Algorithms based on time-expanded formulations for Train Timetabling Problems

Valentina Cacchiani

Department of Electrical, Electronic and Information Engineering
"Guglielmo Marconi",
University of Bologna,
Italy

November 2022

Outline

- Introduction to Train Timetabling
- Models based on time-expanded graphs
- Solution methods
- Generalization to skip-stop planning strategies and passenger-centric objectives

Train Timetabling

Railway Optimization Stages

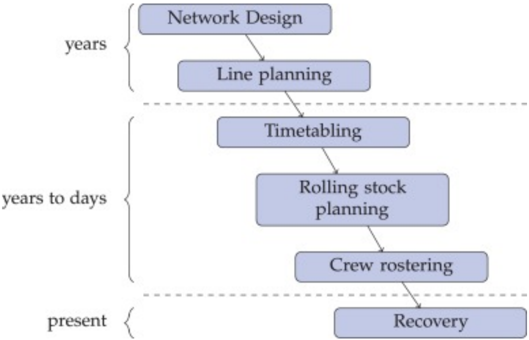


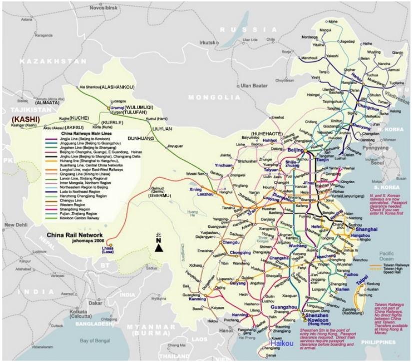
Figure from Lusby, R. M., Larsen, J., Bull, S. (2017).
A survey on robustness in railway planning. European Journal of Operational Research.

Train Timetabling

- It consists of finding an optimal schedule of trains in a railway network satisfying:
  - safety regulations (e.g., minimum headway times between consecutive trains on the same track) and
  - operational constraints (e.g., running times, dwell times, station capacity)
- The schedule is defined by the departure and arrival times of trains at all visited stations
- The objective function depends on the railway company (e.g., schedule as many trains as possible)

Railway infrastructure

- The railway infrastructure consists of a network with:
  - nodes: represent the locations where the trains may interact
  - tracks: connect the nodes and are used by the trains to travel from one node to the next one



Trains to be scheduled

- The trains to be scheduled are determined based on the passenger demand and can be given in input in two different ways:

Trains to be scheduled

- The trains to be scheduled are determined based on the passenger demand and can be given in input in two different ways:
    1. A set of train lines (a route between an origin and a destination station with a specific stopping pattern) and a frequency of the train line

Trains to be scheduled

- The trains to be scheduled are determined based on the passenger demand and can be given in input in two different ways:
  1. A set of train lines (a route between an origin and a destination station with a specific stopping pattern) and a frequency of the train line
  2. An ideal timetable for each train provided by the Train Operator that specifies the departure and arrival times at each visited station of the railway network

Constraints

- minimum headway time between consecutive trains using the same track
- forbid overtaking and crossing of trains on the same track

Constraints

- minimum headway time between consecutive trains using the same track
- forbid overtaking and crossing of trains on the same track
- lower and upper limits on the dwelling time of a train at a station
- lower and upper limits on the running time of a train on a track

Constraints

- minimum headway time between consecutive trains using the same track
- forbid overtaking and crossing of trains on the same track
- lower and upper limits on the dwelling time of a train at a station
- lower and upper limits on the running time of a train on a track
- acceleration and deceleration times when a train stops at a station
- maximum number of trains simultaneously present at a station
- connection constraints for passengers transfers

Constraints

- minimum headway time between consecutive trains using the same track
- forbid overtaking and crossing of trains on the same track
- lower and upper limits on the dwelling time of a train at a station
- lower and upper limits on the running time of a train on a track
- acceleration and deceleration times when a train stops at a station
- maximum number of trains simultaneously present at a station
- connection constraints for passengers transfers
- . . .

Periodic and non-periodic timetabling

- Periodic (or cyclic): the schedule of the trains is repeated every given time period (for example every hour)

Periodic and non-periodic timetabling

- Periodic (or cyclic): the schedule of the trains is repeated every given time period (for example every hour)
- Non-periodic (or non-cyclic): the schedule of the trains is the same every day, it is appropriate for more congested network

Periodic and non-periodic timetabling

- Periodic (or cyclic): the schedule of the trains is repeated every given time period (for example every hour)
- Non-periodic (or non-cyclic): the schedule of the trains is the same every day, it is appropriate for more congested network

- In this talk, we focus on:
  - Starting from an ideal timetable for each train
  - Schedule as many trains as possible and minimize the changes with respect to the ideal timetables

Periodic and non-periodic timetabling

- Periodic (or cyclic): the schedule of the trains is repeated every given time period (for example every hour)
- Non-periodic (or non-cyclic): the schedule of the trains is the same every day, it is appropriate for more congested network

- In this talk, we focus on:
  - Starting from an ideal timetable for each train
  - Schedule as many trains as possible and minimize the changes with respect to the ideal timetables
  - First the non-periodic problem (scheduling trains for a day)

Periodic and non-periodic timetabling

- Periodic (or cyclic): the schedule of the trains is repeated every given time period (for example every hour)
- Non-periodic (or non-cyclic): the schedule of the trains is the same every day, it is appropriate for more congested network

- In this talk, we focus on:
  - Starting from an ideal timetable for each train
  - Schedule as many trains as possible and minimize the changes with respect to the ideal timetables
  - First the non-periodic problem (scheduling trains for a day) and then a periodic problem (scheduling trains for one hour)

Models based on time-expanded graphs

Non-periodic Train Timetabling - one-way line

- $S = \{1, \ldots, s\}$: set of stations
- T: set of trains each with:
  - an assigned importance (e.g., high-speed, local, freight)
  - an ideal timetable

Non-periodic Train Timetabling - one-way line

- $S = \{1, \dots, s\}$: set of stations
- T: set of trains each with:
  - an assigned importance (e.g., high-speed, local, freight)
  - an ideal timetable
- Time discretization (e..g, one minute)

Non-periodic Train Timetabling - one-way line

- $S = \{1, \ldots, s\}$: set of stations
- T: set of trains each with:
  - an assigned importance (e.g., high-speed, local, freight)
  - an ideal timetable
- Time discretization (e..g, one minute)
- The goal is to maximize the total importance of the scheduled trains and minimize the changes to the ideal timetables

Changes to the ideal timetables

Changes can be applied to obtain a feasible timetable (without train conflicts):

Changes to the ideal timetables

Changes can be applied to obtain a feasible timetable (without train conflicts):

- change the departure and/or arrival times of some trains at some of the visited stations → shift
- increase the dwell time of some trains at some of the visited stations → stretch
- cancel (= not schedule) a train

Changes to the ideal timetables

Changes can be applied to obtain a feasible timetable (without train conflicts):

- change the departure and/or arrival times of some trains at some of the visited stations → shift
- increase the dwell time of some trains at some of the visited stations → stretch
- cancel (= not schedule) a train

- Lower and Upper limits are imposed for these changes:
  - maximum shift at the departure station for each train
  - maximum total stretch

Time-expanded graph

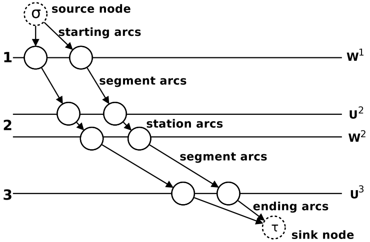
- **Time-Space Graph model** by Caprara, Fischetti and Toth (2002)

Time-expanded graph

- **Time-Space Graph model** by Caprara, Fischetti and Toth (2002)
- time-space graph $G = (V, A)$:

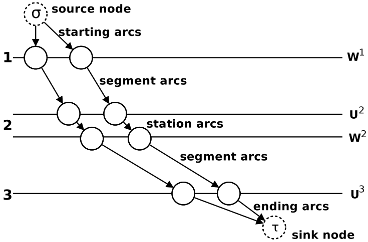
Time-expanded graph

- **Time-Space Graph model** by Caprara, Fischetti and Toth (2002)
- time-space graph $G = (V, A)$:



Time-expanded graph

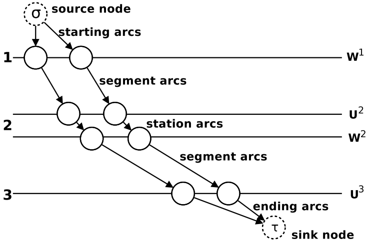
- **Time-Space Graph model** by Caprara, Fischetti and Toth (2002)
- time-space graph $G = (V, A)$:



- V: train departure W^i and arrival U^k times from/at stations ($i \in S \setminus \{s\}$, $k \in S \setminus \{1\}$)

Time-expanded graph

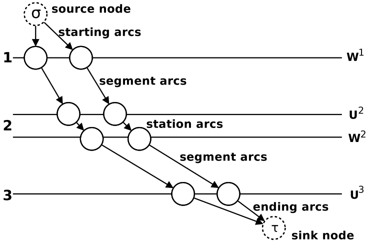
- **Time-Space Graph model** by Caprara, Fischetti and Toth (2002)
- time-space graph $G = (V, A)$:



- V: train departure W^i and arrival U^k times from/at stations ($i \in S \setminus \{s\}$, $k \in S \setminus \{1\}$)
- $A = A^1 \cup, \ldots, \cup A^{|T|}$: starting, segment (travel), station (stop) and ending arcs

Time-expanded graph

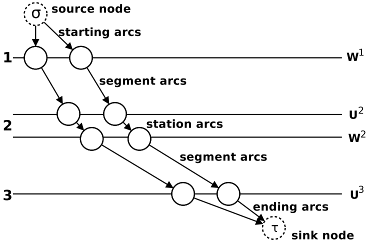
- **Time-Space Graph model** by Caprara, Fischetti and Toth (2002)
- time-space graph $G = (V, A)$:



- V: train departure W^i and arrival U^k times from/at stations ($i \in S \setminus \{s\}$, $k \in S \setminus \{1\}$)
- $A = A^1 \cup, \ldots, \cup A^{|T|}$: starting, segment (travel), station (stop) and ending arcs
- x_a: binary variable equal to 1 iff arc a is selected ($t \in T$, $a \in A^t$)

Time-expanded graph

- **Time-Space Graph model** by Caprara, Fischetti and Toth (2002)
- time-space graph $G = (V, A)$:



- V: train departure W^i and arrival U^k times from/at stations ($i \in S \setminus \{s\}$, $k \in S \setminus \{1\}$)
- $A = A^1 \cup, \dots, \cup A^{|T|}$: starting, segment (travel), station (stop) and ending arcs
- x_a: binary variable equal to 1 iff arc a is selected ($t \in T$, $a \in A^t$)

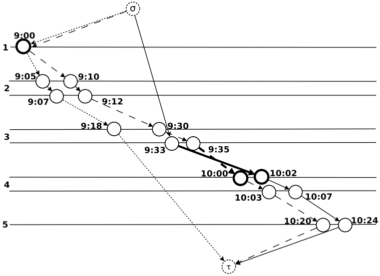
A path in G from σ to τ corresponds to a timetable for a train

An example

Stations	Ideal Timetable A		Ideal Timetable B		Ideal Timetable C	
	Arr. Time	Dep. Time	Arr. Time	Dep. Time	Arr. Time	Dep. Time
1		9:00		9:00		
2	9:05	9:07	9:10	9:12		
3	9:18		9:30	9:35		9:33
4			10:00	10:03	10:02	10:07
5			10:20		10:24	

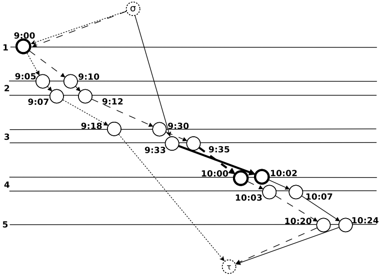
An example

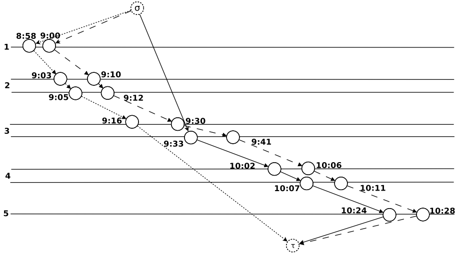
Stations	Ideal Timetable *A*		Ideal Timetable *B*		Ideal Timetable *C*	
	Arr. Time	Dep. Time	Arr. Time	Dep. Time	Arr. Time	Dep. Time
1		9:00		9:00		
2	9:05	9:07	9:10	9:12		
3	9:18		9:30	9:35		9:33
4			10:00	10:03	10:02	10:07
5			10:20		10:24	



An example

Stations	Ideal Timetable *A*		Ideal Timetable *B*		Ideal Timetable *C*	
	Arr. Time	Dep. Time	Arr. Time	Dep. Time	Arr. Time	Dep. Time
1		9:00		9:00		
2	9:05	9:07	9:10	9:12		
3	9:18		9:30	9:35		9:33
4			10:00	10:03	10:02	10:07
5			10:20		10:24	





ILP arc-model

$$\max \sum_{t \in T} \sum_{a \in A^t} p_a x_a$$

- p_a: profit associated with each arc $a \in A$: importance of the train minus penalties for the changes

ILP arc-model

$$\max \sum_{t \in T} \sum_{a \in A^t} p_a x_a$$

- p_a: profit associated with each arc $a \in A$: importance of the train minus penalties for the changes

$$\sum_{a \in \delta_t^+(\sigma)} x_a \leq 1, \qquad t \in T,$$

$$\sum_{a \in \delta_t^-(v)} x_a = \sum_{a \in \delta_t^+(v)} x_a, \qquad t \in T, v \in V \setminus \{\sigma, \tau\},$$

ILP arc-model

$$\max \sum_{t \in T} \sum_{a \in A^t} p_a x_a$$

- p_a: profit associated with each arc $a \in A$: importance of the train minus penalties for the changes

$$\sum_{a \in \delta_t^+(\sigma)} x_a \leq 1, \qquad t \in T,$$

$$\sum_{a \in \delta_t^-(v)} x_a = \sum_{a \in \delta_t^+(v)} x_a, \qquad t \in T, v \in V \setminus \{\sigma, \tau\},$$

$$\sum_{a \in C} x_a \leq 1, \qquad C \in \mathcal{C},$$

$$x_a \in \{0, 1\}, \qquad a \in A.$$

- $\mathcal{C}$: family of maximal subsets C of pairwise incompatible arcs

ILP path-model

$$\max \sum_{t \in T} \sum_{p \in \mathcal{P}^t} \pi_p x_p$$

- x_p: binary variable equal to 1 iff path p is selected $(t \in T,\ p \in \mathcal{P}^t)$
- π_p: profit associated with each path $p \in \mathcal{P}$: importance of the train minus penalties for the changes along the path

ILP path-model

$$\max \sum_{t \in T} \sum_{p \in \mathcal{P}^t} \pi_p x_p$$

- x_p: binary variable equal to 1 iff path p is selected $(t \in T,\ p \in \mathcal{P}^t)$
- π_p: profit associated with each path $p \in \mathcal{P}$: importance of the train minus penalties for the changes along the path

$$\sum_{P \in \mathcal{P}^t} x_p \leq 1, \qquad t \in T,$$

ILP path-model

$$\max \sum_{t \in T} \sum_{p \in \mathcal{P}^t} \pi_p x_p$$

- x_p: binary variable equal to 1 iff path p is selected $(t \in T,\ p \in \mathcal{P}^t)$
- π_p: profit associated with each path $p \in \mathcal{P}$: importance of the train minus penalties for the changes along the path

$$\sum_{P \in \mathcal{P}^t} x_p \leq 1, \qquad t \in T,$$

$$\sum_{p \in \mathcal{I}} x_p \leq 1, \qquad \mathcal{I} \in \mathcal{IP},$$

$$x_p \in \{0, 1\}, \qquad P \in \mathcal{P}.$$

- $\mathcal{IP}$: family of maximal subsets $\mathcal{I}$ of pairwise incompatible paths with incompatibility expressed separately for each station

Solution Methods

Lagrangian-based Heuristic Algorithm

- Proposed in Caprara, Fischetti and Toth (2002)
  extended to a network in Cacchiani, Caprara, Toth (2010)

Lagrangian-based Heuristic Algorithm

- Proposed in Caprara, Fischetti and Toth (2002)
  extended to a network in Cacchiani, Caprara, Toth (2010)
- Applied to the ILP arc-model

Lagrangian-based Heuristic Algorithm

- Proposed in Caprara, Fischetti and Toth (2002)
  extended to a network in Cacchiani, Caprara, Toth (2010)
- Applied to the ILP arc-model
- Incompatibility constraints are relaxed in a Lagrangian way

Lagrangian-based Heuristic Algorithm

- Proposed in Caprara, Fischetti and Toth (2002)
  extended to a network in Cacchiani, Caprara, Toth (2010)
- Applied to the ILP arc-model
- Incompatibility constraints are relaxed in a Lagrangian way
- Subgradient optimization to determine near-optimal Lagrangian multipliers
- Dynamic constraint-generation is used

Lagrangian-based Heuristic Algorithm

- Proposed in Caprara, Fischetti and Toth (2002)
  extended to a network in Cacchiani, Caprara, Toth (2010)
- Applied to the ILP arc-model
- Incompatibility constraints are relaxed in a Lagrangian way
- Subgradient optimization to determine near-optimal Lagrangian multipliers
- Dynamic constraint-generation is used
- During subgradient optimization, iteratively computes a heuristic solution:

Lagrangian-based Heuristic Algorithm

- Proposed in Caprara, Fischetti and Toth (2002)
  extended to a network in Cacchiani, Caprara, Toth (2010)
- Applied to the ILP arc-model
- Incompatibility constraints are relaxed in a Lagrangian way
- Subgradient optimization to determine near-optimal Lagrangian multipliers
- Dynamic constraint-generation is used
- During subgradient optimization, iteratively computes a heuristic solution:
  - Trains are ranked based on the Lagrangian profit (original train profit and Lagrangian penalties)

Lagrangian-based Heuristic Algorithm

- Proposed in Caprara, Fischetti and Toth (2002)
  extended to a network in Cacchiani, Caprara, Toth (2010)
- Applied to the ILP arc-model
- Incompatibility constraints are relaxed in a Lagrangian way
- Subgradient optimization to determine near-optimal Lagrangian multipliers
- Dynamic constraint-generation is used
- During subgradient optimization, iteratively computes a heuristic solution:
  - Trains are ranked based on the Lagrangian profit (original train profit and Lagrangian penalties)
  - Trains are scheduled one by one, choosing the conflict-free path with maximum Lagrangian profit $\rightarrow$ Dynamic Programming

Lagrangian-based Heuristic Algorithm

- Proposed in Caprara, Fischetti and Toth (2002)
  extended to a network in Cacchiani, Caprara, Toth (2010)
- Applied to the ILP arc-model
- Incompatibility constraints are relaxed in a Lagrangian way
- Subgradient optimization to determine near-optimal Lagrangian multipliers
- Dynamic constraint-generation is used
- During subgradient optimization, iteratively computes a heuristic solution:
  - Trains are ranked based on the Lagrangian profit (original train profit and Lagrangian penalties)
  - Trains are scheduled one by one, choosing the conflict-free path with maximum Lagrangian profit → Dynamic Programming
  - Local search procedures to improve the solution found

Branch-and-Cut-and-Price Algorithm

- Proposed in Cacchiani, Caprara, Toth (2008)

Branch-and-Cut-and-Price Algorithm

- Proposed in Cacchiani, Caprara, Toth (2008)
- Applied to the ILP path-model

Branch-and-Cut-and-Price Algorithm

- Proposed in Cacchiani, Caprara, Toth (2008)
- Applied to the ILP path-model
- Solve the LP-relaxation by column generation

Branch-and-Cut-and-Price Algorithm

- Proposed in Cacchiani, Caprara, Toth (2008)
- Applied to the ILP path-model
- Solve the LP-relaxation by column generation
- Pricing problem: determine an optimal path in the time-expanded graph → Dynamic Programming algorithms
- Constraint separation is applied

Branch-and-Cut-and-Price Algorithm

- Proposed in Cacchiani, Caprara, Toth (2008)
- Applied to the ILP path-model
- Solve the LP-relaxation by column generation
- Pricing problem: determine an optimal path in the time-expanded graph → Dynamic Programming algorithms
- Constraint separation is applied
- Branching is applied on the choice of the arcs in the graph

Branch-and-Cut-and-Price Algorithm

- Proposed in Cacchiani, Caprara, Toth (2008)
- Applied to the ILP path-model
- Solve the LP-relaxation by column generation
- Pricing problem: determine an optimal path in the time-expanded graph $\rightarrow$ Dynamic Programming algorithms
- Constraint separation is applied
- Branching is applied on the choice of the arcs in the graph
- Constructive heuristics: LP-based fixing of paths or arcs in the graph

Generalization to include additional real-life features

Skip-stop planning strategies

Skip-stop planning strategies[1]

[1] F. Jiang, V. Cacchiani, P. Toth. Train Timetabling by Skip-Stop Planning in Highly Congested Lines. Transportation Research Part B, 104, 149-174, 2017.

Skip-stop planning strategies[1]

- An additional change to the ideal timetables: it is possible to skip a stop (= not schedule a stop)

[1] F. Jiang, V. Cacchiani, P. Toth. Train Timetabling by Skip-Stop Planning in Highly Congested Lines. Transportation Research Part B, 104, 149-174, 2017.

Skip-stop planning strategies[1]

- An additional change to the ideal timetables: it is possible to skip a stop (= not schedule a stop)
- The case study is the high-speed double-track line Beijing-Shanghai in China

[1] F. Jiang, V. Cacchiani, P. Toth. Train Timetabling by Skip-Stop Planning in Highly Congested Lines. Transportation Research Part B, 104, 149-174, 2017.

Skip-stop planning strategies[1]

- An additional change to the ideal timetables: it is possible to skip a stop (= not schedule a stop)
- The case study is the high-speed double-track line Beijing-Shanghai in China
- The goal is to increase the capacity utilization of the corridor

[1] F. Jiang, V. Cacchiani, P. Toth. Train Timetabling by Skip-Stop Planning in Highly Congested Lines. Transportation Research Part B, 104, 149-174, 2017.

Skip-stop planning strategies[1]

- An additional change to the ideal timetables: it is possible to skip a stop (= not schedule a stop)
- The case study is the high-speed double-track line Beijing-Shanghai in China
- The goal is to increase the capacity utilization of the corridor
- Two sets of trains:
  - existing trains → actual feasible schedule
  - additional trains → ideal timetables

[1] F. Jiang, V. Cacchiani, P. Toth. Train Timetabling by Skip-Stop Planning in Highly Congested Lines. Transportation Research Part B, 104, 149-174, 2017.

Skip-stop planning strategies[1]

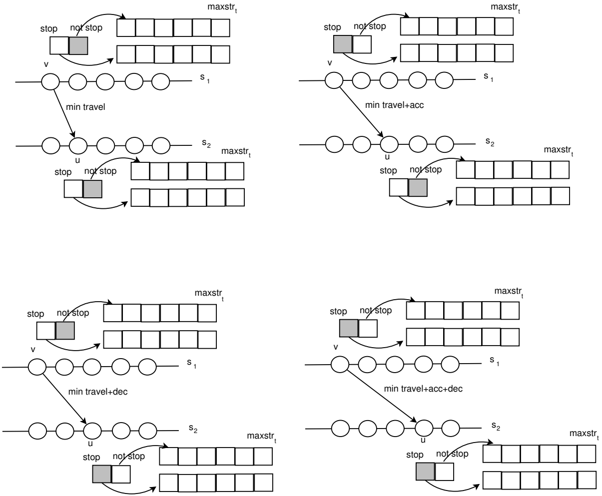
- An additional change to the ideal timetables: it is possible to skip a stop (= not schedule a stop)
- The case study is the high-speed double-track line Beijing-Shanghai in China
- The goal is to increase the capacity utilization of the corridor
- Two sets of trains:
  - existing trains → actual feasible schedule
  - additional trains → ideal timetables
- Acceleration and deceleration times must be taken into account
- maximum number of stops that can be cancelled per train
- no shift for the existing trains

[1] F. Jiang, V. Cacchiani, P. Toth. Train Timetabling by Skip-Stop Planning in Highly Congested Lines. Transportation Research Part B, 104, 149-174, 2017.

Solution method

- ILP arc-model with additional constraints
- Lagrangian-based heuristic algorithm
- Skip-stop strategies (with acceleration and deceleration) are handled by the Dynamic Programming algorithm

Dynamic Programming algorithm



Computational experiments - case study

- Beijing-Shanghai corridor: 29 stations
- 304 existing trains and 42 additional trains

Computational experiments - case study

- Beijing-Shanghai corridor: 29 stations
- 304 existing trains and 42 additional trains

- the maximum number of stops that can be cancelled per train is set to 1
- the maximum stretch is set according to the origin-destination of the train
- the maximum shift is set to ± 10, ± 20 or ± 30 minutes

Computational experiments adding new trains

#trains	shift	#sched	travel	stretch	profit	gap%	time (s)
346 sh±10	109	328(0)	45829	1132(737)	986571	3.72	3857
346 sh±20	294	333(0)	45827	1142(740)	996286	2.98	6153
346 sh±30	415	336(1)	45681	1161(689)	998975	2.95	9732

Table: No stop skipping

Computational experiments adding new trains

#trains	shift	#sched	travel	stretch	profit	gap%	time (s)
346 sh±10	109	328(0)	45829	1132(737)	986571	3.72	3857
346 sh±20	294	333(0)	45827	1142(740)	996286	2.98	6153
346 sh±30	415	336(1)	45681	1161(689)	998975	2.95	9732

Table: No stop skipping

#trains	shift	#sched	travel	stretch	profit	gap%	#sk	time (s)
346 sh±10	115	329(0)	45756	1096(662)	988525	3.66	2	4969
346 sh±20	279	334(0)	45731	1113(648)	997991	2.86	3	7510
346 sh±30	415	337(0)	45752	1192(664)	1003265	2.55	2	11112

Table: With stop skipping

Passenger-centric objectives

Passenger-centric objectives[2]

- Line Planning Problem → frequency of trains for each line in the network

[2] G.J. Polinder, V. Cacchiani, M.E. Schmidt, D. Huisman. An iterative heuristic for passenger-centric train timetabling with integrated adaption times. Computers & Operations Research, 142, 105740, 2022.

Passenger-centric objectives[2]

- Line Planning Problem → frequency of trains for each line in the network
- Regularity (synchronization) constraints between trains of the same line → to provide a regular service to passengers

[2] G.J. Polinder, V. Cacchiani, M.E. Schmidt, D. Huisman. An iterative heuristic for passenger-centric train timetabling with integrated adaption times. Computers & Operations Research, 142, 105740, 2022.

Passenger-centric objectives[2]

- Line Planning Problem → frequency of trains for each line in the network
- Regularity (synchronization) constraints between trains of the same line → to provide a regular service to passengers
- Passengers transfer between trains of different lines to reach their destination

[2] G.J. Polinder, V. Cacchiani, M.E. Schmidt, D. Huisman. An iterative heuristic for passenger-centric train timetabling with integrated adaption times. Computers & Operations Research, 142, 105740, 2022.

Passenger-centric objectives[2]

- Line Planning Problem → frequency of trains for each line in the network
- Regularity (synchronization) constraints between trains of the same line → to provide a regular service to passengers
- Passengers transfer between trains of different lines to reach their destination
- Therefore, trains of different lines have to be synchronized effectively

[2] G.J. Polinder, V. Cacchiani, M.E. Schmidt, D. Huisman. An iterative heuristic for passenger-centric train timetabling with integrated adaption times. Computers & Operations Research, 142, 105740, 2022.

Passenger-centric objectives

- Synchronize trains and achieve regularity $\rightarrow$ minimize the total perceived passenger travel time:

Passenger-centric objectives

- Synchronize trains and achieve regularity → minimize the total perceived passenger travel time: in-train time + transfer time + transfer penalty + adaption time (waiting time at the origin station)

Passenger-centric objectives

- Synchronize trains and achieve regularity $\rightarrow$ minimize the total perceived passenger travel time: in-train time + transfer time + transfer penalty + adaption time (waiting time at the origin station)

Passenger-centric timetabling

- We consider a time period H of one hour (periodic timetabling)

Passenger-centric timetabling

- We consider a time period H of one hour (periodic timetabling)
- Given passengers origin-destination (OD) pairs, we precompute a set of routes for each OD pair k
  (direct travel options and routes with up to a maximum number of transfer options)

Passenger-centric timetabling

- We consider a time period H of one hour (periodic timetabling)
- Given passengers origin-destination (OD) pairs, we precompute a set of routes for each OD pair k
  (direct travel options and routes with up to a maximum number of transfer options)
- d_k: number of passengers of OD pair k
- π: timetabling variables (time of departure and arrival events)

Passenger-centric timetabling

- We consider a time period H of one hour (periodic timetabling)
- Given passengers origin-destination (OD) pairs, we precompute a set of routes for each OD pair k
  (direct travel options and routes with up to a maximum number of transfer options)
- d_k: number of passengers of OD pair k
- π: timetabling variables (time of departure and arrival events)

$$\text{Min}_\pi \sum_{k \in \mathcal{OD}} d_k \cdot R_k(\pi)$$

Such that π is a feasible timetable

passengers take best routes with respect to π

$R_k(\pi)$ avg. perceived travel time of one passenger of OD-pair k $\quad \forall k \in \mathcal{OD}$

Average perceived travel time

$$R_k(\pi) = \frac{1}{d_k} \sum_{v \in V^k} d_k \cdot \frac{L_v^k}{H} \cdot (\gamma_w \cdot W_v^k + Y_v^k) = \frac{1}{H} \sum_{v \in V^k} L_v^k \cdot (\gamma_w \cdot W_v^k + Y_v^k)$$

- W_v^k: adaption time for a route departing in event v towards the destination of OD-pair k
- γ_w: weight of the adaption time
- Y_v^k: in-train time + transfer time on the best route from event v towards the destination of OD-pair k (includes penalties for transfers)
- V^k: set of departure events of these routes for OD pair k (from the origin of k)

Average perceived travel time

$$R_k(\pi) = \frac{1}{d_k} \sum_{v \in V^k} d_k \cdot \frac{L_v^k}{H} \cdot (\gamma_w \cdot W_v^k + Y_v^k) = \frac{1}{H} \sum_{v \in V^k} L_v^k \cdot (\gamma_w \cdot W_v^k + Y_v^k)$$

- W_v^k: adaption time for a route departing in event v towards the destination of OD-pair k
- γ_w: weight of the adaption time
- Y_v^k: in-train time + transfer time on the best route from event v towards the destination of OD-pair k (includes penalties for transfers)
- V^k: set of departure events of these routes for OD pair k (from the origin of k)
- Uniformly distributed passenger arrivals in the hour
- L_v^k: time interval between event v and the previous departure event of a route for OD-pair k

Average perceived travel time

$$R_k(\pi) = \frac{1}{d_k} \sum_{v \in V^k} d_k \cdot \frac{L_v^k}{H} \cdot (\gamma_w \cdot W_v^k + Y_v^k) = \frac{1}{H} \sum_{v \in V^k} L_v^k \cdot (\gamma_w \cdot W_v^k + Y_v^k)$$

- W_v^k: adaption time for a route departing in event v towards the destination of OD-pair k
- γ_w: weight of the adaption time
- Y_v^k: in-train time + transfer time on the best route from event v towards the destination of OD-pair k (includes penalties for transfers)
- V^k: set of departure events of these routes for OD pair k (from the origin of k)
- Uniformly distributed passenger arrivals in the hour
- L_v^k: time interval between event v and the previous departure event of a route for OD-pair k
- The total number of passengers of OD-pair k arriving in each interval L_v^k is $d_k \cdot \frac{L_v^k}{H}$

Solution method

- The problem can be modelled as a Periodic Event Scheduling Problem (PESP) (Serafini Ukovich 1989) with additional constraints to compute the $R_k(\pi)$

Solution method

- The problem can be modelled as a Periodic Event Scheduling Problem (PESP) (Serafini Ukovich 1989) with additional constraints to compute the $R_k(\pi)$
- The PESP model can be solved by a MIP-based heuristic but requires very long computing times for real-life instances

Solution method

- The problem can be modelled as a Periodic Event Scheduling Problem (PESP) (Serafini Ukovich 1989) with additional constraints to compute the $R_k(\pi)$
- The PESP model can be solved by a MIP-based heuristic but requires very long computing times for real-life instances
- Another approach: PESP without infrastructure constraints + algorithm based on a time-expanded formulation

Solution method

- The problem can be modelled as a Periodic Event Scheduling Problem (PESP) (Serafini Ukovich 1989) with additional constraints to compute the $R_k(\pi)$
- The PESP model can be solved by a MIP-based heuristic but requires very long computing times for real-life instances
- Another approach: PESP without infrastructure constraints + algorithm based on a time-expanded formulation
  - Remove from the PESP model all constraints on timetable feasibility $\rightarrow$ allow conflicts between trains

Solution method

- The problem can be modelled as a Periodic Event Scheduling Problem (PESP) (Serafini Ukovich 1989) with additional constraints to compute the $R_k(\pi)$
- The PESP model can be solved by a MIP-based heuristic but requires very long computing times for real-life instances
- Another approach: PESP without infrastructure constraints + algorithm based on a time-expanded formulation
  - Remove from the PESP model all constraints on timetable feasibility $\rightarrow$ allow conflicts between trains
  - Compute passenger-ideal timetables

Solution method

- The problem can be modelled as a Periodic Event Scheduling Problem (PESP) (Serafini Ukovich 1989) with additional constraints to compute the $R_k(\pi)$
- The PESP model can be solved by a MIP-based heuristic but requires very long computing times for real-life instances
- Another approach: PESP without infrastructure constraints + algorithm based on a time-expanded formulation
  - Remove from the PESP model all constraints on timetable feasibility $\rightarrow$ allow conflicts between trains
  - Compute passenger-ideal timetables
  - Make the timetables feasible (no conflicts between trains) by modifying the ideal ones as little as possible

Solution method

- The problem can be modelled as a Periodic Event Scheduling Problem (PESP) (Serafini Ukovich 1989) with additional constraints to compute the $R_k(\pi)$
- The PESP model can be solved by a MIP-based heuristic but requires very long computing times for real-life instances
- Another approach: PESP without infrastructure constraints + algorithm based on a time-expanded formulation
  - Remove from the PESP model all constraints on timetable feasibility → allow conflicts between trains
  - Compute passenger-ideal timetables
  - Make the timetables feasible (no conflicts between trains) by modifying the ideal ones as little as possible → Lagrangian Heuristic (LH)

Solution method

- The problem can be modelled as a Periodic Event Scheduling Problem (PESP) (Serafini Ukovich 1989) with additional constraints to compute the $R_k(\pi)$
- The PESP model can be solved by a MIP-based heuristic but requires very long computing times for real-life instances
- Another approach: PESP without infrastructure constraints + algorithm based on a time-expanded formulation
  - Remove from the PESP model all constraints on timetable feasibility → allow conflicts between trains
  - Compute passenger-ideal timetables
  - Make the timetables feasible (no conflicts between trains) by modifying the ideal ones as little as possible → Lagrangian Heuristic (LH)
  - Evaluate the impact on passenger perceived travel time → feedback mechanism

Feedback mechanism

- After the timetable has been made feasible, some OD-pairs may have a bad perceived travel time

Feedback mechanism

- After the timetable has been made feasible, some OD-pairs may have a bad perceived travel time
- We identify the OD-pairs that got the largest worsening

Feedback mechanism

- After the timetable has been made feasible, some OD-pairs may have a bad perceived travel time
- We identify the OD-pairs that got the largest worsening
- We modify the profit structure by penalizing more the shift at origin and intermediate stations where the service was not regular

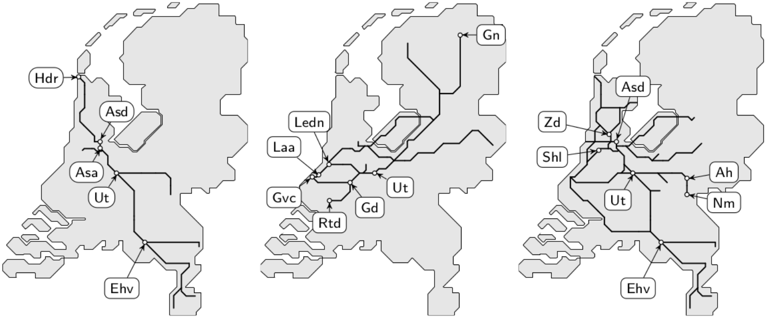
Feedback mechanism

- After the timetable has been made feasible, some OD-pairs may have a bad perceived travel time
- We identify the OD-pairs that got the largest worsening
- We modify the profit structure by penalizing more the shift at origin and intermediate stations where the service was not regular
- Apply again the Lagrangian Heuristic

Computational experiments - case study

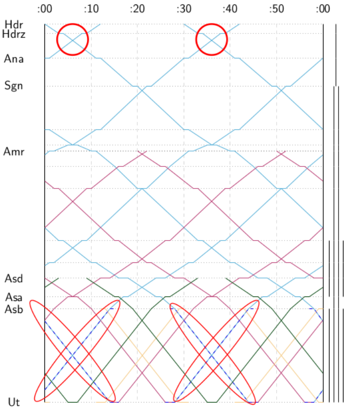
Three case studies of the Dutch railway network (lines of 2019) and one hour period:

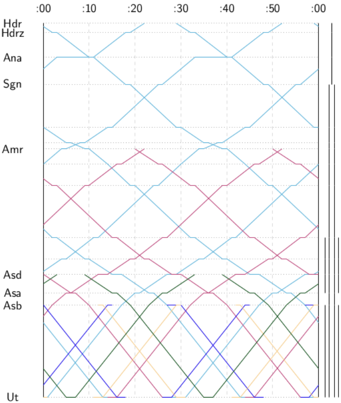
- A2: 34 stations, 20 trains, 891 OD-pairs.
- Rotterdam-Groningen: 77 stations, 60 trains, 3810 OD-pairs.
- Extended A2: 140 stations, 88 trains, 11121 OD-pairs.



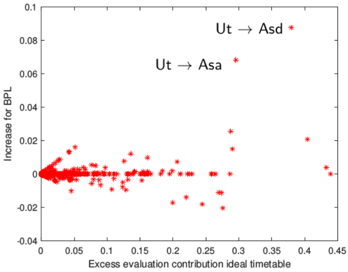
(a) A2 (b) Rotterdam-Groningen (c) Extended A2

A2 instance: ideal vs feasible timetable

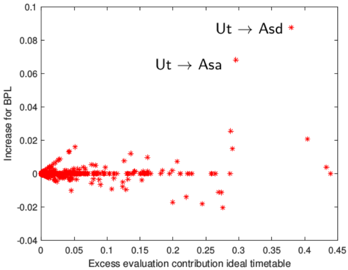


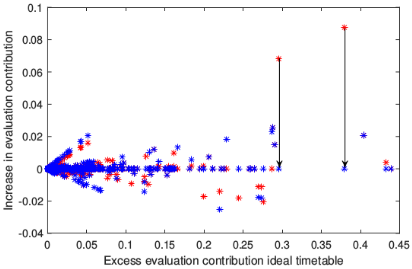


A2 instance: before and after feedback



A2 instance: before and after feedback





Comparison

Instance	Approach	Evaluation value	Time (hours)
A2	Ideal + LH	100.18	$2 + 0.03$

Comparison

Instance	Approach	Evaluation value	Time (hours)
	Ideal + LH	100.18	$2 + 0.03$
	Ideal + LH + FB	100.10	$2 + 0.11$
A2			

Comparison

Instance	Approach	Evaluation value	Time (hours)
A2	Ideal + LH	100.18	2 + 0.03
	Ideal + LH + FB	100.10	2 + 0.11
	Full PESP		
	- After 2.11 hours	105.80	2.11

Comparison

Instance	Approach	Evaluation value	Time (hours)
A2	Ideal + LH	100.18	2 + 0.03
	Ideal + LH + FB	100.10	2 + 0.11
	Full PESP		
	- After 2.11 hours	105.80	2.11
	- After 8 hours	104.88	8

Comparison

Instance	Approach	Evaluation value	Time (hours)
A2	Ideal + LH	100.18	$2 + 0.03$
	Ideal + LH + FB	100.10	$2 + 0.11$
	Full PESP		
	- After 2.11 hours	105.80	2.11
	- After 8 hours	104.88	8
	Lower bound CPLEX	97.09	

Comparison

Instance	Approach	Evaluation value	Time (hours)
A2	Ideal + LH	100.18	$2 + 0.03$
	Ideal + LH + FB	100.10	$2 + 0.11$
	Full PESP		
	- After 2.11 hours	105.80	2.11
	- After 8 hours	104.88	8
	Lower bound CPLEX	97.09	
Rotterdam Groningen	Ideal + LH	100.59	$4 + 0.06$
	Ideal + LH + FB	100.55	$4 + 0.18$
	Full PESP		
	- After 4.18 hours	105.64	4.18
	- After 16 hours	103.69	16
	Lower bound CPLEX	92.72	

Comparison

Instance	Approach	Evaluation value	Time (hours)
A2	Ideal + LH	100.18	2 + 0.03
	Ideal + LH + FB	100.10	2 + 0.11
	Full PESP		
	- After 2.11 hours	105.80	2.11
	- After 8 hours	104.88	8
	Lower bound CPLEX	97.09	
Rotterdam Groningen	Ideal + LH	100.59	4 + 0.06
	Ideal + LH + FB	100.55	4 + 0.18
	Full PESP		
	- After 4.18 hours	105.64	4.18
	- After 16 hours	103.69	16
	Lower bound CPLEX	92.72	
Extended A2	Ideal + LH	101.51	4 + 0.14
	Ideal + LH + FB	101.28	4 + 0.49
	Full PESP		
	- After 4.49 hours	-	4.49
	- After 16 hours	-	16
	Lower bound CPLEX	93.00	

Conclusion

- Time-expanded formulations can be effectively used in heuristic algorithms for real-life case studies
- Efficient timetables can be computed in planning

Conclusion

- Time-expanded formulations can be effectively used in heuristic algorithms for real-life case studies
- Efficient timetables can be computed in planning
- Delays and disruptions can still occur in real-time

Conclusion

- Time-expanded formulations can be effectively used in heuristic algorithms for real-life case studies
- Efficient timetables can be computed in planning
- Delays and disruptions can still occur in real-time
- $\rightarrow$ Andrea D'Ariano will talk about efficient methods for train rescheduling during rail operations

Thank you for your attention