The Restricted Modulo Network Simplex Method for Integrated Periodic Timetabling and Passenger Routing

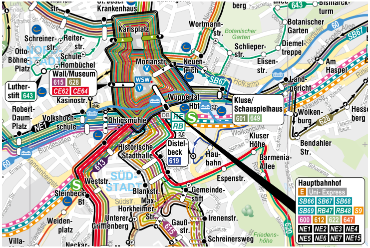
Fabian Löbel, Niels Lindner, Ralf Borndörfer

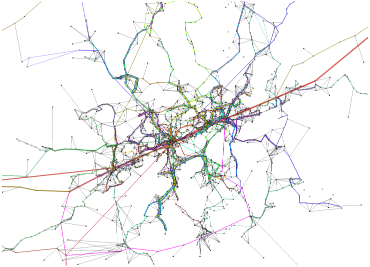
Zuse Institute Berlin

Operations Research 2019 @ Dresden

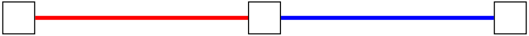
September 5, 2019

Timetable Optimization in Wuppertal



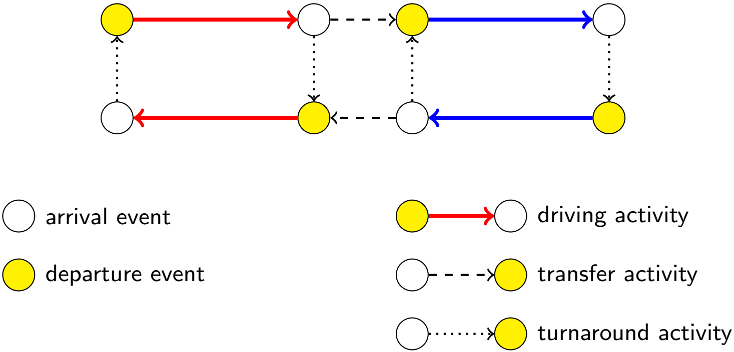


Line Networks → Periodic Timetables



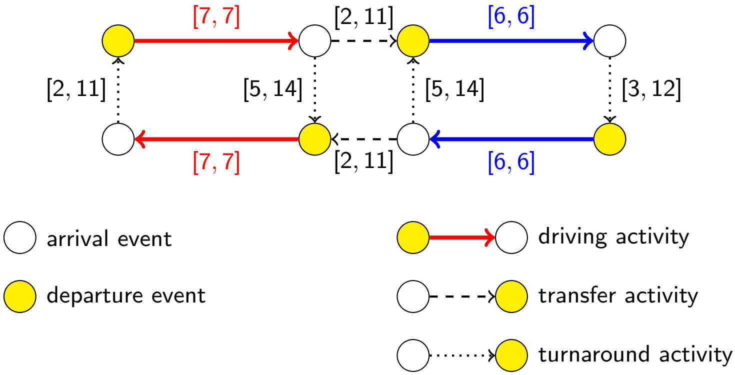
two lines meeting at a common station

Line Networks → Periodic Timetables



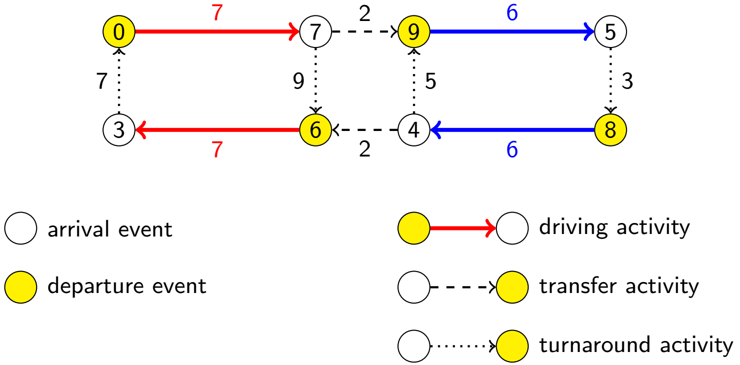
event-activity network model

Line Networks → Periodic Timetables



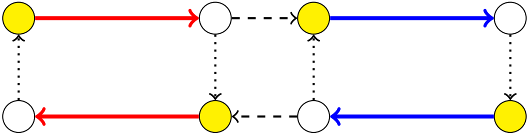
periodic timetabling instance (unweighted), period time $T = 10$

Line Networks → Periodic Timetables



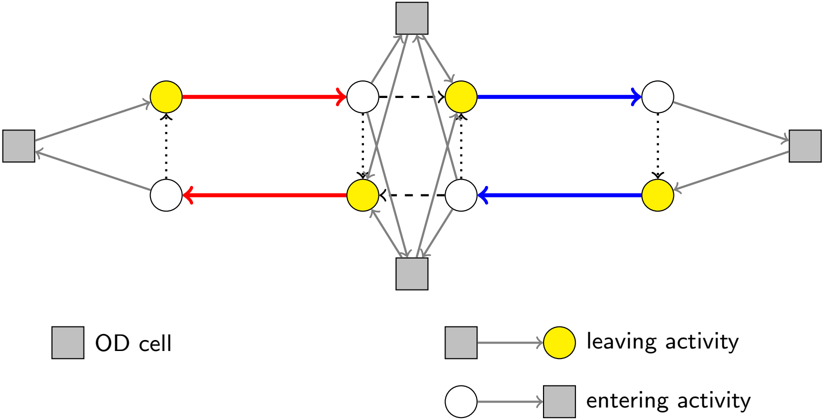
periodic timetable, period time $T = 10$

Integrating Passenger Routing



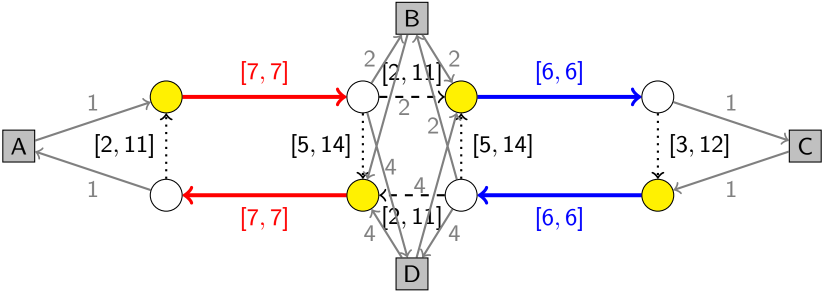
event-activity network model

Integrating Passenger Routing



extended event-activity network

Integrating Passenger Routing



OD matrix	A	B	C	D
A	0	53	81	72
B	59	0	34	22
C	77	34	0	15
D	70	25	18	0

integrated periodic timetabling instance, period time $T = 10$

Integrated Periodic Timetabling Problem

Given

- a digraph G with vertex set $V = V_{\text{event}} \,\dot\cup\, V_{\text{cell}}$ and arc set $A = A_{\text{activity}} \,\dot\cup\, A_{\text{cell}}$, where $A_{\text{activity}} = A \cap (V_{\text{event}} \times V_{\text{event}})$,
- a period time $T \in \mathbb{N}$,
- lower and bounds $\ell, u \in \mathbb{Z}^{A}_{\geq 0}$, $u \geq \ell$, with $u_a = \ell_a$ for $a \in A_{\text{cell}}$,
- an OD matrix $(d_{st}) \in \mathbb{Z}^{V_{\text{cell}} \times V_{\text{cell}}}_{\geq 0}$,

Integrated Periodic Timetabling Problem

Given

- a digraph G with vertex set $V = V_{\text{event}} \dot\cup V_{\text{cell}}$ and arc set $A = A_{\text{activity}} \dot\cup A_{\text{cell}}$, where $A_{\text{activity}} = A \cap (V_{\text{event}} \times V_{\text{event}})$,
- a period time $T \in \mathbb{N}$,
- lower and bounds $\ell, u \in \mathbb{Z}^A_{\geq 0}$, $u \geq \ell$, with $u_a = \ell_a$ for $a \in A_{\text{cell}}$,
- an OD matrix $(d_{st}) \in \mathbb{Z}_{\geq 0}^{V_{\text{cell}} \times V_{\text{cell}}}$,

the **Integrated Periodic Timetabling Problem (IPTP)** is to find a vector $x \in \mathbb{Z}^A$ and a path p_{st} for each OD pair (s, t) s.t.

- $\ell \leq x \leq u$,
- when restricted to $G[V_{\text{event}}]$, x is the periodic tension of a periodic timetable with period time T,
- the intermediate vertices of p_{st} are all in V_{event},
- the total travel time $\sum_{(s,t) \in V_{\text{cell}} \times V_{\text{cell}}} d_{st} \cdot x(p_{st})$ is minimum.

Integrated Periodic Timetabling Problem

More Notation

- Γ: cycle matrix of an integral cycle basis of $G[V_{\text{event}}]$
- D: set of all OD pairs (s,t) with $d_{st} > 0$
- P_{st}: set of all s-t-paths having intermediate vertices only in V_{event}

Mixed Integer Bilinear Program

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{a \in A} w_a x_a \\
\text{s.t.} \quad & \Gamma x \equiv 0 \mod T, \\
& x_a \in [\ell_a, u_a], && a \in A, \\
& w_a = \sum_{(s,t) \in D} \sum_{p \in P_{st} : a \in p} d_{st} f_p, && a \in A, \\
& \sum_{p \in P_{st}} f_p = 1, && (s,t) \in D, \\
& f_p \in \{0,1\}, && p \in P_{st}, (s,t) \in D.
\end{aligned}
$$

Integrated Periodic Timetabling Problem

Some Remarks

- When the path variables f_p are fixed, IPTP becomes a standard periodic timetabling problem (PESP).
- When the tensions x_a are fixed, IPTP becomes a multi-source multi-target shortest path problem.
- In particular, one could relax $f_p \in \{0, 1\}$ to $f_p \in [0, 1]$.
- There are integral variables hidden in the constraint $\Gamma x \equiv 0 \bmod T$.

Integrated Periodic Timetabling Problem

Some Remarks

- When the path variables f_p are fixed, IPTP becomes a standard periodic timetabling problem (PESP).
- When the tensions x_a are fixed, IPTP becomes a multi-source multi-target shortest path problem.
- In particular, one could relax $f_p \in \{0, 1\}$ to $f_p \in [0, 1]$.
- There are integral variables hidden in the constraint $\Gamma x \equiv 0 \bmod T$.

Solving the MIP

- The objective function can be linearized by time expansion (XPESP), turning IPTP into a resource-constrained shortest path problem.
- Passenger paths can be added by column generation.
- Problem: Memory issues on city-sized instances!

Modulo Network Simplex

Idea (Nachtigall, Opitz, 2008)

- Assume that $G[V_{\text{event}}]$ is weakly connected with n events and m activities.
- The projection of the IPTP/PESP polyhedron onto the x_a-space for $a \in A_{\text{activity}}$ is the polytope

$$\mathcal{P} := \operatorname{conv}\{x \in \mathbb{Z}^{A_{\text{activity}}} \mid \Gamma x = Tz, \ell \leq x \leq u\}$$

  for some integer vector z.
- $\mathcal{P}$ is m-dimensional, so any vertex satisfies $m - \operatorname{rank}(\Gamma)$ linearly independent inequalities $\ell_a \leq x_a$ or $x_a \leq u_a$ with equality.
- Linear indepence means that these arcs do not contain a cycle, as the rows of Γ are a basis for the cycle space.
- $m - \operatorname{rank}(\Gamma) = n - 1$, so $x_a = \ell_a$ or $x_a = u_a$ holds along the arcs of some spanning tree.

Modulo Network Simplex

Theorem (Borndörfer, Hoppmann, Karbstein, Löbel, 2017)

Any feasible IPTP instance has an optimal solution with a spanning tree structure, i.e., there is a spanning tree $\mathcal{T}$ of $G[V_{event}]$ such that $x_a \in \{\ell_a, u_a\}$ holds for all activities in T.

Modulo Network Simplex

Theorem (Borndörfer, Hoppmann, Karbstein, Löbel, 2017)

Any feasible IPTP instance has an optimal solution with a spanning tree structure, i.e., there is a spanning tree $\mathcal{T}$ of $G[V_{event}]$ such that $x_a \in \{\ell_a, u_a\}$ holds for all activities in T.

Modulo Network Simplex Outline

1. Start with an initial spanning tree structure $\mathcal{T} = \mathcal{T}_\ell \dot\cup \mathcal{T}_u$.
2. While the objective improves, choose a co-tree arc a' and a tree-arc a on its fundamental cycle w.r.t. $\mathcal{T}$, remove a from $\mathcal{T}$ and insert a' if this is feasible.
3. Try to escape local optima. If successful, go back to 2, otherwise return.

Integrated Modulo Network Simplex

Observation

Computing the change in objective value requires rerouting all passengers.

Integrated Modulo Network Simplex

Observation

Computing the change in objective value requires rerouting all passengers.

Complexity

Rerouting passengers is a shortest path problem ...

- ... for each OD pair
- ... for each non-zero entry of the modulo network simplex tableau

Integrated Modulo Network Simplex

Observation

Computing the change in objective value requires rerouting all passengers.

Complexity

Rerouting passengers is a shortest path problem ...

- ... for each OD pair
- ... for each non-zero entry of the modulo network simplex tableau

Example

Wuppertal (full): 45 158 OD pairs × 74 387 co-tree arcs × 4 313 tree arcs

Integrated Modulo Network Simplex

Observation

Computing the change in objective value requires rerouting all passengers.

Complexity

Rerouting passengers is a shortest path problem ...

- ... for each OD pair
- ... for each non-zero entry of the modulo network simplex tableau

Example

Wuppertal (full): 45 158 OD pairs $\times$ 74 387 co-tree arcs $\times$ 4 313 tree arcs

Conclusion

The shortest path computations should be very fast and rarely used.

Integrated Modulo Network Simplex

Rerouting Variants (Löbel, 2017)

- *integrated*
  for every tableau non-zero
- *hybrid*
  after changing the tree structure
- *iterative*
  when objective has stopped improving
- *fixed*
  after termination of modulo network simplex

high quality
↕
high performance

Integrated Modulo Network Simplex

Rerouting Variants (Löbel, 2017)

- *integrated*
  for every tableau non-zero
- *hybrid*
  after changing the tree structure
- *iterative*
  when objective has stopped improving
- *fixed*
  after termination of modulo network simplex

high quality
↕
high performance

Some OD Preprocessing Ideas

- neglect OD pairs with a direct connection
- neglect OD pairs with low demand
- neglect OD pairs with short worst-case connections

Restricting Passenger Paths

Question

Can we speed up rerouting by decreasing the number of possible paths?

Restricting Passenger Paths

Question

Can we speed up rerouting by decreasing the number of possible paths?

Wuppertal: Some Experiments (Kühner, 2018)

max. number of transfers	0	1	2	3
percentage of OD pairs	24.1	83.7	99.7	100.0
avg. number of paths per OD pair	0.5	6.0	42.3	183.4

percentage of timetables	50	70	80	90	95	100
avg. number of paths per OD pair	1.5	2.4	3.2	4.7	6.5	15.2

Restricting Passenger Paths

Question

Can we speed up rerouting by decreasing the number of possible paths?

Wuppertal: Some Experiments (Kühner, 2018)

max. number of transfers	0	1	2	3
percentage of OD pairs	24.1	83.7	99.7	100.0
avg. number of paths per OD pair	0.5	6.0	42.3	183.4

percentage of timetables	50	70	80	90	95	100
avg. number of paths per OD pair	1.5	2.4	3.2	4.7	6.5	15.2

Idea

Restrict rerouting to a small pool of short paths with ≤ 2 transfers.

Computational Experiments

Set-up

- Test instances: subnetworks of Wuppertal, Karlsruhe, Dutch Intercity
- Initial solution: reference timetable
- Variants: integrated, hybrid, iterative, fixed and restricted integrated
- Restricted integrated modulo network simplex:
  20 shortest paths w.r.t. lower bounds with at most two transfers,
  update pool with actual shortest paths after every base change
- Pivot rule: steepest descent, parallel implementation
- Escaping local optima: single-node cuts
- Time limit: 2 hours wall time

Instances and Results

Instance	Dutch IC	Wuppertal 11	Karlsruhe
Stations	23	82	462
Lines	40	56	115
Events	448	2 166	10 497
Activities	3 791	28 733	84 255
OD pairs	158	21 764	135 177
Initial solution	900 395	1 519 747	4 668 327
Lower bound	868 074	1 373 190	3 844 703
Fixed	883 378	1 503 433	4 568 981
Iterative	883 508	1 502 939	4 563 224
Hybrid	879 213	1 504 797	4 564 298
Integrated	868 647	1 501 858	4 668 327
Restricted Integrated	868 275	1 471 608	4 642 170

Algorithm Analysis

Instance	Method	Wall time [s]	CPU time [s]	Cuts	Gap [%]
Dutch IC	Fixed	5	26	22	1.76
	Iterative	6	37	24	1.78
	Hybrid	6	35	26	1.28
	Integrated	1 023	5 959	45	0.07
	Restricted	36	200	43	0.02
Wuppertal 11	Fixed	61	260	12	9.48
	Iterative	62	288	11	9.45
	Hybrid	53	224	10	9.58
	Integrated	7 200	17 676	3	9.37
	Restricted	7 200	16 986	18	7.17
Karlsruhe	Fixed	675	3 290	35	18.84
	Iterative	951	3 735	34	18.69
	Hybrid	1 182	3 538	32	18.72
	Integrated	7 200	50 473	0	21.42
	Restricted	7 200	30 036	1	20.74

Conclusions

Restricted Integrated Modulo Network Simplex

- superior to the non-restricted integrated method both in running time and quality
- superior to the other methods on small to medium instances
- too slow on larger instances

Conclusions

Restricted Integrated Modulo Network Simplex

- superior to the non-restricted integrated method both in running time and quality
- superior to the other methods on small to medium instances
- too slow on larger instances

Questions

- Can a finer path analysis improve the restriced integrated modulo network simplex further?
- Is it possible to derive better lower bounds, e.g., by a working mixed integer programming approach?

The Restricted Modulo Network Simplex Method for Integrated Periodic Timetabling and Passenger Routing

Fabian Löbel, Niels Lindner, Ralf Borndörfer

Zuse Institute Berlin

Berlin Mathematics Research Center MATH+

ZIB

Operations Research 2019 @ Dresden

September 5, 2019