Locomotive and Wagon Scheduling in Freight Transport

Armin Fügenschuh[1], Henning Homfeld[1], Andreas Huck[2] and Alexander Martin[1]

[1] Technische Universität Darmstadt, Arbeitsgruppe Optimierung
64289, Darmstadt, Schlossgartenstr. 7, Germany
{fuegenschuh,homfeld,martin}@mathematik.tu-darmstadt.de
[2] Deutsche Bahn AG, Konzernentwicklung, GSU 1
60326, Frankfurt, Stephensonstraße 1, Germany
andreas.huck@bahn.de

Abstract. We present a new model for a strategic locomotive scheduling problem arising at the Deutsche Bahn AG. The model is based on a multi-commodity min-cost flow formulation that is also used for public bus scheduling problems. However, several new aspects have to be additionally taken into account, such as cyclic departures of the trains, time windows on starting and arrival times, network-load dependend travel times, and a transfer of wagons between trains. The model is formulated as an integer programming problem, and solutions are obtained using commercial standard software. Computational results for several test instances are presented.

Keywords. Freight Transport, Vehicle Scheduling, Time Windows, Integer Programming.

1 Introduction

Deutsche Bahn AG (DB) is the largest German railway company with 216,000 employees and a turnover of 25 billion Euros in 2005. DB is active in both passenger and freight transportation. Per year, 1.8 billion passengers (72 billion passenger kilometers) and 253 million tons of goods (77 billion ton kilometers) are transported. Moreover, DB is the owner of the German railway system, where DB freight and passenger trains travel 887 million kilometers per year, and external railway companies around 110 million kilometers. The overall length of the railways is 34,000 kilometers, and about 4,400 freight trains and 30,000 passenger trains per day traverse this network [1,2]. All in all, DB's network is considered as one of the most dense and most frequently used railway network in the world.

For the long-term simulations and future predictions of the network load, DB developed a complex simulation tool. The entire simulation tool can be considered as a chain, which decomposes into several components. To this end it is possible to model the normal course of business operations, and to analyze

ATMOS 2006
6th Workshop on Algorithmic Methods and Models for Optimization of Railways
http://drops.dagstuhl.de/opus/volltexte/2006/686

the influence of changing external parameters, such as future demands on goods, possible network expansions or the sensitivity to the price for oil. The components of the tool chain interact by generating output data, which is used as input for other parts of the chain. The entire simulation tool has evolved over the last 5 years, and is still under continuous improvement. In this article we describe the development of a new segment for the tool chain.

Currently, a chain's segment called *train scheduler* is responsible for the generation of trains from individual wagons. As soon as enough wagons are assembled, the train is started. That means, the starting times of the trains are not aligned to some timetable, they just follow the estimated customers' productions and demand peaks in the simulation. Hereby it is assumed that a locomotive for pulling this train is always and immediately available. Moreover, the locomotives are not scheduled.

From the area of public bus transport it is known that routing and scheduling of the vehicles is an important field in the optimization of the operator's business process. For instance, Löbel [3] and Gintner, Kliewer, and Suhl [4] developed models for the scheduling of public buses, which led to significant cost savings in public transport. Moreover, it was noted by Daduna and Völker [5] that in public bus transport an even fewer number of buses is necessary to serve all trips, if the starting times of the vehicles are altered within some small interval. Later, Fügenschuh [6] also included the customers' demands into the optimization such that the scheduling problem of the vehicles is solved together with the starting time problem of the trips, which leads to further reductions of vehicles and costs.

The scope of our research is to carry over these observations to the locomotive scheduling in freight transport of Deutsche Bahn. Several new aspects have to be taken into account, such as cyclic departures of the trains, time windows on starting and arrival times, network-load dependend travel times, and a transfer of wagons between trains. The model presented in this article aims at a support of strategic simulations of the future, for example, simulating the network load in freight transport in the year 2015. The model is formulated as a linear integer programming problem (IP, for short). We give a computational evaluation of the resulting IPs and show whether standard commercial IP solvers (such as ILOG Cplex [7]) are able to handle problem sizes of instances that occur in the context of DB.

The remainder of this article is organized as follows. In Section 2 we describe the problem in greater detail. In Section 3 we provide a stepwise refined model, formulated as an integer programming problem. In Section 4 we present computational results for the different varients of our model using standard software. An outlook to further work is finally given in Section 5. For a survey on combinatorial optimization problems in connection with rail transport we refer to the literature, for instance, the survey articles of Bussieck, Winter, and Zimmermann [8] and Caprara, Fischetti, Toth, and Vigo [9].

2 The Problem Settings

In this section we give details of the integrated scheduling problem and introduce the terminology used at DB.

Wagons. A *wagon* is a rolling stock for freight transport. The wagons have to be delivered between a source and a destination point (goods station) within the network. Large customers produce and/or consume so much goods that they order whole trains. In these cases, the route of the wagons equals the route of the train. Smaller customers order individual wagons. Then the wagons of different customers are assembled to trains and pulled as a whole to an intermediate destination (a shunting yard), where the trains are disaggregated and reassembled to new trains. The trains and the yards where the wagon transfer between trains are known in advance. When changing the starting time of the trains, one has to take care that these transfers still remain feasible.

Trains. A *freight train* (also called *production trip*) consists of several *wagons*. Each train has a *start* and a *destination*, which are goods stations or railroad shunting yards. Also given are *starting times* and *arrival times*. These can be either fixed times or intervals, in which the start or the arrival has to take place. We assume that the trains start cyclical every 24 hours. The *trip duration* is the time difference between start and arrival. The average travel speed of freight trains is not as high as in passenger transport, especially at daytime, when passenger trains always have priority, such that some trips can last up to 3 days. The trains have different length and weight and thus require locomotives with sufficient driving power. In contrast to the problems described by Ahuja et al. [10] or Ziarati [11], a train is always pulled by a single locomotive. At the start a locomotive is attached to the train, and at the destination station it is detached (uncoupled). For both *coupling* processes, a certain train-dependent amount of time has to be taken into account (15 to 30 minutes). At this stage, technical checks and refueling of diesel locomotives are carried out.

Locomotives. DB uses up to 30 different locomotives of several manufacturers. However, the differences between them are often minor, so they are grouped into 3 to 6 *classes* of similar locomotives. The main differences among the classes are the driving power of the engines, and the *traction* (i.e., the motor type, diesel or electrical). Electrical locomotives can only be used on electrical tracks, whereas diesel locomotives in principle can drive everywhere. However, diesel soots the electrical wires, so one wants to avoid their deployment on such tracks. Hence, it is only possible to assign such locomotives to trains that have a sufficient power and the right traction for the track.

Deadheads. A locomotive is either active, i.e., pulling a train, or deadheading, i.e., driving under its own power without pulling a train from the destination station of one train to the start of another train. The duration for a deadhead trip depends on the distance between these two points, and on the class of the locomotive (diesel and electrical might have to use different routes), but not so much on the network load (i.e., independent of daytime or nighttime), because it is assumed that a single locomotive can always be pushed through.

Goals and Objectives. The main goal is to compute feasible starting and arrival times of the trains such that the wagons are transported as fast as possible from their start to the destination within the trains. At intermediate shunting stations the stopover of wagons should not exceed certain limits. The main objective is to reduce operating expenses, that is, to use as few locomotives as possible to pull all trains and, on a subordinate level, to schedule the locomotives in such a way that the deadhead trips are as short as possible.

3 Models

The models we describe in this section can be classified by one main characteristic, the starting time intervals. In the first type of the models, the starting time is given by the pre-scheduler of the tool chain. By this tool, a train is started as soon as enough wagons are assembled. What "enough" in this context means is guided by a local criterion, which is mainly based on the number, total weight, and total length of the wagons. However, this local criterion does not take the availability of locomotives into account. It is simply assumed that a locomotive is always available, if required by some train. A model for scheduling the locomotives under this assumptions is presented in Section 3.1.

On the other hand, there is always a little flexibility in the departure and arrival of the trains, which has to be negotiated with the customers. In Section 3.2 we describe a model where the starting time can vary within a given interval. This model is much more complex, since it has to take care of the syncronization of the train departures and the wagon schedules. A further refinement of this model is given in Section 3.3. Here the trip durations are not constant but dynamically depending on the actual network load.

For all models we use the following notations. Let $\mathcal{V}$ be the set of freight trains (production trips), and let $\mathcal{B}$ be the set of locomotive classes. We introduce a parameter $a_{b,i} \in \{0,1\}$ with $a_{b,i} = 0$ if a class b locomotive cannot pull train i. Let $\mathcal{A} := \mathcal{V} \times \mathcal{V}$ denote the set of all deadhead trips. Denote $a_{b,(i,j)} := a_{b,i} \cdot a_{b,j} \in \{0,1\}$, then we have $a_{b,(i,j)} = 0$ if and only if the deadhead trip from i to j is not feasible for a class b locomotive. Moreover, those $a_{b,(i,j)}$ can be set to zero where the corresponding deadhead trip exceeds a certain length.

3.1 Fixed Starting Times

To begin with, we take the starting and arrival times for the trains as they were computed by the train scheduler of the tool chain. This computation also ensures that individual wagons can transfer between trains. We introduce a decision variable $x_{b,(i,j)} \in \{0, a_{b,(i,j)}\}$ with $x_{b,(i,j)} = 1$ if trains i and j are connected and both served with a locomotive of class b, and $x_{b,(i,j)} = 0$ otherwise. Each train j

must be served with one class of locomotive, that is,

$$\sum_{b \in \mathcal{B}} \sum_{i:(i,j) \in \mathcal{A}} x_{b,(i,j)} = 1. \tag{1}$$

There is a flow conservation in the sense that the cycles of each class b and each trip j must be closed, that is,

$$\sum_{i:(i,j) \in \mathcal{A}} x_{b,(i,j)} = \sum_{k:(j,k) \in \mathcal{A}} x_{b,(j,k)}. \tag{2}$$

The connection of production and deadhead trips is called *cycle*. This notion is justified since each trip has a unique predecessor and a unique successor, and therefore at some point each locomotive will serve the first trip again. We denote by $\lambda_{b,(i,j)}$ the number of locomotives of class b that are additionally necessary due to the connection of i with j. Similar to Liebchen and Möhring [12] this number is computed as

$$\lambda_{b,(i,j)} := \left\lceil \frac{\hat{t}_i - \hat{t}_j + \delta_{b,(i,j)}}{1440} \right\rceil \geq 0. \tag{3}$$

Here $\hat{t}_i, \hat{t}_j$ are the pre-scheduled starting times of trains i and j, respectively. The constant 1440 refers to the number of minutes per day, which is the basis of the cycles (i.e., all trips are repeated on a daily base), and $\delta_{b,(i,j)}$ denotes the total trip and deadhead trip duration, that is,

$$\delta_{b,(i,j)} := \delta_i^{\text{trp}} + \delta_i^{\text{uncpl}} + \delta_{b,(i,j)}^{\text{dhd}} + \delta_j^{\text{cpl}}, \tag{4}$$

where

- δ_i^{trp} denotes the trip duration, i.e., the time the locomotive is active while pulling train i,
- δ_i^{uncpl} denotes the time for uncoupling the locomotive from the train at the arrival,
- $\delta_{b,(i,j)}^{\text{dhd}}$ denotes the time for deadheading from the end of i to the start of train j, and
- δ_j^{cpl} denotes the time for coupling the locomotive to the train at the start of j.

Remark that the driving time δ_i^{trp} is assumed to be independent of the actual class, whereas the deadhead time $\delta_{b,(i,j)}^{\text{dhd}}$ is class dependent (since diesel and electrical might use different routes).

A *capacity* in form of an upper bound B_b on the number of available locomotives of class b can be specified:

$$\sum_{(i,j) \in \mathcal{A}} \lambda_{b,(i,j)} x_{b,(i,j)} \leq B_b. \tag{5}$$

As objective function we want to minimize the total costs defined as

$$\sum_{b\in\mathcal{B}}\sum_{(i,j)\in\mathcal{A}}(\gamma_b^{\mathrm{cls}}\lambda_{b,(i,j)}+\gamma_{b,(i,j)}^{\mathrm{dhd}})x_{b,(i,j)} \tag{6}$$

where

- γ_b^{cls} denotes the costs for a locomotive of class b,
- $\gamma_{b,(i,j)}^{\mathrm{dhd}}$ denotes the costs in connection with deadheading from i to j with locomotive b.

The most important objective is the reduction of the deployed locomotives, and second is the reduction of deadhead costs. Hence $\gamma_b^{\mathrm{cls}} \gg \gamma_{b,(i,j)}^{\mathrm{dhd}}$.

The optimization problem is the minimization of (6) subject to the constraints (1), (2), (5), and the integrality of all $x_{b,(i,j)}$. Due to the cyclic character of the schedules of the locomotives we call this problem the *capacitated cyclic vehicle scheduling problem* (CVSP). In the case of no upper bounds (or $B_b = \infty$) we also speak of the *uncapacitated* CVSP.

For $|\mathcal{B}| = 1$, this problem reduces to a single-commodity minimum-cost flow problem, which can be solved efficiently by polynomial or pseudopolynomial algorithms (for instance by the Hungarian Method, see Ahuja, Magnanti, and Orlin [13] for details). For $|\mathcal{B}| > 1$, the uncapacitated CVSP is a multi-commodity min-cost flow problem, which is known to be NP-hard. Moreover, it is NP-complete to decide whether a feasible solution exists for the capacitated CVSP (see Löbel [3]).

3.2 Variable Starting Times

We change the above model for locomotive scheduling for the case where the starting times of the trains are allowed to be changed within a given interval. The corresponding model is called (capacitated or uncapacitated) *cyclic vehicle scheduling problem with time windows* (CVSPTW). In comparison with the CVSP, the CVSPTW model is more complicated, because we have to take care of the transfer of wagons between trains.

We first introduce bounds on the starting time of the trains. The starting time for $i \in \mathcal{V}$ is denoted by $t_i \in \mathbb{Z}_+$. The set $\mathcal{V}$ is devided into two subsets, $\mathcal{V} = \mathcal{C} \dot\cup \mathcal{S}$. In $\mathcal{C}$ there are all trips i with a connected starting time interval $[\underline{t}_i, \bar{t}_i] \subseteq [0, 1439] \cap \mathbb{Z}$ in which the starting time must lie:

$$\underline{t}_i \le t_i \le \bar{t}_i. \tag{7}$$

In particular, each trip has to start during the first day. If the starting time exceeds this limit, the interval is split into two. This is the case for all trips $i \in \mathcal{S}$. Their starting time must be in $([\underline{t}_i, 1439] \cup [0, \bar{t}_i]) \cap \mathbb{Z}$. We introduce a binary variable $y_i \in \{0, 1\}$ with $y_i = 1$ if the starting time is in $[0, \bar{t}_i]$ (after midnight), and $y_i = 0$ if it is in $[\underline{t}_i, 1439]$ (before midnight). We then obtain the following constraints for the starting times:

$$\underline{t}_i(1 - y_i) \le t_i \le 1439 + (\bar{t}_i - 1439)y_i. \tag{8}$$

As in the CVSP model we introduce decision variables $x_{b,(i,j)} \in \{0, a_{b,(i,j)}\}$ for the deadhead trips, and use the same multi-commodity flow formulation:

$$\sum_{b\in\mathcal{B}} \sum_{i:(i,j)\in\mathcal{A}} x_{b,(i,j)} = 1, \tag{9}$$

and

$$\sum_{i:(i,j)\in\mathcal{A}} x_{b,(i,j)} = \sum_{k:(j,k)\in\mathcal{A}} x_{b,(j,k)}. \tag{10}$$

If $i, j \in \mathcal{V}$ are connected, then the corresponding starting times must be synchronized,

$$t_i + \delta_{b,(i,j)} - 1440 l_{b,(i,j)} \le t_j + 5760(1 - x_{b,(i,j)}). \tag{11}$$

The constant $5760 = 4 \cdot 1440$ reflects the assumption that the train starting time is fixed to the first day, and a trip duration per train is at most 3 days. Since the starting time selection is now integrated in the model, we cannot compute the number of locomotives $\lambda_{b,(i,j)}$ beforehand, as in the case of the CVSP. Instead we introduce a variable $l_{b,(i,j)} \in \mathbb{Z}_+$ which represents the number of additional locomotives due to the connection of i with j.

Finally we have to syncronize those trains $i, j \in \mathcal{V}$ where wagons transfer from one to the other. Since all trains are cyclic, there are in principle no missed transfers. That is, if j departs before the arrival of i, then the wagons have to wait at most 24 hours until the arrival of the next train j. However, long idle waiting times of the wagons are undesired. This is modeled by the following inequality:

$$0 \le (t_j + 1440 q_{i,j}) - (t_i + \delta_i^{\text{trp}} + \delta_{i,j}^{\text{shnt}}) \le 719 + 720 p_{i,j}, \tag{12}$$

where

- $\delta_{i,j}^{\text{shnt}}$ denotes the time for shunting the wagon from i to j,
- $q_{i,j} \in \{0, \dots, 4\}$ is a variable to shift the starting time of j within the same day of the arrival of i, and
- $p_{i,j} \in \{0, 1\}$ is a decision variable with $p_{i,j} = 1$ if and only if the wagons from i to j are idle for more than 12 hours.

In this formulation, the variables $p_{i,j}$ are put into the objective function with a suitable scaling coefficient. In this way it is possible to analyze how many locomotive could potentially be saved if some transfers are missed. It is also possible to set $p_{i,j} := 0$ for all transfers i, j, which gives a hard constraint, i.e., the transfers are much more important than saved locomotives.

The objective is to minimize the total costs defined as

$$\sum_{(i,j)\in\mathcal{A}} \gamma_{i,j}^{\text{idle}} p_{i,j} + \sum_{b\in\mathcal{B}} \sum_{(i,j)\in\mathcal{A}'} (\gamma_b^{\text{cls}} l_{b,(i,j)} + \gamma_{b,(i,j)}^{\text{dhd}} x_{b,(i,j)}), \tag{13}$$

where γ_b^{cls} and $\gamma_{b,(i,j)}^{\text{dhd}}$ are defined as above, and $\gamma_{i,j}^{\text{idle}}$ denotes the costs for idle wagons, i.e., a wagon missing its subsequent train which then has to wait for more than 12 hours. In general, these coefficients reflect the following ordering: Reducing idle wagons is most important, since high contract penalties for late arrivals have to be paid. Second is now the reduction of the deployed locomotives, and the reduction of deadhead costs is moved to the third place.

3.3 Netload-dependent Travel Times

We further refine the above CVSPTW model to the case that the driving time of the trains is not constant, but a function depending on the total network load. At daytime the freight transport has to wait for the passenger transport such that the traveling speed is much lower than at nighttime.

To this end, the whole day is partitioned into a discrete number of time slices $\mathcal{H} = \{1, \ldots, H\}$, that is, $[0, 1439] = \dot{\bigcup}_{h\in\mathcal{H}}[\underline{\tau}_h, \overline{\tau}_h]$. For each train i we introduce decision variables $z_{i,h} \in \{0, 1\}$ with $z_{i,h} = 1$ if and only if the train starts within time slice h. Exactly one slice must be selected:

$$\sum_{h\in\mathcal{H}} z_{i,h} = 1. \tag{14}$$

The slice selection is coupled to the starting time of train i such that the "right" slice h is chosen:

$$\underline{\tau}_h - t_i \leq 1439(1 - z_{i,h}), \tag{15}$$

$$t_i - \overline{\tau}_h \leq 1439(1 - z_{i,h}). \tag{16}$$

Then the trip duration of train i is given by

$$d_i^{\mathrm{trp}} = \sum_{h\in\mathcal{H}} \delta_{i,h}^{\mathrm{trp}} z_{i,h}, \tag{17}$$

where $d_i^{\mathrm{trp}} \in \mathbb{Z}_+$ is a new variable, and $\delta_{i,h}^{\mathrm{trp}}$ is a parameter giving the trip duration of train i when being started in slice h. The actual values of $\delta_{i,h}^{\mathrm{trp}}$ are statistically estimated along historical data. Then in the CVSPTW model, δ_i^{trp} is replaced by d_i^{trp}.

Morover, in case of dynamic trip durations it is desired to specify bounds on the arrival time of some of the trains in addition to the starting time bounds, i.e.,

$$\underline{T}_i \leq t_i + d_i^{\mathrm{trp}} \leq \overline{T}_i, \tag{18}$$

where $[\underline{T}_i, \overline{T}_i] \subseteq [0, 5759] \cap \mathbb{Z}$ is the arrival time interval of train i.

4 Computational Results

The CVSP and the CVSPTW problems are formulated as integer programming problems. Thus one can use standard IP solvers to compute feasible or optimal solutions. For an introduction to integer programming we refer to the literature (Nemhauser and Wolsey [14] for instance). For our computational studies we used

ILOG Cplex 10 [7], one of the currently fastest IP solvers. We made exhaustive tests with the number of parameters that guide the solution process. We made overall best experiences when setting cut generation and probing to the highest level. These settings yield the shortest computation times that are reported in the sequel. The software was running on a Linux server with 16 GByte main memory and 4 dual core AMD Opteron 880 processors running at 2.4 GHz each. Cplex is able to make use out of such environment by parallelizing the branch-and-bound tree.

From the data base of the tool chain we extracted 7 test instances, which we refer to as `A`, `B`, `C`, `D`, `E`, `F`, `G` in the sequel. Here `A` is the smallest instance with 42 trains and 3 classes of locomotives, whereas `G` is the largest, having 1,537 trains and 4 classes of locomotives (for details see the first three columns of Table 1 or Table 2). The instances are related to certain regions within the DB railway network. For example, instance `G` consists of trains mainly from the south of Germany, trains from `A` serve north-south connections, whereas `E` comprises trains from all over Germany.

4.1 Results for the CVSP

At first we take the fixed starting times for the trains in the form they were generated by the train scheduling part of the tool chain. Here each train is assumed to depart as soon as enough wagons are assembled. For most instances the solver was able to find optimal solutions within short amount of time, with exception of `G`, where more than 3 hours of computation time was needed. Note that an instance with around n trains and m classes leads to integer programs with around n^2m many variables, which is more than 3 Mio. for instances `F` and `G`. For the instances from `A` to `G` the solution time grows with the size of the instance (see column 4 of Table 1 or Table 2).

name	trips	classes	time	locomotives	$\sum$	km
A	42	3	1	2/19/2	23	3,808
B	82	3	2	3/34/8	45	52
C	120	4	18	6/35/1/1	43	2,560
D	394	5	27	12/11/21/15/26	85	27,132
E	945	3	480	137/51/45	233	42,057
F	1,507	4	652	28/19/287/52	386	58,446
G	1,537	4	9,502	53/22/26/72	173	2,942

Table 1. Solutions for the uncapacitated CVSP.

The actual amount of locomotives per class that is needed to serve all trips is shown in column 5, the sum of these is given in column 6. Table 1 shows the solutions for the uncapacitated case, that is, it is possible to deploy arbitrarily

name	trips	classes	time	locomotives	$\sum$	km
A	42	3	1	6/15/2	23	958
B	82	3	2	3/26/16	45	52
C	120	4	7	15/26/1/1	43	1,919
D	394	5	27	12/14/22/19/20	87	26,776
E	945	3	2,837	150/38/45	233	35,375
F	1,507	4	4,971	28/19/216/123	386	56,130
G	1,537	4	13,864	53/22/46/52	173	2,838

Table 2. Solutions for the capacitated CVSP.

many locomotives of each class. In contrast, Table 2 shows the solutions of the capacitated case, that is, the number of locomotives per class is limited to the actual stock of DB. In general, the solution times for the capacitated case were higher, in particular for the larger instances `E`, `F`, `G`. However, the total number of locomotives is unchanged (with the exception of instance `D`), only the locomotives per class are different between the capacitated and the uncapacitated case. The last column of Table 1 and Table 2 shows the sum of deadhead trips lengths for all locomotives. Interestingly, the total length (in kilometer) of deadhead trips decrease when switching from the uncapacitated to the capacitated version of the CVSP. This could be ascribed to an increase in diesel locomotives for example, whose deadhead trips are in general shorter.

4.2 Results for the CVSPTW

In the CVSP instances the starting times of all trains were fixed to the times that were computed by the tool chain's scheduler. We now allow the starting times to be altered within a small time window centered around the pre-scheduled starting time, that is, we consider the (uncapacitated) vehicle scheduling problem with time windows (CVSPTW). It turns out that the computation time heavily depends on the actual size of the time windows. Generally speaking, the larger the time window, the more time the solver needs. For all computations we now impose a time limit of 3,600 seconds. We consider the three smallest instances `A`, `B`, and `C`, and take intervals of $\pm 10, \pm 30, \pm 60$, and ± 120 minutes around the current (pre-scheduled) starting time. Moreover, the solver Cplex allows to specify so-called starting solutions, which are integral feasible solutions to the problems that can be generated by other methods (primal heuristics, for instance). For our computations, we take the respective optimal solutions to the (uncapacitated) CVSP (presented in Table 1) as input for the ± 10 instances. Then, we take the optimal (or best feasible) solutions of ± 10 as input for ± 30, and so on. The computational results are shown in Table 3. For the other instances `D` to `G` the solver did not find feasible solutions, or did not even solve the root LP relaxation within the time limit. In column 3 of Table 3 we show the computation time in seconds. An asterisk (*) marks whether the given limit was reached. In those cases the integrality gap (i.e., the distance between the best known solution and the corresponding lower bound) shown in column 4 is non-zero. The last three columns of Table 3 depict the quality of the optimal or best feasible solution. It turns out that there is a significant potential to save locomotives by changing the departure times of the trains. The possible savings are higher the wider the time windows are open. The price one has to pay for this additional flexibility is the increasing solution time.

instance	time w.	time	gap	locomotives	$\sum$	km
A-42-3	$\pm$ 10	4	0.00 %	2/18/2	22	3, 931
A-42-3	$\pm$ 30	30	0.00 %	2/14/2	18	3, 731
A-42-3	$\pm$ 60	$3,600^*$	11.69 %	2/11/2	15	2, 516
A-42-3	$\pm$120	$3,600^*$	27.78 %	2/10/2	14	1, 640
B-82-3	$\pm$ 10	79	0.00 %	3/27/14	44	52
B-82-3	$\pm$ 30	183	0.00 %	3/23/16	42	52
B-82-3	$\pm$ 60	$3,600^*$	5.32 %	3/15/19	37	6
B-82-3	$\pm$120	$3,600^*$	9.59 %	2/14/16	32	1, 591
C-120-4	$\pm$ 10	753	0.00 %	5/35/1/1	42	2, 301
C-120-4	$\pm$ 30	$3,600^*$	21.84 %	6/32/1/1	40	3, 608
C-120-4	$\pm$ 60	$3,600^*$	38.14 %	6/30/1/1	38	3, 120
C-120-4	$\pm$120	$3,600^*$	79.06 %	6/29/1/1	37	2, 882

Table 3. CVSPTW with different time window sizes.

Note that in these computations the number of missed transfers of wagons is not shown. This is due to the fact that the corresponding variables $p_{i,j}$ are fixed to their lower bounds beforehand. This reflects the fact that a fast transfer of wagons is always more important than saved locomotives.

4.3 Results for the refined CVSPTW

Finally we consider the (uncapacitated) CVSPTW with netload-dependend travel times for the trains. This is the most evolved model in our hierarchy and it comes with no surprise that the solution times here are even higher than for the CVSPTW with constant traveling times. As before, we take the best feasible solution of an instance with a smaller time window as integer starting solution for the instance with the next bigger time window. Our results are summarized in Table 4. Depending on the starting time, the total travel time varies between 70 % and 130 % of the constant travel time. The bias of the travel time is estimated using historical data.

The results in Table 4 give an impression on the current state of problem sizes that can be solved using Cplex out of the box, with some altered parameter settings. Since we are far from solving even medium-size instances to optimality,

instance	time w.	time	gap	locomotives	$\sum$	km
A-42-3	$\pm$ 10	5	0.00 %	2/18/2	22	3,808
A-42-3	$\pm$ 30	34	0.00 %	2/13/2	17	2,510
A-42-3	$\pm$ 60	3,600*	19.38 %	2/11/2	15	2,516
A-42-3	$\pm$120	3,600*	36.99 %	3/9/2	14	2,125
B-82-3	$\pm$ 10	102	0.00 %	3/24/17	44	98
B-82-3	$\pm$ 30	213	0.00 %	3/19/18	40	52
B-82-3	$\pm$ 60	3,600*	9.41 %	3/20/13	36	6
B-82-3	$\pm$120	3,600*	16.56 %	2/18/11	31	1,649
C-120-4	$\pm$ 10	1,033	0.00 %	5/35/1/1	42	2,301
C-120-4	$\pm$ 30	3,600*	22.60 %	8/30/1/1	40	3,083
C-120-4	$\pm$ 60	3,600*	41.10 %	7/29/1/1	38	3,168
C-120-4	$\pm$120	3,600*	79.06 %	6/29/1/1	37	4,191

Table 4. CVSPTW with netload-dependent travel times.

one can try to use heuristic reductions of the problem's complexity. A first idea in this respect is to remove those deadhead trips that are above a certain limit. The hope is that sufficiently many long deadhead trips are removed by this, such that the remaining instance is smaller and computationally easier to solve, and on the other hand the solution is not too far away from the optimal solution. Similar as above, the best feasible solution of one instance with a tighter bound on the maximal deadhead length can be used as input for another instance with a larger bound. This we use here for the `C` instances, which cannot be solved to optimality within the given time limit. Our results are shown in Table 5. Column 3 of this table contains the upper bound on the deadhead trip length $\delta^{\mathrm{dhd}}_{b,(i,j)}$. If $\delta^{\mathrm{dhd}}_{b,(i,j)}$ exceeds the respective bound then $a_{b,(i,j)}$ is set to zero. The last three columns of Table 5 show the deviation from the optimality (for those instances `A` and `B` where the optimal solution is known, cf. Table 4). For convenience we repeat in the first row of each block `A`, `B`, `C` the results from Table 5 (with an upper bound of ∞, which is equivalent to no upper bound). As one can see, the loss is only a small one, whereas the solution time (given in column 4) is much lower now.

5 Conclusions and Further Work

In this article we presented new models for the strategic locomotive scheduling. These models were in part inspired by similar optimization problems in public bus transport. However, several new rail-specific requirements emerged such that the models could not be directly carried over. The models were formulated as integer programs. Thus commercial standard software for their solution could be applied. The evaluation of the capability of this software was part of our project. It turned out that for the cyclic vehicle scheduling without time

instance	time w.	bound	time	gap	locomotives	$\sum$	km
A-42-3	±30	∞	34	0.00 %	2/13/2	17	2,510
A-42-3	±30	600	27	0.00 %	3/12/2	17	958
A-42-3	±30	300	9	0.00 %	3/12/2	17	958
A-42-3	±30	100	2	0.00 %	3/16/2	21	6
A-42-3	±30	0	2	0.00 %	3/16/3	22	0
B-82-3	±30	∞	213	0.00 %	3/19/18	40	52
B-82-3	±30	100	39	0.00 %	3/25/12	40	52
B-82-3	±30	50	39	0.00 %	3/23/14	40	52
B-82-3	±30	10	24	0.00 %	3/24/14	41	6
B-82-3	±30	0	22	0.00 %	4/23/15	41	0
C-120-4	±30	∞	$3,600^*$	21.08 %	4/34/1/1	40	2,459
C-120-4	±30	600	$3,600^*$	18.63 %	4/34/1/1	40	2,459
C-120-4	±30	300	$3,600^*$	18.54 %	5/33/1/1	40	2,113
C-120-4	±30	100	$3,600^*$	13.85 %	6/37/1/1	45	486
C-120-4	±30	0	$3,600^*$	12.28 %	7/44/2/1	54	0

Table 5. CVSPTW with upper bounds on the deadhead trips.

windows, the software was able to solve even larger instances. As soon as time windows enter the scene, the sizes where global optimal solutions were computed was reduced by some orders of magnitude. Our further work thus aims at an improvement of the model formulation, where we want to develop and implement primal heuristics, problem-specific cutting planes, and branching rules such that larger models can be routinely solved to optimality. Also a further refinement of the presented model is on our agenda. An example in this direction is the inclusion of deadheading locomotives which are included in trains and thus do not need own power and staff for operating.

Acknowledgements. We thank Dr. Gerald Pfau (Deutsche Bahn AG), Jörg Wolfner (Universität Dortmund, Lehrstuhl Verkehrssysteme und -logistik), and Andreas Ginkel (Universität Göttingen) for fruitful and productive discussions. Last but not least we would like to thank the anonymous reviewers for their various helpful comments.

References

1. Deutsche Bahn: Geschäftsbericht des Konzerns. Deutsche Bahn AG, Potsdamer Platz 2, 10785 Berlin (2005)
2. Railion Deutschland: Geschäftsbericht. Railion Deutschland AG, Rheinstraße 2, 55116 Mainz (2005)
3. Löbel, A.: Optimal Vehicle Scheduling in Public Transit. Shaker Verlag, Aachen (1997)
4. Gintner, V., Kliewer, N., Suhl, L.: Solving large multi-depot multi-vehicle-type bus scheduling problems in practice. OR Spectrum **27** (2005) 507 – 523
5. Daduna, J., Völker, M.: Fahrzeugumlaufbildung im ÖPNV mit unscharfen Abfahrtszeiten. Der Nahverkehr **11** (1997) 39 – 43
6. Fügenschuh, A.: The Integrated Optimization of School Starting Times and Public Transport. Logos Verlag, Berlin (2005)
7. ILOG Ltd.: ILOG Cplex 10 Solver Suite. Technical report, ILOG Cplex Division, 889 Alder Avenue, Suite 200, Incline Village, NV 89451, USA (2006)
8. Bussieck, M., Winter, T., Zimmermann, U.: Discrete optimization in public rail transport. Mathematical Programming **79** (1997) 415 – 444
9. Caprara, A., Fischetti, M., Toth, P., Vigo, D.: Algorithms for railway crew planning. Mathematical Programming **79** (1997) 125 – 141
10. Ahuja, R., Liu, J., Orlin, J., Sharma, D., Shughart, L.: Solving real-life locomotive scheduling problems. Transportation Science **39** (2002) 503 – 517
11. Ziarati, K.: A heuristic to find cyclical planning solution for locomotive assignment problems. In: Proc. 1st Nat. Ind. Eng. Conf., Iran, Sharif University of Technology (2001)
12. Liebchen, C., Möhring, R.: The modeling power of the periodic event scheduling problem: Railway timetables – and beyond. Technical Report 2004/20, Technische Universität Berlin (2004)
13. Ahuja, R., Magnanti, T., Orlin, J.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, New Jersey (1993)
14. Nemhauser, G., Wolsey, L.: Integer and Combinatorial Optimization. Wiley Interscience, New York (1999)