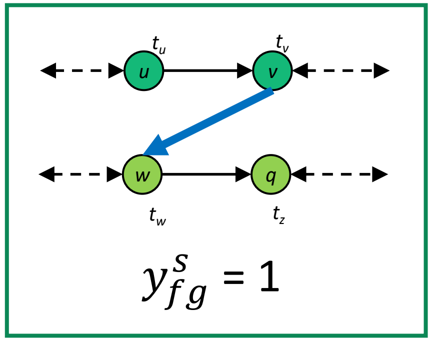
- Big-M
- Path-and-Cycle

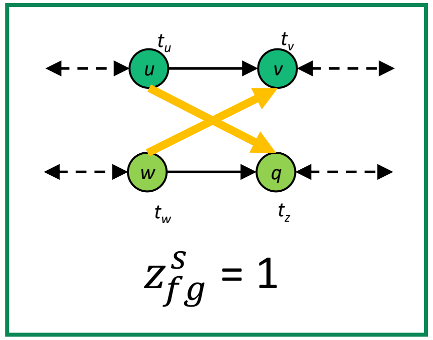


A big-M formulation

- Precedence constraints







$$t_u - t_q \geq M\,(x_{fg}^{s} - 1)$$

$$t_w - t_v \geq M\,(y_{fg}^{s} - 1)$$

$$t_v - t_w \geq M\,(z_{fg}^{s} - 1)$$
$$t_q - t_u \geq M\,(z_{fg}^{s} - 1)$$

- Selection constraints

$$x_{fg}^{s} + y_{fg}^{s} + z_{fg}^{s} = 1$$

- Capacity constraints

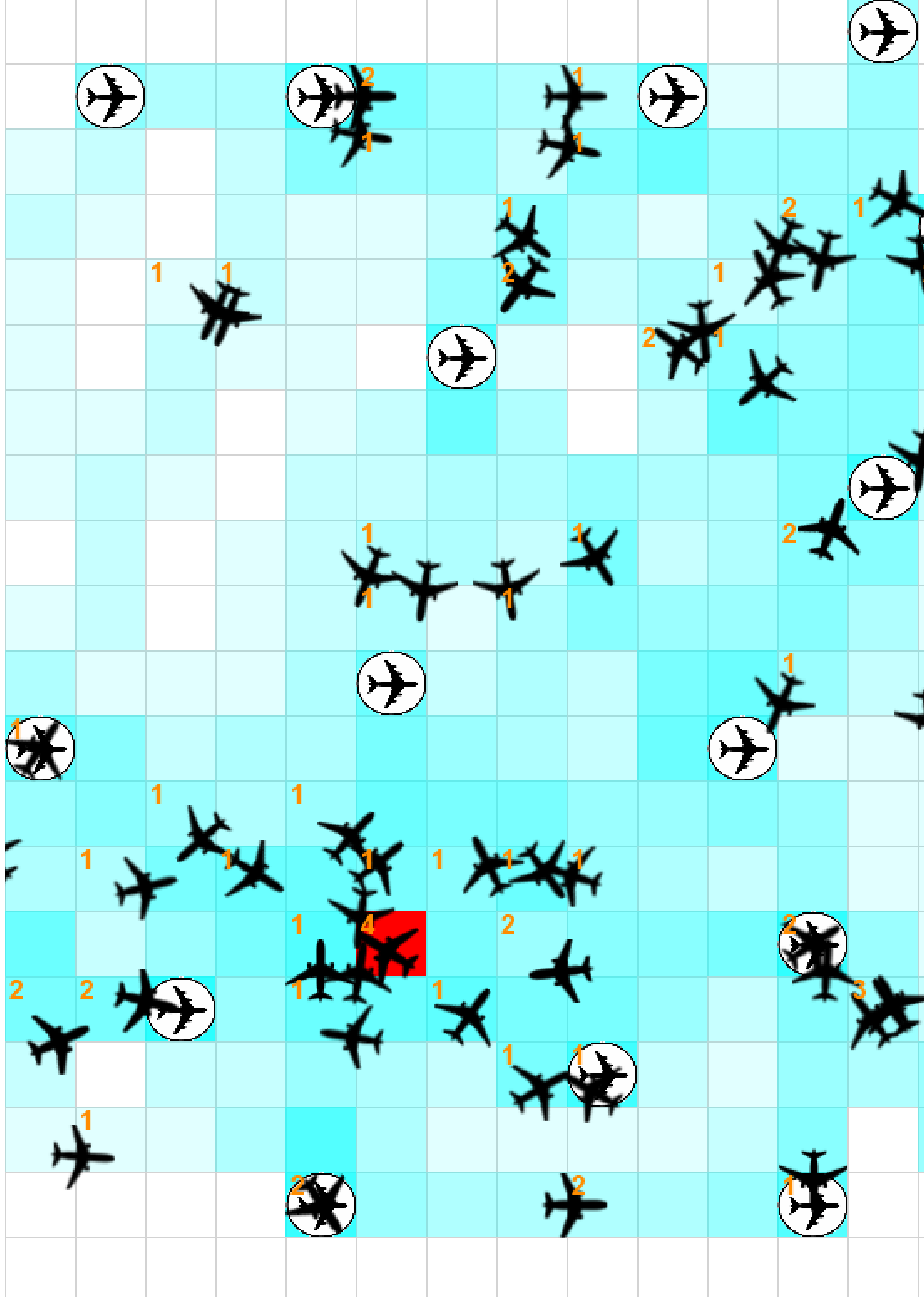
$$\sum_{\{f,g\}\in K} z_{fg}^{s} \leq \binom{C_s + 1}{2} - 1, \quad K \subseteq F, |K| = C_s + 1$$

The path-and-cycle formulation

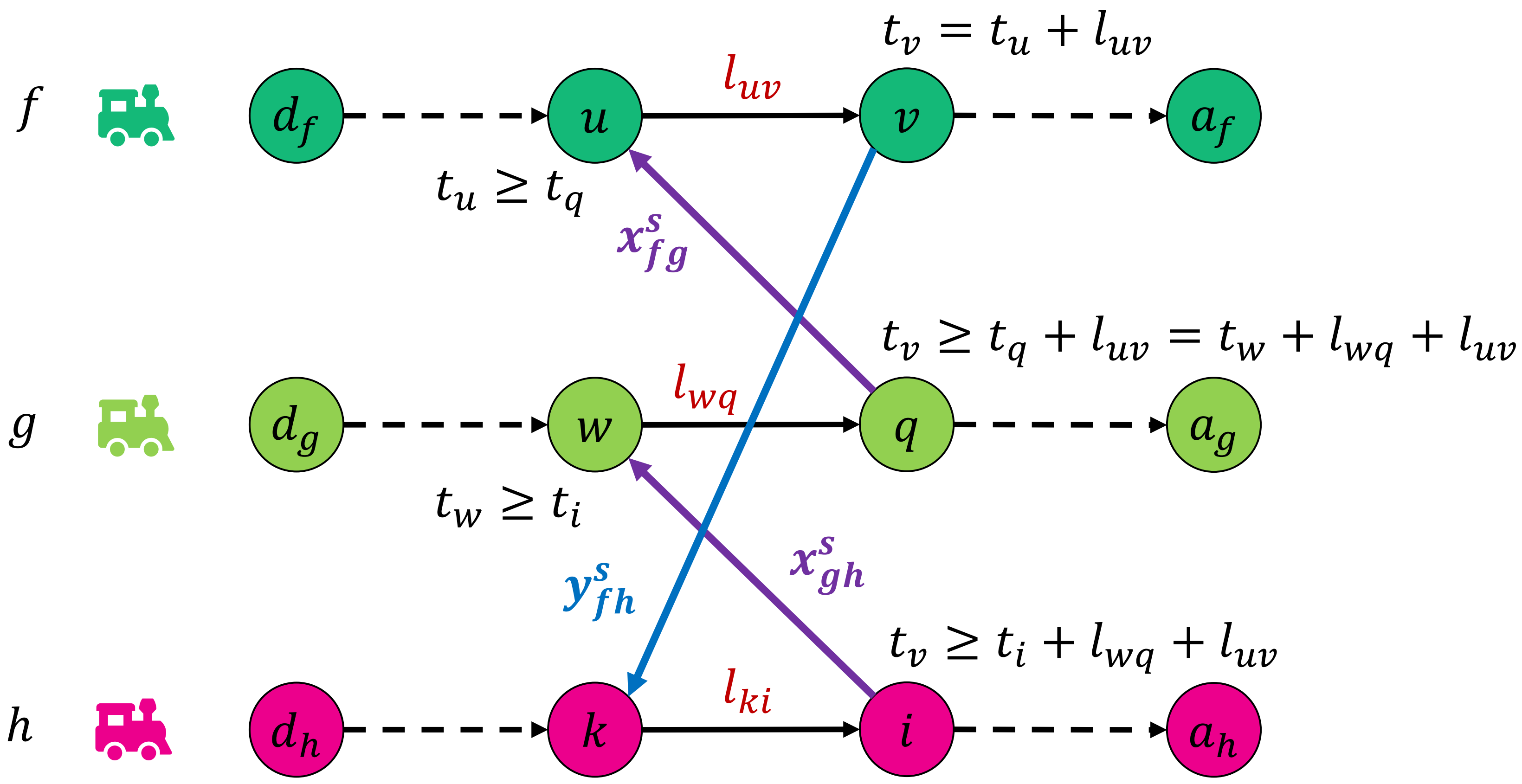
- Obtained from strengthening and lifting the constraints of a Benders' reformulation
- No large coefficients, but non-compact
- Constraints corresponds to basic graph structures: path and cycles

Lamorgese, L. and Mannino, C., 2019. A noncompact formulation for job-shop scheduling problems in traffic management. Operations Research, 67(6), pp.1586-1609.

Sartor, G. and Mannino, C., 2018. The path&cycle formulation for the hotspot problem in air traffic management. ATMOS 2018, August 23–24, 2018, Helsinki, Finland.

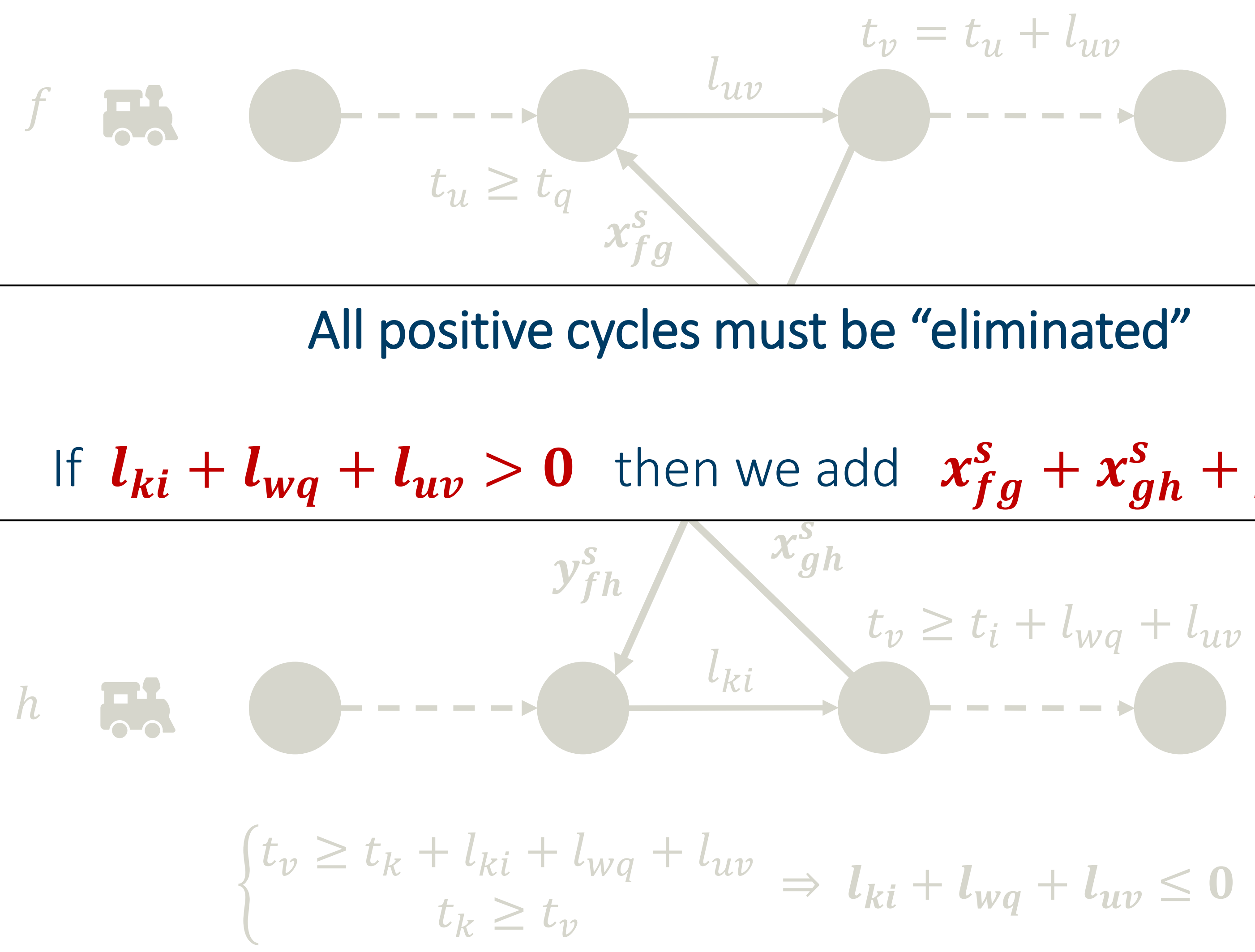


The path-and-cycle formulation



$$\begin{cases} t_v \geq t_k + l_{ki} + l_{wq} + l_{uv} \\ t_k \geq t_v \end{cases} \Rightarrow \boldsymbol{l_{ki} + l_{wq} + l_{uv} \leq 0}$$

The path-and-cycle formulation

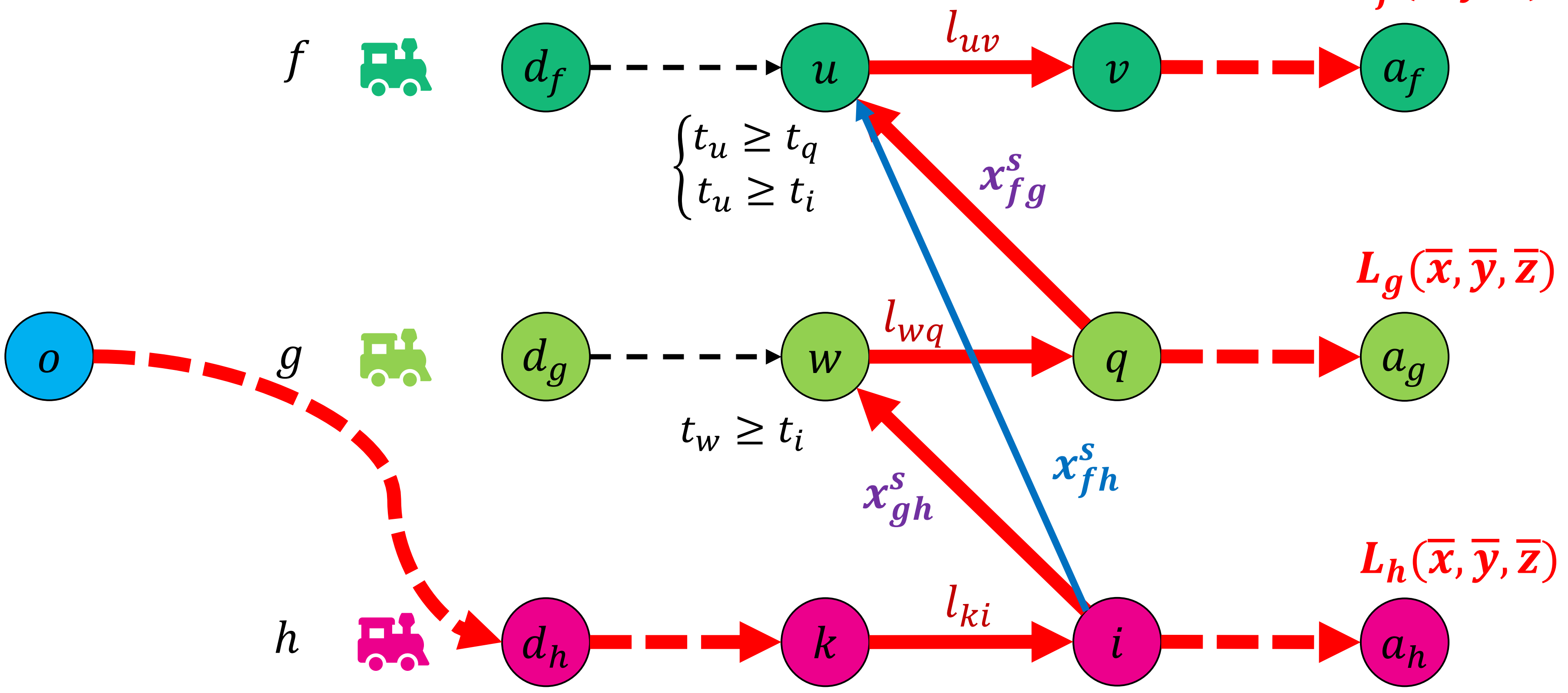


All positive cycles must be "eliminated"

If $\boldsymbol{l_{ki} + l_{wq} + l_{uv} > 0}$ then we add $\boldsymbol{x^s_{fg} + x^s_{gh} + y^s_{fh} \leq 2}$

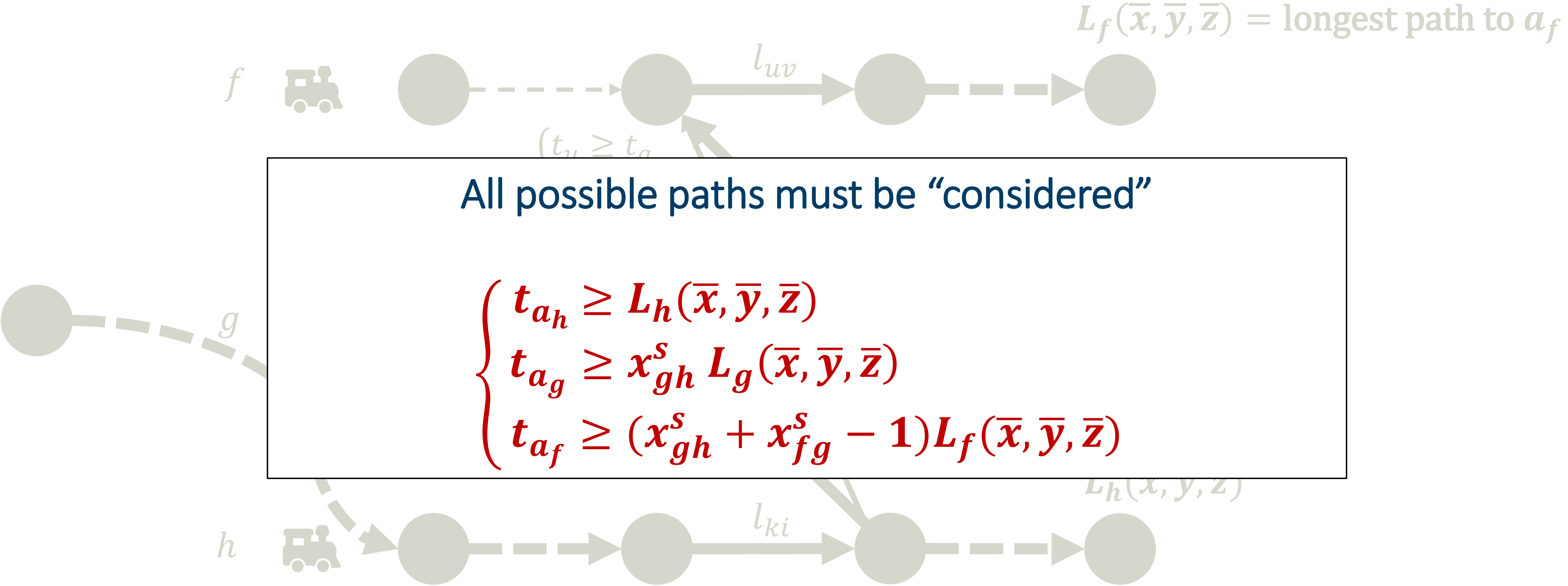
The path-and-cycle formulation

$L_f(\bar{x}, \bar{y}, \bar{z})$ = longest path to a_f



$$\begin{cases} t_i = t_k + l_{ki} \\ t_q = \max\{t_w + l_{wq}, t_i\} \\ t_v = \max\{t_u + l_{uv}, t_q, t_i\} \end{cases} \Rightarrow$$ **Longest path tree rooted in o**

The path-and-cycle formulation



All possible paths must be "considered"

$$\begin{cases} t_{a_h} \geq L_h(\bar{x}, \bar{y}, \bar{z}) \\ t_{a_g} \geq x^s_{gh} \, L_g(\bar{x}, \bar{y}, \bar{z}) \\ t_{a_f} \geq (x^s_{gh} + x^s_{fg} - 1) L_f(\bar{x}, \bar{y}, \bar{z}) \end{cases}$$

$$\begin{cases} t_i = t_k + l_{ki} \\ t_q = \max\{t_w + l_{wq}, t_i\} \\ t_v = \max\{t_u + l_{uv}, t_q, t_i\} \end{cases} \Rightarrow$$ **Longest path tree rooted in o**

The path-and-cycle formulation

- Cycle constraints $$Q\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{z}\end{pmatrix} \leq \boldsymbol{q}$$
- Path constraints $$H_f\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{z}\end{pmatrix} \leq \mu_f \boldsymbol{h_f} - \mathbf{1} \quad f \in F$$
- Capacity constraints $$\sum_{\{f,g\}\in K} z_{fg}^{s} \leq \binom{C_s+1}{2} - 1, \quad K \subseteq F, |K| = C_s + 1$$

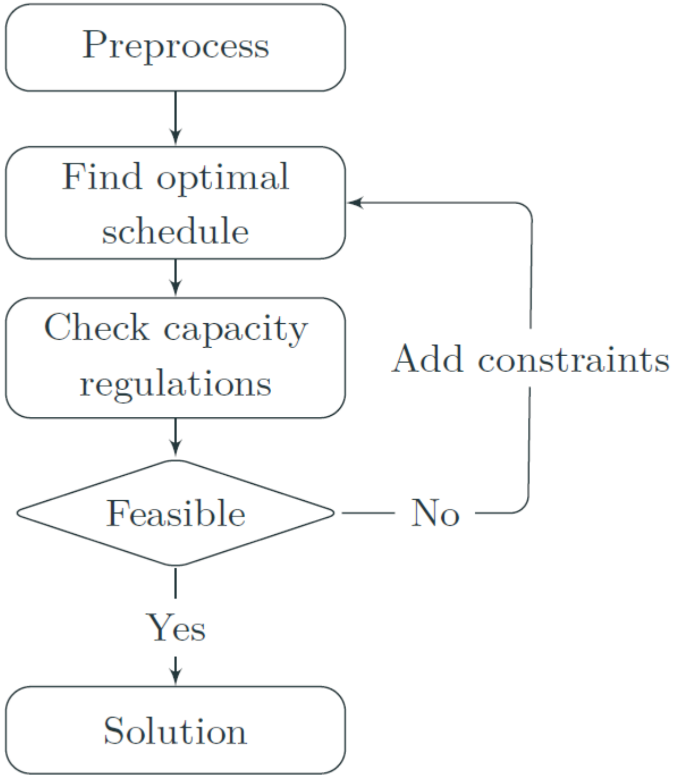
This can also be substituted/extended with disjunctive cuts associated with infeasibility cycle sets

Leutwiler, F. and Corman, F., 2022. A logic-based benders decomposition for microscopic railway timetable planning. European Journal of Operational Research, 303(2), pp.525-540.

An efficient MIP solution method

min	$c(\boldsymbol{t})$	*Minimize some function of the travel time*
s.t.:	$A\boldsymbol{t} \leq b$	*Speed and time constraints*
	$C\boldsymbol{x} + D\boldsymbol{t} \leq e$	*All other operational and disjunctive constraints*
	$\boldsymbol{x} \in \{0,1\},\ \boldsymbol{t} \in \mathbb{R}_+$	

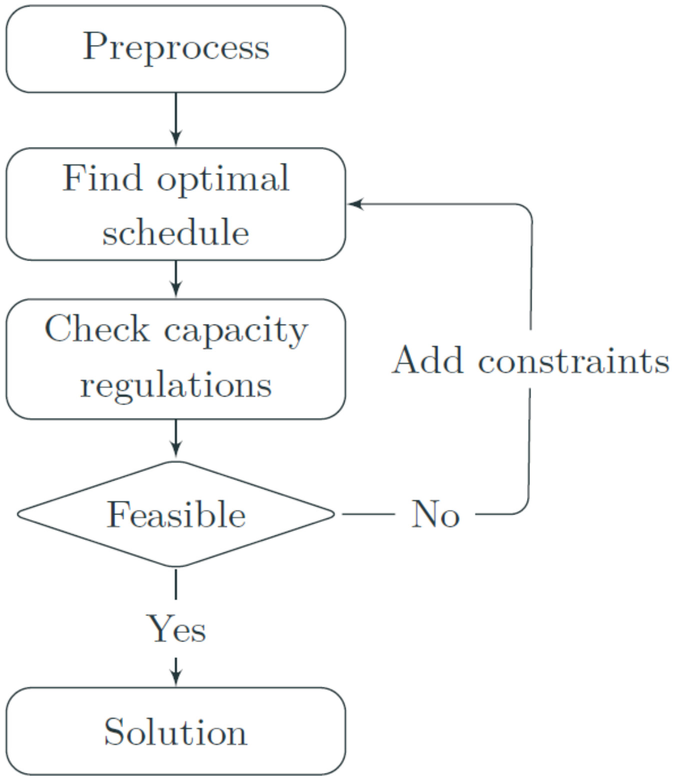
Every variable or constraint is generated on the fly when needed



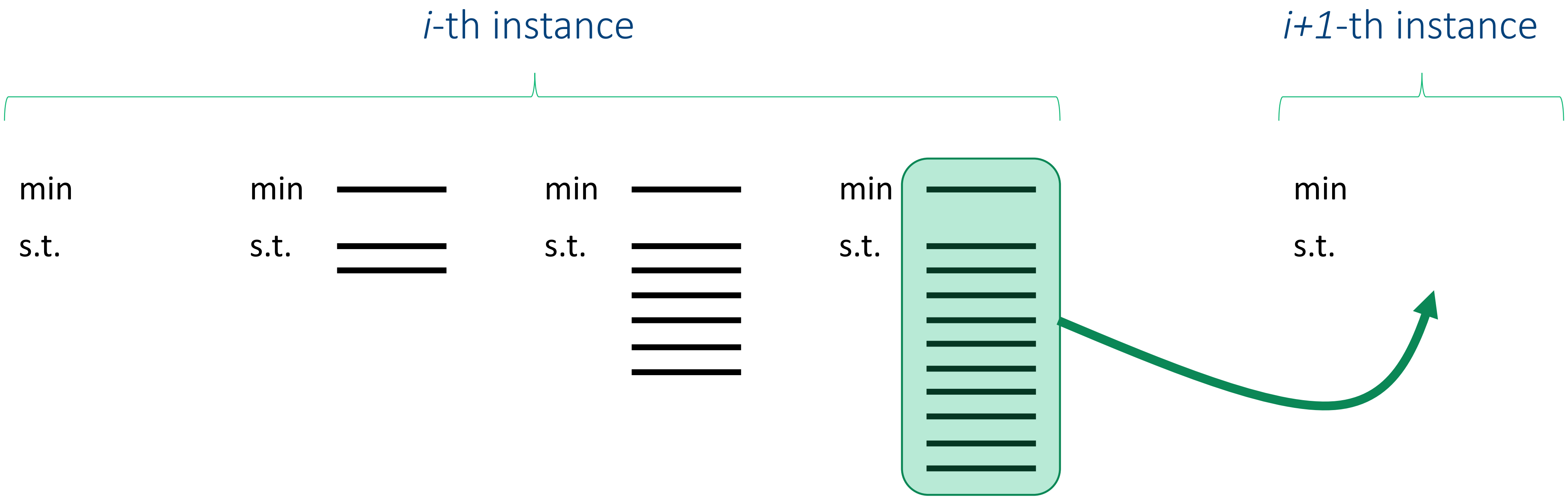
An efficient MIP solution method

- Pros
  - We can use a very fast and efficient MILP solver
  - We consider only the most relevant conflicts
  - We can solve large real-life instance in nearly real-time
  - Relevant conflicts can be learned from history

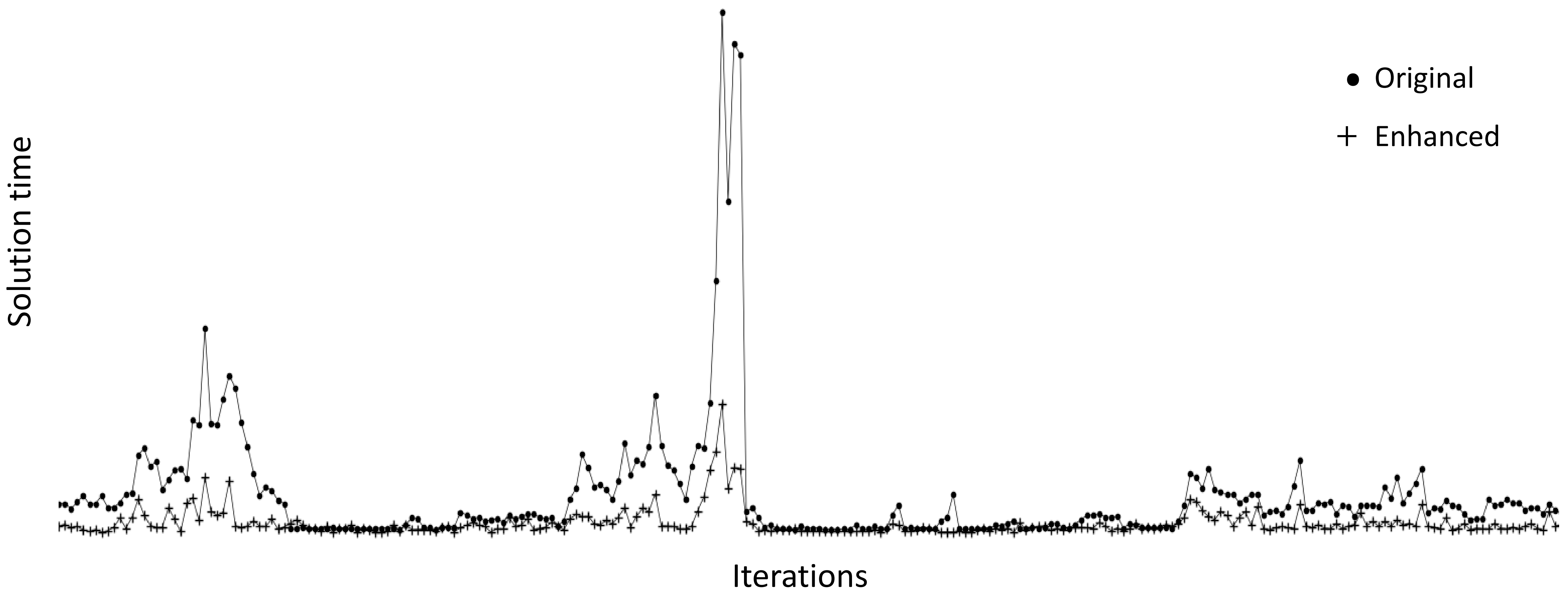
- Cons
  - It does not provide a feasible solution until it finds the optimal one
  - There is no obvious way to easily scale it



An efficient MIP solution method



Reusing parts of the previous polyhedron

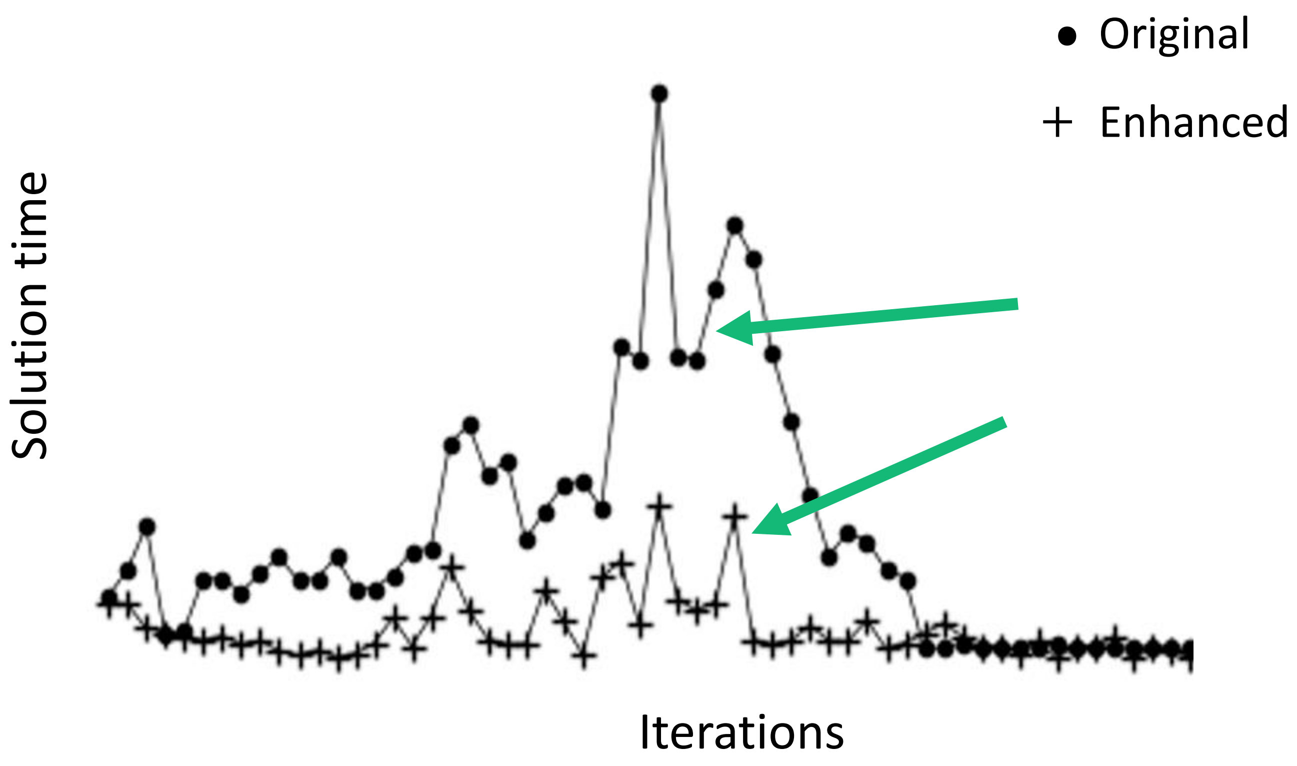


Reusing parts of the previous polyhedron

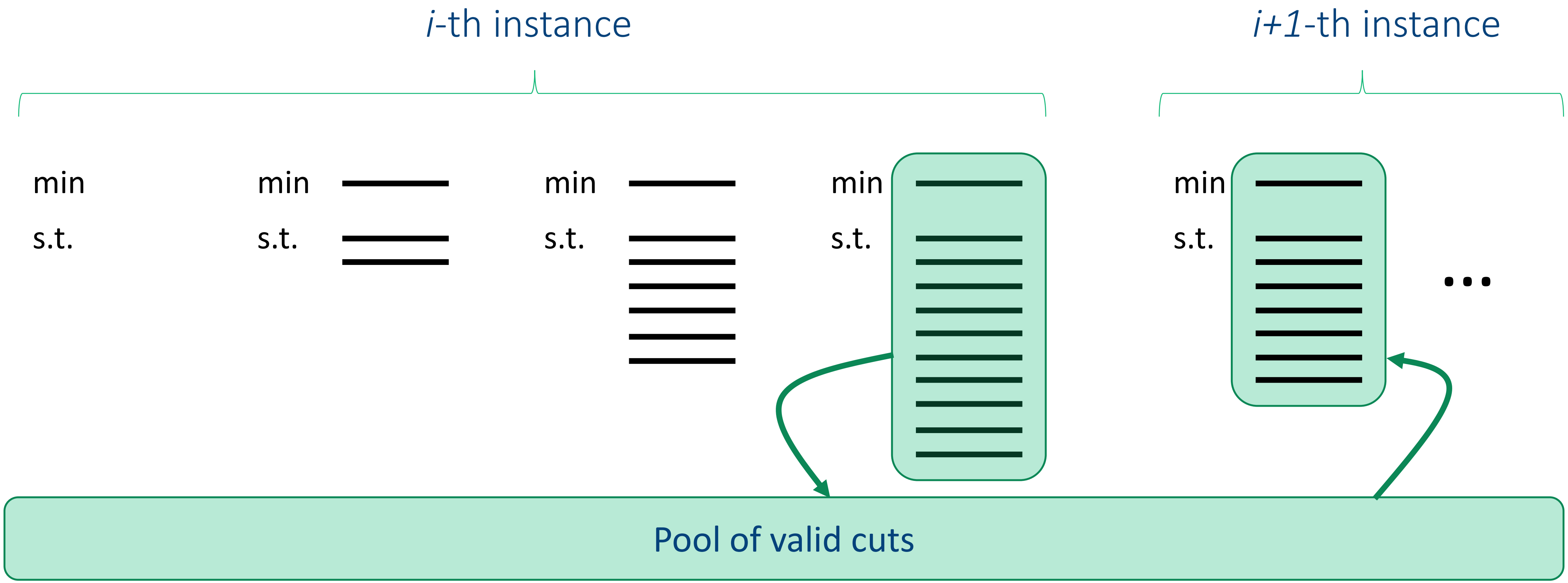
- When new train conflicts appear, the solution time **jumps up** for both formulations

- But at the next iteration, the solution time of the «enhanced» formulation **goes back down**, because now it learnt how to solve those conflicts

- Can we "shave off" the jumps by **learning** valid, relevant conflicts based on the current situation?



Reusing parts of the previous polyhedron



Sartor, G., Mannino, C. and Bach, L., 2019, September. *Combinatorial Learning in Traffic Management*. In International Conference on Machine Learning, Optimization, and Data Science (pp. 384-395). Springer, Cham.

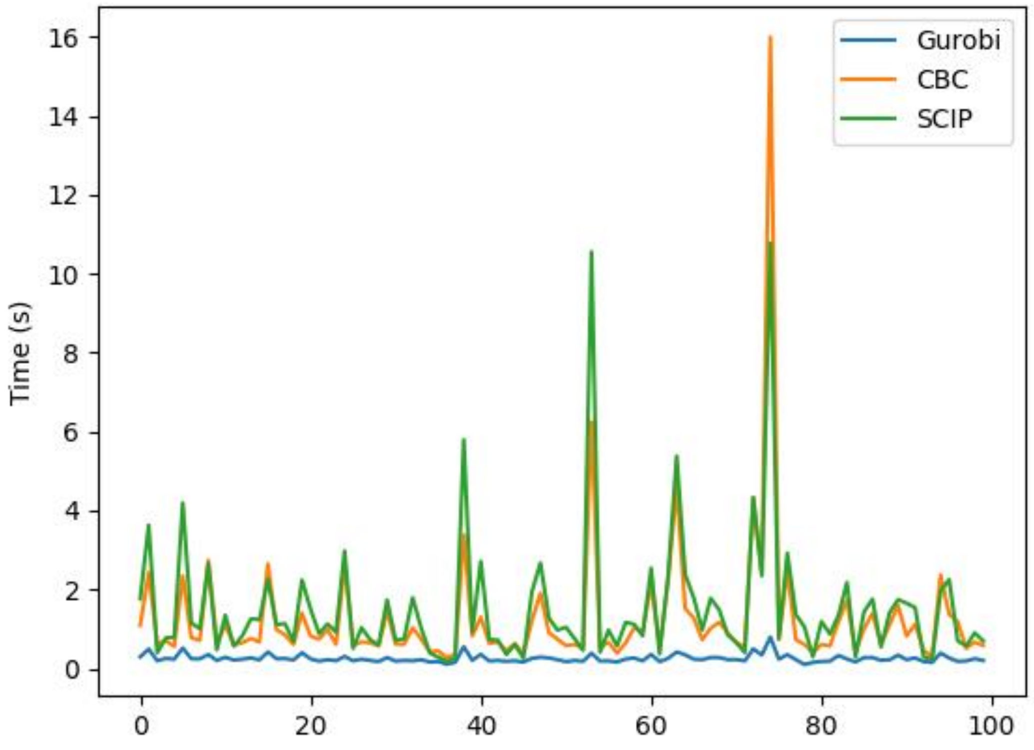
Leutwiler, F., Filella, G.B. and Corman, F., 2023. Accelerating logic-based Benders' decomposition for railway rescheduling by exploiting similarities in delays. Computers & Operations Research, 150, p.106075.

Why do we need Gurobi?

Gurobi	CBC	SCIP
0.3 s ± 0.1	1.3 s ± 1.6	1.5 s ± 1.3

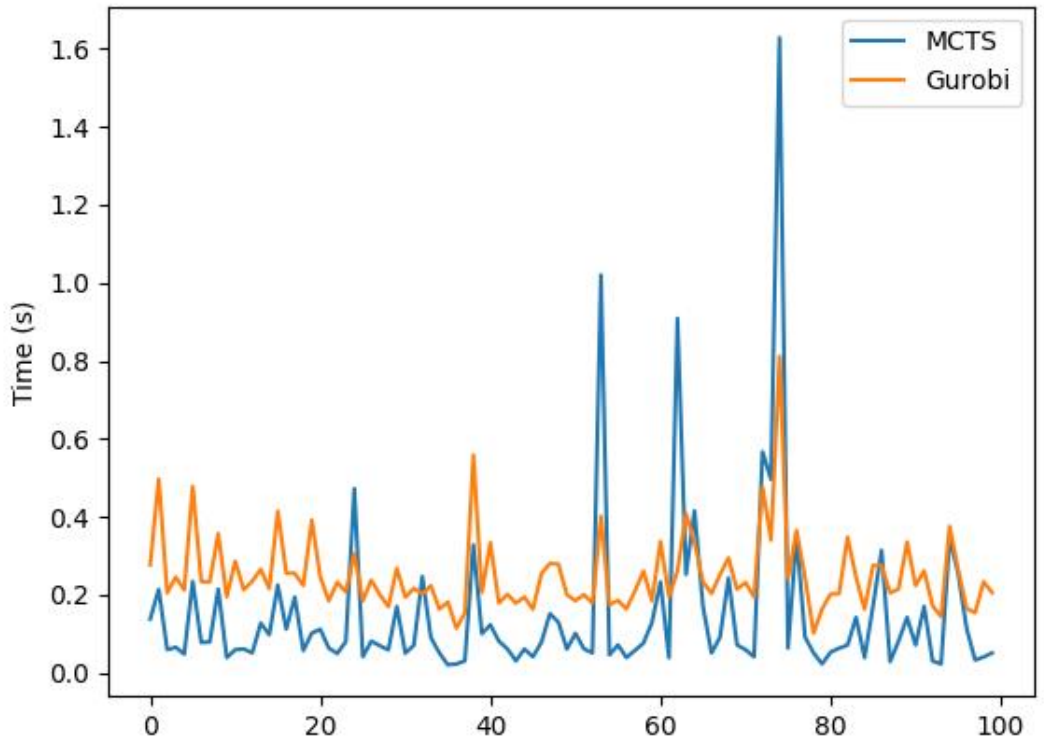
- Gurobi is about **5 times** faster than CBC or SCIP in small, but realistic dispatching instances

- Gurobi is also much more "stable"



But do we really need Gurobi?

- SINTEF has developed a custom Monte Carlo Tree Search algorithm that outperforms Gurobi branch and bound default algorithm
- Each node of the tree corresponds to a selection of arcs, and all graph operations are done efficiently using a custom longest-path algorithm with lazy path augmentations
- Easier to scale and customize compared to using off-the-shelf optimization solvers



- But yeah, we may still need Gurobi… ☺

Dal Sasso, V., Lamorgese, L., Mannino, C., Onofri, A. and Ventura, P., 2021. The tick formulation for deadlock detection and avoidance in railways traffic control. Journal of Rail Transport Planning & Management, 17, p.100239.

Croella, A.L., Dal Sasso, V., Lamorgese, L., Mannino, C. and Ventura, P., 2022. Disruption management in railway systems by safe place assignment. Transportation science, 56(4), pp.938-952.

Challenges ahead

1. How to improve models and theory?
   - Decomposition techniques
   - Reformulations
   - Valid cuts

2. How to incorporate Machine Learning?
   - Predict the importance of each potential train conflict
   - Predict the best way to resolve a train conflict

3. How to conquer industrial applications?
   - Deal with large infrastructures and dynamic input data
   - Deal with complex constraints and objectives

4. How to foster international collaborations?
   - You'll see in a second…

Part FOUR

SINTEF

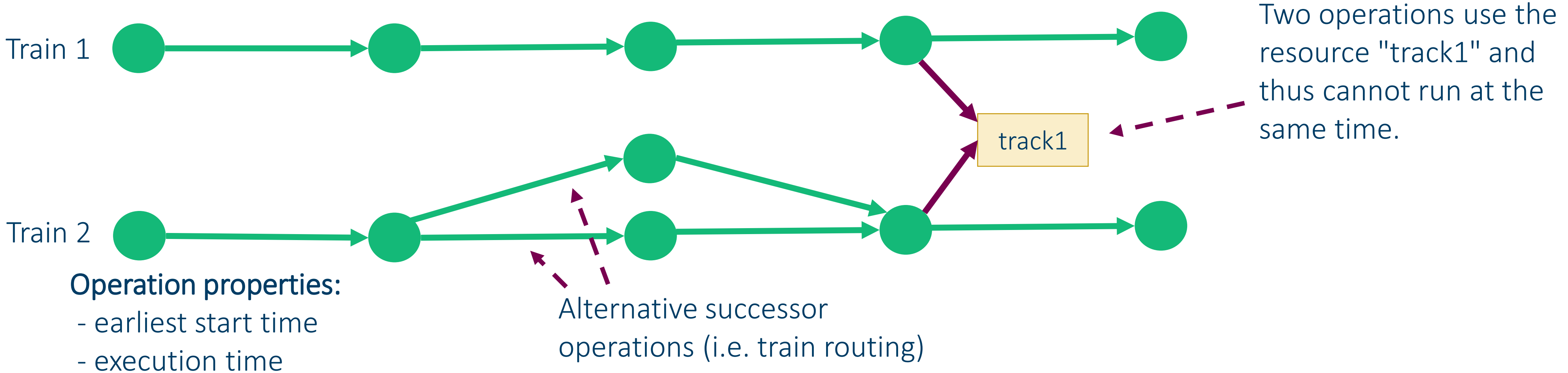
The DISPLIB library

- **Which dispatching algorithm is the best one?**
- Most academic papers use one-off data sources and a small number of instances with special characteristics.
- Why don't we have a benchmark set?
  (like MIPLIB, PDDL, SAT competitions, CVRPLIB)

- In response to the need of researchers for access to a realistic, diverse, and comprehensive set of instances, SINTEF is creating the **DISPLIB**!
- ...and arranging a **competition**!

DISPLIB problem definition

- Each node is a train operation with resource usage, e.g., a train traversing a section of the railway.
- The route of a train is a set of nodes forming a path through the graph of operations
- The schedule of a train is given by assigning start and end time to each operation of the selected route.
- A schedule is feasible if the exclusive access given to the resources of each operation is not violated.
- **Objective:** select operations and schedule them to minimize sum of delays.
- Formalized into a JSON file format (documentation available soon!)



Operation properties:
- earliest start time
- execution time
- due time

DISPLIB competition

- Currently collecting problem instances from researchers around the world. If you want to contribute, **get in touch with us**!

giorgio.sartor@sintef.no

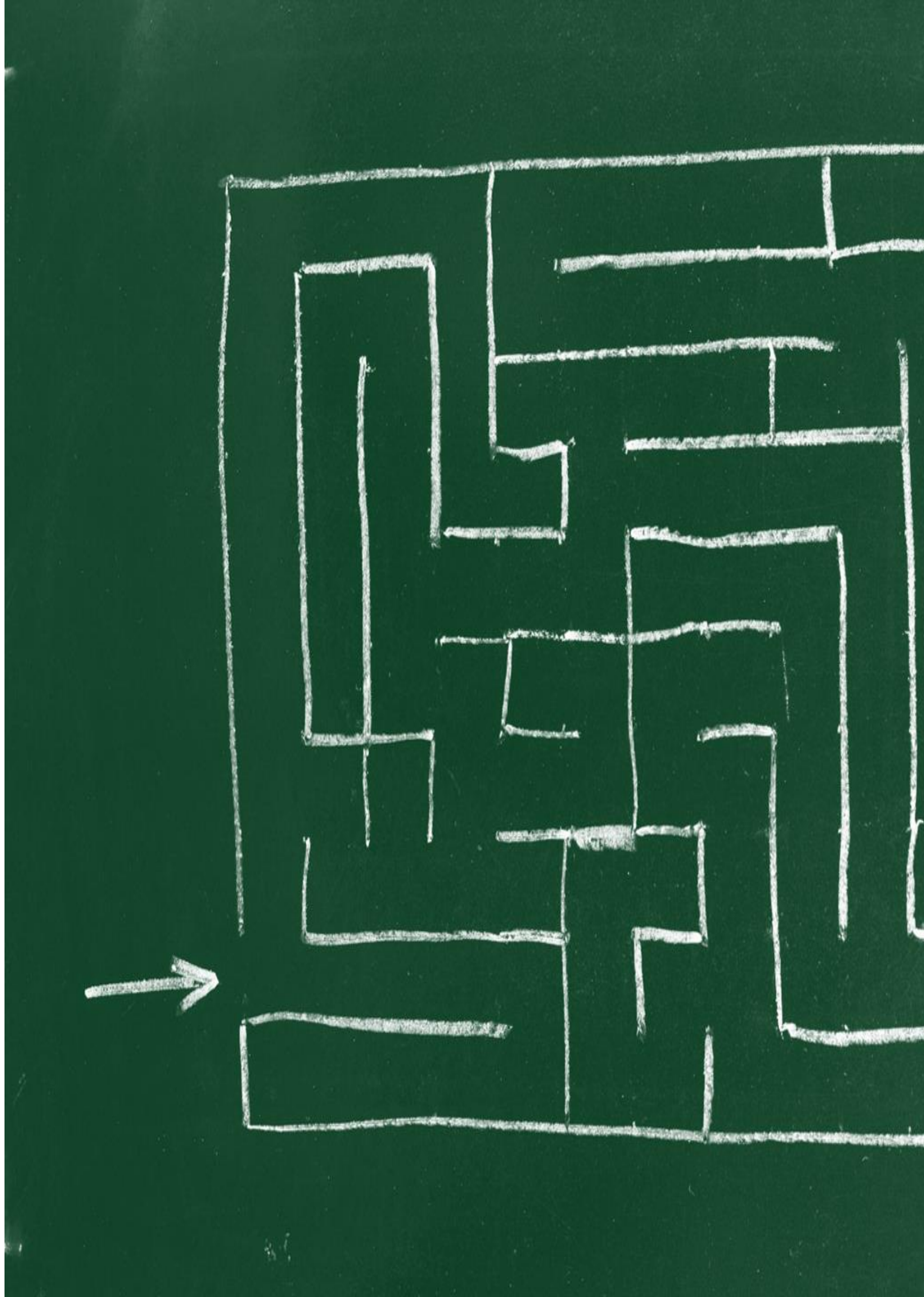
bjornar.luteberget@sintef.no

- Competition will run in Sep 2024 – May 2025, official announcement at ODS 2024.

- The authors of the three best algorithms will be invited to present at ODS 2025.

- Anyone can participate – **put your problem-solving skills to the test on an important real-world problem!**

https://displib.github.io/



SINTEF

Technology for a better society