Maximizing Trains Platformed, for a given Timetable

Peter Sels[1,2,3], Thijs Dewilde[1], Dirk Cattrysse[1], Pieter Vansteenwegen[4]

[1]Katholieke Universiteit Leuven,
Centre for Industrial Management/Traffic & Infrastructure,
Celestijnenlaan 300a, 3001 Heverlee, Belgium

[2]Logically Yours BVBA,
Plankenbergstraat 112 bus L7, 2100 Antwerp, Belgium
e-mail: sels.peter@gmail.com, corresponding author

[3]Infrabel, Department of Network Access,
Frankrijkstraat 91, 1070 Brussels, Belgium

[4]Ghent University, Department of Industrial Management,
Technologiepark 903, 9052 Zwijnaarde, Belgium

June 23, 2011

Table of Contents

Task

Belgian Infrastructure Management Company: Infrabel:

"Maximizing Trains Platformed, for a given Timetable"

Resulting Schedule Implies:

Station Capacity Consumption

Fixed: Constants:

Infrastructure, Timetable, Train Lines, Halting Pattern

To Determine: Variables

Platform, In- & Out-Routing, Corresponding sub-times, (in)feasible trains

Specifics:

One Day of Traffic

Reality

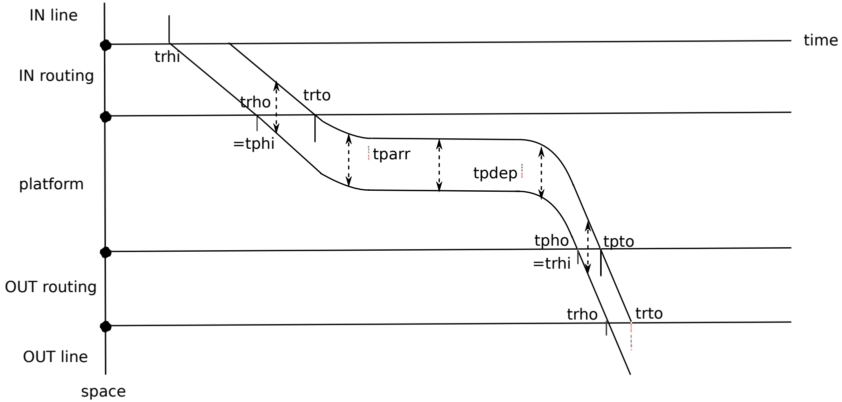

Figure: Train Occupation Time Points and Durations Between Them

Model

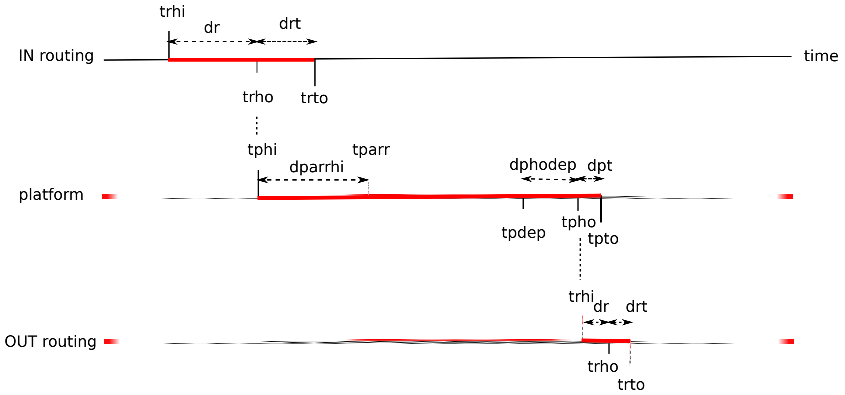

Figure: Calculations are done from platform intervals towards routing intervals.

Mixed Integer Linear Programming

- Train Platforming Problem, but with fixed times
- Mixed Integer Linear Programming approach, optimal
- Fictive platform connected to fictive routings, always feasible

Goal Function:

- Minimize:

$$\begin{aligned} g(op_{o,p}) &= \sum_{o \in O_{INI}} C2F_{INI}\, ch2f_o + C2OR_{INI}\, ch2or_o \\ &+ \sum_{o \in O_{SUP}} C2F_{SUP}\, ch2f_o + C2OR_{SUP}\, ch2or_o. \end{aligned} \tag{1}$$

Constraints:

- Infrastructure: Connectivity Platforms-Routings, Routing Conflicts
- Timetable: Fixed Platform Times, Train Run Times, Train Lengths

Infrastructure

- L is the set of lines of both sides of a station, both in and out lines
- P is the set of platforms of a station
- R is the set of routings from lines towards the platforms, and from platforms to lines
- $\forall p \in P : R_p$ is the set of routings that are connected to platform p
- $r2p : R \to P : r \mapsto p$ is the mapping that for each routing r, gives the platform p it is connected to
- $dep : R \times R \to \{0, 1\} : (r_0, r_1) \mapsto dep_{r_0,r_1}$ defines route conflict pairs

Train Activities

- O is the set of occupations to be mapped on platforms
- M is the set of all movements, where several movements can belong to the same occupation
- M_{IN} is the set of IN movements
- M_{OUT} is the set of OUT movements
- $\forall o \in O : M_o$ is the set of movements for an occupation o
- $m2o : M \rightarrow O : m \mapsto o$ is the mapping that for each movement m, gives the occupation o it is belongs to

Note that occupations of any complexity are supported: stop, pass, split, merge, split & merge, ...

Variables

- $\forall o \in O$ we define the variables
    - $o2p_o$ as the platform $p \in P$ chosen for occupation o
    - $\forall p \in P : op_{o,p}$ as the boolean that is true iff $o2p_o = p$
- $\forall o \in O : \forall m \in M_o$ we define the variables
    - $m2r_m$ as the routing $r \in R$ chosen for movement m
    - $\forall r \in R : omr_{o,m,r}$ as the boolean that is true iff $m2r_m = r$

Constraints: Allocation

- Occupation to Platform Boolean Integer Variable Relation

$$\forall o \in O : \begin{cases} \sum_{p \in P} op_{o,p} & = & 1 \\ \sum_{p' \in P} op_{o,p'} \cdot p' & = & o2p_o. \end{cases} \quad (2)$$

- Movement to Routing Boolean Integer Variable Relation

$$\forall o \in O : \forall m \in M_o : \begin{cases} \sum_{r \in R} mr_{o,m,r} & = & 1 \\ \sum_{r' \in R} mr_{o,m,r'} \cdot r' & = & m2r_m. \end{cases} \quad (3)$$

- Relation between occupation to platform and movement to routing allocation boolean variables:

$$\forall o \in O : \forall m \in M_o : mr_{o,m,r} \implies op_{m2o_m, r2p_r} \quad (4)$$

or/and equivalently:

$$\forall o \in O : \forall p \in P : op_{o,p} \implies \sum_{r \in R_p} mr_{o,m,r} = 1 \quad (5)$$

Constraints: Separation

- Separate pair of platform occupation intervals if they are on the same platform resource:

$$
\begin{array}{l}
\forall o_0 \in O : \forall o_1 \in O : o_0 \prec o_1 : \\
[otLoLbC_{o_0}, otHiUbC_{o_0}] \cap \\
[otLoLbC_{o_1}, otHiUbC_{o_1}] \neq \phi : \\
\forall p_0 \in P_{o_0} : \forall p_1 \in P_{o_1} : p_1 = p_0 :
\end{array}
\left\{
\begin{array}{c}
op_{o_0,p_0} \wedge op_{o_1,p_1} \\
\implies o0sepo1_{o_0,o_1} .
\end{array}
\right.
\tag{6}
$$

- Separate pair of movement routing intervals if they are on the same routing resource:

$$
\begin{array}{l}
\forall m_0 \in M : \forall m_1 \in M : m_0 \prec m_1 : \\
[mtLoLbC_{m_0}, mtHiUbC_{m_0}] \cap \\
[mtLoLbC_{o_1}, mtHiUbC_{o_1}] \neq \phi : \\
\forall r_0 \in R_{m_0} : \forall r_1 \in R_{m_1} : dep_{r_0,r_1} :
\end{array}
\left\{
\begin{array}{c}
mr_{o_0,m_0,r_0} \wedge mr_{o_1,m_1,r_1} \\
\implies m0sepm1_{m_0,m_1}
\end{array}
\right.
\tag{7}
$$

Schedule Graph

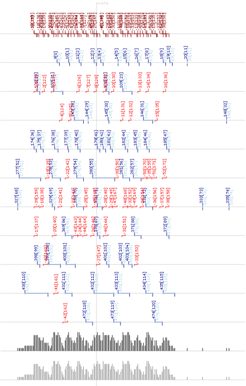

Figure: Occupation Time Interval Graph for Mechelen Station and Peak Original and Optimized Traffic

[Original schedule in dark colors. Optimized schedule in light colors.] * [Red for initial, blue for supplementary traffic.]

Solver Response Times Table I

Station #P,#R,#O	Solver on Machine	#Con-straints	#Var-iables	Time (h,m,s)
Bergen 7,128,178	Cplex v12.2 on Apple 4C 2.3-3.2 GHz	103362	33473	291.34s
	Gurobi v4.5.1 on Apple 4C 2.3-3.2 GHz	109955	37175	451.35s
	Cplex v11.2 on HP 2C 3.16 GHz	103365	33474	1216s
	Xpress v7.2 on HP 2C 3.16 GHz	109955	37175	OM at 3100s
	Gurobi v4.5.1 on HP 2C 3.16 GHz	109955	37175	4422s
Brugge 10,198,68	Cplex v12.2 on Apple 4C 2.3-3.2 GHz	34384	10958	3.54s
	Gurobi v4.5.1 on Apple 4C 2.3-3.2 GHz	35627	11717	23.5s
	Cplex v11.2 on HP 2C 3.16 GHz	34384	10958	305s
	Xpress v7.2 on HP 2C 3.16 GHz	35627	11717	11s
	Gurobi v4.5.1 on HP 2C 3.16 GHz	35627	11717	84s
Dender-leeuw 9,206,211	Cplex v12.2 on Apple 4C 2.3-3.2 GHz	154553	47187	37.57s
	Gurobi v4.5.1 on Apple 4C 2.3-3.2 GHz	159226	50074	26.79s
	Cplex v11.2 on HP 2C 3.16 GHz	154553	47187	621s
	Xpress v7.2 on HP 2C 3.16 GHz	159226	50074	11s
	Gurobi v4.5.1 on HP 2C 3.16 GHz	159226	50074	2651s

Table: Optimization Execution Times I

#P = number of real platforms. #R = number of real routings. #O = number of occupations, initial and supplementary together. Cplex Matrix dimensions are already (slightly) reduced ones. n.a. = not available (not enough patience limit). OM = Out of Memory.

Solver Response Times Table II

Station #P,#R,#O	Solver on Machine	#Con-straints	#Var-iables	Time (h,m,s)
Leuven 14,324,256	Cplex v12.2 on Apple 4C 2.3-3.2 GHz	105709	36242	303s
	Gurobi v4.5.1 on Apple 4C 2.3-3.2 GHz	113119	40676	224.92s
	Cplex v11.2 on HP 2C 3.16 GHz	105709	36242	321s
	Xpress v7.2 on HP 2C 3.16 GHz	113119	40676	3600s (at 3%)
	Gurobi v4.5.1 on HP 2C 3.16 GHz	113119	40676	1569
Mechelen 10,170,121	Cplex v12.2 on Apple 4C 2.3-3.2 GHz	9959	4756	0.09s
	Gurobi v4.5.1 on Apple 4C 2.3-3.2 GHz	12201	6292	3.1s
	Cplex v11.2 on HP 2C 3.16 GHz	9959	4756	4.3s
	Xpress v7.2 on HP 2C 3.16 GHz	12201	6293	7s
	Gurobi v4.5.1 on HP 2C 3.16 GHz	12201	6293	17s

Table: Optimization Execution Times II

#P = number of real platforms. #R = number of real routings. #O = number of occupations, initial and supplementary together. Cplex Matrix dimensions are already (slightly) reduced ones. n.a. = not available (not enough patience limit). OM = Out of Memory.

Solver Times Graph

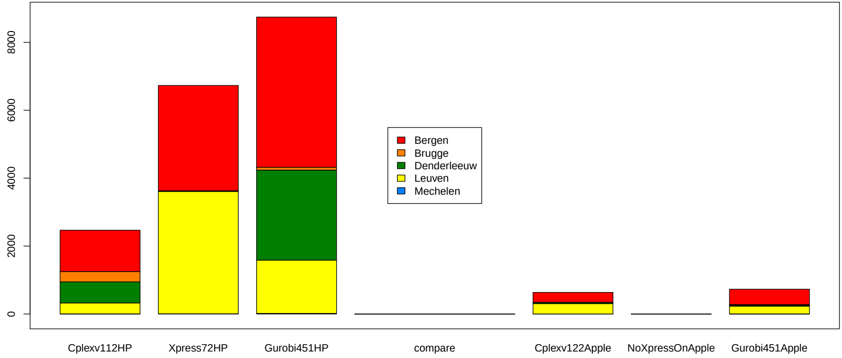

Figure: Intel Core2 Duo, 1.4 GB per process (left) to QuadCore i7, 8GB per process (right) comparison. Serious Improvement across all Solvers.

Conclusions & Further Work

- Conclusions: Business Results
  - *timetable based* capacity (cfr UIC code 406)
  - *fixed time* platforming
  - robust via separation times
  - platforming (& routing) in less than 5 minutes per station
  - both conservative and progressive options
  - improved estimation in practice
  - user surprised about platforming 'cleverness/inventiveness'
  - business problem solved
- Conclusions: Research Method: Compared To Billonnet (2003)
  - also fixed time TPP
  - routing choice influences timing and potential conflicts
  - so our conflicts are conditional, 10 to 100 times bigger problems
  - no (conditional) conflict clique heuristics
  - we cover full search space i.o. subset
  - we use real world data i.o. randomized
  - half the solver time

Further Work

- interface
  - availability of data of all stations
- model: consider passenger transfers
  - strive for transfers between neighboring platforms
  - weigh with passenger numbers (flows)

Questions?

sels.peter@gmail.com

References I

- UIC (2004) UIC CODE 406: Capacity *UIC* **IV(1)**, 1–21.
- Caprara, A. and Kroon, L. 1 2007 Transportation, Handbooks in Operations Research and Management Science Publisher, 1 edition.
- Caprara, A. and Galli, L. (2007) *Proceedings of the 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems* , 49–61.
- Zwaneveld, P. J. e. a. (1996) Routing Trains through a Railway Station: Model Formulation and Algorithm *Transportation Science* **30(1)**, 181–194.
- Zwaneveld, P. J. (1997) Railway Planning and Allocation of Passenger Lines. Ph. D. Thesis Rotterdam School of Management.
- Kroon, L. and Romeijn, H. (1997) Scheduling and Platforming Trains at Busy Complex Stations *EJOR* **98(1)**, 485–498.

References II

De Luca Cardillo, D. and Mione, N. (1999) k L-List tau Colouring of Graphs *EJOR* **106(1)**, 160–164.

Billonnet, A. (2003) Using Integer Programming to Solve the Train Platforming Problem *Transportation Science* **37(1)**, 213–222.