TTPlib 2008 – a library for train timetabling problems

B. Erol[1], M. Klemenz[2], T. Schlechte[1], S. Schultz[3] & A. Tanner[4]
[1]*Berkan Erol and Thomas Schlechte, Zuse Institute Berlin (ZIB), Germany*
[2]*Marc Klemenz, IVE Hannover, Germany*
[3]*Sören Schultz, SFWBB TU Berlin, Germany*
[4]*Andreas Tanner, WIP TU Berlin, Germany*

Abstract

We introduce (TTPlib), a data library for train timetabling problems that can be accessed at *http://ttplib.zib.de*. In version 1.0, the library contains data related to more than 100 scenarios. Most instances result from the combination of macroscopic railway networks and several train request sets for the German long distance area containing Hanover, Kassel and Fulda, short denoted by HA-KA-FU. In this paper, we introduce the data concepts of TTPlib, describe the scenarios included in the library and provide a free visualization tool TraVis.
Keywords: train timetabling problem, data library, optimization.

1 Introduction

Mathematical optimization has greatly advanced many planning procedures in railway industry, like public transport network design, vehicle routing and dispatching or crew scheduling – however, the problem of computing new timetables from scratch is still unsolved. It is computationally not too difficult to verify that a given timetable is implementable, or to extend a given timetable incrementally by additional trains, but the computation of a timetable that is optimal in some mathematically well-defined sense is provably hard.

For similar problems of mathematical optimization, the establishment of *standard models* and *standard problem libraries* have contributed to the success in problem solving. Such libraries exists for the famous Traveling Salesman Problem,

WIT Transactions on The Built Environment, Vol 103, © 2008 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR080591

see [1], as well as for general Mixed Integer Programs, see [2] but, to our knowledge, unfortunately not for the train timetabling problem.

There are two main contributions of this paper: First, we propose a model for the timetabling problem that relies on a set of *slot requests*, that is, a set of trains that would be desirable to run, among which a subset has to be computed that is implementable on the one hand, and optimal with respect to some target function on the other hand. Our target functions are expressed in monetary terms, by attaching some value to every slot request. Additionally, we allow some flexibility with respect to arrival and departure times, and deviation from the optimal time is penalized.

This model is general enough to encompass a broad set of problems treated in the literature with a broad range of solution approaches: Mixed-integer linear programming methods, meta-heuristics, approximate methods, evolutionary algorithms or hybrid methods of all these, and therefore suitable as a standard model for timetabling.

Second, we present the newly installed since May 2008 *Train Timetabling Problem Library* (TTPlib), a library of standardized test instances for timetabling on a macroscopic railway network. It encompasses problem instances, together with information on the best known solutions and proven bounds. The complete library is freely accessible from the website [3]. Currently, all solutions contained in the library were produced with our optimization modul TS-OPT. This software was developed at Zuse Institute Berlin and is able to solve train timetabling problems to optimality by using state of the art linear and integer programming techniques.

We invite other teams to contribute new problem instances, tighter bounds or even better solutions.

In addition, we also provide 3-dimensional visualization tool TraVis, which is based on [4], available for free download. You are allowed to retrieve TraVis, especially [4], only for research purpose as a member of a non-commercial and academic institution.

Even though the library is premeditate to macroscopic train timetabling, we cannot predict which extensions and integrations are useful in future versions.

In addition we want to mention that such a standardization is already done on a microscopic level for rolling stock, infrastructure and timetables, see [5]. Our work can be considered as a first step to natural extension of RailML to a macroscopic level.

2 TTPlib conception

The level of detail of a railway infrastructure or railway operation model depends on the quality and accuracy requirements for generating appropriate results and, of course, on the availability and reliability of the data. For long term and strategic planning problems high accuracy data is not manageable, might not exists or is even not providable on time. Also formal and legal reasons might prohibit free access to high detailed infrastructure data.

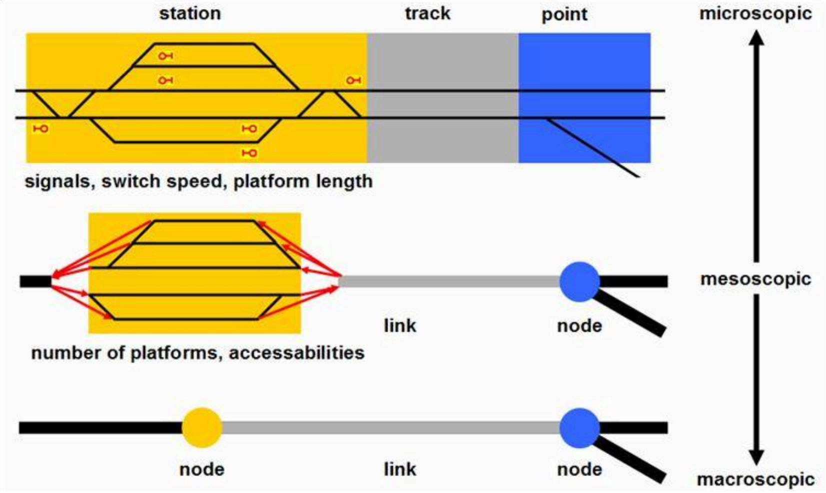
station
track
point
microscopic
signals, switch speed, platform length
mesoscopic
link
node
number of platforms, accessabilities
node
link
node
macroscopic

Figure 1: From microscopic and macroscopic models.

These are reasons why different models of scale has been established:

- *microscopic models* require high detailed data to produce reliable and high quality results, i.e. for running time calculation and the simulation of timetables and railway operations
- *mesoscopic models* are produced if no microscopic data is available
- *macroscopic models* embrace coarse and aggregated structures; real-world applications are vehicle circulation, long term traffic and strategic infrastructure planning

Figure 1 illustrates the differences of these models; showing the level of detail of a railway node and link system. The train timetabling problem – considered by various authors in the literature, see [6] and [7], as a combinatorial optimization problem – has to be tackled on a macroscopic level. We deliberately separated the input data of a train timetabling problem; reflecting the trend to perceive infrastructure providers and train operators as distinct institutions. An instance of the TTPlib train timetabling problem, see Figure 2, consists of two parts:

- a *railway infrastructure*, which specifies train types, train sets, stations, tracks and some detailed corresponding planning data
- a *set of train requests*, which defines the demanded trains to route through the network and their *monetary value* attached to them by the train operator

The train timetabling problem is then to find – given a railway network and a set of train requests – a conflict free set of train routes with maximum value.

2.1 TTPlib macroscopic railway data

As already mentioned, we assume a railway infrastructure on a macroscopic level. A railway network consists of stations and connecting tracks. A track is directed

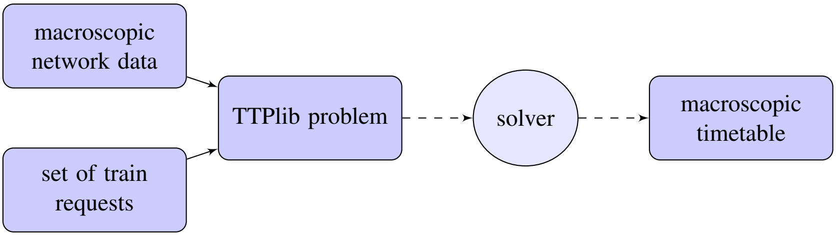

Figure 2: Planning process and idea of TTPlib instance.

and can only be approached by one train at each time. By definition overtaking is not possible on a single track and only allowed inside stations; but it is possible to declare additional points for line crossings in the network, so called pseudo-stations, to deal with overtaking on the railroad.

Beside these artificial stations, dead-end (only one side) and standard stations (exactly two sides to enter and leave) can be defined. In addition to the start and end station of each track, information which sides of the stations are connected by this track are needed.

Trains must belong to a specified aggregated macroscopic train type. A tree structure helps to build up the hierarchy of these sets, see Figure 3. For each train type all properties of the higher level train type are valid; analogously restrictions of the higher level train type has to be fulfilled, as well as the train type specific ones.

All times are discretized to some defined unit (given in number of seconds, i.e. we use a macroscopic discretization of 60 seconds). A train route on a macroscopic view is a path in discretized time and space through a suitable digraph. The nodes of the digraph correspond to events of the stations in time. Three possible events inside a normal station are considered – arrival, departure (not considered in pseudo-stations) and passing for each side.

This gives an intuition of the nodes (tupels of station, side, node-type and train-type) of the scheduling digraph. These nodes are suitably connected by directed arcs - representing certain activities. Three different kinds are covered by our model, driving on tracks, making a turnover and waiting inside a station. A turnover inside a station specifies the minimal time a train of this type needs to enter and leave this station in the same direction or on the same side.

Obviously, driving times highly depend on the status (velocity, weight etc.) of the train at the beginning and at the end of the considered track. We assume a fixed valid maximal passing velocity per track start and end. With respect to the introduced events at each station, we distinguish between four different main drive possibilities (modes):

1. stop at start point of the track and stop at the end,
2. stop at start point of the track and pass at the end,
3. pass at start point of the track and stop at the end,
4. pass at start point of the track and pass at the end.

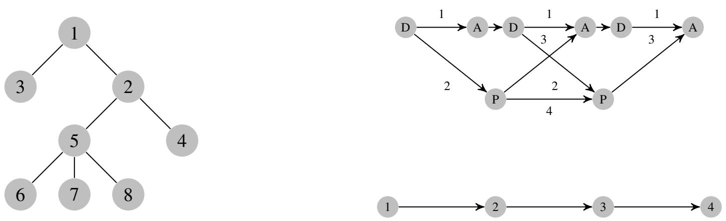

Figure 3: Example of train type tree and digraph.

In Figure 3 such a exemplary event-activity or scheduling digraph for a corridor-network from station 1 to 4 is shown. We use node labels (A) for arrival, (D) for departure and (P) for passing; all labels of the arcs denote drive mode.

Instead of block section exclusivity, which is handled on a highly detailed microscopic stage, we ensure track safety with respect to given minimal headway times at departure. The minimal abiding difference of the departure times between two consecutive trains is defined using the respective headway times. For each driving possibility on a track we need the information of the next possible time a train (depending on the train type) can enter this track - driving in the same direction. Obviously, there may be situations, where you need to specify headway times between different tracks (or at arrival). For instance:

- driving arcs on tracks of opposite directions (so called single tracks),
- driving arcs on tracks that intersect,
- restrictions of meeting point (tunnels, short sidings),

We deliberately decided to ignore these special cases in the instances for the first version of TTPlib. But our macroscopic infrastructure scheme includes a natural possibility to define these additional headways.

Another type of operational constraints, which has to be satisfied on a macroscopic level, are station capacities. At any discretized point in time, given station capacities has to be fulfilled. These define the maximal number of trains of a given train type that can be handled; we distinguish between:

1. platform capacity,
2. running capacity,
3. overall capacity.

Platform capacity specifies the maximal number of simultaneously stopping, while running capacity restricts the number of simultaneously passing trains of a given train type. Finally, overall capacity limits the number of trains that can be handled at one time unit in this station (independently of their driving mode). The screenshot of TraVis in Figure 4 visualizes a macroscopic train route in space and time on the left and the described headways of a single track on the right.

2.2 TTPlib train requests

Second input of an instance of the train timetabling problem is a set of requested train slots.

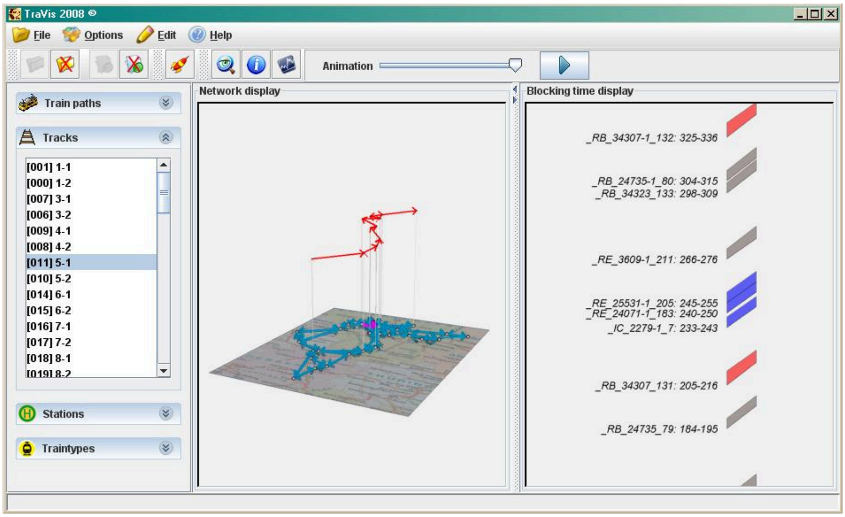

Figure 4: Train route through the network (left) and single track headway display (right).

A slot request describes the desired course of a train through the network. A train is identified by TrainName and TrainNumbers. A train type – taken from the leafs of the train type tree from the infrastructure file – should be specified as well. This is the way we distinguish between train sets and 'real' train types. Finally, a *basic value* is attached to the train.

Specification of departure and final stations are mandatory. Optionally, in-between stops may be specified. To reflect the fact that there is usually some flexibility in the planning of a train with respect to arrival and departure times, a request specified *time windows* is set around the *optimal time*, within which arrival and departure at specified stops may be scheduled. Diversion from the optimal times within the bounds of the time window may be penalized in a linear manner as follows:

Let the request specify the arrival or departure at some station by the *time-value specification* (`OptimalTime`, `MinimalTime`, `LeftSlope`, `MaximalTime`, `RightSlope`) Let `ScheduledTime` be the time that the arrival or departure at some stop is actually scheduled. The penalty incurred at this time-value specification is then computed by

$$\text{penalty} = \begin{cases} (\texttt{OptimalTime} - \texttt{ScheduledTime}) \cdot \texttt{LeftSlope} \\ \qquad \text{if } \texttt{MinimalTime} \leq \texttt{ScheduledTime} < \texttt{OptimalTime} \\ (\texttt{ScheduledTime} - \texttt{OptimalTime}) \cdot \texttt{RightSlope} \\ \qquad \text{if } \texttt{OptimalTime} < \texttt{ScheduledTime} \leq \texttt{MaximalTime} \end{cases} \tag{1}$$

The value of a scheduled train as given in a *solution* (see below) is defined as the basic value minus the sum of all penalties that the train 'collects' during its course.

We allow 'forcing' a train into a solution by assigning to it a special flag `fixed`. If this flag is set, this train has to be scheduled for a computed solution to be feasible.

Optionally, for intermediate stops, minimal and maximal *dwelling time*, that is, the time that a train should wait at the platform, can be specified. This is useful for passenger trains that possibly need a longer stop at some stations where many passengers enter or leave the train.

2.3 TTPlib solution

For the solution of a train timetabling problem - a macroscopic feasible timetable – it is sufficient to specify the scheduled trains and their route through the network. Accordingly, for each train request, we expect only a sorted list of stations with arrival and departure time, and a list of connecting tracks. Furthermore objective values of each path and of the entire solution, user-specific station and track labels, and valid upper bounds can be optionally specified.

3 TTPlib scenarios

TTPlib involves – beside some artificial instances, which we refrain from describing here in detail – a set of realistic macroscopic train timetabling problems for the German long distance railway network region of HA-KA-FU. These scenarios were created during the project *Railway Slot Allocation*. We provide some simplified macroscopic infrastructure data sets of this area, consisting of 10 train types, 37 stations, 120 tracks leading to 720 driving and 4320 headway times. These data was generated by a prototype of InfraGen, a tool developed at SFWBB to gain macroscopic data from meso- or microscopic ones. Network hakafu_stations additionally deals with 1110 station capacities.

To prove that the approximated running times can be confirmed, and that the train paths are feasible without conflicts, a macroscopic timetable must be mapped back onto a high accuracy model; in the idealized setting the starting point of the analysis. The feasibility of a macroscopic timetable can then be certified under consideration of buffer times, or minimum headways, within the mesoscopic model. As an example, when conflicts from super-imposed block occupation times of two trains are remaining, a conflict protocol is produced. Thus the macroscopic model can be adjusted by changing the allowed headways or disaggregation of train types. In Figure 5 the idealized workflow of this evaluation process for the HA-KA-FU instances is shown. In our project we did mesoscopic simulation in RailSys®, see [8], to derive and calibrate our macroscopic model. Due to the lack of free available data, we could not perform microscopic simulations.

All request sets of HA-KA-FU are based on an original timetable of the year 2002 in that area. Considering all trains of a normal weekday during 8:00 to 10:00,

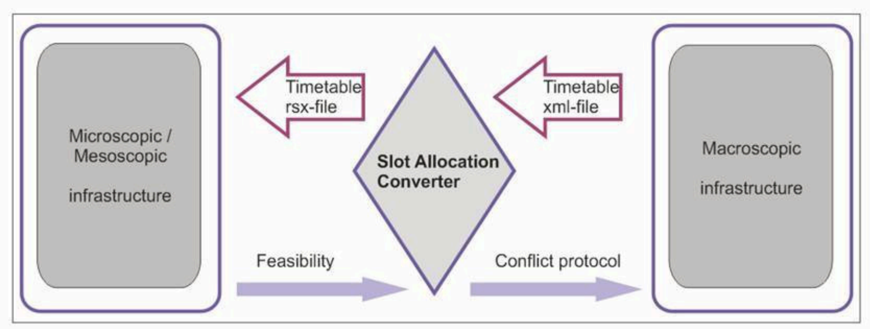

Figure 5: Systematic evaluation of macroscopic models.

Table 1: Main scenarios of TTPlib 2008.

infrastructure	request set	#trains	time	description
wheel	req11	288	–	artificial instance
hakafu_simple	req37	334	8-12	added cargo traffic
hakafu_simple	req39	358	8-12	added regional passenger traffic
hakafu_simple	req41	382	8-12	added interregional passenger traffic
hakafu_simple	req43	319	8-12	added mixed traffic
hakafu_simple	req45	344	8-12	basis timetable

8:00 to 12:00 and 8:00 to 16:00 o'clock. Moreover we provide request sets with additional comprehensible demands; All parameters of the specified objective or utility function are chosen w.r.t. the official German railway price system or w.r.t. realistic demand models, details on them can be found in [9] and [10].

Table 1 lists exemplary the sizes of some main scenarios and Figure 6 shows the considered railway networks.

4 Conclusions

The goal of TTPlib is to provide realistic infrastructure data to the railway community, which can be used to compare different approaches, mathematical models or optimization algorithms. In addition, we wanted to create a place for sharing not only data but also other information. Therefore TTPlib provides a collection of references on train timetabling, a list of upcoming conferences in the field, and a mailing list.

Furthermore TraVis, which is an academic visualization tool developed at ZIB, has been made available for free download. On the one hand visualizations can

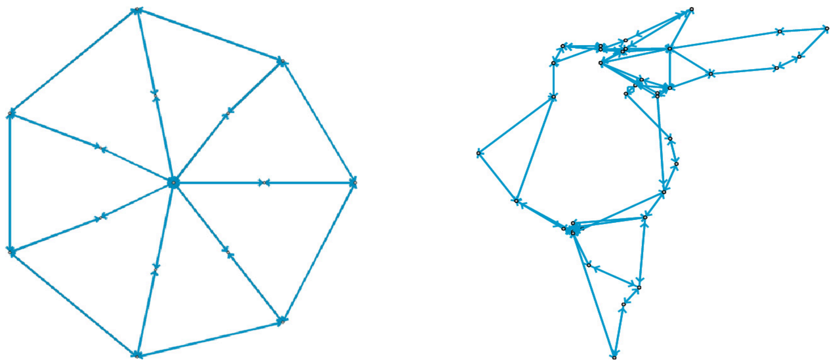

Figure 6: Railway network examples wheel and Ha-Ka-Fu.

help users to illustrate and understand their problems; on the other hand it supports the process of developing solution algorithms and gaining deeper insights.

Everybody in the railway community is invited to use the TTPlib scenarios in publications, to download and use TraVis, to submit solutions or bounds of TTPlib instances, as well as references to new papers, or conference announcements, either via the web interfaces at http://ttplib.zib.de or at the e-mail address ttplib-webmaster@zib.de.

We hope that TTPlib and TraVis will help to develop and compare railway timetabling approaches; improving the planning process of timetables and in consequence the quality of timetables in a challenging railway future.

Acknowledgements

Special thanks to the initiators and contributors of TTPlib, who have helped build up the library by providing railway infrastructure instances, sophisticated train requests or solutions. In particular, this includes Ralf Borndörfer, Martin Balser, Andreas Gille, Marek Grochowski, Martin Grötschel, Gottfried Ilgmann, Sascha Lukac, Kay Mitusch, Annette Mura, Klemens Polatschek, Alfons Radtke, Anke Reuther, Thomas Siefer, Jürgen Siegmann and Christian Weise.

We also thank Berkan Erol and Mathias Kinder for steady development on TraVis growing into a helpful instrument.

This work was funded by the Federal Ministry of Economics and Technology (BMWi), project *Railway Slot Allocation (in german Trassenbörse)*, grant 19M7015B.

References

[1] Reinelt, G., Tsplib - a traveling salesman problem library. *ORSA Journal on Computing*, **3**, pp. 376–384, 1991.

[2] Achterberg, T., Koch, T. & Martin, A., MIPLIB 2003. *Operations Research Letters*, **34(4)**, pp. 1–12, 2006. ZIB-Report 05-28.

[3] Ttplib-homepage, 2008. Http://ttplib.zib.de.

[4] Javaview-homepage, 2008. Http://www.javaview.de.

[5] Railml-homepage, 2008. Http://www.railml.org.

[6] Brännlund, U., Lindberg, P., Nou, A. & Nilsson, J.E., Railway timetabling using langangian relaxation. *Transportation Science*, **32(4)**, pp. 358–369, 1998.

[7] Caprara, A., Fischetti, M. & Toth, P., Modeling and solving the train timetabling problem. *Operations Research*, **50(5)**, pp. 851–861, 2002.

[8] Railsys-homepage, 2008. Http://www.railsys.de.

[9] Borndörfer, R., Grötschel, M., Lukac, S., Mitusch, K., Schlechte, T., Schultz, S. & Tanner, A., An auctioning approach to railway slot allocation. *Competition and Regulation in Network Industries*, **1(2)**, pp. 163–196, 2006.

[10] Reuther, A., *Kombinatorische Auktionen und ihre Anwendungen im Schienenverkehr*. Master's thesis, Technische Universität Berlin, 2005.