Math. Meth. Oper. Res. (2005) 62: 281–295
DOI 10.1007/s00186-005-0001-0

ORIGINAL ARTICLE

Thomas Lindner · Uwe T. Zimmermann

Cost optimal periodic train scheduling

Received: January 2003 / Revised version: February 2005 / Published online: 8 October 2005

Abstract For real world railroad networks, we consider minimizing operational cost of train schedules which depend on choosing different train types of diverse speed and cost. We develop a mixed integer linear programming model for this train scheduling problem. For practical problem sizes, it seems to be impossible to directly solve the model within a reasonable amount of time. However, suitable decomposition leads to much better performance. In the first part of the decomposition, only the train type related constraints stay active. In the second part, using an optimal solution of this relaxation, we select and fix train types and try to generate a train schedule satisfying the remaining constraints. This decomposition idea provides the cornerstone for an algorithm integrating cutting planes and branch-and-bound. We present computational results for railroad networks from Germany and the Netherlands.

1 Introduction

The train schedule constitutes the backbone of public rail transport planning. A train schedule consists of the *arrival* and *departure times* of the lines at certain points of the network. Different purposes require information on different levels of aggregation of network structure. In particular, data points may include stations (high degree of aggregation) or switches and important signal points (low degree of

T. Lindner
Siemens AG, Ackerstraße 22,
D-38126 Braunschweig, Germany
E-mail: thomas.dr.lindner@siemens.com

U. T. Zimmermann (✉)
Institut für Mathematische Optimierung, Technische Universität Braunschweig,
Pockelsstraße 14, D-38106 Braunschweig, Germany
E-mail: u.zimmermann@tu-bs.de

aggregation). For the German railroad network depending on the degree of aggregation, we have to handle 8000–27000 data points. In general, schedules for public transport are periodical, i.e., the schedule is repeated after a certain *time period*.

Different departments of the railroad companies formulate various goals in connection with schedule planning. The sales and marketing departments give priority to travelers' requests, e.g., short travel time, direct connections and, if necessary, switching of trains on the same platform with short waiting time. The controlling, management and logistics departments pay much more attention to cost related aspects and ask for efficient management of rolling stock and personnel resources. Rolling stock is a scarce resource, rules implied by contracts and legal requirements set tight limitations for planning. High traffic load at critical points in the network and security requirements add a lot of operational constraints. Goals and requirements are obviously in conflict. Moreover, external factors like political decisions influence the planning process.

2 Model

A railroad network is usually modeled by a graph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of edges. Nodes represent stations or important network points like switches, and edges represent railroad tracks connecting these points. Train scheduling is based on a known line plan that defines the lines, i.e., the paths in the network that have to be served by trains within some fixed period T. This set of lines is denoted by $\mathcal{R}$. For a fixed line plan, we have to construct a train schedule which assigns feasible departure and arrival times to the nodes of all lines, cf. Figure 1.

2.1 Feasibility

Many requirements on train schedules can be modeled by so-called "periodical interval constraints". Here is a short example. At some station travelers want to

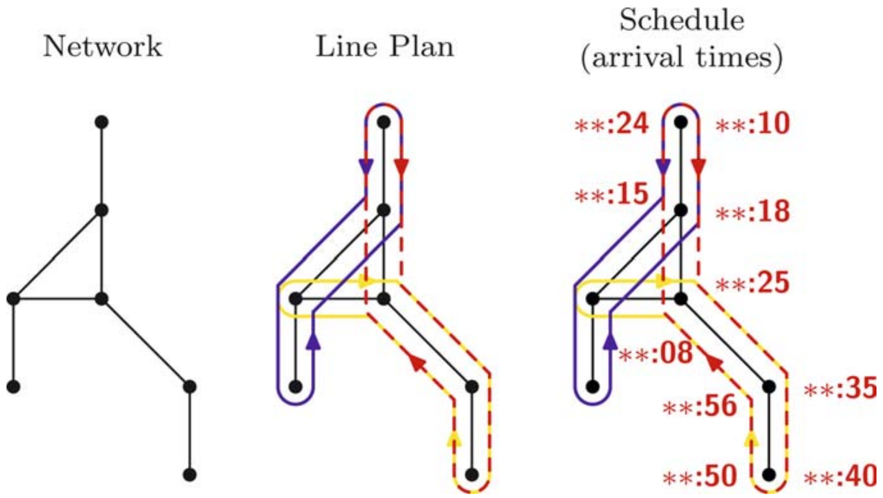

Fig. 1 Network, line plan, and schedule for one line

change from line 1 to line 2. Therefore, in a feasible train schedule the difference between the arrival of a train of line 1 and the departure of the corresponding train of line 2 has to stay within a certain interval. If the difference is too small travelers may fail to reach the train of line 2. On the other hand, if the difference is too large, the waiting time at the station will be inconvenient. Let the schedule period be $T = 60$ min. If 8–15 min are considered to be acceptable, then 68–75 min are acceptable as well since arrivals and departures repeat every hour. We model such a periodical interval constraint in the following way. If a_1^S is the arrival time (in minutes) of some train of line 1 at station S then we know that the arrival time of any train of line 1 at station S is $a_1^S + z \cdot 60$ for some $z \in \mathbf{Z}$. Similarly, the departure time d_2^S of some train of line 2 at station S fixes the departure times of all trains of line 2. Therefore, the time difference requirement for all trains can be modeled by

$$8 \leq d_2^S - a_1^S - z \cdot 60 \leq 15, \quad z \in \mathbf{Z}. \tag{1}$$

Many other requirements can be modeled in a similar way, e.g., headway times between trains on the same track and travel times of trains between stations. The problem of finding a feasible schedule, i.e., a schedule satisfying a set of periodical interval constraints is called periodical event scheduling problem (PESP) and was introduced by Serafini and Ukovich (1989). There are several different algorithmic approaches for solving PESPs, e.g., implicit enumeration methods (Schrijver and Steenbeek 1994; Serafini and Ukovich 1989) and a cutting plane method by Odijk (1996). Voorhoeve (1993) is based on constraint propagation.

2.2 Cost optimization

So far we have considered only the feasibility of a schedule. Such feasibility models present several problems for a practical application:

- Practical instances may be infeasible. From a theoretical point of view, this is not a problem but in practice some schedule *has* to be generated, even in this situation. In order to make the instance feasible some constraints have to be relaxed. However, it is neither clear which constraints should be relaxed nor how they should be relaxed. Any practical algorithm requires rules for such decisions.
- Furthermore, generating feasible schedules does not provide information on the quality of the found schedules. In practice there are many criteria for evaluating feasible schedules (Cordeau 1998).

One criterion of increasing importance is cost. A model for evaluating operational costs of *line plans* is introduced by Claessens (1994), Claessens et al. (1998). We propose an adaptation of this model for operational costs of *train schedules*. The model includes the following cost:

- *Fixed cost per schedule period, per motor unit, and per coach*: This includes depreciation cost, capital cost, fixed maintenance cost or cost for overnight parking.
- *Cost per distance, per motor unit, and per coach*: Examples are energy and maintenance cost.

The model is quite flexible and allows to cover further cost aspects.

In our model, we consider operational cost depending on the type of the train assigned to a line. We assume that the railroad company wants to find a low cost assignment. A solution of our model proposes a minimum cost assignment of *train types* to lines. A train type is characterized by:

- cost, capacity of coaches, bounds on the number of coaches; and
- speed

Let $\mathcal{T}$ denote the set of train types. Since train types run at different speed, feasibility of a schedule depends on the assigned train types. A solution of our model provides a minimum cost assignment of different train types with a feasible schedule.

We propose a mixed integer linear programming (MIP) model containing the following classes of variables:

$w_{r,\varrho,c}$	Line $r \in \mathcal{R}$ uses train type $\varrho \in \mathcal{T}_r$ with $c \in \{\underline{W}, \dots, \overline{W}\}$ coaches (binary variable) ,
$a^v_{r,\mu}$	Arrival time of one train of line r, with direction μ, at station v,
$d^v_{r,\mu}$	Departure time of one train of line r, with direction μ, at station v,
z	Vector of integers for the PESP constraints.

Here, $\mathcal{T}_r$ denotes the set of train types that can be assigned to a line $r \in \mathcal{R}$. $\underline{W}$ is the minimal number of coaches of a train, $\overline{W}$ the maximal number. The direction μ can either be 0 or 1, corresponding to the two directions of each line. The complete model, referred to as minimum cost scheduling problem (MCSP), is shown in Figure 2.

$$\min \sum_{r\in\mathcal{R}} \sum_{\varrho\in\mathcal{T}_r} \sum_{c=\underline{W}}^{\overline{W}} \lceil \hat{t}_{r,\varrho}/T \rceil \cdot (C^{\text{fix}}_\varrho + c \cdot C^{\text{fixC}}_\varrho) w_{r,\varrho,c} + d_r \cdot (C^{\text{km}}_\varrho + c \cdot C^{\text{kmC}}_\varrho) w_{r,\varrho,c}$$

$$\sum_{r\in\mathcal{R}, r\ni e} \sum_{\varrho\in\mathcal{T}_r} \sum_{c=\underline{W}}^{\overline{W}} \mathfrak{C}_\varrho \cdot c \cdot w_{r,\varrho,c} \geq N_e \qquad \text{for each } e \in E$$

$$\sum_{\varrho\in\mathcal{T}_r} \sum_{c=\underline{W}}^{\overline{W}} w_{r,\varrho,c} = 1 \qquad \text{for each } r \in \mathcal{R}$$

$$\sum_{\varrho\in\mathcal{T}_r} \sum_{c=\underline{W}}^{\overline{W}} \underline{\Delta}^{vv'}_\varrho w_{r,\varrho,c} \leq a^{v'}_{r,\mu} - d^v_{r,\mu} \leq \sum_{\varrho\in\mathcal{T}_r} \sum_{c=\underline{W}}^{\overline{W}} \overline{\Delta}^{vv'}_\varrho w_{r,\varrho,c} \qquad \text{for each } r \in \mathcal{R},\ (v,v') \in r$$

$$\text{PESP constraints}$$

$$\begin{aligned} w_{r,\varrho,c} &\in \mathbf{Z} && \text{for each } r \in \mathcal{R},\ \varrho \in \mathcal{T}_r,\ c \in \{\underline{W},\dots,\overline{W}\} \\ a^v_{r,\mu} &\in \mathbf{R} && \text{for each } r \in \mathcal{R},\ v \in r,\ \mu \\ d^v_{r,\mu} &\in \mathbf{R} && \text{for each } r \in \mathcal{R},\ v \in r,\ \mu \\ z &\ \text{integer vector} \end{aligned}$$

Fig. 2 Mixed integer linear program for minimum cost scheduling

We will now explain the parts of the model in detail. The objective function summarizes all costs for the chosen trains. Its first part contains the fixed costs. The estimated cycle time (i.e., the time required for a train to run from one end of the line to the other end and back) of train type ϱ along line r is denoted by $\hat{t}_{r,\varrho}$. The actual number of trains of type ϱ needed to operate line r is then given by $\lceil \hat{t}_{r,\varrho}/T \rceil$. C_ϱ^{fix} denotes the fixed cost for one engine of type ϱ, C_ϱ^{fixC} the fixed cost for one coach. The second part of the sum contains the mileage cost. C_ϱ^{km} denotes the cost per km (1 km = 1000 m) for one engine of type ϱ, C_ϱ^{kmC} the cost per km for one coach. Here, d_r denotes the length of line r.

With the first constraint class, we assure that the capacity of trains is sufficiently large to carry all travelers. The capacity of one train of type ϱ is the capacity of one coach, denoted by $\mathfrak{C}_\varrho$, multiplied by the number of coaches $c \cdot w_{r,\varrho,c}$ of the train. Summation over all trains describes the available capacity, which must be at least the required capacity N_e for a network edge e.

With the second class of constraints, we assure that exactly one train type and one number of coaches is chosen for each line. These two classes of constraints involve only train types and numbers of the coaches of these trains, but no time variables.

The third constraint class ensures a feasible travel time for the trains of line r. Here, the minimum travel time for a train of type ϱ from a station v to the next station v' is denoted by $\underline{\Delta}_\varrho^{vv'}$, the maximum time by $\overline{\Delta}_\varrho^{vv'}$.

All other constraints are periodical interval constraints describing a PESP. For each such constraint an integer variable models the periodicity of the schedule as in inequality (1). All these and all further integer variables in the remaining PESP constraints (e.g., waiting times at stations, headway etc.) are collected in a vector z.

The MCSP contains subproblems that are very difficult to solve. The optimal choice of train types and the optimal choice of numbers of coaches are both NP-hard problems, and the generation of a feasible schedule (the PESP) is NP-complete (Garey and Johnson 1979; Lindner 2000; Serafini and Ukovich 1989). On the other hand, the MIP model precisely states which train schedule we consider as feasible and which cost we want to minimize. Moreover, the MIP formulation enables easy addition of other requirements, e.g., the available number of engines and coaches of a certain type, without destroying the structure of the model. The high flexibility of MIP models is very helpful when modeling and solving real world applications.

3 Solution algorithm

The MCSP model is intended for strategic and tactical railroad planning, i.e., for long or medium term decisions rather than for day-by-day operations. In order to provide a helpful evaluation tool for different network or train type scenarios, only short computation times, say some minutes, are acceptable when solving the model for real world MCSP instances. Furthermore, the planner can only use standard computer equipment. For our test data like the InterCity or the InterRegio network of Germany and the Netherlands, the direct application of some commercial MIP solver resulted in several hours or even days of computation time on a 400 MHz Pentium II PC. For some instances, its main memory of 256 MB was not sufficient. In the following, we present a more sophisticated approach for solving MCSP.

3.1 Decomposition

Our approach is based on decomposition of the MCSP as proposed by the structure of the matrix of all objective function coefficients and all constraint coefficients in Figure 3. Only shaded blocks contain nonzero entries for the respective variables and the corresponding class of constraints.

With the exception of the w-variables in the travel time data constraints, the matrix has two-block diagonal structure. The first block represents the problem of minimizing costs subject to requirements on the trains that can be assigned to lines. This subproblem has only binary variables and is called minimum cost train problem (MCTP). Its optimal solution assigns train types to the lines and defines the number of coaches in each assigned train or, shortly, it assigns trains to lines. However, it is not clear whether a feasible train schedule for these trains exists at all. In fact, for some fixed train type assignment, the remaining constraints form the second subproblem, called the feasible schedule problem (FSP). FSP is a PESP.

For solving MCSP we propose a branch-and-bound method based on these two subproblems. In particular, the MCTP is used as relaxation of the MCSP.

3.2 Branch-and-bound method

In our branch-and-bound tree, each node represents an MCSP for which only a certain subset of train type assignments is allowed. We will now describe the resulting method in detail.

Choice of the relaxation: As a relaxation in each branch-and-bound node, we use the MCTP instance corresponding to the allowed train type assignments. At the root node, for each line r, all train types $\varrho \in \mathcal{T}_r$ are allowed. By using some ideas described below, we are able to solve the MCTP instance for practical networks with a commercial MIP solver in a few seconds.

Feasibility check: Let the optimal solution of the relaxation be given by the vector $\boldsymbol{w}$ of the train types and numbers of coaches. We now need to find out whether the FSP constraints can be satisfied for $\boldsymbol{w}$. Since this problem is a PESP, we will address solution methods for PESP instances later in this section.

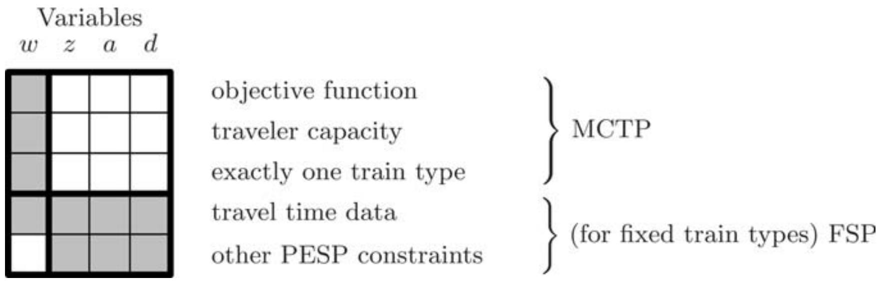

Fig. 3 Structure of objective function and constraints of the MCSP

Choice of division/partition: If the FSP instance is infeasible, we need to change at least one of the train types. Let ϱ_r be the train type for line $r \in \mathcal{R}$, and let $\rho := |\mathcal{R}|$. An obvious way to divide the set $\mathcal{T}_1 \times \cdots \times \mathcal{T}_\rho$ of possible (but not necessarily feasible) combinations for train types is given by

$$\mathcal{T}_1 \times \cdots \times \mathcal{T}_\rho \backslash (\varrho_1, \ldots, \varrho_\rho) = (\mathcal{T}_1 \backslash \{\varrho_1\}) \times \mathcal{T}_2 \times \cdots \times \mathcal{T}_\rho \cup \cdots \cup \mathcal{T}_1 \times \cdots \times \mathcal{T}_{\rho-1} \times (\mathcal{T}_\rho \backslash \{\varrho_\rho\}). \tag{2}$$

According to this scheme, $|\mathcal{R}|$ new problems have to be generated in this case. In order to keep the size of the branch-and-bound tree small we would prefer to replace $\mathcal{R}$ by a "smaller" set of lines $\hat{\mathcal{R}} \subset \mathcal{R}$ which is still causing the conflict. The identification of a minimal set is NP-hard again, but some heuristical approach turned out to be helpful for our practical instances. We will shortly address it later.

An algorithmic description of the branch-and-bound method is given as Algorithm 1. We remark that this decomposition requires solving an NP-hard relaxation as well as an NP-complete feasibility check at every node. Moreover, even for practical MCSP instances, no polynomial bound for the number of nodes is known.

Algorithm 1 Branch-and-bound algorithm for the MCSP

```
$c_* := \infty$; $\mathcal{L} := \{\{\mathcal{T}_1 \times \cdots \times \mathcal{T}_\rho\}\}$; $l_{\mathcal{T}_1 \times \cdots \times \mathcal{T}_\rho} := -\infty$
loop
  if $\mathcal{L} = \emptyset$ then
    Stop. If $c_* = \infty$, the problem is infeasible. Otherwise, an optimal solution is given by
    $\boldsymbol{w}_*, \boldsymbol{a}_*, \boldsymbol{d}_*, \boldsymbol{z}_*$.
  end if
  Choose $\mathcal{T}' \in \mathcal{L}$.
  $\mathcal{L} := \mathcal{L} \backslash \{\mathcal{T}'\}$
  if the MCTP for $\mathcal{T}'$ is infeasible then
    continue
  end if
  Let an optimal solution of the MCTP be defined by the vector $\boldsymbol{w}$ with optimal value $c$.
  if $c \geq c_*$ then
    continue
  end if
  if the FSP for $\boldsymbol{w}$ is feasible with solution vectors $\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{z}$ then
    $c_* := c$; $\boldsymbol{w}_* := \boldsymbol{w}$; $\boldsymbol{a}_* := \boldsymbol{a}$; $\boldsymbol{d}_* := \boldsymbol{d}$; $\boldsymbol{z}_* := \boldsymbol{z}$
    $\mathcal{L} := \mathcal{L} \backslash \{\hat{\mathcal{T}} \mid l_{\hat{\mathcal{T}}} \geq c_*\}$
    continue
  end if
  Let $\hat{\mathcal{R}}$ be a set of lines leading to the infeasibility of the FSP for $\boldsymbol{w}$ and let $\varrho_i$ be the train
  type for line $r_i \in \mathcal{R}$ in the MCTP solution defined by $\boldsymbol{w}$.
  for $i = 1$ to $\rho$ do
    if $r_i \in \hat{\mathcal{R}}$ then
      Let $\mathcal{T}^*$ denote $\mathcal{T}_1 \times \cdots \times \mathcal{T}_{i-1} \times \mathcal{T}_i \backslash \{\varrho_i\} \times \mathcal{T}_{i+1} \times \cdots \times \mathcal{T}_\rho$.
      $l_{\mathcal{T}^*} := c$
      $\mathcal{L} := \mathcal{L} \cup \{\mathcal{T}^*\}$
    end if
  end for
end loop
```

4 Solving MCTP instances

Unfortunately, when directly applied to MCTP of real world size, commercial solvers may require too much computation time. For its repeated use within the branch-and-bound scheme we propose adding suitable classes of cutting planes. We will now present such a class.

Proposition 1 *For $n \geq 2$, let $e \in E$ be an edge served by the lines $r_1, \dots, r_n \in \mathcal{R}$. We consider $\delta_1, \dots, \delta_{n-1} \in \{1, \dots, N_e - 1\}$ with $\delta := \sum_{i=1}^{n-1} \delta_i < N_e$. Then*

$$\left(\sum_{i=1}^{n-1} \sum_{\varrho \in \mathcal{T}_{r_i}} \sum_{c \in \{\underline{W}_\varrho, \dots, \overline{W}_\varrho\},\ c \cdot \mathfrak{C}_\varrho \geq \delta_i} w_{r_i,\varrho,c} \right) + \sum_{\varrho \in \mathcal{T}_{r_n}} \sum_{c \in \{\underline{W}_\varrho, \dots, \overline{W}_\varrho\},\ c \cdot \mathfrak{C}_\varrho > N_e - \delta} w_{r_n,\varrho,c} \geq 1 \tag{3}$$

is a valid inequality for the MCTP.

Proof Let w denote a feasible binary solution of MCTP. If the binary variables in (3) sum up to less than 1, each of them has value 0. Therefore any line r_i, $i < n$, can only contribute a capacity less than δ_i, and the line r_n can contribute at most the capacity $N_e - \delta$. This results in

$$\begin{aligned} \sum_{\substack{r \in \mathcal{R} \\ r \ni e}} \sum_{\varrho \in \mathcal{T}_r} \sum_{c=\underline{W}_\varrho}^{\overline{W}_\varrho} c \cdot \mathfrak{C}_\varrho \cdot w_{r,\varrho,c} &= \sum_{i=1}^{n} \sum_{\varrho \in \mathcal{T}_{r_i}} \sum_{c=\underline{W}_\varrho}^{\overline{W}_\varrho} c \cdot \mathfrak{C}_\varrho \cdot w_{r_i,\varrho,c} \\ &< \sum_{i=1}^{n-1} \delta_i + N_e - \delta = N_e, \end{aligned} \tag{4}$$

which is a contradiction to the traveler capacity inequality of the MCTP. □

For our practical instances many of these cuts are violated by the respective LP relaxation, although the edges are mostly served only by a few lines. Even for $n = 2$, there are too many such violated inequalities to add all of them. However, for $n = 2$, we can restrict our attention to a smaller set of cuts. Let $\mathfrak{C}_i^1 < \cdots < \mathfrak{C}_i^{k_i}$ denote all potential train capacities which can be assigned to line r_i, $i = 1, 2$, i.e., all values $c \cdot \mathfrak{C}_\varrho$ for some train c, ϱ. The following proposition shows, that it is sufficient to consider cuts for k_1 values of δ_1.

Proposition 2 *Let $e \in E$ be an edge only served by the two lines r_1 and r_2 and let $\boldsymbol{w}$ denote a feasible solution for the LP relaxation of the MCTP. If $\boldsymbol{w}$ violates a cut from* (3) *for some $\delta_1 \in \{1, \dots, N_e - 1\}$ then it also violates a cut from* (3) *for some $\hat{\delta}_1 \in \{\mathfrak{C}_1^1 + 1, \dots, \mathfrak{C}_1^{k_1} + 1\}$.*

Proof Obviously, $\delta_1 > \mathfrak{C}_1^1$ (otherwise no $\boldsymbol{w}$ can violate a cut from (3)). Therefore either $\delta_1 \geq \mathfrak{C}_1^{k_1} + 1$ or $\mathfrak{C}_1^i + 1 \leq \delta_1 \leq \mathfrak{C}_1^{i+1}$ for some index $i \in \{1, \dots, k_1 - 1\}$.

In the first case, let $\hat{\delta}_1 := \mathfrak{C}_1^{k_1} + 1$. Then $\boldsymbol{w}$ violates the corresponding cut as

$$\sum_{\varrho\in\mathcal{T}_{r_1}} \sum_{c\in\{\underline{W}_\varrho,\dots,\overline{W}_\varrho\},\ c\cdot\mathfrak{C}_\varrho\geq\hat{\delta}_1} w_{r_1,\varrho,c} + \sum_{\varrho\in\mathcal{T}_{r_2}} \sum_{c\in\{\underline{W}_\varrho,\dots,\overline{W}_\varrho\},\ c\cdot\mathfrak{C}_\varrho>N_e-\hat{\delta}_1} w_{r_2,\varrho,c}$$
$$= \sum_{\varrho\in\mathcal{T}_{r_2}} \sum_{c\in\{\underline{W}_\varrho,\dots,\overline{W}_\varrho\},\ c\cdot\mathfrak{C}_\varrho>N_e-\hat{\delta}_1} w_{r_2,\varrho,c} \leq \sum_{\varrho\in\mathcal{T}_{r_2}} \sum_{c\in\{\underline{W}_\varrho,\dots,\overline{W}_\varrho\},\ c\cdot\mathfrak{C}_\varrho>N_e-\delta_1} w_{r_2,\varrho,c} < 1.$$

Otherwise, choose $\hat{\delta}_1 := \mathfrak{C}_1^i + 1$. Then $\boldsymbol{w}$ violates the corresponding cut as

$$\sum_{\varrho\in\mathcal{T}_{r_1}} \sum_{c\in\{\underline{W}_\varrho,\dots,\overline{W}_\varrho\},\ c\cdot\mathfrak{C}_\varrho\geq\hat{\delta}_1} w_{r_1,\varrho,c} + \sum_{\varrho\in\mathcal{T}_{r_2}} \sum_{c\in\{\underline{W}_\varrho,\dots,\overline{W}_\varrho\},\ c\cdot\mathfrak{C}_\varrho>N_e-\hat{\delta}_1} w_{r_2,\varrho,c}$$
$$= \sum_{\varrho\in\mathcal{T}_{r_1}} \sum_{c\in\{\underline{W}_\varrho,\dots,\overline{W}_\varrho\},\ c\cdot\mathfrak{C}_\varrho\geq\delta_1} w_{r_1,\varrho,c} + \sum_{\varrho\in\mathcal{T}_{r_2}} \sum_{c\in\{\underline{W}_\varrho,\dots,\overline{W}_\varrho\},\ c\cdot\mathfrak{C}_\varrho>N_e-\hat{\delta}_1} w_{r_2,\varrho,c}$$
$$\leq \sum_{\varrho\in\mathcal{T}_{r_1}} \sum_{c\in\{\underline{W}_\varrho,\dots,\overline{W}_\varrho\},\ c\cdot\mathfrak{C}_\varrho\geq\delta_1} w_{r_1,\varrho,c} + \sum_{\varrho\in\mathcal{T}_{r_2}} \sum_{c\in\{\underline{W}_\varrho,\dots,\overline{W}_\varrho\},\ c\cdot\mathfrak{C}_\varrho>N_e-\delta_1} w_{r_2,\varrho,c}$$
$$< 1.$$

□

For larger values of n, Proposition 2 lists a reasonable subset of the cuts. In the following, we discuss the quality of the cuts from (3).

In particular, we consider a small graph $\hat{G}$ consisting only of two nodes and one connecting edge e served by only two lines r_1 and r_2. We list all potential train capacities for line r_i according to variables $w_{r_i,\varrho,c}$ in increasing order as $\mathfrak{C}_i^1 \leq \cdots \leq \mathfrak{C}_i^{k_1}$. Here, contrary to the definition used in Proposition 2, some successive values will be identical when the same capacity can be obtained from different combinations of train types and numbers of coaches.

Proposition 3 *For the small graph $\hat{G}$ the polyhedron P described by the constraints*

$$\sum_{\varrho\in\mathcal{T}_{r_1}} \sum_{c=\underline{W}_\varrho}^{\overline{W}_\varrho} w_{r_1,\varrho,c} = 1, \qquad \sum_{\varrho\in\mathcal{T}_{r_2}} \sum_{c=\underline{W}_\varrho}^{\overline{W}_\varrho} w_{r_2,\varrho,c} = 1, \qquad w \geq 0$$

and the cuts from (3) *for all values of $\hat{\delta}_1$ as described in Proposition* 2 *is integral.*

Proof We show that the constraint matrix is an interval matrix and thus is totally unimodular (Nemhauser and Wolsey 1988). Since the right hand sides of the constraints are integral, the proposition follows.

We reorder the columns of the constraint matrix by ordering the corresponding variables for line r_1 by increasing capacities $\mathfrak{C}_1^i$ and the corresponding variables for line r_2 by decreasing capacities. The reordered constraint matrix is an interval matrix (bound constraints are omitted):

$$\begin{array}{c c}
\begin{array}{ccccccccccc}
\mathfrak{C}_1^1 & \mathfrak{C}_1^2 & & \dots & & \mathfrak{C}_1^{k_1} & \mathfrak{C}_2^{k_2} & & \dots & \mathfrak{C}_2^2 & \mathfrak{C}_2^1
\end{array} & \\
\left(\begin{array}{ccccccccccc}
1 & 1 & 1 & \dots & 1 & 1 & 0 & 0 & \dots 0 & 0 & 0 \\
0 & 0 & 0 & \dots & 0 & 0 & 1 & 1 & \dots 1 & 1 & 1 \\
0 & 1 & \dots & \dots & \dots & \dots & \dots & \dots & 1 & 0 \dots & 0 \\
0 & 0 & 1 & \dots & \dots & \dots & \dots & \dots & \dots 1 & 0 & 0 \\
 & & & & & \text{etc.} & & & & &
\end{array}\right) &
\begin{array}{l}
\sum_\varrho \sum_c w_{r_1,\varrho,c} = 1 \\
\sum_\varrho \sum_c w_{r_2,\varrho,c} = 1 \\
\text{from Proposition 2} \\
\text{from Proposition 2} \\
\text{from Proposition 2}
\end{array}
\end{array}$$

□

We can compare P with the polyhedron P' defined by the LP relaxation of the corresponding MCTP instance. Due to the full set of cut constraints, every integer point of P satisfies the traveler capacity constraints of the MCTP. Thus it is a feasible solution of the MCTP and its relaxation P'. Vice versa, all integer points from P', i.e., all feasible solutions of the MCTP belong to P since the cuts are valid for MCTP (cf. Proposition 1). Therefore, for the small graph $\hat{G}$, the integral polyhedron P is the convex hull of the feasible solutions of the MCTP. In other words, the cuts as described by Proposition 2 are sufficient to obtain a linear description of the convex hull of the MCTP.

For general instances, we may only conclude that if there is an edge served by exactly two lines, there are no "better" cuts for the MCTP which can be derived from the required edge capacity and the potential train capacities assignable to the two lines serving that edge.

In general, we can only refer to some positive experience for practical problems. In our implementation of the solution algorithm for MCTP, violation of all cuts from Proposition 1 for $n = 2$ is checked (which can be no more than $|E| \cdot |\mathcal{R}| \cdot (|\mathcal{R}| - 1)$ inequalities).

5 Solving FSP instances

Solving an FSP instance is equivalent to finding an integral solution to a set of linear inequalities. Again, the straightforward application of some commercial MIP solver leads to unacceptably large computation times. On the other hand, we already observed that FSPs are PESPs. We already mentioned that there are several algorithms for solving PESP instances known from the literature. However, these algorithms also seem to be too slow for instances of practical sizes.

One such method is the algorithm proposed by Serafini and Ukovich (1989). We have developed a modified version of this algorithm which enables us to solve much larger PESP instances and in particular those instances arising from our practical problems. We now describe a generalized version of the algorithm given in Serafini and Ukovich (1989). We then point out the differences between the original algorithm and our version.

The algorithm of Serafini and Ukovich works on the so-called *PESP event graph*. This directed graph $G = (V, A)$ is defined as follows: For each periodical event such as the arrival or departure of a line at a station there is a node $v \in V$, and for each periodical interval constraint, there is an arc $a \in A$. The goal of the algorithm is now to assign a *potential* $\boldsymbol{\pi} \in \mathbf{R}^{|V|}$ to the nodes such that for each arc the corresponding interval constraint is satisfied. Let $\Re$ be the node arc incidence matrix of the graph, and let $\boldsymbol{l}$ and $\boldsymbol{u}$ be the vectors of lower and upper (periodical) interval bounds. Then a solution of the PESP instance is represented by a pair $(\boldsymbol{\pi}, \boldsymbol{z})$ such that

$$\boldsymbol{l} \leq \Re^T \boldsymbol{\pi} - T\boldsymbol{z} \leq \boldsymbol{u}, \qquad (\pi, 2) \in \mathbf{R}^{|V|} \times \mathbf{Z}^{|A|}. \tag{5}$$

For an arc $a \in A$, the value z_a is called the *modulo parameter* of a.

Let $\mathcal{T}$ be an arbitrary spanning tree of G. In Serafini and Ukovich (1989) it is shown that if there is a solution to (5), then there is also a solution to (5) where all components of $\boldsymbol{z}$ corresponding to $\mathcal{T}$ are equal to 0. Note that by fixing $\boldsymbol{z}\,|_{\mathcal{T}} = \boldsymbol{0}$, the number of possible values of z_a for a non-tree arc a is bounded. Therefore, it is possible to enumerate all values of $\boldsymbol{z}$ with $\boldsymbol{z}_{|\mathcal{T}} = \boldsymbol{0}$. The algorithm of Serafini and Ukovich is, in some sense, such an enumeration procedure. A generalized version of this method is shown as Algorithm 2.

Algorithm 2 Generalized Serafini-Ukovich algorithm

```
Choose a spanning tree $\mathcal{T}$. Set $S := A(\mathcal{T})$.
$z_a := 0$ for all $a \in S$
Determine a feasible potential $\boldsymbol{\pi}$ for the graph $(V, S)$ with the fixed modulo parameters.
**loop**
  Choose non-tree arc $a \in A \backslash S$.
  $Z_a := \{z_a \mid z_a$ is a feasible modulo parameter for $a$ if only arcs $a' \in S$ are considered, but
  with their fixed modulo parameters$\}$
  **if** $Z_a \neq \emptyset$ **then**
    Choose a $z \in Z_a$.
    $z_a := z$
    $S := S \cup \{a\}$
    Determine a feasible potential $\boldsymbol{\pi}$ for the graph $(V, S)$ with the fixed modulo parameters.
    **if** $S = A$ **then**
      Stop. $\boldsymbol{\pi}$ is a feasible potential for the PESP instance, and $\boldsymbol{z}$ is a corresponding vector
      of modulo parameters.
    **end if**
  **else**
    Perform backtracking: Choose another value from $z_{a'}$ for the arc $a'$ whose modulo param-
    eter was fixed in the iteration before. If this is not possible, perform further backtracking.
    Delete the corresponding arcs from $S$. If there are no more modulo parameters for the
    arc whose modulo parameter was fixed in the first iteration, stop. The PESP instance is
    infeasible.
  **end if**
**end loop**
```

Some steps of the algorithm should be discussed more carefully. A feasible initial potential for (V, S), S being the arc set forming the spanning tree, always exists and can be found trivially. Determining a feasible potential for (V, S) after adding some arcs to S is done by solving a shortest path problem with a modified Dijkstra procedure (for details, see Serafini and Ukovich 1989).

A crucial part of the algorithm is the selection of an arc $a \in A \backslash S$ in order to fix the modulo parameter z_a. This choice has much influence on the structure of the enumeration search tree and thus on the solution time of the algorithm. In order to get small values for $|z_a|$ and thus only few subproblems in the enumeration tree one should try to identify the ‘most restrictive’ constraints and choose these arcs as soon as possible. Serafini and Ukovich (1989) suggested sorting the arcs by the interval width $u_a - l_a$ at the initialization and then always choosing an arc a with the smallest value $u_a - l_a$.

For our problem instances we have changed this selection rule: we recall that after choosing a spanning tree $\mathcal{T}$, the inequalities of (5) imply bounds on the number of possible modulo parameters for each non-tree arc and thus on the number of subproblems in the enumeration tree. In fact, added to the tree each non-tree arc induces a unique cycle. Along the cycle, one may exploit the inequalities of (5) together with the fact that $z\,|_{\mathcal{T}} = \boldsymbol{0}$ in order to compute such bounds on z_a. Let b_a be the bound on the number of modulo parameters for arc a. We propose to choose an arc a_* with

$$b_{a_*} = \min_{a \in A \setminus S} b_a$$

for modulo parameter fixing. This reduces the enumeration tree remarkably and thus leads to the required speedup. Another possibility is to calculate more exact bounds b_a during the algorithm and re-sort the arcs according to the new bounds. A detailed discussion of arc selection rules is presented in Lindner (2000).

In view of the use of the FSP in the branch-and-bound method, it is important to analyze infeasible FSP instances. Infeasibility leads to a branching step with the generation of new subproblems. In order to keep the size of the branch-and-bound tree as small as possible, it seems to be preferable to generate only few subproblems. Therefore, when identifying an infeasible FSP instance, we try to generate a small set of lines that already causes the infeasibility. The main idea of the heuristic approach is to use the following observation: when some initial tree leads to an infeasible non-tree arc then only changes for lines between the root of the tree and the end points of the infeasible arc may resolve the conflict. For a more detailed information on the approach we have to refer to Lindner (2000).

6 Computational results

Test data for our algorithms are real networks from the German railroad company Deutsche Bahn (DB) and from the railroad company of the Netherlands Nederlandse Spoorwegen (NS). For the German railroad, we used network data, line plans, origin destination matrices, and cost data for the InterCity (IC) and InterRegio (IR) networks. For the Dutch railroad, we obtained the respective data for the InterRegio (IR), InterCity (IC), and AggloRegio (AR) supply networks.

Some characteristics of these instances are displayed in Table 1. For all instances, four different train types are considered.

As an example, the InterCity network of the Netherlands is shown in Figure 4. Cost optimal lines for this and other Dutch networks were generated in Bussieck (1998).

Table 2 shows the overall results for our test networks. We use a 400 MHz Pentium II PC with 256 MB main memory for our experiments. The MIP instances are solved with CPLEX, version 6.5.2. With the exception of the Dutch AggloRegio network, which is a very large network, our branch-and-bound approach is very fast. Even when we enforce arrivals and departures suitable for travelers changing trains (by adding periodical interval constraints for connections at important

Table 1 Some characteristics of the railroad networks

	DB-IC	DB-IR	NS-IC	NS-IR	NS-AR
Number of nodes	90	297	36	38	122
Number of edges	107	384	48	40	134
Number of lines	31	89	25	21	117
Average number of edges per line	7.5	5.9	5.0	5.8	4.2

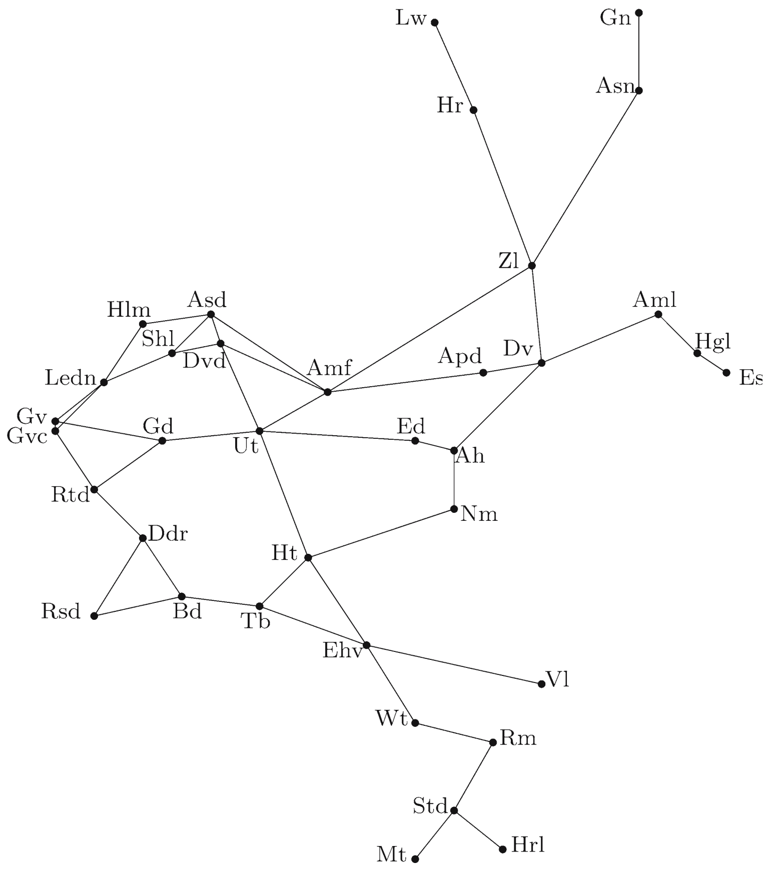

Fig. 4 Intercity network of the Netherlands

stations – we stopped at 40 constraints, because then the running time increased much, but the gain in traveller connection was marginal), we generate nearly optimal train schedules within 5 min.

The number of branch-and-bound nodes varied from 200 (NS-IR) to 30000 (DB-IC). The main time was spent on solving MCTP instances (90%). Almost all FSP instances could be solved or proved to be infeasible in 10 s. The first solution for NS-AR was generated after half an hour.

We will now discuss the effect of the cutting planes from Section 4 for solving our MCTP instances. As example instances, we have chosen the MCTP instances from the root node of our branch-and-bound algorithm applied to the test networks.

Table 2 Results for test instances without and with added constraints

Instance	DB-IC	DB-IR	NS-IC	NS-IR	NS-AR
Number of added constraints	0	0	0	0	0
Verified optimum found in	219 s	4 s	30 s	33 s	0:47 h
Optimality gap after 5 min	0%	0%	0%	0%	–*
Instance	DB-IC	DB-IR	NS-IC	NS-IR	NS-AR
Number of added constraints	40	40	40	40	40
Verified optimum found in	9:31 h	122 s	14:00 h	1:20 h	1:24 h
Optimality gap after 5 min	0.10%	0%	0.39%	0.27%	–*

*No solution in 5 min

Table 3 Results for MCTP

Inst.	#Con.	#Var.	#≠ 0	Root gap (%)	Time (S)
DB-IC	74	735	3435	0.9	1
DB-IR	69	471	1376	0.3	1
NS-IC	57	793	3237	2.5	20
NS-IR	41	545	2741	1.5	2
NS-AR	178	2221	9328	2.1	865

Table 4 Results for the MCTP using cutting planes

I	# Iter.	# Cuts	Root gap (%)	Time (S)	I	# Iter.	# Cuts	Root gap (%)	Time (S)
DB-IC	3	7	0.6	1	DB-IR	2	18	0.05	1
NS-IC	10	134	0.6	9	NS-IR	10	94	0.9%	4
NS-AR	9	123	1.1	934					

In Table 3, the number of rows, columns and non-zeros of the MIPs, the relative gap between the optimal LP solution and the optimal MIP solution and the solution time for MCTP are given.

The solver CPLEX uses a MIP preprocessor for reducing the MIP size. The numbers from Table 3 were obtained after the preprocessing step.

The effect of using the cutting planes we have discussed can be seen in Table 4. There, the number of cutting plane iterations before starting the MIP branch-and-bound process, the number of used cuts, the relative gap between the LP solution and the MIP solution and the solution time (cutting plane algorithm + MIP solution) are given.

We can see that for our instances, the use of cutting planes provides better LP bounds and an additional acceleration in some cases (while in the other cases the solution time does not increase much).

7 Conclusions

In this paper, we derived a mixed integer linear programming formulation of a cost optimization train scheduling problem. A direct solution of problem instances of practical size with a commercial MIP solver turned out to be impossible because of the tremendous sizes of the instances. Instead, we proposed a decomposition-based branch-and-bound approach which produces solutions of acceptable quality within a few minutes. Therefore, with our model, traffic planners may interactively generate or evaluate different scenarios.

In view of the further development of commercial solvers we verified our conclusions by experiments with a more recent CPLEX version (8.0). Although with this version our MCTP instances could be solved faster than with our cutting plane algorithm, the general improvement in solving MCTP is still not sufficient to directly attack the full MIP instances of MCSP.

For larger networks, we propose a strategy of regional decomposition in order to get "good" solutions. At first, the network is partitioned into several regional networks which allow fast optimization. Then, the corresponding partial train schedules are combined to a train schedule of the complete network. The combination step will require some interactive corrections which may be supported by computerized tools. In manual schedule planning, regional decomposition is a well established approach. With the help of the LP relaxation of the MCTP, which can be calculated in a few seconds even for large networks, we can determine an upper bound for the quality of the solutions constructed by the regional decomposition.

Acknowledgements We would like to express our thanks for encouragement, support and data made available by Adtranz Signal (Braunschweig), Dr. M. Krista (Lineas, Braunschweig) and Prof. Dr. L. Kroon (Railned, the Netherlands). We are also grateful to Dr. M. Lübbecke (TU Berlin) for comparing our algorithms to different CPLEX versions.

References

Bussieck MR, Winter T, Zimmermann UT (1997) Discrete optimization in public rail transport. Math Program 79(1–3):415–444

Bussieck MR, Krista M, Wiegand K-D, Zimmermann UT (1997) Linienoptimierung – Modellierung und praktischer Einsatz. In: Hoffmann K-H, Jäger W, Lohmann T, Schnuck H (eds) Mathematik – Schlüsseltechnologie für die Zukunft, Springer, Berlin Heidelberg New York

Bussieck MR (1998) Optimal lines in public rail transport. PhD thesis, Technische Universität Braunschweig

Claessens MT (1994) De kost-lijnvoering. Master's thesis, University of Amsterdam

Claessens MT, van Dijk NM, Zwaneveld PJ (1998) Cost optimal allocation of rail passenger lines. Eur J Oper Res 110(3):474–489

Cordeau J-F, Toth P, Vigo D (1998) A survey of optimization models for train routing and scheduling. Transport Sci 32(4):380–404

Garey MR, Johnson DS (1979) Computers and intractability — a guide to the theory of NP-completeness. Freeman, San Fransisco, USA

Lindner T (2000) Train schedule optimization in public rail transport. PhD thesis, Technische Universität Braunschweig

Nachtigall K (1998) Periodic network optimization and fixed interval timetables. Habilitationsschrift. Universität Hildesheim

Nemhauser GL, Wolsey LA (1988) Integer and combinatorial optimization. Wiley, New York

Odijk MA (1996) A constraint generation algorithm for the construction of periodic railway timetables. Transport Res B 30(6):455–464

Schrijver A, Steenbeek A (1994) Dienstregelingontwikkeling voor Railned. Technical Report, Centrum voor Wiskunde en Informatica

Serafini P, Ukovich W (1989) A mathematical model for periodic scheduling problems. SIAM J Disc Math 2(4):550–581

Voorhoeve M (1993) Rail scheduling with discrete sets. Unpublished report, Eindhoven University of Technology, The Netherlands