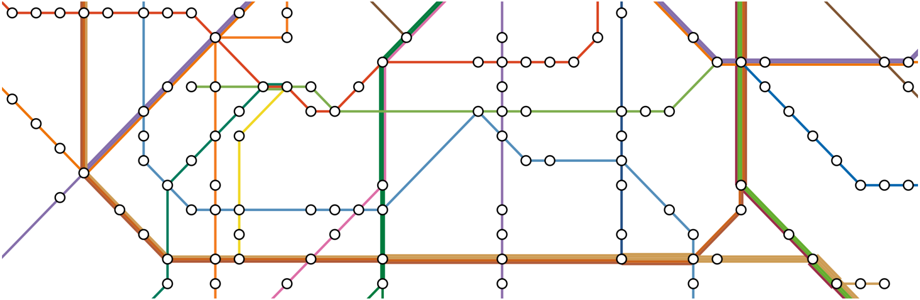
Mathematical Aspects of Public Transportation Networks

Niels Lindner



June 11, 2018

Chapter 3

Periodic Timetabling

§3.4 Modulo Network Simplex

Periodic timetabling

Let $(\mathcal{E}, T, \ell, u, w)$ be a PESP instance, Γ a cycle matrix of an integral cycle basis of $\mathcal{E}$, $\mu = \mu(\mathcal{E})$.

Cycle&slack MIP formulation

$$\begin{array}{lrr} \text{Minimize} & w^T y & \\ \text{s.t.} & \Gamma y - Tz = -\Gamma \ell, & \text{(cycle periodicity)} \\ & 0 \leq y \leq u - \ell, & \text{(periodic slack)} \\ & z \in \mathbb{Z}^\mu. & \text{(cycle offset)} \end{array}$$

Definition

Let $m = \#E(\mathcal{E})$. Define the **integer slack polytope** Q as

$$Q := \operatorname{conv}\{(y, z) \in \mathbb{R}^m \times \mathbb{Z}^\mu \mid \Gamma y - Tz = -\Gamma \ell,\ 0 \leq y \leq u - \ell\}.$$

Then any optimal objective value for the above MIP is attained by some vertex of Q.

Spanning tree structures

Assume that $\mathcal{E}$ is weakly connected.

Theorem (Nachtigall, 1996)

Let $(y, z) \in Q$. Then (y, z) is a vertex of Q if and only if there is a spanning tree T of $\mathcal{E}$ s.t. for all $e \in E(T)$ holds $y_e = 0$ or $y_e = u_e - \ell_e$.

Definition

A **spanning tree structure** is a spanning tree T together with a decomposition $E(T) = T_\ell \dot\cup T_u$.

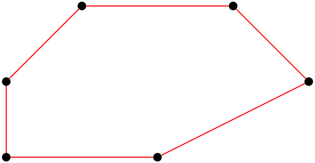
Remarks

- For every optimal objective value, there is a PESP solution coming from a spanning tree structure.
- A spanning tree structure gives always a periodic timetable, however, this does not necessarily respect the bounds.

Vertices of polyhedra

Let A be a $p \times q$-matrix of rank q, and let $b \in \mathbb{R}^p$. Consider the polyhedron

$$P := \{x \in \mathbb{R}^q \mid Ax \leq b\}.$$



Theorem

For a point $x \in P$, the following are equivalent:

(1) *x is a vertex of P.*

(2) *x cannot be written as $x = \lambda x_1 + (1 - \lambda) x_2$ for all $\lambda \in (0, 1)$ and $x_1 \neq x_2 \in P$.*

(3) *There is an index set $I \subseteq \{1, \dots, p\}$ such that $\mathrm{rk}(A_I) = q$ and $A_I x = b_I$.*

More polytopes

Definition

Define for $z \in \mathbb{Z}^\mu$ the polytope

$$Q_z := \{y \in \mathbb{R}^m \mid \Gamma y = Tz - \Gamma \ell,\ 0 \leq y \leq u - \ell\}.$$

Lemma

Suppose that $(y, z) \in Q$. Then (y, z) is a vertex of Q if and only if y is a vertex of Q_z.

Proof.

For $y_1, y_2 \in Q_z$ and $\lambda \in [0, 1]$,

$$y = \lambda y_1 + (1 - \lambda) y_2 \quad \Leftrightarrow \quad (y, z) = \lambda(y_1, z) + (1 - \lambda)(y_2, z). \qquad \square$$

Proof of the spanning tree structure theorem

Proof.

($\Leftarrow$) Let $(y, z) \in Q$ be a feasible PESP solution. Suppose that there is a spanning tree T satisfying $y_e = 0$ or $y_e = u_e - \ell_e$ for all $e \in E(T)$.

Since (y, z) is feasible, we have $\Gamma y = Tz - \Gamma\ell$. Moreover, we have $y_e = 0$ or $y_e = u_e - \ell_e$ for the $m - \mu$ edges of $E(T)$. In particular, $\mu + m - \mu = m$ inequalities of the system

$$\begin{aligned} \Gamma y &= Tz - \Gamma\ell \\ Iy &\geq 0 \\ Iy &\leq u - \ell \end{aligned}$$

are satisfied with equality, where I is the $m \times m$-identity matrix.

It remains to show that these m rows form a rank m subsystem. To this end, observe that if any of the rows corresponding to the spanning tree structure is a non-trivial linear combination of the rows of Γ, then this produces a non-trivial cycle inside T. Conclusion: y is a vertex of Q_z.

Proof of the spanning tree structure theorem

Proof (cont.)

($\Rightarrow$) Let (y, z) be a vertex of Q, i.e., y is a vertex of Q_z. Then there is a rank m subsystem of

$$\begin{aligned} \Gamma y &= Tz - \Gamma\ell \\ Iy &\geq 0 \\ Iy &\leq u - \ell \end{aligned}$$

satisfied by y with equality.

In particular, we can choose $m - \mu = n - 1$ edges e with $y_e = 0$ or $y_e = u_e - \ell_e$ such that the corresponding rows are linearly independent from the rows of Γ, where $n = \#V(\mathcal{E})$. This means that the subgraph induced by these edges does not contain any cycle, and is hence a forest. But any forest with $n - 1$ edges and at most n vertices has precisely n vertices and is hence a spanning tree. $\square$

Orthogonality of incidence and cycle matrix

In fact, we made use of the following two facts:

Let G be a weakly connected digraph on n vertices, m edges, and $\mu = m - n + 1$. Consider the incidence matrix $A \in \{-1, 0, 1\}^{n \times m}$ and a cycle matrix $\Gamma \in \{-1, 0, 1\}^{\mu \times m}$ of an arbitrary cycle basis of G.

Lemma
There is a one-to-one correspondence between invertible $(n-1) \times (n-1)$-submatrices of A and spanning trees of G.

Lemma
$\Gamma \cdot A^T = 0$.

Proof.
Exercise. □

Minimum cost network flows

The **minimum cost network flow** problem is the following:

Let $G = (V, E)$ be a directed graph with incidence matrix $A \in \{-1, 0, 1\}^{V \times E}$. Given non-negative cost and capacities $c, u \in \mathbb{R}^E$ and a balance $b \in \mathbb{R}^V$ satisfying $\sum_{v \in V} b_v = 0$, find

$$\min\{c^T f \mid Af = b,\ 0 \leq f \leq u\}.$$

Theorem

Let f be a feasible flow. Then f is a vertex of $\{f \in \mathbb{R}^E \mid Af = b,\ 0 \leq f \leq u\}$ if and only if there is a spanning tree T such that for all $e \notin E(T)$ holds $f_e = 0$ or $f_e = u_e$.

Proof.

As before: Linear subsystem of rank m satisfied with equality
$\leftrightarrow$ $m - n + 1$ edges with $f_e \in \{0, u_e\}$, linearly independent from $E(T)$. $\square$

Network simplex algorithm

The *network simplex algorithm* exploits the fact that there is always an optimal solution coming from a spanning tree structure.

Network Simplex (Dantzig, 1951)

1. Compute an initial feasible spanning tree structure $E(T) = T_\ell \dot\cup T_u$.
2. While there is an improving co-tree edge e:
   - Select one such edge e.
   - Add e to T.
   - Remove some other edge of the fundamental circuit of e from T.
   - Update the tree structure.

Theorem (Orlin, 1995)

The network simplex algorithm can be improved to a strongly polynomial-time algorithm solving minimum cost network flow problems to optimality.

Modulo network simplex

Modulo Network Simplex (Nachtigall/Opitz, 2008)

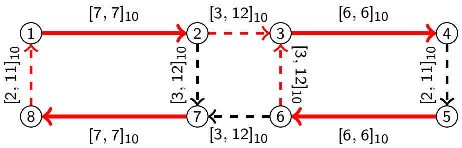
1. Compute an initial feasible spanning tree structure $E(T) = T_\ell \dot\cup T_u$.
2. While there is an improving co-tree edge e:
   - Select one such edge e.
   - Add e to T.
   - Remove some other edge of the fundamental circuit of e from T.
   - Update the tree structure.

Remarks

- There are several pivoting strategies for choosing the edges.
- In contrast to minimum cost network flows, there is no easy optimality condition. Hence the algorithm might get stuck in a local optimum.
- In practice, this happens a lot.

Modulo network simplex: Example

Consider the following PESP instance $(\mathcal{E}, T, \ell, u, w)$ with $T = 10$ and $w = 1$ for all transfer activities.



The cycle matrix w.r.t. the red spanning tree is

	12	23	34	56	63	78	81	27	45	67	y
γ_{27}	1	0	0	0	0	1	1	1	0	0	1
γ_{45}	0	0	1	1	1	0	0	0	1	0	3
γ_{67}	1	1	0	0	-1	1	1	0	0	1	1

Start with the tree structure $T_\ell = \{12, 23, 34, 56, 63, 78, 81\}$ and $T_u = \emptyset$. This produces a feasible initial solution with slack 5.

Modulo network simplex: Example

The possible moves are given by the non-zero entries of the cycle matrix.

	12	23	34	56	63	78	81	27	45	67	y
γ_{27}	1	0	0	0	0	1	1	1	0	0	1
γ_{45}	0	0	1	1	1	0	0	0	1	0	3
γ_{67}	1	1	0	0	-1	1	1	0	0	1	1

Suppose we replace the tree edge 81 by the co-tree edge 27 with $y_{27} = 0$. Then γ_{27} is replaced by γ_{81}, but cycle and slack do not change. The cycle γ_{45} remains untouched. Finally, γ_{67} loses 78, 81 and 12, and receives 27 in backward direction. The slack on γ_{67} reduces to 0.

	12	23	27	34	56	63	78	81	45	67	y
γ_{81}	1	0	1	0	0	0	1	1	0	0	1
γ_{45}	0	0	0	1	1	1	0	0	1	0	3
γ_{67}	0	1	-1	0	0	-1	0	0	0	1	0

Modulo network simplex: Remarks

Remarks

- In principle, one has to compute the change in weighted slack for every feasible move.
- The usual rule is "steepest descent": Take the biggest improvement in terms of the objective value.
- Another rule is to choose a column (tree edge) with minimal number of non-zero entries, thereby reducing the number of touched cycles.
- The algorithm terminates if no improving move is found.
- A strategy to escape local minima is to not only look for *fundamental cuts* of the spanning trees, but also for more general cuts. However, this requires a recomputation of the spanning tree structure.
- In practice, the modulo network simplex achieves a few big improvements in its first iterations and only minor improvements later on. On the other hand, MIP software needs more time, but produces better quality solutions in the long run.

Chapter 3

Periodic Timetabling

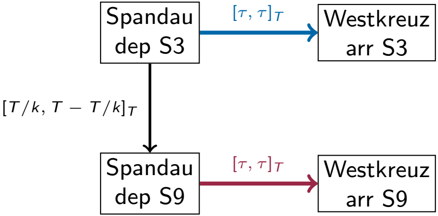
§3.5 Further topics

Headway activities: Equidistribution

Suppose we have k lines using the same edge of a line network with the same travel time τ.

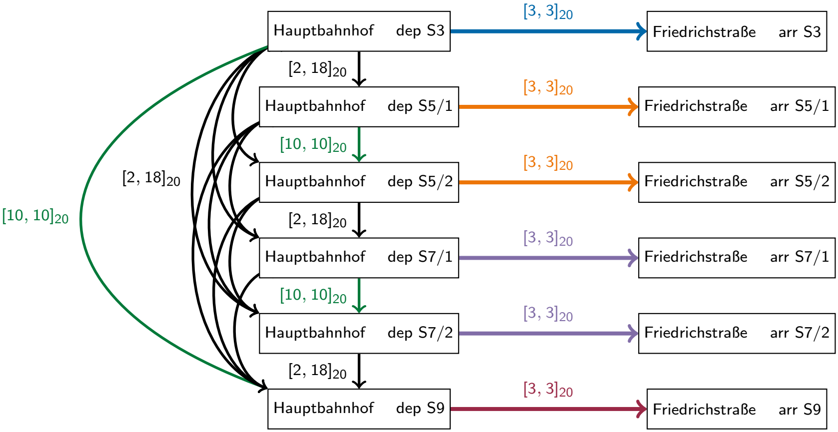
It is often desired to distribute these lines equidistantly, i.e., the time between two subsequent departures should be precisely T/k.

This is modeled as follows between every pair of such lines:



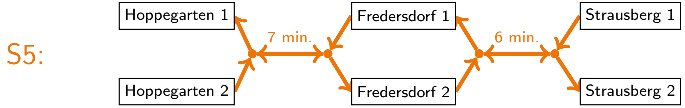
Headway activities: Track headway

The same method models safety constraints, e.g., if there must be a minimum gap between two consecutive trains on a track:

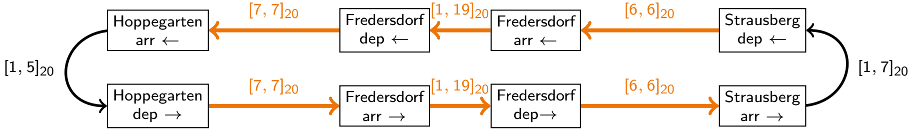


Headway activities: Single tracks

Sometimes a track is used by trains in both directions. A train is not allowed to enter the track while another train uses the track in opposite direction.



If it takes τ minutes to travel the single track, then add an arrival-departure activity at one of the ends with the interval $[0, T - 2\tau]_T$. This may be combined with minimum headway times h to obtain $[h, T - 2\tau - h]$.



At least 10 units of slack must be distributed along this cycle. Minimizing waiting time at Fredersdorf, this fixes the timetable of the whole line S5.

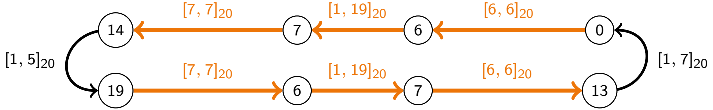
Symmetric timetables

Consider an event-activity network $\mathcal{E}$ associated to a line network.

Definition

- A pair $(v, w) \in V(\mathcal{E})^2$ of events is called **complementary** if
  - v is a departure event,
  - w is an arrival event,
  - v and w belong to the same station of the line network,
  - v and w belong to opposite directions of the same line.
- A periodic timetable $\pi \in [0, T)^{V(\mathcal{E})}$ is **symmetric** with **symmetry axis** $s \in [0, T/2)$ if for all pairs of complementary events (v, w) holds $[\pi_v + \pi_w]_T = 2s$.

Example (symmetry axis: 6.5)



Symmetric timetables

Advantages

- Suppose we have a symmetric timetable. Consider a transfer activity (v_1, w_2) from an arrival event of line 1 to a departure event of line 2. Let w_1 and v_2 be the complementary events to v_1 and w_2, respectively. Then

$$\begin{aligned}[\pi_{w_2} - \pi_{v_1}]_T &= [\pi_{w_2} - \pi_{v_1} + \pi_{v_2} + \pi_{w_1} - \pi_{v_2} - \pi_{w_1}]_T \\ &= [\pi_{v_2} - \pi_{w_1} + 2s - 2s]_T \\ &= [\pi_{v_2} - \pi_{w_1}]_T.\end{aligned}$$

  In other words, the transfer from line 1 to line 2 and in the opposite direction take the same amount of time (modulo T).
- If a symmetric timetable is to be constructed, then any direction of of a line is implied by the opposited direction. I.e., the PESP needs only half the number of events.

Symmetric timetables

Disadvantages

- Infeasibility: Sometimes headway constraints do not allow a symmetric timetable.

	S2	S25	S2	S25	S25	S2	S25	S2
Zepernick (crossing)			0			0		
Tegel (crossing)		0					0	
Bornholmer Straße	12	13	2			18	7	8
Nordbahnhof	18	19	8	9	11	12	1	2
Priesterweg	16	17	6	7	13	14	3	4

S-Bahn Berlin: Taking the same symmetry axis for both lines violates minimum headway times between S2 and S25 [Liebchen, 2004].

- Suboptimality: A symmetric timetable does not need to be optimal in terms of total travel time.

Symmetric timetables

Usage

- The big European train operators use a symmetry axis between 28 and 30 mod 60.
- S-Bahn Berlin does not use symmetric timetables.
- Heidekrautbahn (RB27 of Niederbarnimer Eisenbahn) uses a symmetry axis of $\approx 10 \bmod 60$, as the only important connection is the transfer to S-Bahn Berlin.
- When timetables of suburban transportation networks are planned independently from trains, there is usually no common symmetry axis for all lines.

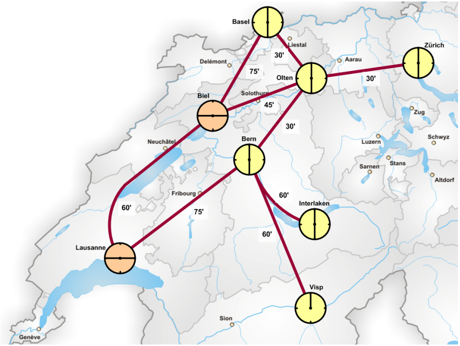
IFIT: Integrated fixed-interval timetables

- In an **i**ntegrated **f**ixed-**i**nterval **t**imetable (German: integraler Taktfahrplan/ITF), certain stations are selected as **hubs**.
- At a hub, *all* lines meet at the same time.
- In particular, both directions of a line meet at the same time, so an IFIT is necessarily symmetric.
- Moreover, all lines at a hub share the same symmetry axis s.
- The time when all lines meet at a hub is either s ("zero hub") or $s + T/2$ ("half hub").
- The travel time between two hubs is (approximately) an integer multiple of $T/2$.
- Without loss of generality, it can be assumed that $s = 0$.
- An IFIT is very restrictive, so that computing an IFIT is not hard – however, the suitable infrastructure must be available.

IFIT

Examples

- The railway networks of Switzerland and the Netherlands are IFITs.
- The Berlin night network is (partially) an IFIT including S-Bahn, U-Bahn, buses and trams.



Bernese_media, based on Karte_Schweiz.png by Tschubby, CC BY-SA 3.0

IFIT

Advantages

- short transfer times in all directions
- departure and arrival times are easy to memorize
- simple to create

Disadvantages

- suitable infrastructure required
- increase of waiting times in stations
- needs extremely good punctuality
- neglects passengers needing longer transfer times
- does not necessarily minimize total travel time

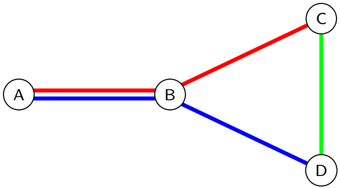
Passenger flow

Question

Where do the weights for a PESP instance come from?

Answer

Usually from the number of passengers using the corresponding activity. One technique to compute the weights is to give an *origin-destination* matrix whose entries are the number of passengers that want to travel from a certain origin to a certain destination:



	A	B	C	D
A	0	23	56	0
B	26	0	112	8
C	48	124	0	22
D	0	8	23	0

Passenger flow

The weights on each activity in the event-activity network are then computed by some routing strategy, e.g.,

- use the lower bounds as lengths for a shortest path problem
- use a fixed timetable and compute shortest routes

Clearly, the choice of routes is affected by these strategies, which in turn has an impact on the timetable which is to be optimized. For the optimal timetable, the chosen routes are not necessarily the shortest ones!

Integrated Periodic Timetabling and Passenger Routing

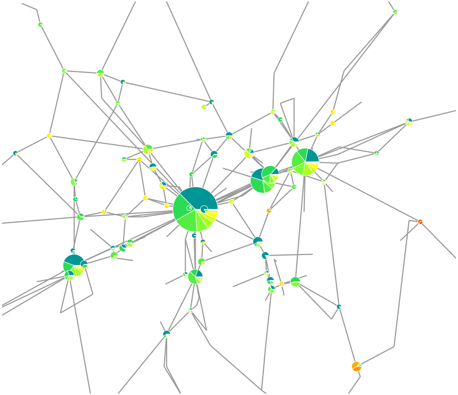
Given a PESP instance $(\mathcal{E}, T, \ell, u)$ without weights and an origin-destination matrix $D = (d_{ij})$, find

$$\min \left\{ \sum_{p \in P_{i,j}} \sum_{a \in E(p)} f_p x_a \;\middle|\; \begin{array}{lll} \Gamma x \equiv_T 0 & \ell \leq x \leq u & \\ \sum_{p \in P_{ij}} f_p = d_{ij} & f_p \geq 0 & \text{for all } ij \end{array} \right\},$$

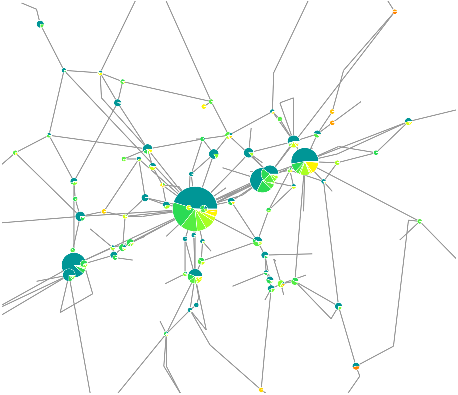
where P_{ij} is the set of all i-j-paths and Γ is the cycle matrix of an integral cycle basis of $\mathcal{E}$. This is a quadratic optimization problem!

Integrated Timetabling and Routing: Wuppertal

reference timetable



optimized timetable



2.1% improvement in total travel time
8.9% improvement in total transfer time

[Borndörfer/Hoppmann/Karbstein, 2017]

Organization

The **lecture on June 18** takes place in the ZIB meeting room **3028** (Rundbau, first floor).

Written exams (60 minutes):

- Tuesday, August 7, 2018 – 10.00am – ZIB seminar room 2006
- Monday, October 8, 2018 – 10.00am – ZIB seminar room 2006

The last problem set will be a test exam, which will be discussed in the last lecture on July 16.

There will be no tutorial on July 19 – instead, you are invited to join the *Variants of Shortest Path Problems* event (9.00am, ZIB lecture hall).