Solving train scheduling problems: reformulations, decompositions, and practice.

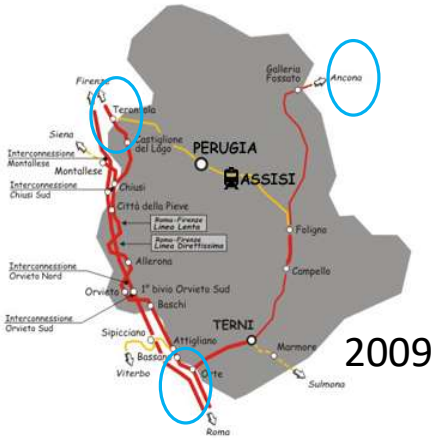
Carlo Mannino
Dept. of mathematics and cybernetics, SINTEF
Siemens Mobility

Aussois, January 2025

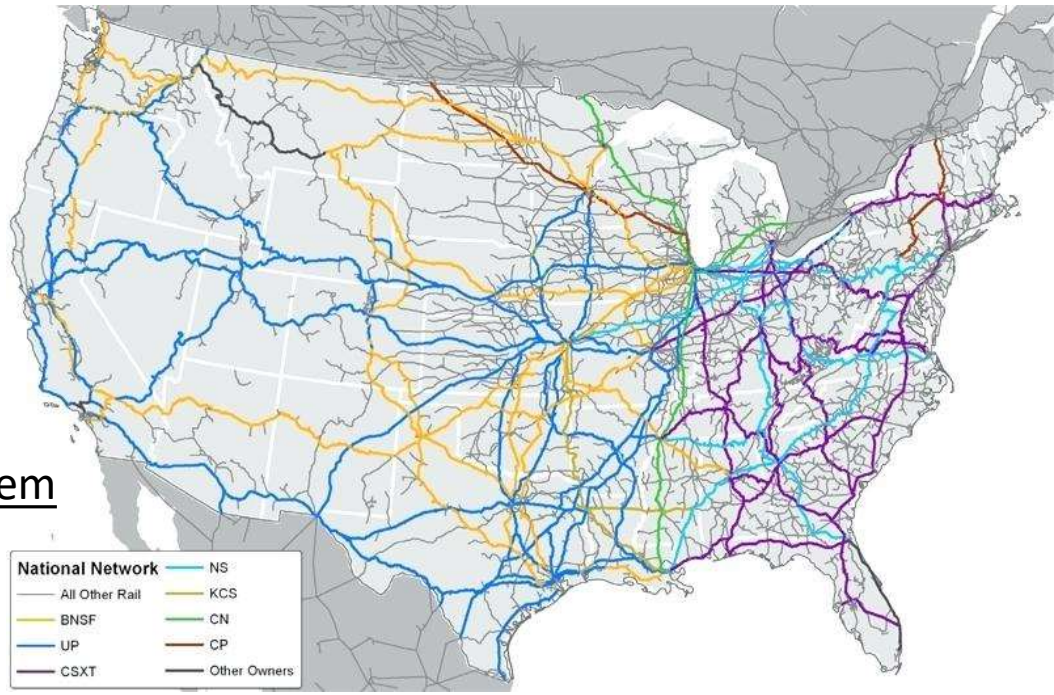
Photo: David Gubler

Technology for a better society

The challenge: dispatching in large networks



2009: REGIONAL LINES IN ITALY



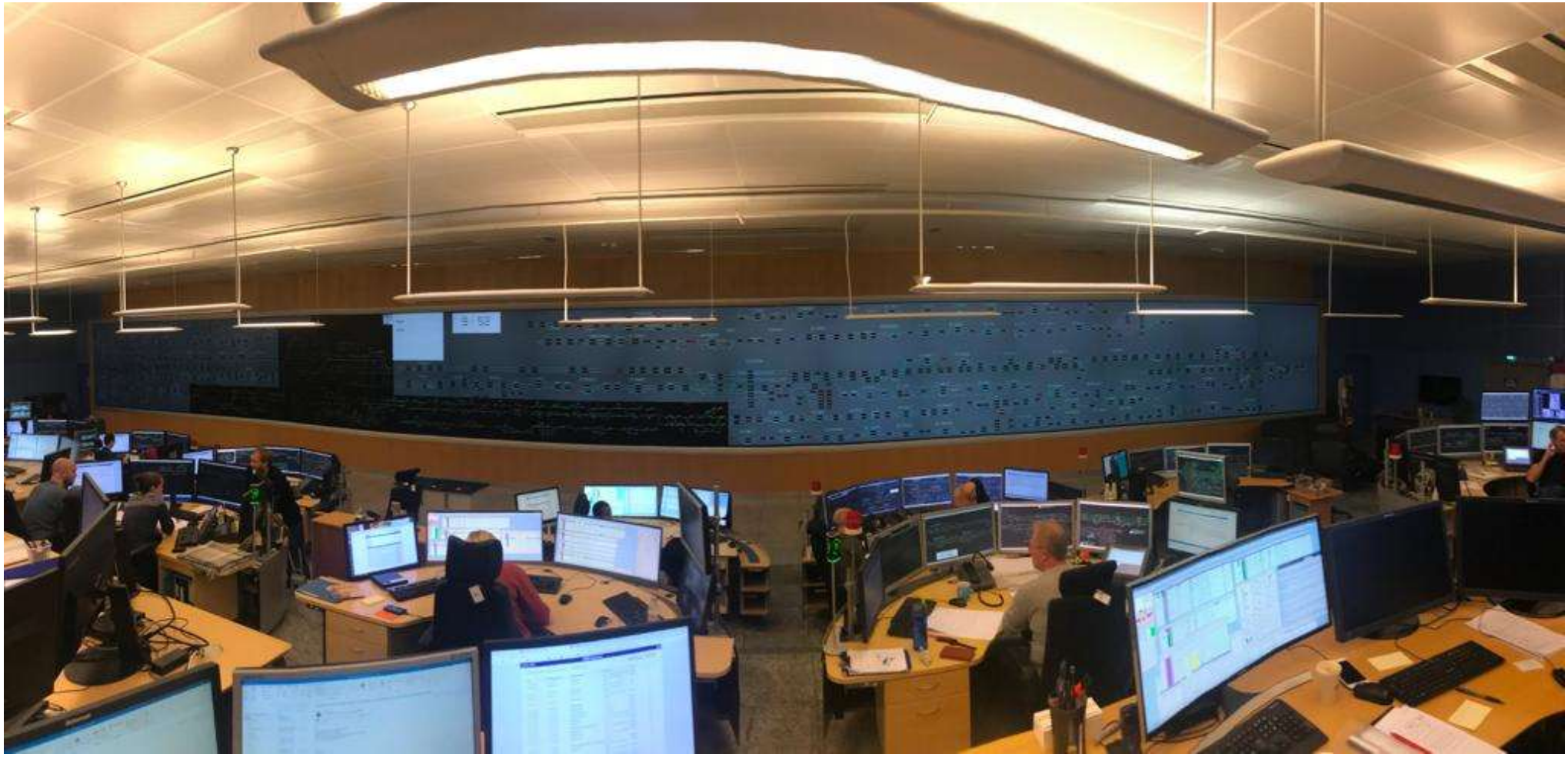
In 2017 we were asked to design and develop a system to dispatch trains in Union Pacific railway network.

UP: Largest railroad in North America
32k miles, 23 states, 8.3k locomotives

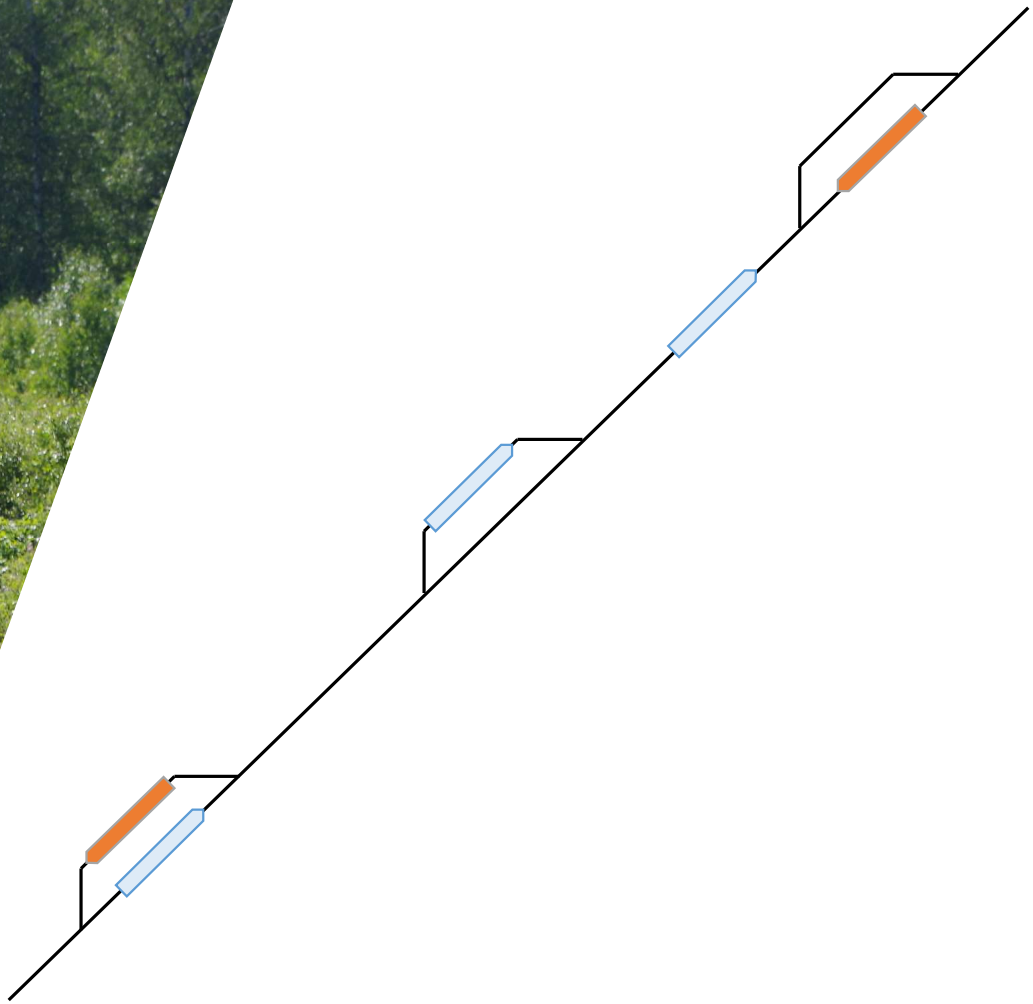
Union Pacific: blue network.

About 1000 trains running at any time any day of the year

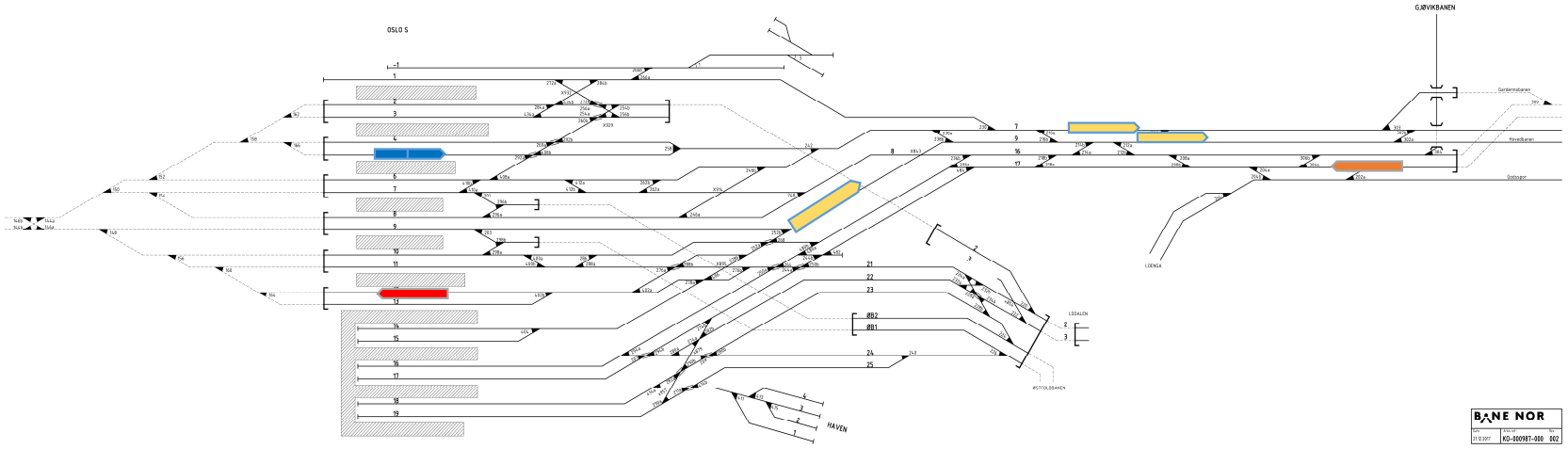
Dispatchers at work




SINTEF
ATC
BALISER
81-7
Hil
L4



Train routing and scheduling (TRS)

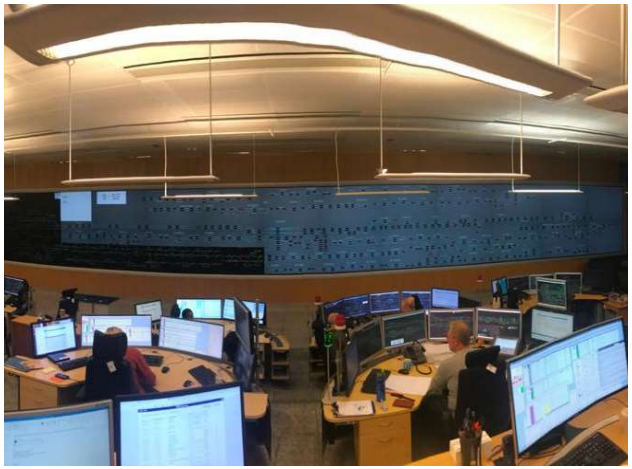


Roughly, **the train routing and scheduling problem** (**TRS**):

- **Given**: railway network, set of trains, traffic and time requirements, connections, …
- **Find**: a route for each train and a schedule of the movement of each train along its route which minimizes the deviation from requirements

Versions of TRS

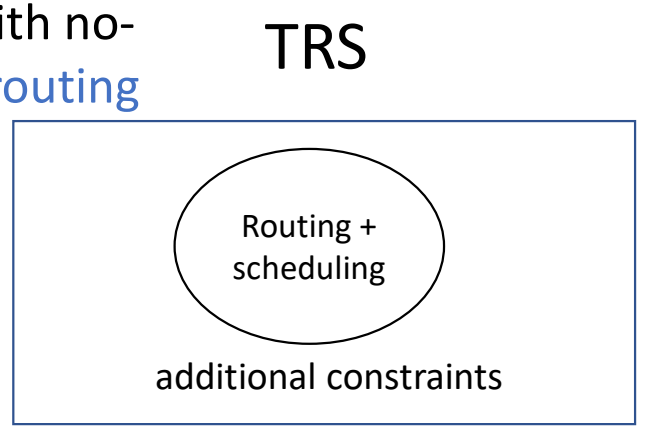
Operational (real-time)
Train dispatching (rescheduling)



Tactical/Strategical
train timetabling

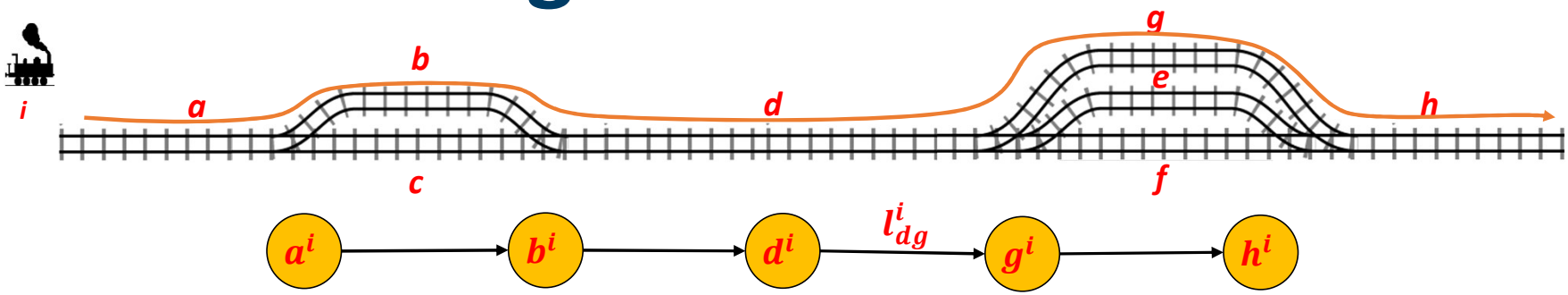
TRS is difficult

- TRS: Very hard to solve practical instances
  1. The core is a job-shop scheduling problem (NP-hard) with no-wait blocking constraints (Mascis & Pacciarelli 2002) + routing
  2. In practice, additional rules and constraints
  3. Very large practical instances
  4. Short computing time available (e.g. dispatching: 10-30 sec)

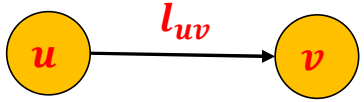


Modelling train movements

Modelling train movements

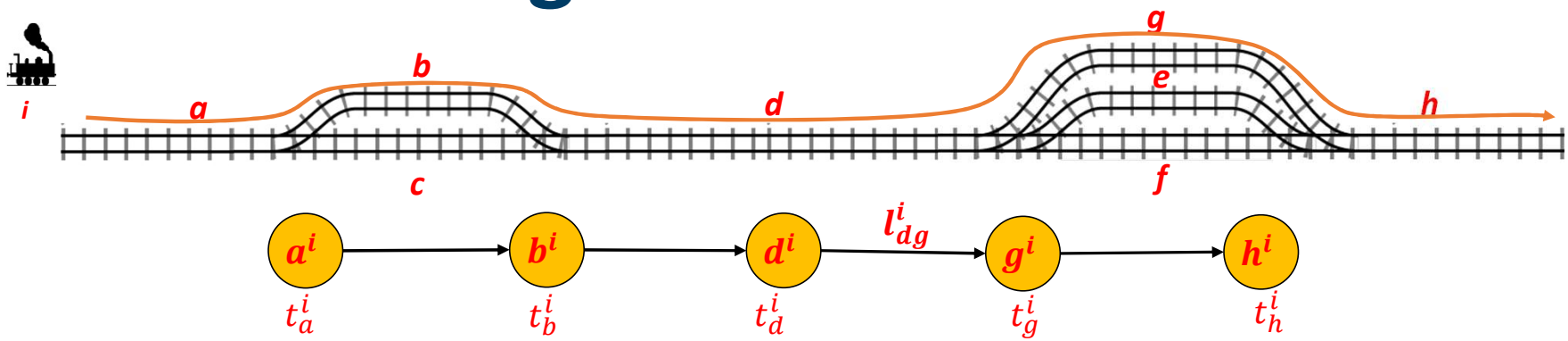


- Tracks are segmented into *track sections* (each can accommodate at most one train)
- The train movement is a sequence of track sections, represented by a **directed path** (graph)
- A node corresponds to (the event of) entering **a train *entering a track section***
- An arc (u, v) represents that v is the track section following u



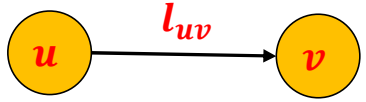
- The length l_{uv} of (u, v) is the miniming running time of the train from u to v

Modelling train movements

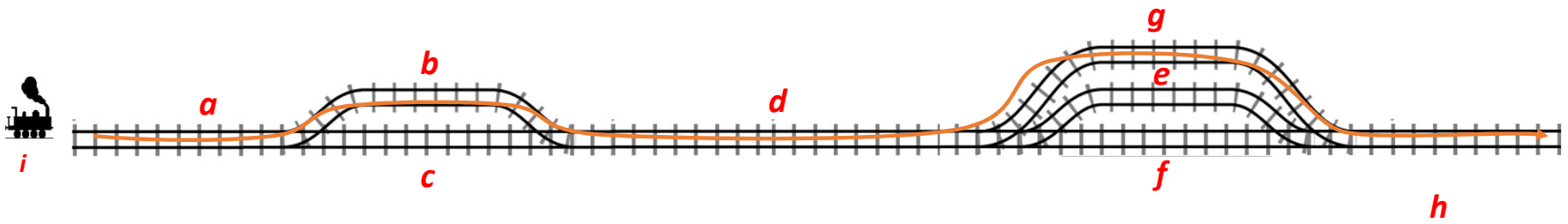


- A variable t_u is associated with node u representing **the time the train enters the track section**
- Arc (u, v) with length l_{uv} implies a time precedence constraint!

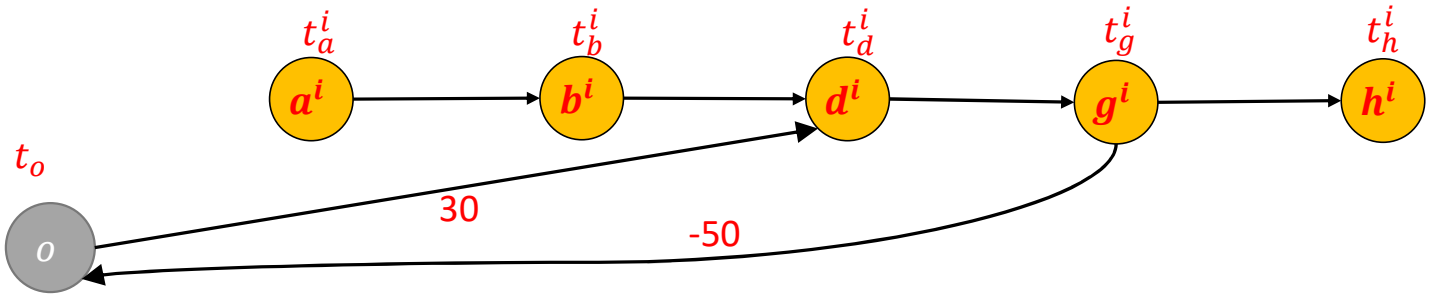
$$t_v - t_u \geq l_{uv}$$



The time origin

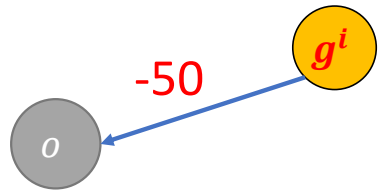


- We have an extra node o representing the start t_o of the planning horizon

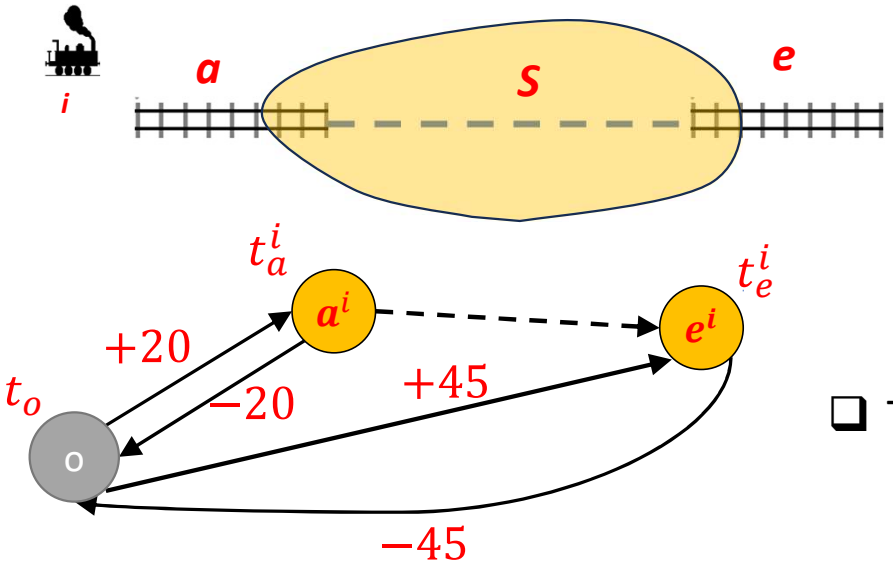


- Deparure from b not before 30: Event d^i starting not before 30: $\boldsymbol{t_d^i - t_o \geq 30}$

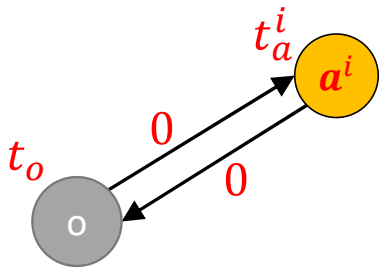
- Arrival in g not after 50: $t_g^i - t_o \leq 50 \rightarrow \boldsymbol{t_o - t_g^i \geq -50}$



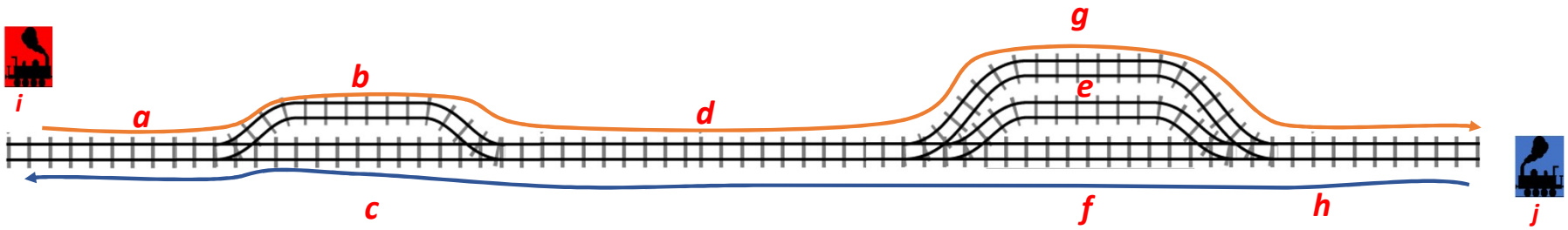
Timetable arcs



- A timetable provides arrival (departure) times of trains in certain subnetworks.
- Represented in the TRS graph by a pair of antiparallel arcs of opposite lengths.
- Train i arrives in station S at time 20 ($t_a^i - t_o = 20$)
  $t_a^i - t_o \geq 20$ and $t_a^i - t_o \leq 20$
- Train i leaves S at time 45 ($t_e^i - t_o = 45$)
- OBS: Train i already in section a (starting now): $t_a^i - t_o^i = 0$



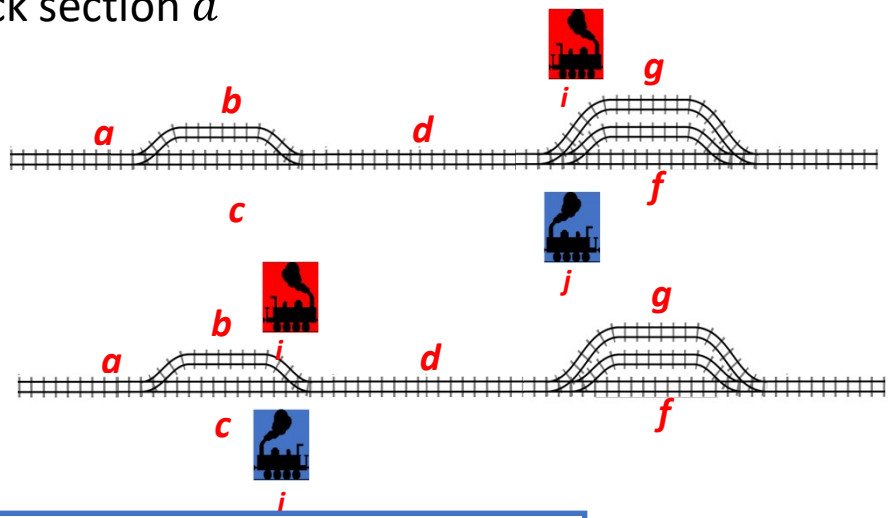
Conflict



- Conflict: both train i and train j wants to access track section d

Either i enters g before j enters d

$$t_d^j - t_g^i \geq 0$$



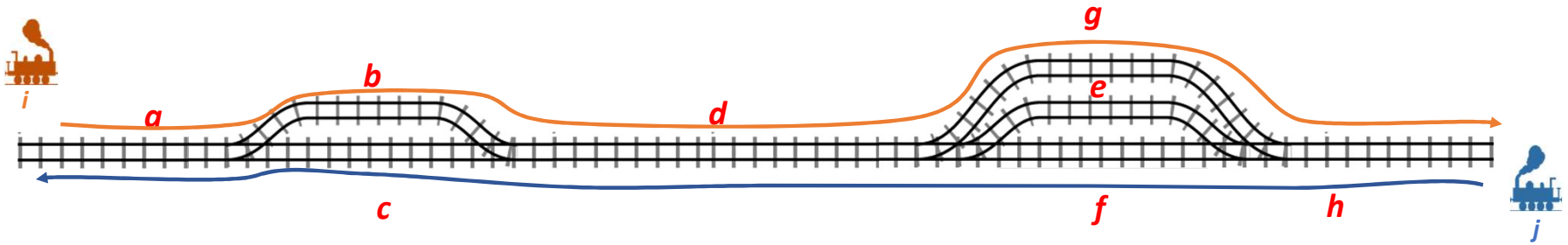
Or j enters c before i enters d

$$t_d^i - t_c^j \geq 0$$

$$t_d^j - t_g^i \geq 0 \bigvee t_d^i - t_c^j \geq 0$$

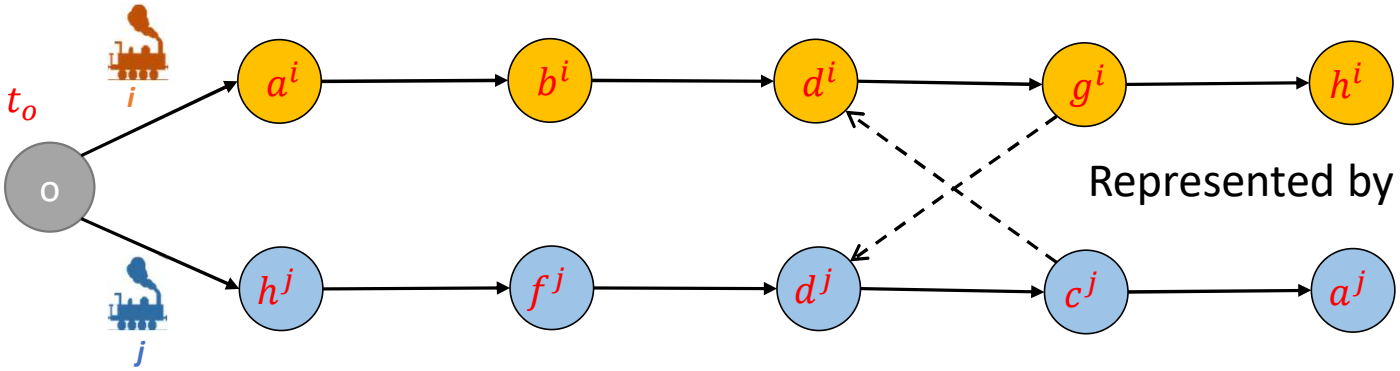
Disjunctive (precedence) constraint

Disjunctive graph



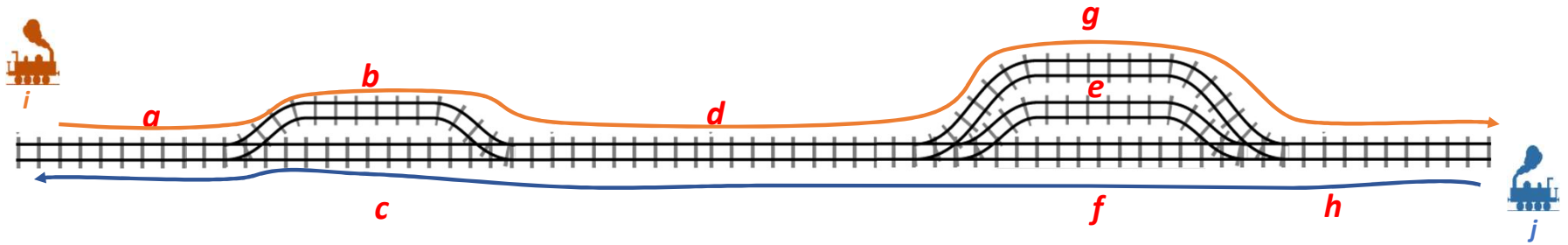
$$t_d^j - t_g^i \geq 0 \bigvee t_d^i - t_c^j \geq 0$$

Disjunctive precedence constraint

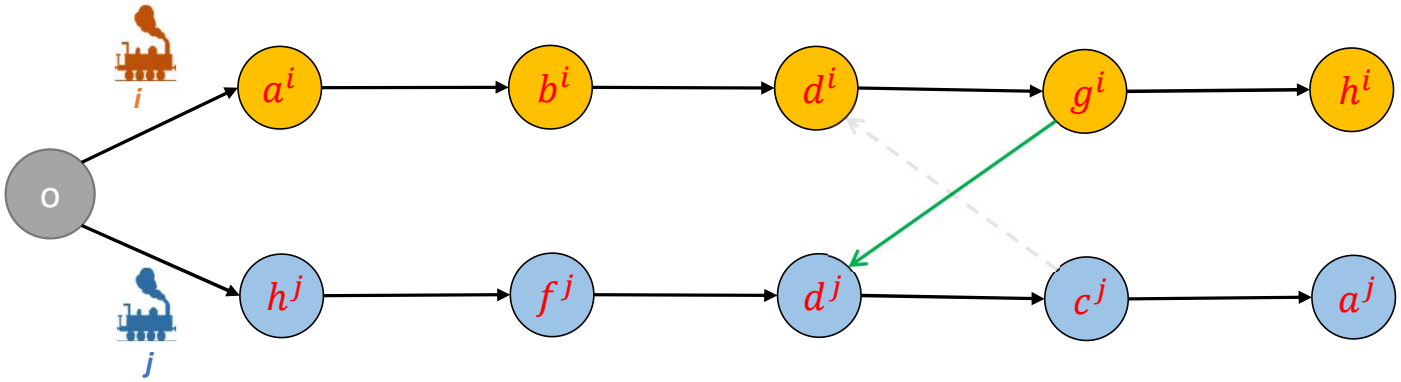


Represented by a pair of *disjunctive* arcs

"Solving" Conflicts



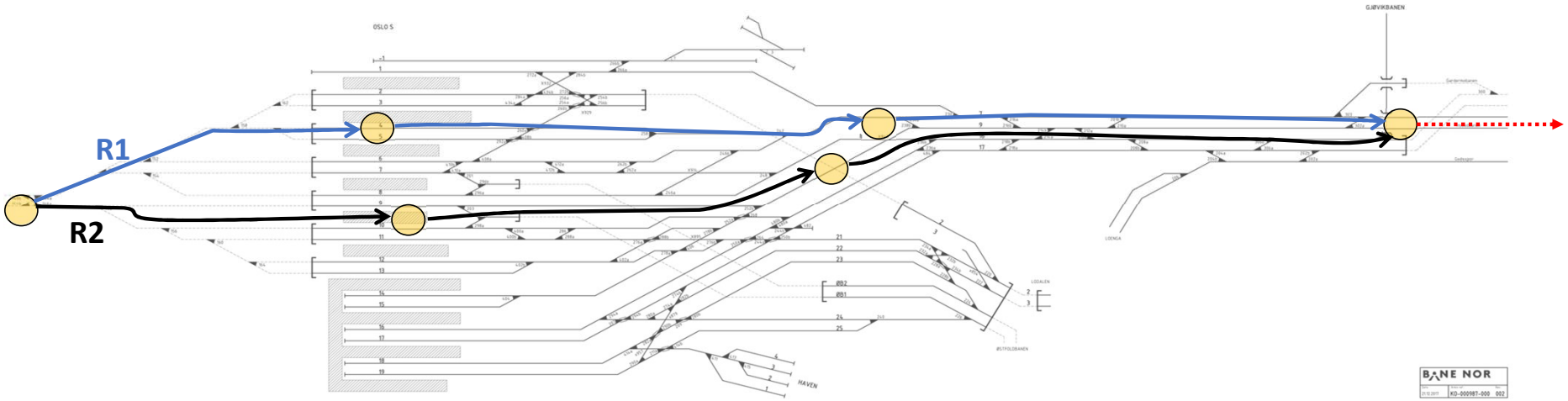
- Solving a conflict means deciding which term in $t_d^j - t_g^i \geq 0$ **OR** $t_d^i - t_c^j \geq 0$ to satisfy
- Equivalent to selecting an arc and dropping the other



train i goes first

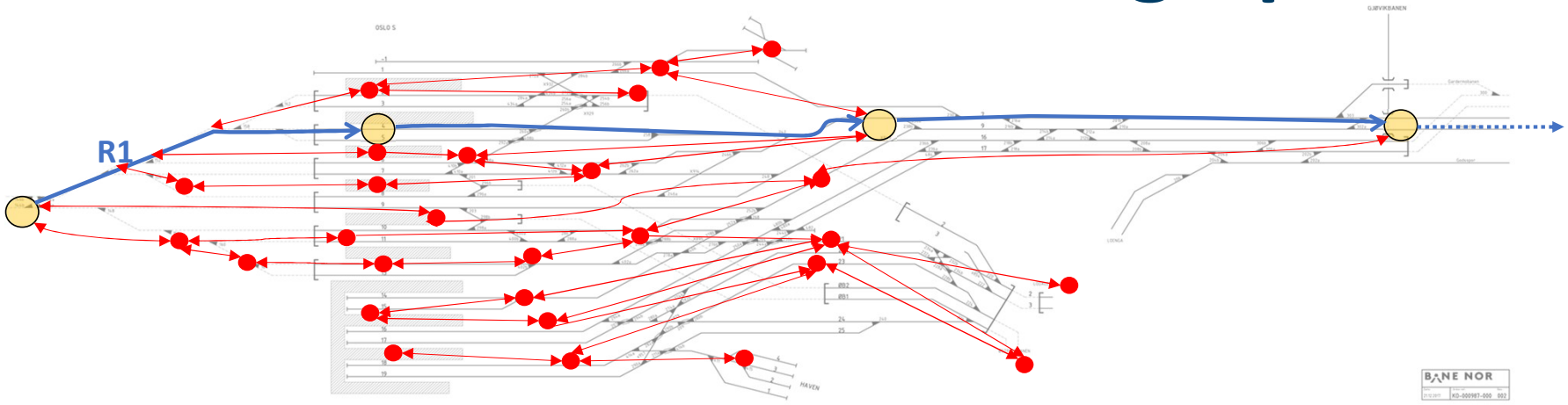
$t_d^j - t_g^i \geq 0$

Alternative routes



- Two alternative routes for the same train in Oslo station

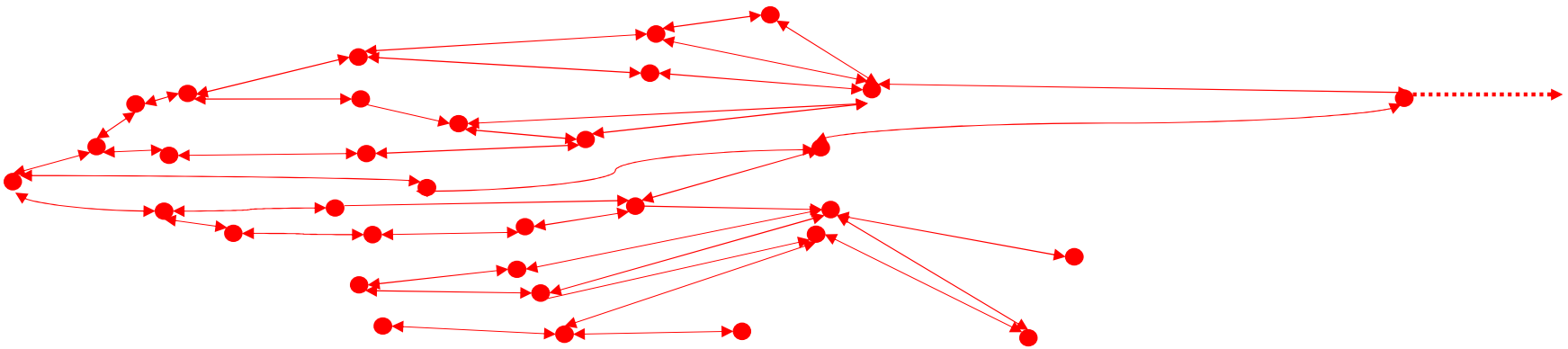
The rail network graph



- Underlying rail network graph (with alternative *routing arcs* A^R)

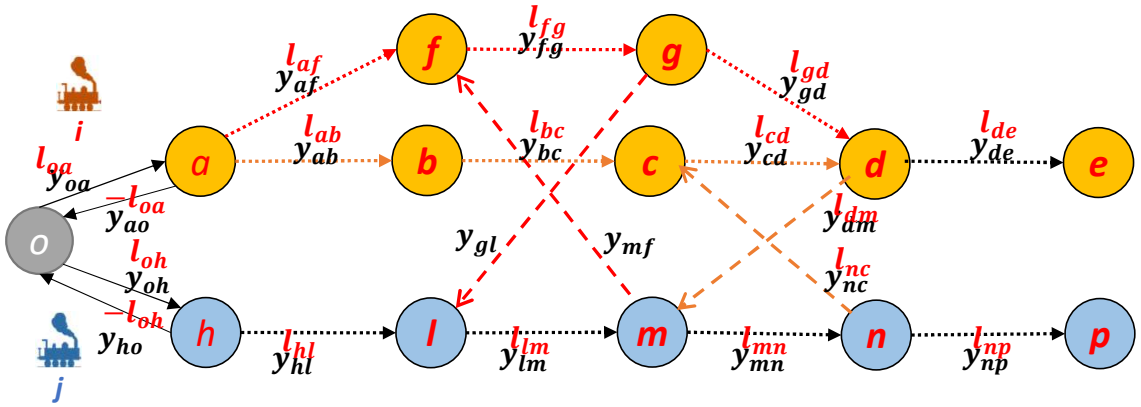
The rail network graph

- Networks with many routing options blow up computational complexity



A MIP formulation

An instance of TRS: TRS Graph



- The TRS graph $G = (V, A)$ contains all **routing, conflict** and **timetable arcs**
- Extra node o for **time synchronization**
- $l \in R^A$ **length vector**
- $G = (V, A), l$ **instance of TRS**

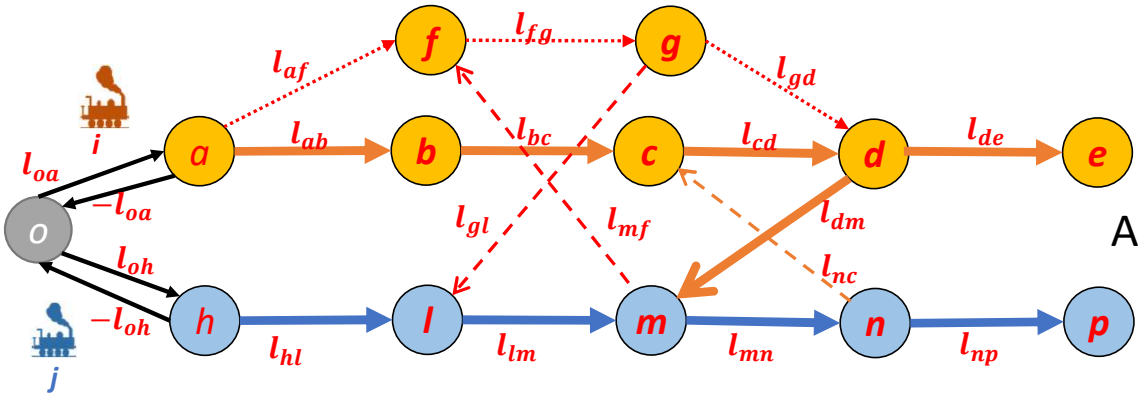
Routing and conflict-resolution decisions $\Leftrightarrow$ selecting arcs in G

Mathematical model: $a \in A$, $y_a \in \{0,1\}$, ($y_a = 1$ if arc chosen)

Y set of candidate solutions $y \in \{0,1\}^A$ (can be described with linear constraints)

$\bar{y} \in Y$ incidence vector of arcs $A(\bar{y}) \subseteq$ A (*feasible selection of arcs*)

The big-M trick



A feasible selection of arcs (solid arcs)

$y_{uv} = 1$ = selecting arc (u, v) = imposing a time precedence constraint

$$t_v - t_u \geq +l_{uv} \text{ - (1- } y_{uv})M$$

$y_{uv} = 0$ = " deleting" (u, v)

M large positive constant

Big-M formulation for TRS

$G = (V, A), l$ instance of TRS, Y feasible (incidence vectors of) selections

$$\begin{aligned} \min \quad & f(t) \\ & t_v - t_u \geq l_{uv} - (1 - y_{uv})M \qquad (u, v) \in A \\ & t \in \mathbb{R}^V, y \in Y \subseteq \{0,1\}^A \end{aligned}$$

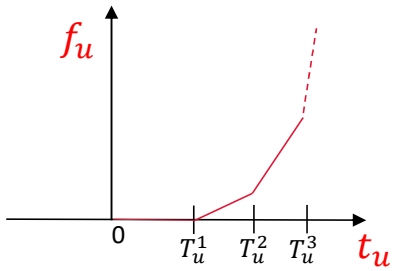
- For $\bar{y} \in Y$ TRS reduces to the following program (linear if $f(t)$ is linear):

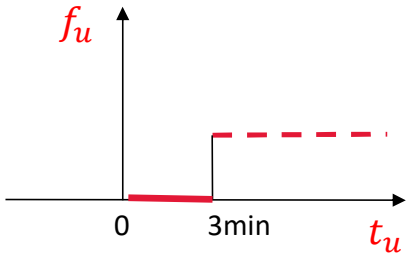
Sched($\bar{y}$)

$$\begin{aligned} \min \quad & f(t) \\ & t_v - t_u \geq l_{uv} \qquad (u, v) \in A(\bar{y}) \\ & t \in \mathbb{R}^V \end{aligned}$$

On the objective function $f(t)$

- Only computed in a few, special points (e.g. terminal stations $V^* \subset V$)
- $f(t)$ is separable: $f(t) = \sum_{u \in V^*} f_u(t_u)$
- Typically $f_u(t_u)$ is non-decreasing *(regular)*.





- We only consider regular, separable cost functions

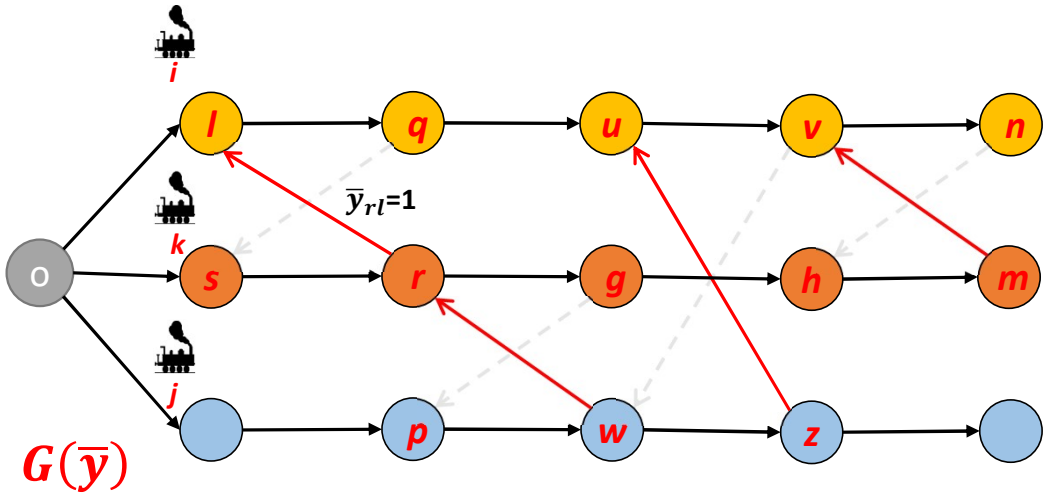
Important facts on feasible solutions

TRS graph and scheduling

❑ Choose $\bar{y} \in Y$: the *TRS graph* reduces to a standard graph $G(\bar{y}) = (V, A(\bar{y}))$

Sched($\bar{y}$)

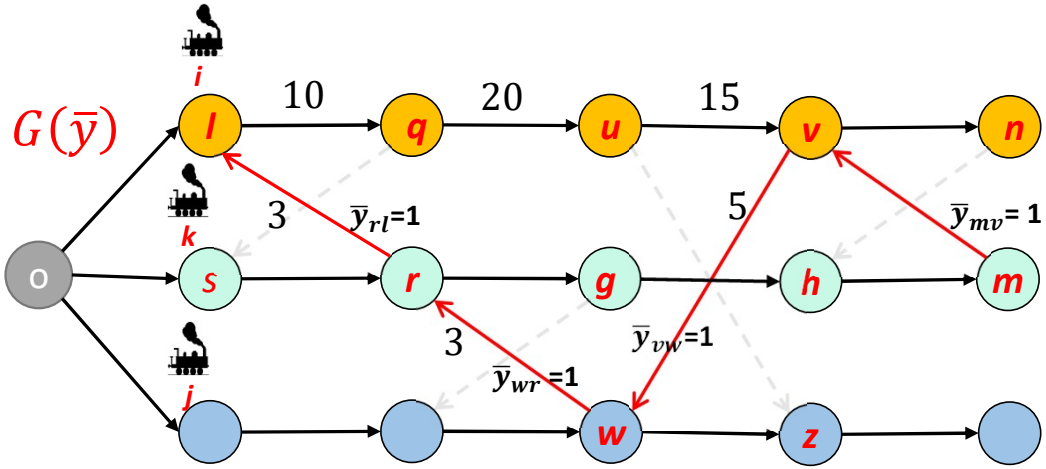
$$\min \quad f(t)$$
$$t_v - t_u \geq l_{uv} \qquad (u,v) \in A(\bar{y})$$
$$t \in \mathbb{R}^V$$



❑ How does $G(\bar{y})$ relate to the associated scheduling problem Sched($\bar{y}$)?

Feasibility

For $\bar{y} \in Y$, Sched($\bar{y}$) has a solution, if and only if $G(\bar{y})$ does not contain a directed cycle C of positive length $l(C)$.

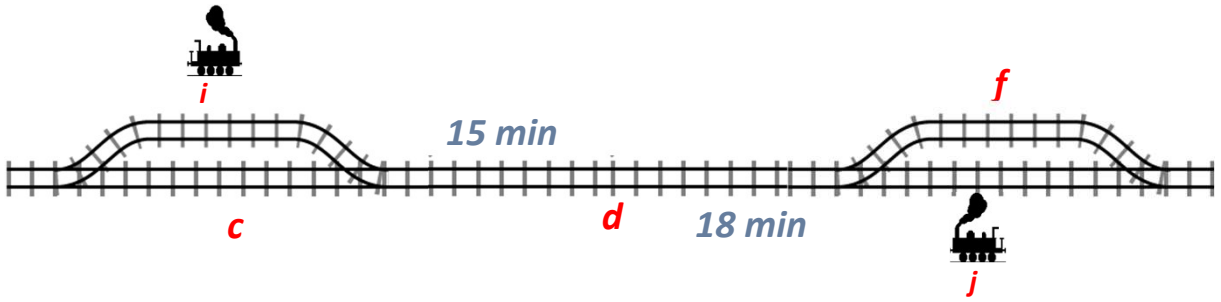


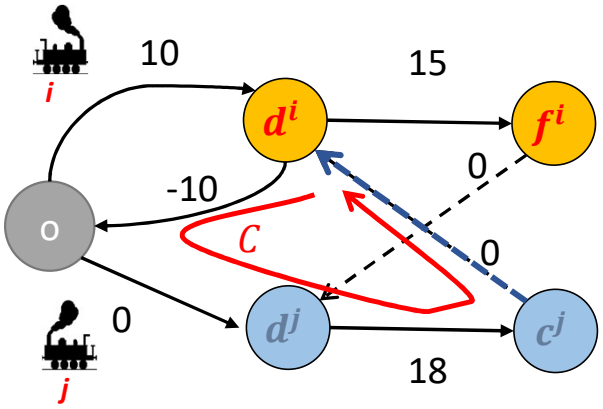
Sched($\bar{y}$)

$$\min \quad f(t)$$
$$t_v - t_u \geq l_{uv} \qquad (u,v) \in A(\bar{y})$$
$$t \in \mathbb{R}^V$$

$C = \{(lq), (qu), (uv), (vw), (wr), (rl)\}, \qquad l(C) > 0$

Example of infeasible solution





- Current time is **09:00**
- Train *i* leaves the station **at 9:10** (exactly)
- Train *j* can leave the station at any time from now
- **Suppose** j wins the conflict on d

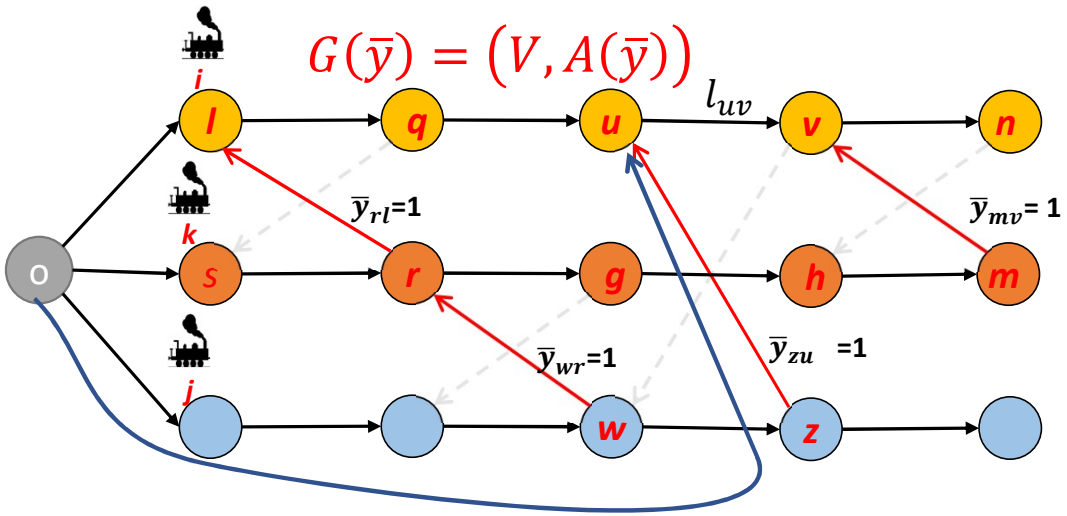
j wins → pick edge (c^j, d^i)

Cycle $C = \{c^j d^i, d^i\ o, o d^j, d^j c^j\}$, **length 8 > 0**

Building feasible schedules

❑ Instance $G = (V, A), l$. Solution $\bar{y} \in Y, G(\bar{y})$ no positive directed cycles.

. $\bar{t}_u^* =$ length of longest path from o to $u \in V$ in $G(\bar{y})$ is feasible for Sched($\bar{y}$)



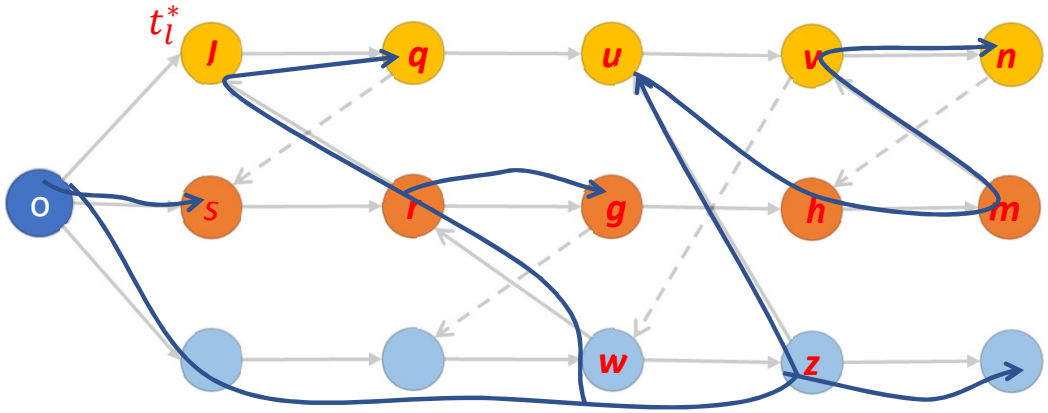
Sched($\bar{y}$)

$$\begin{aligned} \min \quad & f(t) \\ & t_v - t_u \geq l_{uv} \qquad (u, v) \in A(\bar{y}) \\ & t \in \mathbb{R}^V \end{aligned}$$

Optimal solutions

- $\bar{y} \in Y$, $G(\bar{y})$ no positive dicycles. For $u \in V$, $\bar{t}^*_u =$ length of longest ou-path in $G(\bar{y})$

If $f(t)$ is non-decreasing then $\bar{t}^*$ is an optimal solution for Sched($\bar{y}$)



Sched($\bar{y}$)

$$\min \quad f(t)$$
$$t_v - t_u \geq l_{uv} \qquad (u,v) \in A(\bar{y})$$
$$t \in \mathbb{R}^V$$

A reformulation for the TRS problem

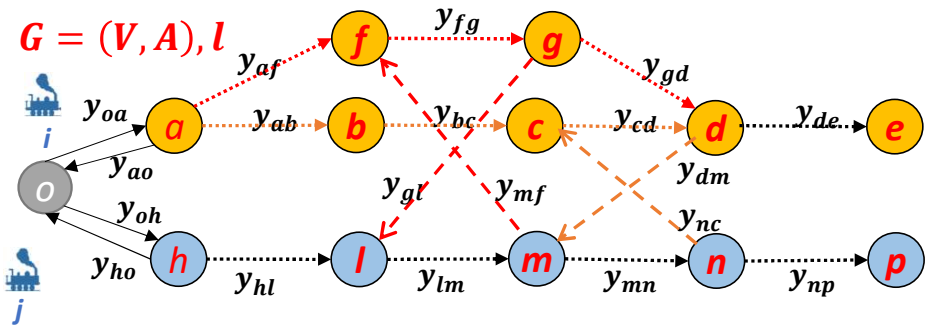
- Assuming $f(t)$ is non-decreasing, we can restate TRS as:

> *Find $y^* \in Y$, such that $G(y^*)$ has no positive directed cycles and the cost $f(t^*)$ of the schedule t^* associated with the longest path tree H^* in $G(y^*)$ is minimum.*

- This also leads to a non-compact, non-big-M reformulation (*Path&Cycle,* Lamorgese and M., 2019)
- The variables are again the arc variables $\boldsymbol{y}$. But the constraints are associated with the positive directed cycles and the longest path trees of $G(V, A), l$.

Solving instances of TRS

Solving TRS by Branch & Bound



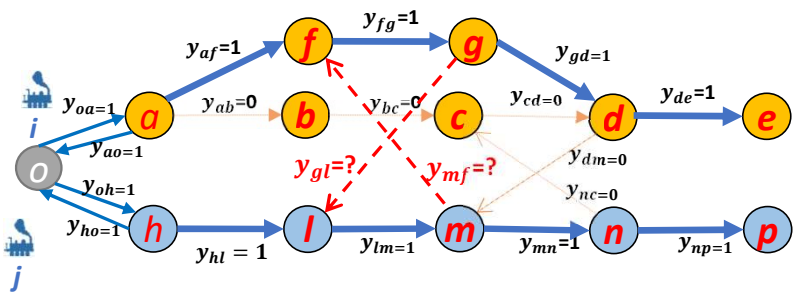
$$\begin{aligned} \min \quad & f(t) \\ & t_v - t_u \geq l_{uv} - (1 - y_{uv})M \qquad (u, v) \in A \\ & t \in \mathbb{R}^V, y \in Y \subseteq \{0, 1\}^A \end{aligned}$$

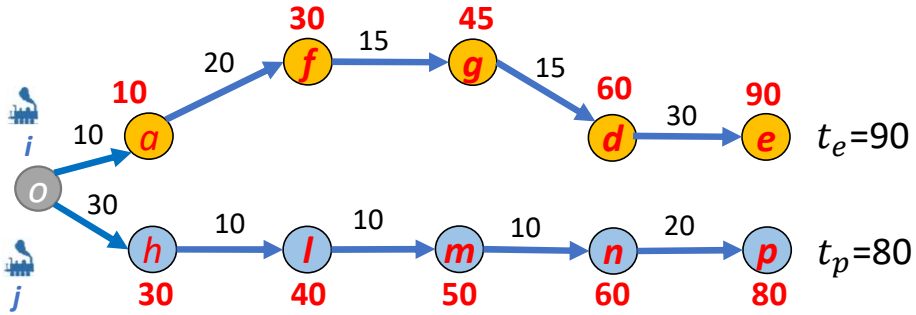
- We assume $\boldsymbol{f(t)}$ **non-decreasing**, e.g. linear $\boldsymbol{c^T t}$, with $\boldsymbol{c \geq 0}$
- MIP formulations can be solved by B&B, either ad-hoc or using off-the-shelf MIP-solvers
- In any node q of the B&B tree we have some y variables fixed to 0 or 1. I.e. we have selected $A^q \subseteq A$ arcs and excluded some other arcs.
- The associated MIP contains some standard precedence constraints and some big-M constraints. Solving linear relaxation = neglecting all residual big-M constraints.
- Computing the linear relaxation amounts to compute a longest path tree in $G(A^q)$

Into the branching node

$G^q = (V^q, A^q), l^q$

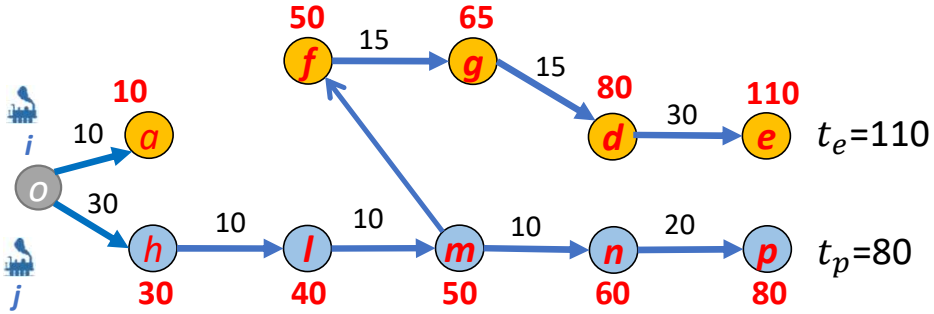
Branch node *q* : Blue arcs selected, red arcs still free





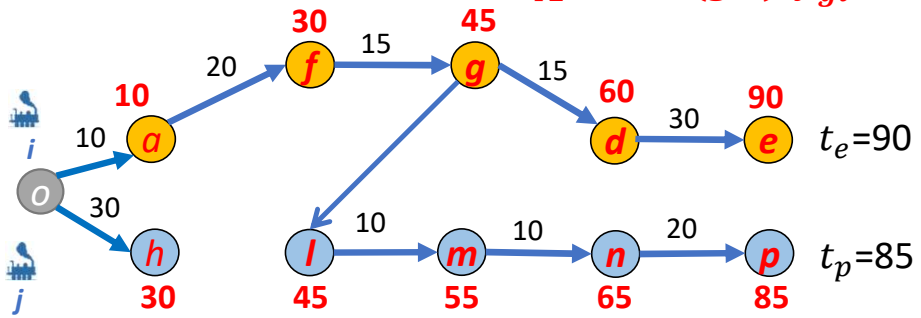
New branch on conflict $y_{mf} + y_{gl} = 1$

Child q_1, select $(m, f), y_{mf}$=1



Longest path tree. Cost = $t_p + t_e = 190$

Child q_2, select $(g, l), y_{gl}$=1



Longest path tree. Cost = $t_p + t_e = 175$

}

Incremental longest path tree

- Binary variables are associated with conflict- or routing-decisions (timetable arcs are fixed)
- At each branching node we either branch on a conflict or on a routing decision
- In either case, we add to the parent graph one or few arcs
- A positive cycle identification (infeasibility) or a partial feasible schedule with associated lower bound is computed effectively by **incremental longest path computations**.
- We extended previous results for incremental topological ordering (Haupler et al, 2011)

}

Is B&B good enough?

A modest example

An instance with **6 trains** (submitted to MIPLIB)

- TRS graph G contains 1954 nodes
- On average 213 alternative routes per train

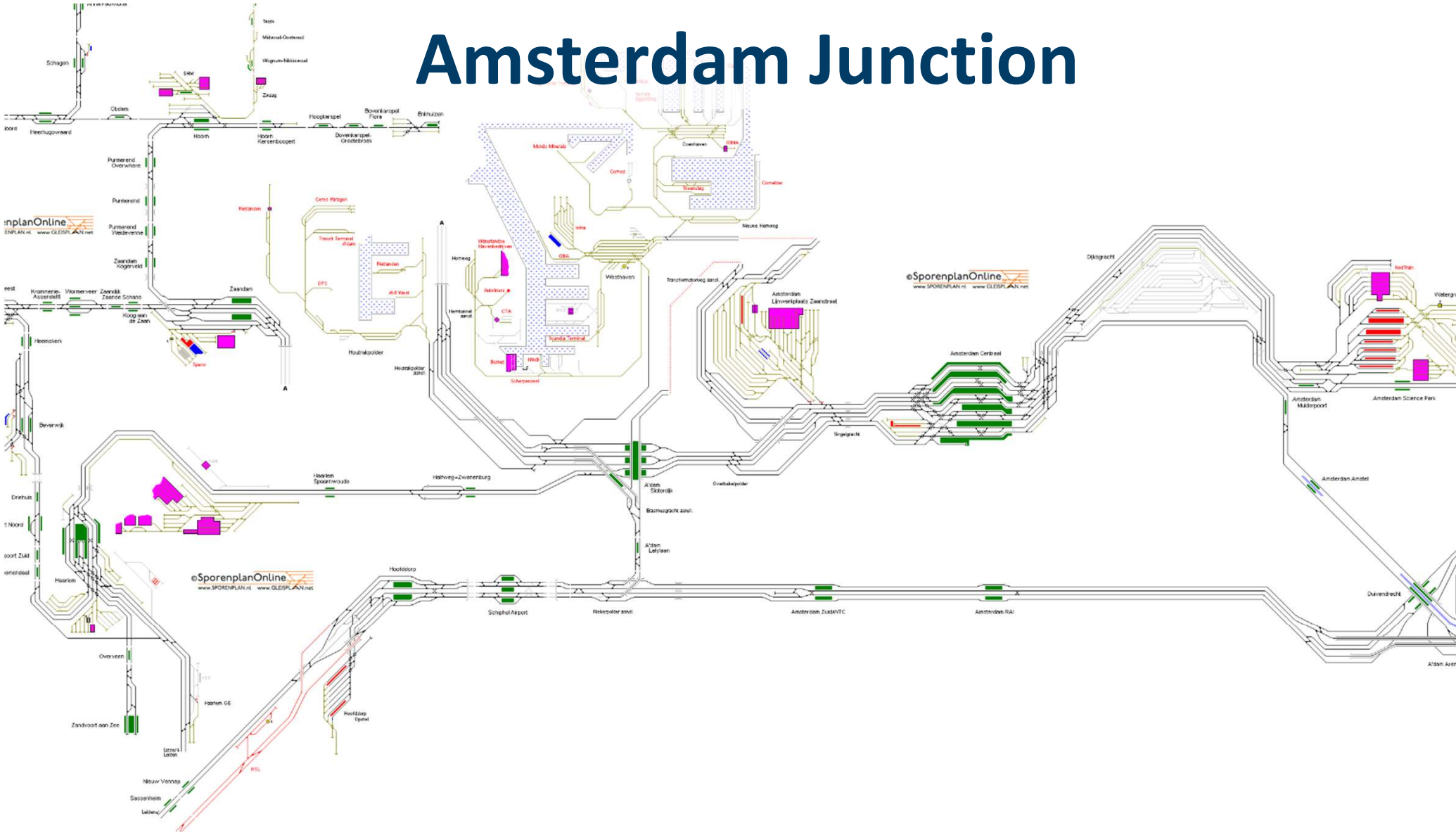
Natural MIP model
- 1900 continuous variables (schedule t)
- 23000 binary variables (arc variables y)
- 61000 constraints, 300000 non-zeros

After 10 minutes computation Gap 100%.
(Gurobi 9.5, default settings, 12 cores)

In practice, we have 22 seconds …

Decomposition

Amsterdam Junction



Logic Benders' reformulation

- We adopt a more general version of classical Benders' decomposition

$$\begin{aligned} \min \quad & f(x) \\ & Ax \le b \\ & Bx + Cw \le e \\ & x \in X, w \in W \end{aligned}$$

$X \subseteq Z^{q_x} \times R^{s_x}, W \subseteq Z^{q_w} \times R^{s_w}$

both x and w are mixed integer vectors

- Objective and a subset of constraints not involving w
- The target is to project out w variables and reformulate the problem as

$$\begin{aligned} \min \quad & f(x) \\ & Ax \le b \\ & Dx \le d \quad \text{feasibility cuts} \\ & x \in X \end{aligned}$$

- Is this convenient in our case?
- How to choose x, w?
- How to generate the feasibility cuts

Master/worker decomposition

$$\text{(TRS)} \quad \begin{aligned} \min \quad & f(x) \\ & Ax \le b \\ & Bx + Cw \le e \\ & x \in X, w \in W \end{aligned}$$

$$(P) \quad \begin{aligned} \min \quad & f(x) \\ & Ax \le b \\ & Dx \le d \quad \text{feasibility cuts} \\ & x \in X \end{aligned}$$

Constructed iteratively by row generation

1. Solve reformulation with subset $D^k x \le d^k$ of feasibility cuts (**restricted master** P^k)
2. If P^k infeasible **STOP (TRS infeas)**. **Otherwise** let x^k optimal solution to P^k
3. **If** x^k can be extended to a solution (x^k, w^k) of TRS, **STOP** ((x^k, w^k) is optimal)
4. **Else** identify valid constraint $\delta^k x \le \Delta^k$ violated by x^k. Generate $D^{k+1} x \le d^{k+1}$, P^{k+1}
5. Iterate

Points 3., 4. and 5. is called **worker (sub)problem:**

Does there exist $w^k \in W$ such that $Cw^k \le e - Bx^k$?

$$(P^k) \quad \begin{aligned} \min \quad & f(x) \\ & Ax \le b \\ & D^k x \le d^k \\ & x \in X \end{aligned}$$

$$\delta^k x^k > \Delta^k$$

$$(P^{k+1}) \quad \begin{aligned} \min \quad & f(x) \\ & Ax \le b \\ & D^k x \le d^k \\ & \delta^k x \le \Delta^k \\ & x \in X \end{aligned}$$

How to choose x and w

$$\begin{aligned} \min \quad & f(x) \\ & Ax \le b \\ & Bx + Cw \le e \\ & x \in X, w \in W \end{aligned}$$

$$\begin{array}{c|ccc|ccc} \multicolumn{3}{c}{x} & & \multicolumn{3}{c}{w} \\ \multicolumn{3}{c|}{\mathrm{A}} & & \multicolumn{3}{c}{0} \\ \hline B^1 & 0 & \cdots & 0 & C^1 & 0 & \cdots & 0 \\ 0 & B^2 & \cdots & 0 & 0 & C^2 & \cdots & 0 \\ 0 & \ddots & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & \cdots & 0 & B^q & 0 & \cdots & 0 & C^q \end{array}$$

worker problem

Does there exist $w^k \in W$ such that $Cw^k \le e - Bx^k$?

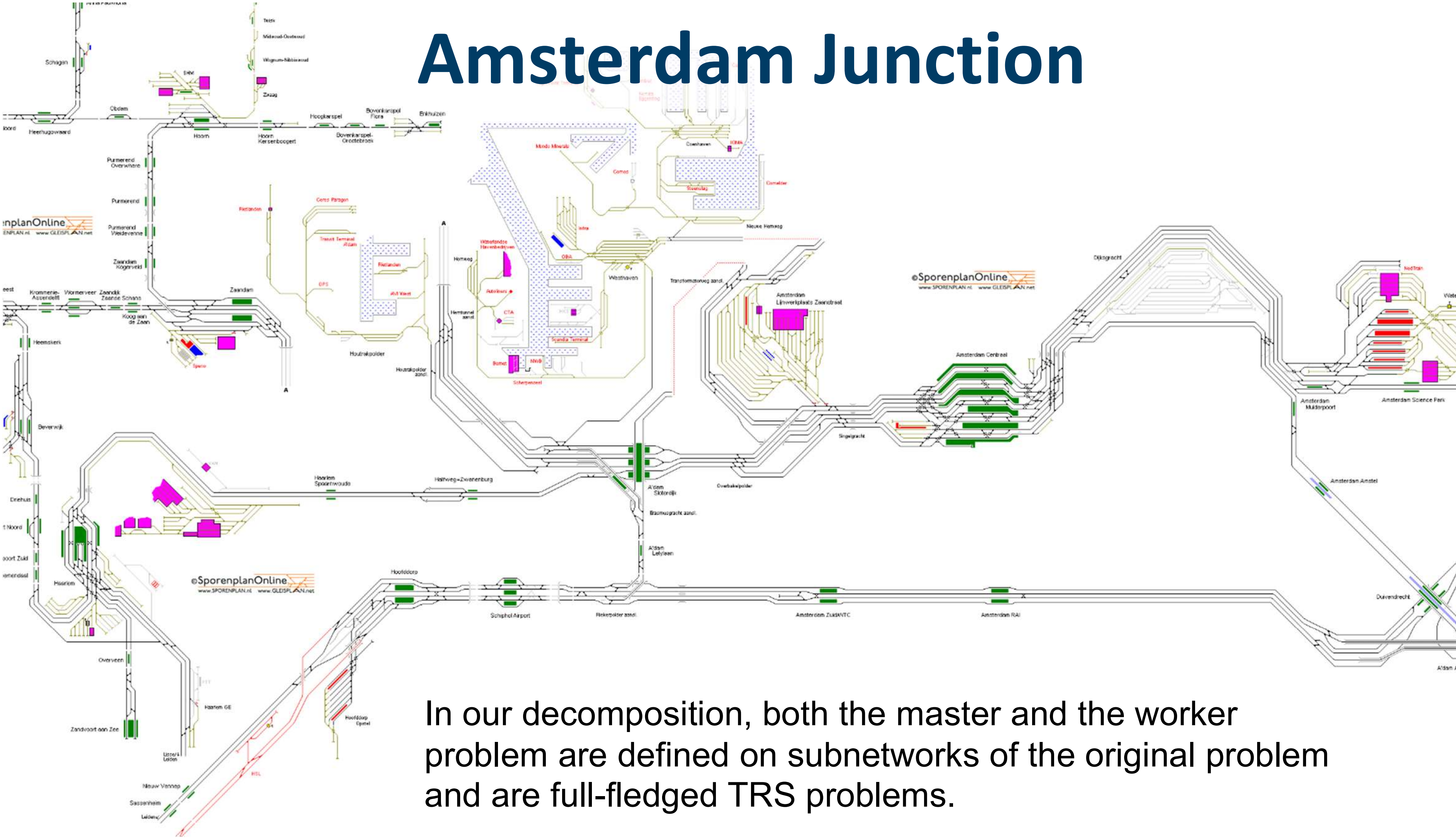
C decomposes into several, non-overlapping and small blocks → the worker problem decomposes into independent subproblems, solved individually → Workers solve quickly

B decomposes into small, non overlapping blocks, each corresponding to a C-block → each worker subproblem "shares" few variables with master program → Strong infeasibility cuts

A small → the master program is small → Master solves quickly

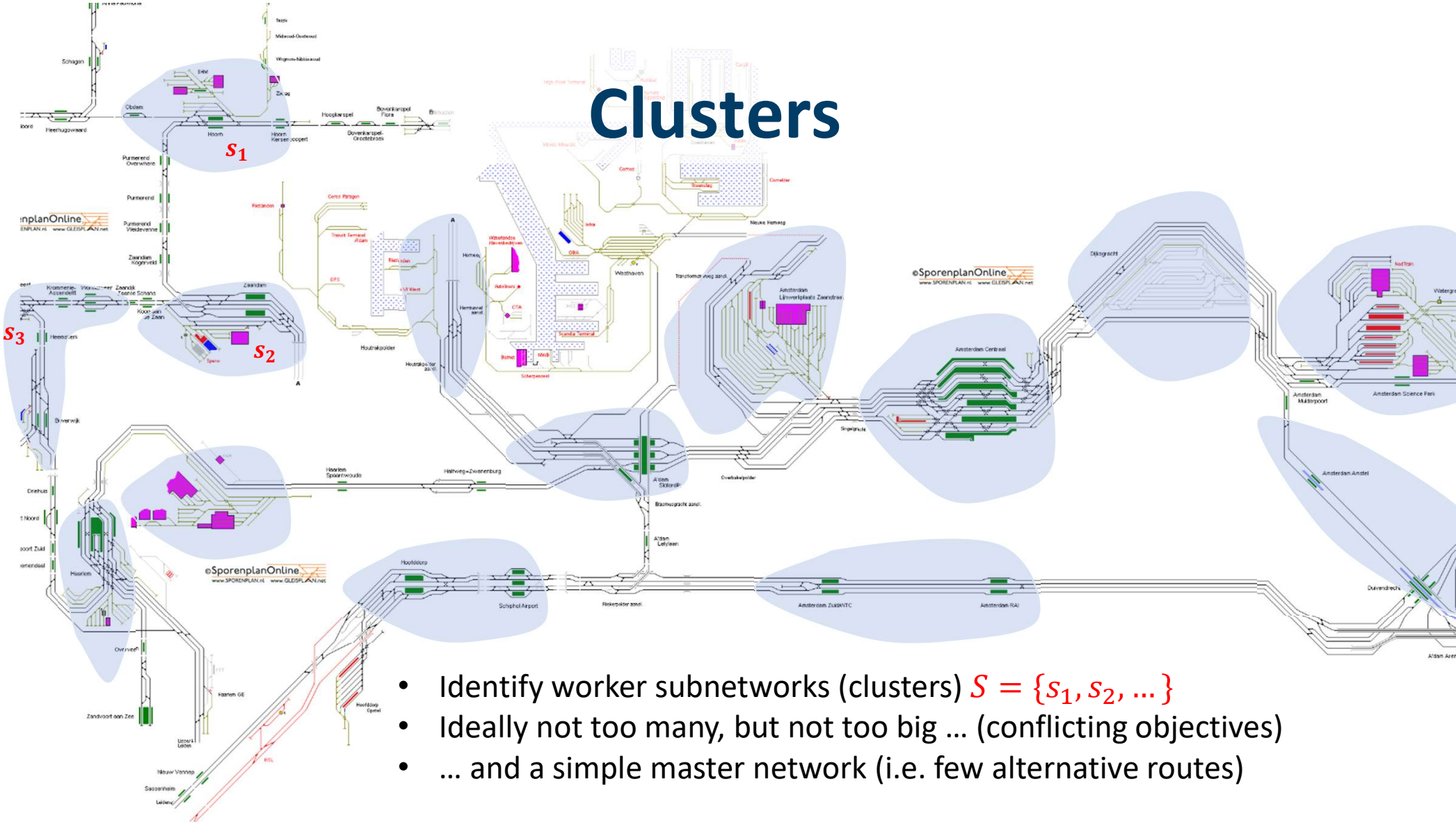
We identify x and w, thus A, B, C, b, e by **spatial decomposition**

Amsterdam Junction

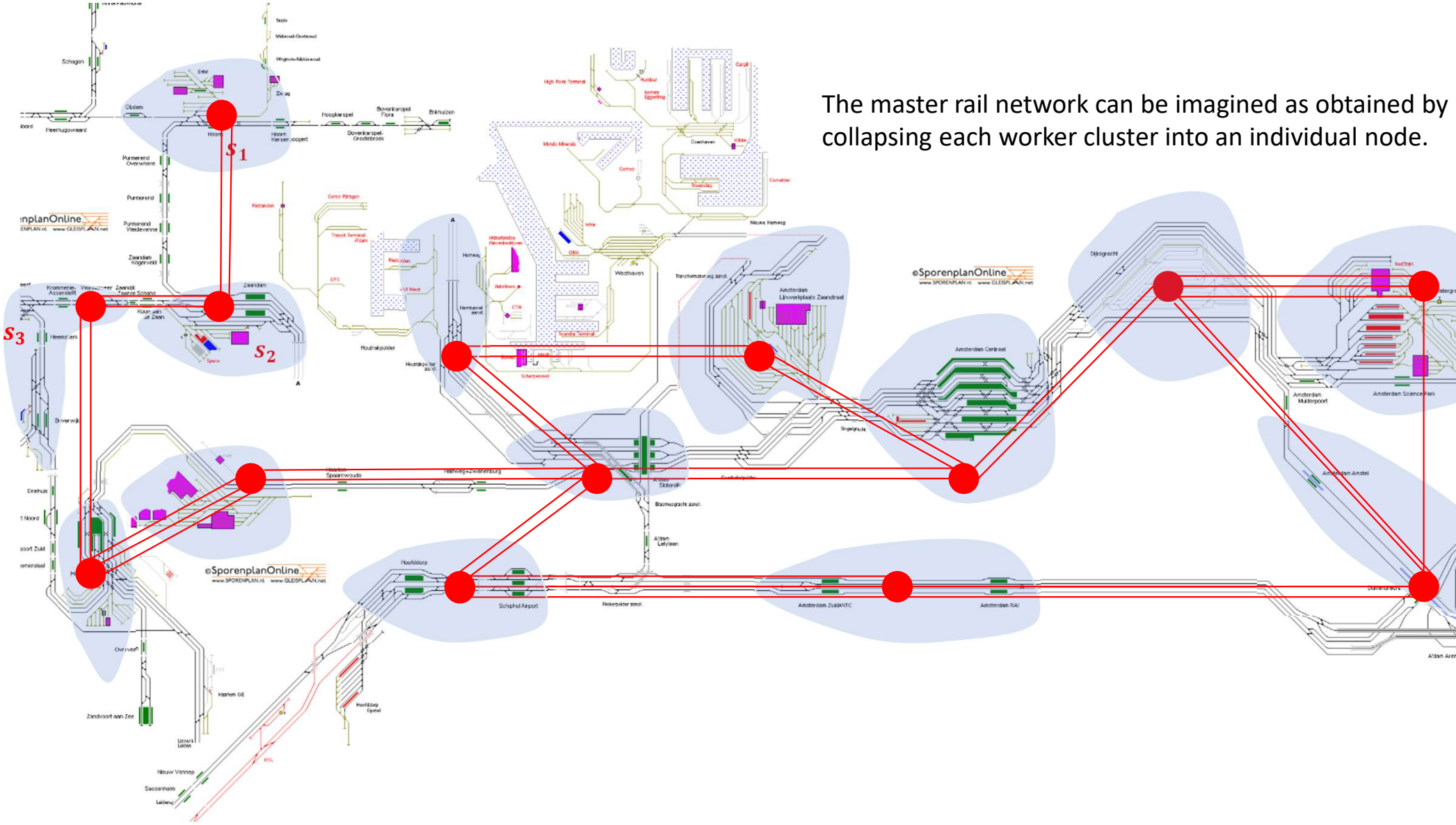


In our decomposition, both the master and the worker problem are defined on subnetworks of the original problem and are full-fledged TRS problems.

Clusters



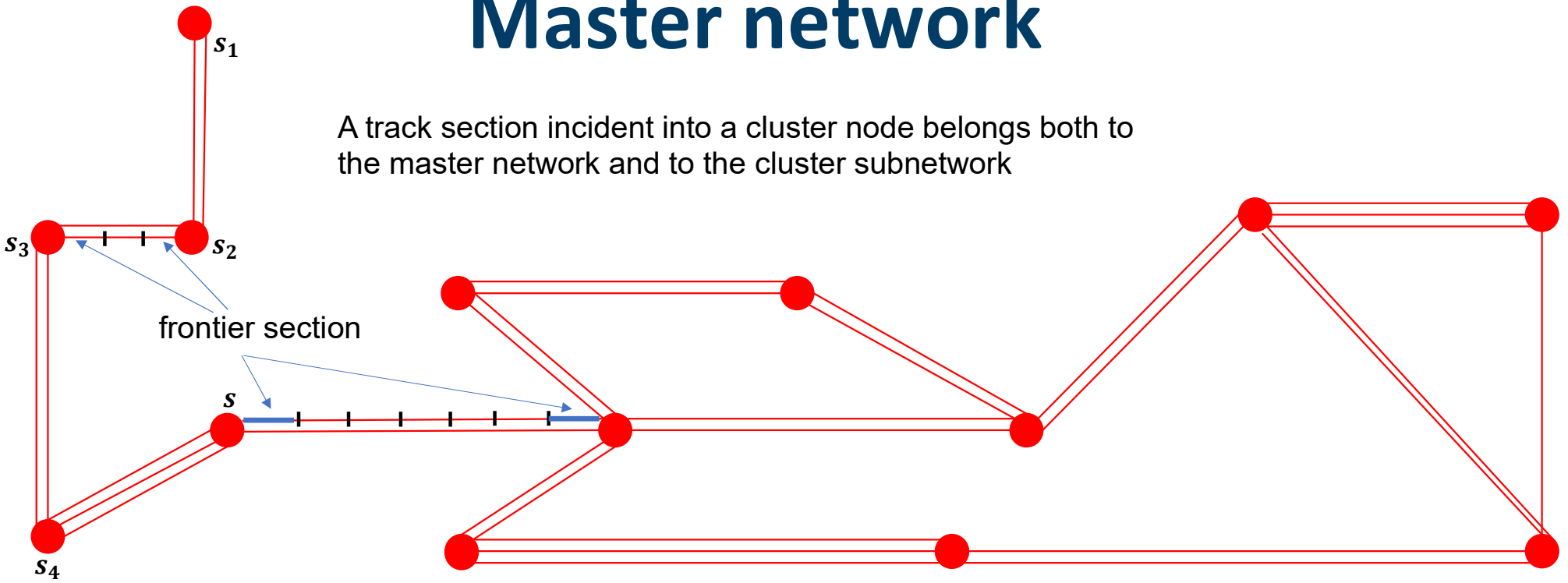
- Identify worker subnetworks (clusters) $S = \{s_1, s_2, \dots\}$
- Ideally not too many, but not too big … (conflicting objectives)
- … and a simple master network (i.e. few alternative routes)



The master rail network can be imagined as obtained by collapsing each worker cluster into an individual node.

Master network

A track section incident into a cluster node belongs both to the master network and to the cluster subnetwork

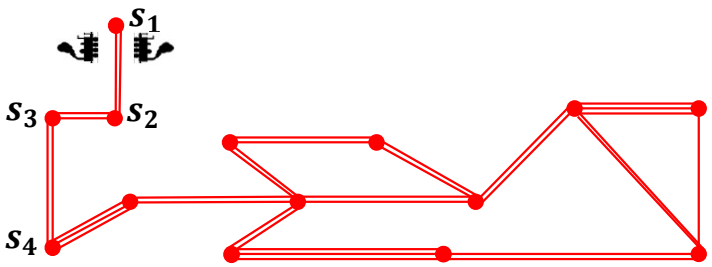


A schedule variable associated with a train and such frontier section belongs both to the master and to the worker

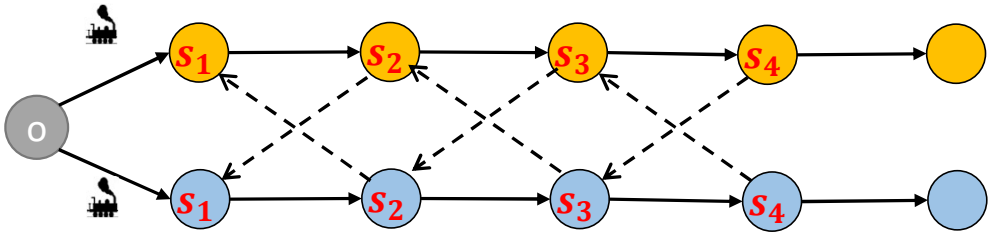
Spatial decomposition: master

- The master problem amounts to solving TRS on a collapsed railway network G^N

$$\begin{aligned}\min\quad & f(t)\\ & t_v^N - t_u^N \geq l_{uv}^N - (1 - y_{uv}^N)M \qquad (u,v) \in A^N\\ & \dots\\ & \text{Feasibility cuts}\\ & \dots\\ & t^N \in \mathbb{R}^{V^N}, y^N \in Y^N \subseteq \{0,1\}^{A^N}\end{aligned}$$



Master Graph $G^N = (V^N, A^N)$

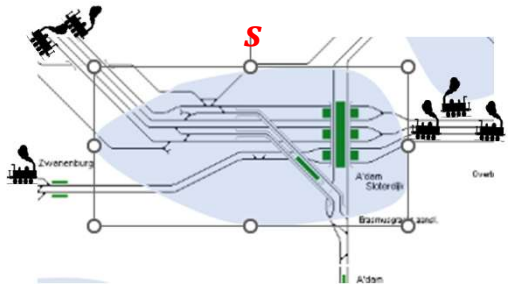


- t^N: master schedule solution
- For each cluster and each train, t^N provides arrival and departure times in the cluster
- t_s^N *master (tentative) timetable* for cluster s

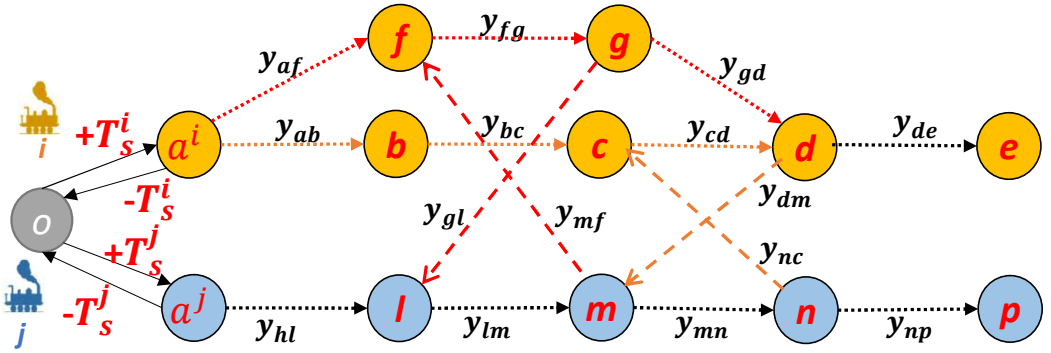
Worker subproblem s

Worker subproblem: Is the master timetable T feasible for (each) cluster s?

T_s^i arrival time of train i in cluster s in current master solution (here we neglect departure times)



Cluster s



Worker TRS graph associated with s and master timetable T

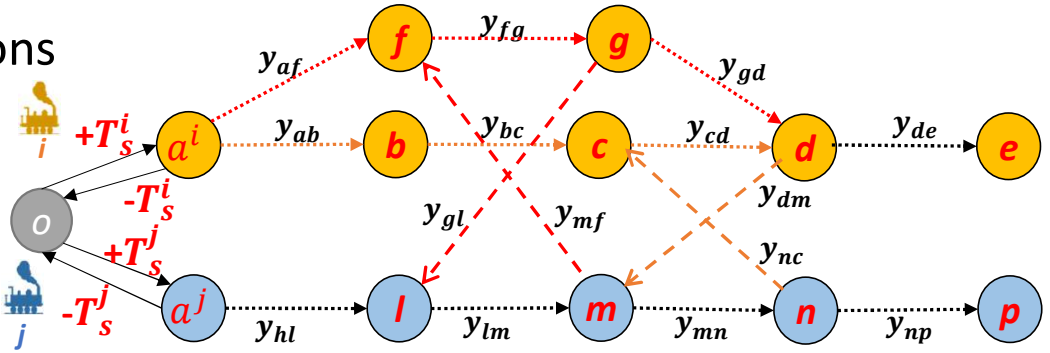
If worker infeasible, what is the feasibility cut violated by the current master timetable T?

Cycle Obstructions and feasibility cuts

Cycle obstructions

TRS Instance $G = (V, A), l$, Y feasible selections

- Let $\bar{y} \in Y$. If $G(\bar{y})$ contains a positive dicycle, then $\mathrm{Sched}(\bar{y})$ is infeasible.



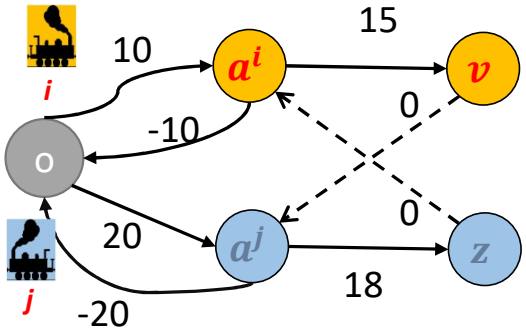
*An instance $G = (V, A), l$ of TRS is **infeasible** iff, **for all** $\bar{y} \in Y$, $G(\bar{y})$ has a positive dicycle*

- If TRS is infeasible, then there exists a **family** $\bar{\Omega} = \{C^1, C^2, \dots\}$ of **positive dicycles** of G, l such that for every selection $\bar{y} \in Y$, $G(\bar{y})$ contains (at least) a cycle of $\bar{\Omega}$.

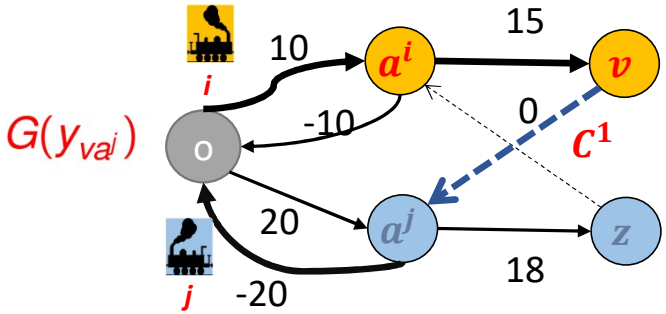
- We call $\bar{\Omega} = \{C^1, C^2, \dots\}$ a **cycle obstruction** (for $G = (V, A), l$)

Example of cycle obstruction

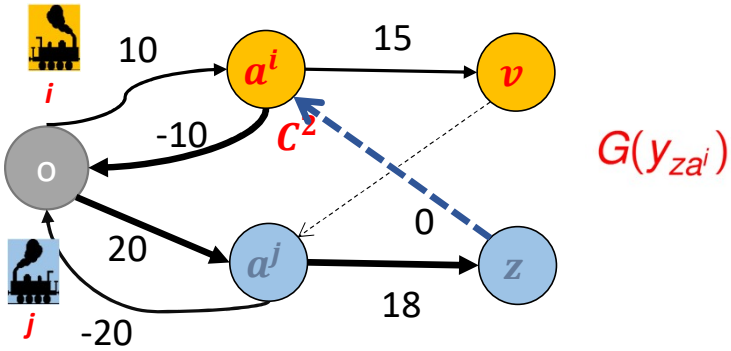
"Timetable"
$T^i = 10$, $T^j = 20$



Every solution contains
either (v, a^j) **or** (z, a^i)
(every other arc is fixed)



$l(C^1) = 10 + 15 + 0 - 20 = 5 > 0$



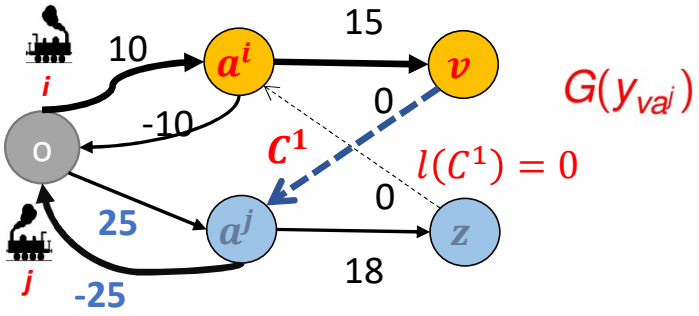
$l(C^2) = 20 + 18 + 0 - 10 = 28 > 0$

cycle obstruction $\bar{\Omega} = \{C^1, C^2\}$

Avoiding cycle obstructions

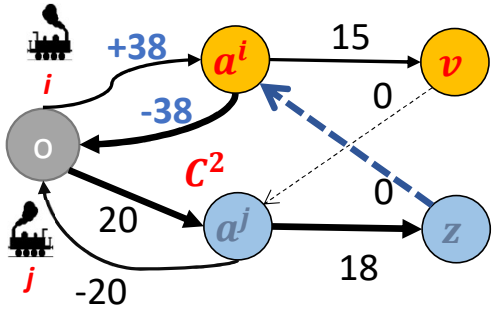
- Cycle obstructions depend on the lengths l of the arcs of the instance $G = (V, A), l$
- In the example, if we modify the timetable, we can get rid of some positive cycles

New arrival "Timetable"
T^i = 10, T^j = ~~20~~25



$G(y_{va^i})$

New arrival "Timetable"
T^i = ~~10~~38, T^j = 20

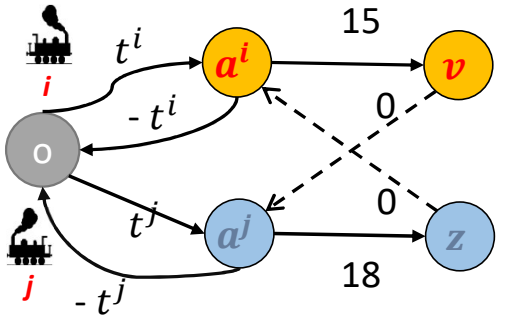


$G(y_{za^i})$

$l(C^2) = 20 + 18 + 0 - 38 = 0$

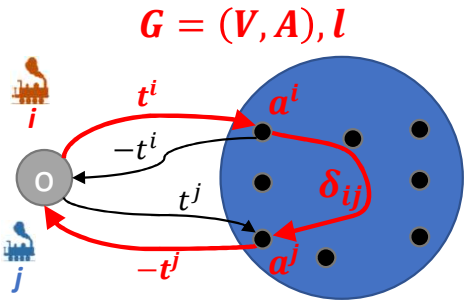
Any feasible (arrival) timetable t must satisfy one of two constraints:

$$t_a^j - t_a^i \geq 15 \bigvee t_a^i - t_a^j \geq 18$$



Preventing positive timetable cycles

❑ To prevent a positive «arrival» cycle to appear, we can constraint the arrival times of the trains involved



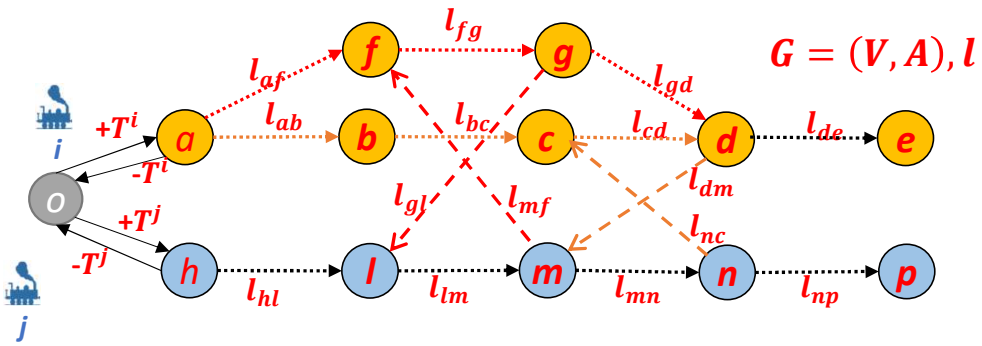
- t^i, t^j arrival times of train i, j
- a^i, a^j arrival nodes of train i, j
- δ_{ij} length of a maximum length path P_{ij} from a^i to a^j in $G = (V, A), l$

The directed cycle $(o, a^i), P_{ij}, (a^j, o)$ has **non-positive length** if

$$t^i - t^j + \delta_{ij} \leq 0 \quad \Rightarrow \quad \boldsymbol{t^j - t^i \geq \delta_{ij}}$$

time-precedence constraint

Feasibility of timetables

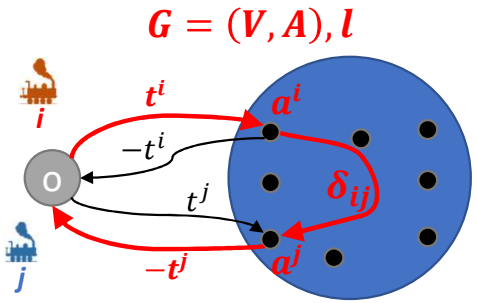


Timetable $\tau = T^1, T^2$

$\Omega = \{C^1, C^2, \dots\}$ **cycle obstruction**

- If a timetable $\tau = T^1, T^2, \dots$ is infeasible then there exists (at least) a cycle obstruction $\Omega = \{C^1, C^2, \dots, C^q\}$ associated with τ.
- We are only interested in cycles containing timetable arcs, i.e. through the origin.
- We may assume timetable cycles are simple, and Ω is minimal.
- WANT: find a timetable $\Delta = t^1, t^2, \dots$ so that Ω **is not a cycle obstruction** of Δ.
- At least one of the cycles in Ω should be non-positive w.r.t. Δ

Disjunctive feasibility cut



Non "positiveness" constraint

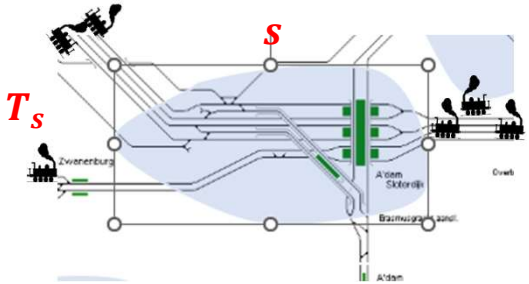
$t^j - t^i \geq \delta_{ij}$

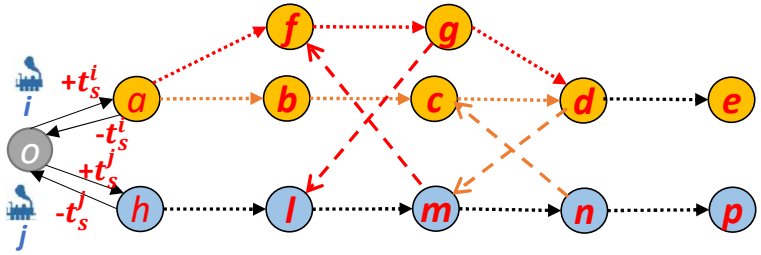
- Cycle obstruction $\Omega = \{C^1, C^2, \dots, C^q\}$ timetable cycles of Ω
- Each cycle only involves two timetable arcs, associated with two trains
- $(i_1, j_1), (i_2, j_2), \dots, (i_q, j_q)$ pairs of trains involved in the obstruction
- OBS: With n trains, the number q of cycles of Ω is at most $n \times (n-1)$
- Let $\Delta = t^1, t^2, \dots, t^n$ be the arrival time of train $1, \dots, n$
- Since at least one cycle of Ω must be non-positive in a feasible schedule, then the next ***k*-disjunctive constraint** is valid (Leutwiler & Corman 2023)

$$t^{j_1} - t^{i_1} \geq \delta_1 \vee t^{j_2} - t^{i_2} \geq \delta_2 \vee \dots \vee t^{j_q} - t^{i_q} \geq \delta_q$$

Worker subproblem

- Is the current master timetable t_k infeasible for some cluster $s \in S$?





- Worker s Infeasible: there exists a cycle obstruction $\Omega^s = \{C^1, C^2, \dots, C^q\}$ for t_k.

- To avoid at least one of such positive cycles, **the master timetable** must satisfy a q-disjunctive constrai

$$t_s^{j_1} - t_s^{i_1} \geq \delta_1 \vee t_s^{j_2} - t_s^{i_2} \geq \delta_2 \vee \dots \vee t_s^{j_k} - t_s^{i_k} \geq \delta_k$$

Remark: each term is a time-precedence constraint!

Master program: feasibility cuts

Each feasibility cut: *q-disjunction* of precedence constraints.

$$
\begin{aligned}
\min \quad & f(t) \\
& t_v^N - t_u^N \geq l_{uv}^N - (1 - y_{uv}^N)M \qquad (u,v) \in A^N \\
& \dots \\
& t_{j_1}^N - t_{i_1}^N \geq \delta_1 \bigvee \cdots \bigvee t_{j_q}^N - t_{i_q}^N \geq \delta_q \qquad \Omega = \{C_1, \dots, C_q\} \in \mathcal{O} \\
& \dots \\
& t^N \in \mathbb{R}^{V^N}, y^N \in Y^N \subseteq \{0,1\}^{A^N}
\end{aligned}
$$

$$
\begin{cases}
t_{j_1}^N - t_{i_1}^N \geq \delta_1 - (1 - y_{i_1 j_1}^{\Omega})M \\
\dots \\
t_{j_q}^N - t_{i_q}^N \geq \delta_q - (1 - y_{i_q j_q}^{\Omega})M \\
y_{i_1 j_1}^{\Omega} + \cdots + y_{i_q j_q}^{\Omega} = 1
\end{cases}
\qquad \Omega \in \mathcal{O}
$$

The "nature" of the formulation is unchanged after including the feasibility cuts

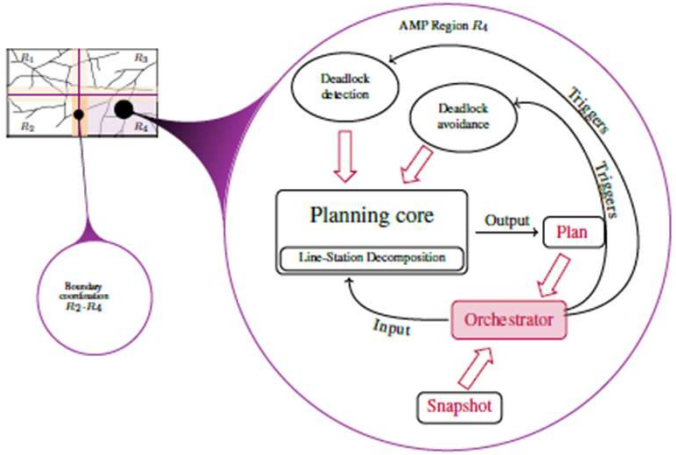
Include in *G* the "cycle" arcs $A^O = \{(\mathrm{i}_1, \mathrm{j}_1)^{\Omega}, \dots, (i_q, j_q)^{\Omega}, \Omega \in O\}$, the associated *y* variables and constraints

$$
\begin{aligned}
\min \quad & f(t) \\
& t_v - t_u \geq l_{uv} - (1 - y_{uv})M \qquad (u,v) \in A^N \cup A^{\mathcal{O}} \\
& t \in \mathbb{R}^V, y \in Y^N \cap Y^{\mathcal{O}} \subseteq \{0,1\}^{A^N \cup A^{\mathcal{O}}}
\end{aligned}
$$

}

Conclusions

The automatic dispatching system was deployed in all 12 administative regions of UP network in the fall 2021.



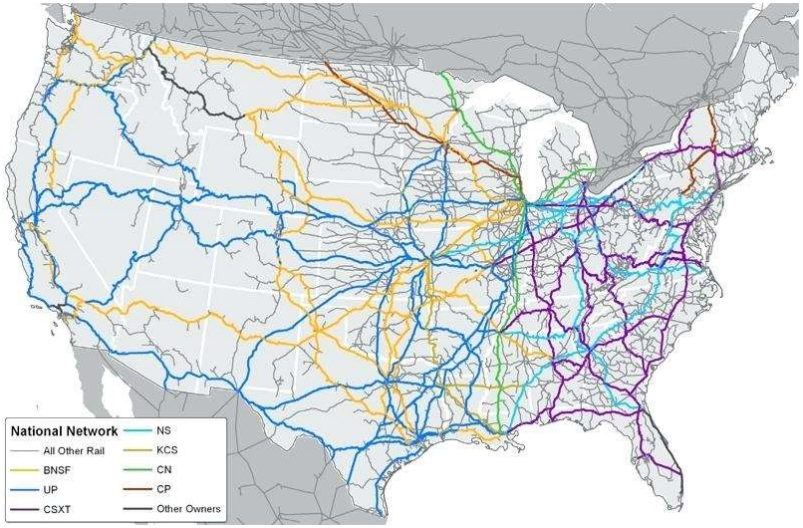
It required the development of many communicating algorithms and software components

}

Conclusions

Since then, we started new challenging projects with UP:

- **Block design**: how to group commodities and send them from origin to destination
- **Train design**: how to group the blocks and form trains
- **Deadlock detection**
- **Network decomposition**



Joint effort: Annese, Boccia, Dal Sasso, Lamorgese, Luteberget, **M.**, Onofri, **Oriolo**, Piacentini, Russo Russo, **Sartor**, **Ventura** (and others)

Very complex problems, very large instances asking for the development of new combinatorial optimization and integer programming methods.

- Lamorgese, L., & Mannino, C. (2015). An exact decomposition approach for the real-time train dispatching problem. *Operations Research*, *63*(1), 48-64.
- Lamorgese, L., & Mannino, C. (2019). A noncompact formulation for job-shop scheduling problems in traffic management. *Operations Research*, *67*(6), 1586-1609.
- Leutwiler, F., Corman, F. (2021). A logic Benders' decomposition for microscopic railway timetable plalnning, *EURO 2021 conference,* Athens, July 2021.
- Mascis, A., & Pacciarelli, D. (2002). Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, *143*(3), 498-517.
- Haeupler, B., Kavitha, T., Mathew, R., Sen, S. and Tarjan, R.E., 2012. Incremental cycle detection, topological ordering, and strong component maintenance. *ACM Transactions on Algorithms (TALG)*, *8*(1), pp.1-33.