A Case Study in Periodic Timetabling

Christian Liebchen and Rolf H. Möhring

Technische Universität Berlin, Institut für Mathematik,
Straße des 17. Juni 136, D-10623 Berlin

Abstract

In the overwhelming majority of public transportation companies, designing a periodic timetable is even nowadays largely performed manually. Software tools only support the planners in evaluating a periodic timetable, or by letting them comfortably shift sets of trips by some minutes, but they rarely use optimization methods. One of the main arguments against optimization is that there is no clear objective in practice, but that many criteria such as amount of rolling stock required, average passenger changing time, average speed of the trains, and the number of cross-wise correspondences have to be considered.

This case study will demonstrate on the example of the Berlin underground (BVG) that all these goals can be met if carefully modeled, and that timetables constructed by optimization lead to considerable improvements.

Our approach uses the Periodic Event Scheduling Problem (PESP) with several add-ons concerning problem reduction and strengthening. The resulting integer linear programs are solved with the CPLEX MIP-Solver. We have been able to construct periodic timetables that improve the current timetable considerably. For any of the above criteria, we have been able to identify global lower and upper bounds. Our favorite timetable improves the current BVG timetable in each of these criteria.

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

Introduction

Public transportation companies usually plan their service hierarchically. The construction of a timetable is typically done after the infrastructure and the lineplan are fixed. Furthermore, to operate a given timetable, vehicle and crew assignments must be defined. As the timetable is the last result in this planning process that is published, it has a central role.

There are several software tools that support vehicle assignment by really calculating new solutions on their own. But software tools which are used for timetable construction — such as HASTUS (GIRO Inc., Montréal, CA), FBS (iRFP, Frankenheim, D), BERTA (IVU Traffic Technologies AG, Berlin, D), MICROBUS (IVU Traffic Technologies AG, Berlin, D), VISUM ÖV 7.0 (PTV AG, Karlsruhe, D), ptv interplan (PTV AG, Karlsruhe, D), and solutions by TLC GmbH, Berlin, D — are limited to only modifying interactively an already existing timetable.

In our study we demonstrate the practical relevance of mathematical optimization techniques in periodic timetabling. We investigated the weak traffic time of the Berlin Underground network. There, we have been able to model any of the – stepwisely identified – requirements given by practitioners. Our final timetable improves the one currently operated in each of the given criteria.

We model the periodic timetable optimization task as a Periodic Event Scheduling Problem. As there is a choice in deriving an MIP formulation, we propose a systematical approach for selecting a good formulation which is based on finding an "optimized" graph representation of the problem via short cycle bases. Moreover, we strengthen this good formulation by adding valid inequalities of well-known types. An analysis of running times demonstrates the benefit of these two strategies.

The Periodic Event Scheduling Problem – And some Add-Ons

A periodic timetable works with an abstract period T and assigns point of time $\pi_i \in [0, T)$ to any relevant event i. These include arrivals and departures of directed traffic lines at stations, but also other events resulting e.g. from safety conditions. Events are coupled by constraints on the *difference* $\pi_j - \pi_i$ of any two events' i, j points of time π_j, π_i.

In a straightforward graph-theoretic interpretation, the events i define the nodes of a directed graph G, and the arcs of this graph are derived from the constraints on the differences $\pi_j - \pi_i$.

For example, if we want to ensure a travel time of at least ℓ_{ij} but at most u_{ij} time units from event i to event j, a naive way to model this would be

$$\ell_{ij} \leq \pi_j - \pi_i \leq u_{ij}. \tag{1}$$

Note that we may interpret π_i as node potentials of the graph G, and that restrictions of type (1) impose a Feasible Differential Problem on G [Roc84]. A solution π of (1) is called a *feasible potential*.

As long as $\pi_i < T - \ell_{ij}$, everything is fine. If not, there would be no value for $\pi_j \in [0, T)$ satisfying the above constraint. Nevertheless, a departure π_i at time $T - \ell_{ij}$ and arrival π_j at time 0 *periodically* make sense. Hence, the Feasible Potential Problem is too restrictive for our purpose. This is why we should allow to add an integer multiple p_{ij} of the period time T to every constraint. Doing so, restriction (1) becomes

$$\ell_{ij} \leq \pi_j - \pi_i + p_{ij}T \leq u_{ij}. \tag{2}$$

For a given graph G with lower and upper bounds on the arcs, a node potential $\pi = (\pi_i)_{i \in V(G)}$is said to be *periodically feasible*, if for every arc $a = (i, j)$ an integer p_{ij} satisfying (2) can be found. The problem to find a periodically feasible potential for a given graph is called the *Periodic Event Scheduling Problem (PESP)*. It was introduced by Serafini and Ukovich [SU89] and shown by them to be $\mathcal{NP}$-complete.

An alternative way to look at potentials are tensions. For a given node potential π, the corresponding tension x is defined on every arc $a = (i, j)$ by $x_{ij} = \pi_j - \pi_i$. We call a tension *periodically feasible* for a given set of constraints, if it may be derived from a periodically feasible potential in this way. Note that we can easily check if a given tension is periodically feasible by constructing a feasible potential along the arcs of some spanning tree. If it is not feasible, conflicts would only arise on non-tree arcs.

Moreover, we know how to characterize periodically feasible tensions implicitly [Nac98]. Consider a spanning tree of G and a corresponding so-called *strictly fundamental cycle base* which is given by the $k = m - n + 1$ fundamental cycles induced by adding the non-tree edges to the tree. The cycle-arc incidence matrix Γ of such a strictly fundamental cycle basis is called the *network matrix* of G [Sch86]. Then, periodically feasible tensions x are exactly the solutions to the following linear system:

$$\left.\begin{aligned} \Gamma x &= pT \\ \ell \leq x &\leq u \\ p &\,\text{integer.} \end{aligned}\right] \tag{3}$$

Note that there is only one integer variable p_C for every fundamental cycle C in this formulation. So we have saved $n - 1$ integer variables in comparison with (2). For the remaining variables p_C, we may impose box constraints by exploiting Odijk's [Odi94] inequalities. Let C be a fundamental cycle with incidence vector $\gamma = \gamma^+ - \gamma^-$, then

$$\underline{p}_C := \left\lceil \frac{\gamma^+\ell - \gamma^- u}{T} \right\rceil \leq p_C \leq \left\lfloor \frac{\gamma^+ u - \gamma^-\ell}{T} \right\rfloor =: \overline{p}_C \tag{4}$$

are valid bounds. The number of possible values for the variable p_C is then

just $\overline{p}_C - \underline{p}_C + 1$. As we want to keep this value small for every fundamental cycle, we will investigate these terms in more detail below.

To get a flavor of the bounds $\underline{p}_C$ and $\overline{p}_C$, we omit the rounding operation in equation (4), and the division by T as well. Then, we obtain

$$\overline{p}_C - \underline{p}_C \approx \gamma^+ u - \gamma^- \ell - \gamma^+ \ell + \gamma^- u = \gamma \cdot (u - \ell).$$

Hence, the number of possible integer vectors p essentially depends on the sum of the *arc-widths* $u - \ell$ of the fundamental cycles. Since we are still free in choosing the spanning tree, we are going to look for spanning trees that have a "short" fundamental cycle base, where the length of a cycle basis is the sum of the cycles' arc widths. Short cycle bases have been exploited by Kroon and Peeters [KP01] in minimizing the rolling stock. Unfortunately, it is $\mathcal{NP}$-hard to compute a strictly fundamental cycle basis of minimum weight, see Deo et al [DPK82].

Our case study confirms that the choice of a short cycle basis is indeed important. We have therefore implemented several heuristics to find a good cycle base. The simplest one is to compute a minimal spanning tree (MST) according to the arc-widths, and then take its strictly fundamental cycle base. Deo et al [DKP95] introduce two heuristics that seem to be more sophisticated: In UV (unexplored vertices), they grow the spanning tree by adding nodes that are adjacent to many non-tree nodes. In NT (non-tree edges), they grow the tree by selecting nodes that induce many non-tree edges in the current forest. Since every non-tree edge closes a fundamental cycle, they hope to get many short fundamental cycles from nodes added at the beginning, and only few long fundamental cycles from the last nodes.

An more general approach for constructing periodically feasible tensions would be to relax the requirement on the cycle basis to be strictly fundamental. Then, we may use Horton's [Hor87] polynomial time algorithm for solving the minimal cycle basis problem. If the cycle basis constructed by Horton's algorithm permits to express every cycle of some strictly fundamental cycle basis as an *integer* linear combination, then we would have a way to reformulate (3) as follows:

- Let Γ be the network matrix for some spanning tree.
- Let B be the cycle-arc incidence matrix of another cycle basis.
- Because of the basis property of B, we find a regular Q, with $\Gamma^T = B^T Q$.
- Solve

$$\left.\begin{array}{l} Bx = qT \\ \ell \le x \le u \\ q \,\text{integer}. \end{array}\right]$$

- Define $p := Q^T q$.

Then x and p fulfill $\Gamma x = pT$. As q was required to be integral, p is integral

if Q has only integer entries. Hence, in this case, the vector x that has been computed using B and q — thus with best possible box constraints — instead of Γ and p, is a periodically feasible tension.

Clearly, a cycle base whose cycle-arc incidence matrix B can be written as $B = [UA]$ with a regular upper triangular $\{0,1\}$-matrix U permits every cycle to be written as such an integer linear combination. Hartvigsen and Zemel [HZ89] call these bases (generalized) fundamental. But they also give examples for cycle bases that are not fundamental in this sense.

Railway systems usually have many safety constraints to be met, and so the decision problem (3) is interesting on its own right – and often hard enough to solve. However, for bus systems, or underground systems like the one in Berlin, no safety restrictions have to be obeyed because every line is operated on its proper track. In such systems, every timetable vector π is feasible. Here, we *only* have to select a *good* timetable from a huge number of feasible ones.

Usual optimization criteria are total or maximum passenger waiting time, amount of rolling stock required, and average speed of the lines. Any of these criteria may be modelled by a linear objective function c – as long as we forbid line changes of vehicles, which would lead to a quadratic objective function [LP02]. The optimization task then can be formulated as

$$\left.\begin{aligned} \min\ & c(x-\ell) \\ \text{s.t. } & \Gamma x = pT \\ & \ell \le x \le u \\ & p \,\text{integer}. \end{aligned}\right] \tag{5}$$

We have chosen MIP-solver of CPLEX as our tool to solve the integer linear programs representing the PESP instances. The MIP-solver of CPLEX is based on branch-and-bound, and uses the LP-relaxation of the PESP and its generated subproblems as lower bounds. So we must ensure that the LP-relaxation in sufficiently strong to generate good lower bounds. This is not the case with system (5), as we may choose x equal to ℓ, and the p variables will always compensate the T on the right-hand-side. Hence, the initial LP-relaxation always has zero as its minimum.

For this reason, it is very important to add valid inequalities to the problem formulation, and hereby improve the lower bound. There are two important classes of valid inequalities. The first one are Odijk's inequalities that resulted in the box constraints for the integer variables in equation (4). In their most general form, they may be formulated for an elementary cycle C with incidence vector γ as

$$\left\lceil \frac{\gamma^+\ell - \gamma^-u}{T} \right\rceil \le y^T p \le \left\lfloor \frac{\gamma^+u - \gamma^-\ell}{T} \right\rfloor, \tag{6}$$

where y is the solution of $\Gamma^T y = \gamma$. The second important class of valid inequalities has been introduced by Nachtigall [Nac96]. These cutting planes are also defined for every elementary cycle. With $b = (-\gamma^T \ell) \bmod T$, they are

$$(T-b) \cdot \gamma^+(x-\ell) + b \cdot \gamma^-(x-\ell) \geq b(T-b). \tag{7}$$

A Case Study for the Berlin Underground

We first give a rough description of the Berlin Underground network. Then, we introduce the objectives that are important to the BVG. Finally, we report our computational results.

The Berlin Underground Network

The total length of the tracks is more than 144 km. There is a total of 170 stations, and 19 stations of them allow line changes. The average trip length of the passengers is slightly less than 6 km, or about eight stations. During one day, there are about 1.3 million passengers.

The Berlin Underground is only one part of the very complex public transportation network of Berlin. It also encompasses regional trains, fast trains, trams and busses. The timetable planning is done hierarchically, and only trams and busses depend on the timetable of the underground. This implies that the underground has to take into account the timetable of the fast train network.

The fast train and the underground networks have about 20 stations in common. In most of them, there are rather long walks between the two systems. Hence, there are no reliable values for the time required by passengers to change platforms. For this reason, traditionally only system changes within those station where fast train and underground share platforms are considered when planning the underground. Further interactions between the two systems are only considered in a later fine-tuning phase.

During peak-hours and on weekdays, all lines are operated at least every five minutes. Only in the evening hours and on weekends, the period length is extended to $T = 10$ minutes, the *weak traffic time*. This weak traffic time is the target of our study. In it, each of the nine lines is operated on its own track. As there are no single tracks in the Berlin Underground network, no safety conditions have to be obeyed. The planning precision is half a minute.

The passengers' flow has been given by weights — ranging from 25 to 1900 — for any of the 168 possible change activities. We have to mention that these data stem from 1997, and that passenger flows might have changed due to the installation of new regional and fast train lines. Unfortunately, this is the most current data that was available to us.

Objectives given by the BVG

In order to give an idea of the modeling power of the PESP, we describe the iterative process of formulating the mathematical model in detail. As practitioners are likely not familiar with the PESP, we start with the most obvious constraints such as travel times and change activities of the passengers. But the scenario based on these constraints only produced a solution that violated an important criterion not named so far. Because of the flexibility of the PESP, we have been able to incorporate all the iteratively identified restrictions of very different types. Sometimes, we thought of this process as being a "*verbal cutting-plane algorithm*"...

The main objective has been to reduce passenger waiting time. As the current timetable defines additional stopping times in four stations, we tried to insert such additional stopping times most gainfully for the changing passengers.

The result has been a timetable which reduced the weighted average waiting time of the changing passengers from more than 34% of the period time of ten minutes down to less than 17%. For the 24 most important relations, the average waiting time even decreased from more than 30% to less than 4%, which translates to an average waiting time of less than 30 seconds. Recall here that we always ensure a certain time for changing platforms and only minimize the time exceeding this required changing time.

This first timetable was very passenger-friendly but required 78 trains instead of only 71 in the current timetable. Thus it was not acceptable for the BVG. This led to a model that incorporated also vehicle waiting time besides passenger waiting time.

The second timetable produced by our system still represented a significant improvement for the changing passengers. The average waiting time decreased by more than one minute, both in total and on the 24 most important relations. And even more interesting, this timetable required only 68 trains, which turned out to be a global lower bound and thus the minimum number of trains.

Surprisingly for us, this was also not acceptable for the BVG. The timetable proposed for the line `U1` is reported in table 1. Since the minimal turning times

Table 1
Times proposed for line `U1` in timetable 2

direction A	Station	direction B
20:09:30	Warschauer Straße	21:35:00
20:50:00	Krumme Lanke	20:55:00

for the trains are exactly met in both endpoints, the timetable for this line is likely to be very instable if the line is operated with only nine trains. For this reason, we were obliged to add one minute to the (theoretically) minimal turning time at both endpoints of this line, as well as at both endpoints of

line U7, which is the longest line requiring 55 minutes for one direction.

The third timetable added one train on line U1, and the passengers got a small benefit out of it. But this timetable still violated some — up to here — unknown constraints. Four out of the ten most important relations showed the situation in table 2. These connections were considered to be much too

Table 2
Situation on four very important relations in timetable 3

Source	Station	Destination	(1)	(2)
U9 Rath. Steglitz	Leopoldplatz	U6 Alt-Tegel	90	0
U7 Rudow	Hermannplatz	U8 Wittenau	90	0
U2 Ruhleben	Alexanderplatz	U5 Hönow	60	0
U2 Ruhleben	Zoolog. Garten	U9 Osloer Str.	90	0
(1): min. changing time (sec) (2): effective waiting time				

instable. To avoid this, we added 30 or 60 seconds to the minimal changing time for every relation in the third precision of our model.

As we do not want to penalize connections in the real timetable that could have been planned with the potentially instable original minimal changing time, we counted an objective value as zero for connections resulting in an effective waiting time of 09:00 minutes or more. By that, the current timetable's average waiting time decreases to 28.2%, and to 23.6% on the 24 most important relations.

The fourth timetable again required 69 trains, and implied an average waiting time of 22.7%, resp. 15.2% on the most important relations, subject to the extended minimum changing times. This timetable did violate another – almost political – constraint, again not mentioned so far. Recall, that there is one station (Wuhletal), where underground and fast train share their platforms. The timetable proposed for line U5 has been the one given in table 3. Only the slightly more important cross-wise correspondence in direction B has

Table 3
Times proposed for line U5 in timetable 4

direction A	Station	direction B
20:00:00	Hönow	21:14:00
20:10:30	Wuhletal	21:03:30
20:33:30	Alexanderplatz	20:40:30

been adequately respected. In order to offer a good correspondence in direction A as well, line U5 must have been in Wuhletal two minutes later. But this conflicts with the minimal turning time of seven minutes for the trains

in Alexanderplatz. Thus, with the timetable for the fast train as given, good correspondences in both directions can only be offered if one additional train for line U5 is funded by the Berlin government – and indeed it is. Hence, the new constraints ensure good correspondences in every station where different lines share a platform. Besides the correspondence with the fast train, there are two other stations within the Berlin Underground network where good cross-wise correspondences for four pairs of directed lines now became obligatory. In Mehringdamm, where U6 and U7 meet, this implies one additional train for one of these two lines as well.

This is why the fifth timetable requires 71 trains. With the increased flexibility, passenger waiting times have been reduced to 20.2% for the whole network and to only 10.6%, or about one minute, on the TOP 24.

This timetable was criticized as not being sufficiently balanced. The situation in Berliner Straße, where U7 and U9 meet, is given in table 4, and representative for this phenomenon. For the current settings, this situation

Table 4
Situation in Berliner Straße proposed in timetable 5

Source	Arrival	Destination	Departure	Weight
U7 (A)	20:08:00	U9 (A)	20:16:30	900
U7 (A)	20:08:00	U9 (B)	20:16:30	300
U7 (B)	20:08:30	U9 (A)	20:16:30	400
U7 (B)	20:08:30	U9 (B)	20:16:30	400
U9 (A)	20:06:30	U7 (A)	20:08:00	700
U9 (A)	20:06:30	U7 (B)	20:08:30	200
U9 (B)	20:06:30	U7 (A)	20:08:00	700
U9 (B)	20:06:30	U7 (B)	20:08:30	900

is even locally optimal for line U9, because an earlier passage in direction A forces an earlier passage of direction B as well, due to the turning times. Besides, for this line an additional train is *not* accepted. The crux is that optimal solutions of linear programs are attained in the vertices of polyhedra, where as many constraints as possible are tight. Finally, we were asked to avoid large deviations like 00:00 against 06:30 for this and two other stations, where passenger flows are as balanced as here. This has been modelled by setting the minimal changing time to an artificial value of four minutes. This prefers timetables that bundle both directions of the same line, but separate the two lines.

The discussion also showed that the lack of balance was not the only problem here. Since the currently published timetable offers the connection from line U7 (A) to line U9 (A) with arrival 20:01 and departure 20:05, one fears

that an immediate change to timetable 5 results in an unacceptable amount of complaints. Finally, we introduced explicit constraints such that none of the most important connections faces a deterioration of more than two minutes — and hereby limited flexibility considerably.

The last timetable computed still improves the average waiting time compared to the currently valid timetable from 34.6% to 30.6%, on the TOP 24 from 30.1% to 23.1%. In this final scenario, we refer to the minimal changing times originally given by the BVG. Although no additional vehicles are required, our timetable ought to be more stable than the one currently operated: turning times respect a buffer of 30 or 60 seconds added to the theoretically minimal turning times, and the four correspondences at shared platforms within the Berlin Underground network respect 30 seconds of additional minimal changing time each. Furthermore, there is only one connection with only the minimum amount of changing time in our timetable, but 11 in the current timetable. On the TOP 24, only the current timetable offers three relations with waiting times of `07:00` minutes or more. On the TOP 90, only along three relations passengers have to wait at least three minutes more than in the current timetable, none of them belongs to the TOP 24. But seven resp. three relations were improved by at least that amount. Last but not least, the average speed of the trains even increases a little bit, because we inserted only five minutes of additional stopping time.

The question, if there still rest further restrictions not formulated so far, has been answered by an assistant of the BVG:

No k.o.-criterion is violated.

Running Times

The scenarios in the case study have been computed by relaxing the 80 lightest change arcs. Doing so, we neglected 10% of the changing passengers, the ones who use relations with the two smallest weights. But the search space reduced notably: the product of the number of possible values of the integer variables

$$\prod_{c \in \mathcal{C}} (\overline{p}_C - \underline{p}_C + 1), \tag{8}$$

decreased from more than 10^{90} to less than 10^{60}. This allowed running times between 200 and 1000 seconds for the MIP-solver of CPLEX on a 440 MHz machine, using the problem formulation refined by adding some of the mentioned valid inequalities.

For the last but one scenario, we analyze the running times in more detail. The full graph has 61 nodes and 243 arcs, after redundancies were eliminated in a preprocessing phase. We start by comparing the different approaches of constructing a fundamental cycle base, and then demonstrate how we profit from the valid inequalities.

To compare the heuristics for constructing a short fundamental cycle base, we list the potential dimension of the search space (8) in table 5. In two

heuristics, we only consider the cardinality of the basic cycles, in two others, we include also the width of the cycles by summing up the spans of the arcs. We tested every heuristic on the models with 83%, 90%, and 100% of the changing passengers. Without adding valid inequalities, only the branch-and-bound tree

Table 5
Quality of different fundamental cycle bases

Heuristic	Focussing on 83% Search Space	Focussing on 90% Search Space	Full Graph Search Space
MST (card.)	$1.58 \cdot 10^{46}$	$2.67 \cdot 10^{56}$	$6.48 \cdot 10^{95}$
MST (spans)	$1.52 \cdot 10^{40}$	$5.82 \cdot 10^{49}$	$2.24 \cdot 10^{91}$
NT (card.)	$3.54 \cdot 10^{44}$	$6.14 \cdot 10^{52}$	$4.66 \cdot 10^{94}$
UV (spans)	$8.94 \cdot 10^{45}$	$4.55 \cdot 10^{52}$	$3.58 \cdot 10^{90}$

of MST (spans) on 83% did not exceed the memory limit of 768MB and was solved optimally within 423 seconds. The others reached the memory limit after more than two hours, with optimality gaps of at least twenty percent. We should mention that a minimal cycle basis of the 90% scenario limits the search space to only $7.90 \cdot 10^{34}$, but our minimal basis is not even fundamental in the generalized sense.

This shows that the choice of the heuristic for building the spanning tree is of relevance. Roughly speaking, the arcs' spans should be taken into account. But since this preprocessing step has negligible cost, several heuristics might be tried before fixing one problem formulation for the MIP-solver.

The valid inequalities have a big impact on the solution capabilities. In table 6, we list the running times of the MIP-solver on formulations obtained by adding up to 100 or 200 cuts of the two relevant types (6) and (7). Generating the valid inequalities took at most 420 seconds. In rating the lower bounds, observe that randomly generated cycles were considered in some steps.

One conclusion that may be taken from table 6 is that a very bad cycle basis causes longer solution times even for the problem formulation strengthened by cuts. Another observation is that, among the more sophisticated heuristics, UV comes up with the longest solution times, but we have no explanation for this phenomenon.

The results clearly show that adding valid inequalities significantly enlarges the set of problems that we can solve to optimality. There is only one heuristic for which we were able to solve the 83% instance in the original formulation. However, with some valid inequalities added, every heuristic allows a solution time of less than eight minutes for the 90% instance.

With the practical experience obtained by this analysis, we attacked the network in which only 5% of the changing passengers, or 51 change arcs, are relaxed. We were able to solve even this scenario optimally after 5558 seconds,

Table 6
Solution times for PESP with valid inequalities added

		Focussing on 83%		Focussing on 90%	
Heuristic	Cuts	Lower Bound	Time	Lower Bound	Time
MST (card.)	100	81.04%	4271	79.19%	11990
MST (spans)	100	95.84%	52	81.05%	365
NT (card.)	100	95.64%	202	79.60%	679
UV (spans)	100	96.00%	467	80.06%	5129
MST (card.)	200	96.89%	48	97.14%	432
MST (spans)	200	96.22%	50	96.88%	299
NT (card.)	200	96.52%	69	96.35%	172
UV (spans)	200	97.24%	108	96.27%	359

although the LP lower bound has not been very tight (80.87%). However, the optimal solutions for the 90% scenario and the 95% scenario are the same, only in the 83% scenario passengers have to wait 5.4% longer. In a final run, we omitted only the 29 relations having smallest weight (1.3%). The valid inequalities allowed to respect the memory limit of 768MB even for this scenario. After 11 hours, this instance has been solved optimally, and the previous solution still has been improved – by 0.5 permill. . .

The picture about the MST heuristics is not as conclusive as this study may indicate. We have other traffic networks from practice in which the MST heuristics are always dominated by the unweighted NT and UV heuristics.

Conclusions

We showed that optimized timetables can be significantly better than manually constructed timetables, and this in several hard criteria such as amount of rolling stock or average waiting time of changing passengers. But there are also many rather soft criteria, such as balance and stability which practitioners take into account as well. Fortunately, even these can be modelled by the PESP. This is why we are convinced that automatic timetable generation will sooner or later become relevant in practice. We are currently aware only of two software tools for automatic timetable generation which include optimization techniques to some extent: CADANS (ORTEC Consultants bv, Gouda, NL) and the latest release of VISUM ÖV 8.0 (PTV AG, Karlsruhe, D) [LN02].

From a theoretical point of view, we raised the question of systematically looking for short fundamental cycle bases, in order to keep the search space

for a MIP-solver small. Although on the example investigated in this study the differences have been rather small, we consider short cycle basis to be very important for other instances. Finally, the benefit of additional valid inequalities once more has been clearly demonstrated.

Acknowledgement

We want to thank the current and former personnel of the BVG for their support, in particular Hans-Christian Kaiser. It has been very time consuming, but important, to analyze our timetables with respect to the operational requirements of the BVG. Furthermore we want to thank Leon Peeters for several worthwhile hints concerning relevant literature.

References

[DPK82] N. Deo, M. Prabhu, and M.S. Krishnamoorthy. *Algorithms for Generating Fundamental Cycles in a Graph.* ACM Trans. Math. Software, Vol. 8 No. 1, 1982

[DKP95] N. Deo, N. Kumar, and J. Parsons. *Minimum-Length Fundamental-Cycle Set Problem: A New Heuristic and an SIMD Implementation.* Technical Report CS-TR-95-04, Univ. Central Florida, Orlando, 1995

[HZ89] D. Hartvigsen and E. Zemel. *Is Every Cycle Basis Fundamental?* Journal of Graph Theory, Vol. 13, 1989

[Hor87] J.D. Horton. *A Polynomial-Time Algorithm to Find the Shortest Cycle Basis of a Graph.* SIAM J. Comput., Vol. 16 No. 2, 1987

[KP01] L. Kroon and L. Peeters. *A Cycle Based Optimization Model for the Cyclic Railway Timetabling Problem.* In Computer-Aided Scheduling of Public Transport, Springer, 2001

[LN02] C. Liebchen and K. Nökel. *Überführung eines mathematischen Modells zur Taktversatzoptimierung in die Praxis.* Tagungsbericht Heureka '02, 2002

[LP02] C. Liebchen and L. Peeters. *Some Practical Aspects of Periodic Timetabling.* In Operations Research Proceedings 2001, Springer, 2002

[Nac96] K. Nachtigall. *Cutting Planes for a Polyhedron Associated with a Periodic Network.* Institutsbericht IB 112-96/17, Deutsche Forschungsanstalt für Luft- und Raumfahrt e.V., 1996

[Nac98] K. Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables.* Habilitation thesis, University Hildesheim, 1998

[Odi94] M. Odijk. *Construction of periodic timetables, Part I: A cutting plane algorithm* Report 94-61, TU Delft, 1994

[Roc84] R.T. Rockafellar. *Network Flows and Monotropic Optimization.* Wiley, 1984

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, 1986

[SU89] P. Serafini and W. Ukovich. *A Mathematical Model for Periodic Scheduling Problems.* SIAM J. Disc. Math., Vol. 2 No. 4, 1989