Discrete optimization in public rail transport\textsuperscript{*}

M. R. Bussieck, T. Winter, U. T. Zimmermann\textsuperscript{†}

September 17, 1997

1 Introduction

Fewer than twenty years after the invention of the steam engine in 1785, the first steam driven locomotives were constructed. During the first decades of the nineteenth century, rail traffic increased drastically, especially in coal mining, and for freight transportation. However, the railroad companies quickly started passenger transportation. In the 1830’s, most European railroads were already being built in order to transport passengers.

For the efficient use of the new technology in this early age of railroad systems, it was soon necessary to develop detailed plans for the schedules of the public transportation services. The planning process started with the definition of the routes and lines of the railroad network. To attract customers, regularly serviced routes between stations had to be guaranteed. Assignment and dispatch of locomotives, railroad carriages, and personnel became more and more involved. These problems of the early ages of rail traffic continue to be relevant for modern railroad systems.

Starting with local railroad networks which were connected over the following years, long distance routes were introduced whose names are now famous, for example the “Orient Express” from Paris to Istanbul, the “Golden Flyer” from London to Paris, and the “Train Blue” from Paris southbound to the Côte d’Azur. Each of the stations of these long distance trains were themselves starting points for regional railroad connections to minor centers in Europe and for local railroad services. These railroad networks, each independently governed, constituted a huge railroad network all over Europe, although there was neither systematic and global planning, nor optimization and network construction at the European level. In North America, the expansion to the west and the construction of transcontinental railroads were closely connected.

These days, the number of passengers transported on modern railroads is increasing [CvZ95] while the use of trains for freight transport is decreasing\textsuperscript{1}. New train systems like the TGV in France, the ICE in Germany, and the express system of the Japanese National Railways which started with the “Shinkansen” railroad lead to a new age of passenger transport. Because of these improved services, passengers can travel fast and comfortably to the desired destinations. Due to an increasing mobility and due to congested roads, public railroad networks are still expanding, like the regional net of New York City [Mid97]. In Europe, there are plans to improve the international railroad infrastructure [vW96].

With the integration of computers in the planning process, discrete optimization methods became available for the optimization of international, national, and local public rail transport. A growing number of railroad problems became solvable using modern optimization techniques and the increasing power of computers. Research programs sponsored by public authorities like the RAIL21-Project in the Netherlands or traffic oriented interdisciplinary projects in Germany indicate this process.

\textsuperscript{*}a Web page with examples quoted in this survey can be found at http://www.math.tu-bs.de/mo/ismp.html
\textsuperscript{†}TU Braunschweig, Abt. für Mathematische Optimierung, Pockelsstrasse 14, D-38106 Braunschweig, Germany
\textsuperscript{1}A significant example for this process is the decision of the German mail company Deutsche Post AG to stop rail transport and to start transport by truck.
Planning Rail Traffic

Improvements in the process of planning traffic, e.g. the development and evaluation of operation plans based on analytic models, are stimulated by the increasing success of railroad passenger transport. However, models from railroad freight transport are of limited use, since they are based on quite different assumptions, e.g. freight trains are dispatched on demand rather than according to a train schedule.

In densely populated countries like the Netherlands, Switzerland, or Belgium and for long distance railroad services in other countries, e.g. in Germany, a regular interval oriented, periodic or cyclic train schedule forms the backbone [Hoo96] of the railroad transport system. Such train schedules based on a system of lines with fixed frequencies were introduced at the beginning of the twentieth century in urban public transport. The train schedule serves as the most important interface to the passengers using the railroad transport system. It defines the cornerstone for all subsequent planning tasks. Because of the complexity of the railroad transport system, the process of planning traffic is hierarchically organized, and several stages have to be passed before the train schedule can be created.

Network Planning

The design of a railroad network from scratch is impossible in a historically grown infrastructure but due to high speed trains, new stations, capacity extension and reduction the railroad network is changing. In particular, possible decisions have to be evaluated, recommended or rejected. In fact, such decisions are mainly based on political reasoning.

Public transport companies offer several railroad subsystems to meet the requirement of their customers. Typically, fast long-distance trains (Intercity), trains connecting district towns (Inter-regio), and regional commuter trains operate in the railroad network. The set of stations where a train of such a subsystem should stop has to be determined. This decision is based on technical constraints as well as on an economical analysis.

Models for railroad passenger transport have to include passenger demand data. A convenient and useful representation of these data is the origin-destination matrix (OD-matrix). How to find a reasonable estimation of the OD-matrix from counting passengers on the edges of a railroad network or from spot tests with passenger interviews is one of the first problems.

Line Planning

Lines are the fundamentals of periodically scheduled systems. A line is a route in the railroad network connecting two terminal stations. The frequency of a line is the number of trains that serve this route in a fixed time interval (e.g. in one hour). The line optimization problem consists in choosing a set of operating lines and its frequencies to serve the passenger demand and to optimize some given objective.

Train Schedule Generation

The generation of train schedules is divided into two parts. The first part consists in finding a regular periodic train schedule based on a proposed line plan and corresponding fixed frequencies of the lines. The arrival and departure times of the trains have to satisfy certain regulations, e.g. the safety conditions defined by the company. An objective for this construction is to minimize the total transit time of the passengers, i.e. the time spent by the customer in the system, including traveling time and waiting time. The second part, the domain of the experienced human planner, consists in adjusting the proposed regular train schedule to meet an abundance of local requirements (rush hours, splitting of lines, etc.) and peculiarities.
Schedules: Rolling Stock and Personnel

All itineraries have to be served by a train consisting of a locomotive (or railcar) and some carriages. Difficulties in scheduling the rolling stock include the restricted assignment of the material, the maintenance, and the location of the depots. All trains have to be equipped with a crew including engine-driver and accompanying staff. At this planning stage, the cost can be computed, and hence the objective is to minimize the cost subject to all cumulated conditions and requirements.

Real Time Traffic

After all the strategic and tactical planning, one has to ensure the realization of the schedules. For example, external influences will result in delayed trains. Minor irregularities in the execution of a schedule may imply severe disturbances, and it may well be necessary to recompute some schedules online.

Hierarchy of Planning

The planning tasks were described above within a hierarchy of subtasks, i.e. the generation of OD-matrices was followed by line planning, line planning was followed by train schedule generation, and train schedule generation was followed by scheduling of rolling stock and personnel. This top-down approach has certain advantages. The complete problem decomposes into subproblems of manageable size which can be solved by using currently available methods and hardware. In addition to the technical advantages, this decomposition supports the various planning time intervals which arise from another classical subdivision consisting of strategic, tactical, and operational procedures [Aum65]. Operational decisions reflect the day-by-day activities and the disturbances when executing the schedules. Tactical planning addresses resource allocation for the period from one to five years ahead. Most of the problems in railroad transport systems reviewed in this paper occur at the tactical level. Strategic planning focuses on resource acquisition for the period from five to fifteen years ahead. Network planning problems may be viewed as the main strategic issues, but, in order to evaluate possible strategic alternatives, the subsequent stages including at least line planning and train schedule generation have to be considered.

The disadvantages of the hierarchical planning are obvious, since the optimal output of a subtask which serves as the input of a subsequent task, will not result, in general, in an overall optimal solution.

Methodology and Overview

In this survey we consider recent developments and the use of mathematical programming methods in public rail transport planning within the last 5–10 years. We will focus on some new aspects of the planning process, and on some new results which lead to more comprehensive planning and optimization of railroad network systems. This survey does not claim to be comprehensive, however we strive to present some of the most challenging problems in public rail transport which these days can be modeled and solved by discrete optimization techniques. We discuss in particular the computation of the line plans, train schedules, and schedules of rolling stock. A special survey of crew scheduling methods for railroad systems can be found in [CFT+97] in this issue.

Since the estimation of the OD-matrix is a crucial basic input for these schedules, we briefly review some approaches based on mathematical programming. In view of the scope and size of this survey we cannot discuss many other problems occurring in rail transport, like pricing, seat allocation, etc.

2 Determining the Passenger Demand

In order to plan railroad services, one needs reliable information on the number of the passengers traveling from each location to any other location in the network along certain paths. Such
information on the traveling demand on these paths is based on respective traffic counts and spot tests of passenger interrogations on some links of the network. Link related data suffer from inherent inconsistencies, e.g., traffic counts are not collected simultaneously. However, the railroad companies need reliable estimates for the origin-destination demands.

Let $b^k$ be the vector of the available link data consisting of $k$ observations on each link, $A$ the matrix describing the traveling paths in the network, and $\tau$ the traveling demand from station to station which should satisfy $A\tau = b^k$. A column of $A$ corresponds to a convex combination of traveling paths connecting two stations. Due to the inconsistencies in the link data, the equation system does not admit a solution. Thus, one has to search for an estimate $\tau$ minimizing some function of the residual $r^k := (b^k - A\tau)$. Note that $A$ has more columns than rows since there are more origin-destination pairs than link data in practice. Thus, the estimation problem will not have a unique solution. Mainly, such estimates are derived from a statistical perspective using least-squares, maximum-likelihood or related methods (e.g., cf. [CN87]). However, in recent articles the influence mathematical programming techniques is increasing.

Jörnsten and Wallace [JW93] discuss how to select a unique well-defined estimate despite of the unavoidable inconsistencies within the link data. Not all link data are of the same quality. Jörnsten and Wallace propose weights $q_k^e$ representing the reliability of the $k$-th link data item on link $e$. Then, they solve a two-stage estimation problem. First, let $T$ denote the set of all solutions minimizing the penalty function

$$\sum_{e} \sum_{k} q_k^e (d_+^e (r_k^e)_+ + d_-^e (r_k^e)_-)$$

where $r_k^e := (b^k - A\tau)_e$ is the residual on link $e$ for observation $k$ and $(r_k^e)_+$ (resp. $(r_k^e)_-$) denotes the positive (resp. negative) part of $r_k^e$. If we consider the $L_1$-problem, and fix the weights $d_+^e$ and $d_-^e$ on positive and negative parts to 1 for all links $e$, this problem is solved by linear programming. Secondly, the unique solution $\tau$ is determined by solving $\min \{ G(\tau) \mid \tau \in T, \sum_{ij} \tau_{ij} = T \}$. Here, $T \in \{ \sum \tau_{ij} \mid \tau \in T \}$ denotes the expected number of passengers traveling in the system and $G(\tau)$ is an entropy function, e.g., $G(\tau) = \sum_{ij} \tau_{ij} \log \tau_{ij}$. This problem is solved by applying parametric programming in $T$.

Barbour and Fricker [BF94] discuss a heuristic approach for an estimate of an origin-destination matrix. This greedy-like method is based on shortest augmenting paths in the underlying network. Barbour and Fricker choose a shortest path between any OD-pair and send as many passengers as possible through these paths. The number of these passengers is bounded by the minimum link count on the connecting shortest paths. Using some reduction step to avoid infeasibility, they can repeat this approach for each OD-pair.

Sherali, Sivanandam, and Hobeika [SSH94] present a linear programming approach for the OD-matrix estimation problem considering all paths between an OD-pair. For consistent link data the sum of all traveler flows should be equal to the link data on each arc. Each path has a certain travel cost. Using a path longer than the shortest path is penalized by additional costs (big-M). Sherali et al. discuss some modifications of their model to consider inconsistent link data, and to use prior information about the OD-matrix. They solve the flow problem using a column generation technique for some test instances from the literature and apply it to a part of the Northern Virginia road network.

### 3 Line Planning

The line plan is the first basic schedule in the hierarchy of traffic planning (cf. section 1) for a railroad system with periodically recurrent trains. The first paper on the analysis of line planning that we are aware of is the one by Patz [Patz25] who investigates how the number of empty seats depends on the choice of lines. The rare literature [Patz25, PW76, Die78, CI92, PR95, BK95, CV95, ZBW97, BHW97] distinguishes between cost and service aspects of line planning. Here, we review recent work based on mathematical programming methods for both objectives.
The service offered by a railroad company usually consists of three different train systems IC (InterCity), IR (InterRegio), and AR (AggloRegio). The trains of a line belong to exactly one of these subsystems, hence the line plan can be split in (three) disjoint parts. According to this decomposition the underlying railroad network forms an undirected graph \( G = (V, E^c \cup E^\theta \cup E^{ar}) \), where the nodes in \( V \) represent the stations and an edge \( uv = e \in E^\theta, \theta \in \mathcal{S} = \{IC, IR, AR\} \) stands for a direct link of station \( u \) and \( v \) in subsystem \( \theta \) (cf. figure 1 (A)). Such a link can be composed of several tracks of the underlying network. For simplicity, we assume symmetric data. The models and methods can easily be extended to the asymmetric case. Due to security resp. economy requests, the number of trains on an edge \( e \in E^\theta \), is bounded from above and below by \( \overline{\varphi}_e^\theta \) and \( \underline{\varphi}_e^\theta \) (cf. figure 1 (B)). The union of selected subsets \( \mathcal{L}^\theta \) of paths in \((V, E^\theta)\), \( \theta \in \mathcal{S} \), defines the set of all possible lines, denoted by \( \mathcal{L} \). The length \( \lambda_l \) of a line \( l \in \mathcal{L} \) is the sum of the lengths of the underlying tracks of \( l \). The cycle time \( t_l \) is the time between two consecutive departures of trains of a line \( l \). Suitable cycle times for a fixed period \( T \) (e.g. 60 min) are those which lead to integral frequencies \( x_l = T/t_l \). The NP-complete line planning problem [BKZ95] consists in choosing appropriate frequencies \( x_l \in \mathbb{Z}_+ \) for \( l \in \mathcal{L} \) satisfying

\[
\overline{\varphi}_e^\theta \leq \sum_{l \in \mathcal{L}^\theta, e \in l} x_l \leq \underline{\varphi}_e^\theta \tag{1}
\]

for every edge \( e \in E^\theta, \theta \in \mathcal{S} \) (cf. figure 1 (C)).

![Part of the German network](image1)

![Bounds on the frequency](image2)

![A valid line plan](image3)

Figure 1:

We present two approaches to the line planning problem. Claessens, van Dijk, and Zwaneveld [CvZ95] focus on minimum cost line plans whereas Bussieck, Kreuzer, and Zimmermann [BKZ95] suggest a model improving the comfort of the travelers. In any approach, travelers, as the clients of the railroad company, have to be taken into account in the design of the line plan. Information on the number of travelers between certain stations of the railroad network is collected in an origin-destination matrix (cf. section 2). Using a method called system split [Olt94, BO94], one can compute OD-matrices for each subsystem \( \theta \in \mathcal{S} \) as well as the number of travelers on each edge \( e \in E^\theta \). The system split is based on the reasonable assumption that a traveler will change from a slow train to a faster train (e.g. from AR to IR or IC) at the earliest possible point of his journey and, vice versa, will leave the faster train as late as possible, if it is necessary to travel in a slower train at the end of the journey. After the system split, we know the number of travelers \( \varphi_e \) on an edge \( e \in E^\theta \) as well as the number of travelers \( \tau_{ij}^\theta \) between \( i \) and \( j \) in subsystem \( \theta \in \mathcal{S} \).

The minimum cost approach proposed in [CvZ95] admits a variable train length \( y_l \) for each line, i.e. \( y_l \in \mathbb{Z}_+ \) denotes the number of carriages for each train in line \( l \). A line plan with frequencies \( x \) and train length \( y \) provides sufficient capacity if

\[
\sum_{l \in \mathcal{L}^\theta, e \in l} \gamma^\theta \gamma^l x_l \geq \varphi_e \tag{2}
\]
holds for every edge $e \in E^\theta$, $\theta \in \mathcal{S}$, where $\gamma^\theta$ denotes the number of seats or, in general terms, the capacity of one carriage of subsystem $\theta$. The objective is to minimize the cost of the line plan in period $T$ which can be expressed by

$$
\sum_{e \in E^\theta} \sum_{i \in \mathcal{L}^e} \left( \eta^\theta \lambda_i y_{i \ell} x_{i j} + \kappa^\theta y_{i \ell} \right).
$$

Here, for one carriage of system $\theta$, $\eta^\theta$ denotes the cost per unit length and $\kappa^\theta$ denotes the cost per period $T$. Furthermore, $\mu_i$ is the turnaround time of line $l$ divided by period $T$, where the turnaround time is the time between two consecutive departures of a tram from the start station including traveling and waiting times. Hence $[\mu_i x_{i j}]$ is the number of trains that are necessary to establish line $l$ with frequency $x_{i j}$ in period $T$. The objective (3) can easily be extended by including a cost term for locomotives. Total cost decreases if we aggregate the inequalities (1) resp. (2) for parallel edges $e$ of different subsystems. The reason for this improvement is the possibility of moving capacity from expensive, fast IC trains to cheaper but slower AR trains. Of course, such an aggregation prohibits the decomposition of the problem into independent subproblems.

In contrast to the minimum cost approach, Busseeck et al. focus on improving the line plan from the traveler’s perspective. A major inconvenience is the necessity to change lines, hence the total number of changes should be minimized or, even simpler, the number of direct travelers, i.e. travelers not changing trains, should be maximized. The behaviour of the travelers in the network is modeled as follows. For each pair of stations $i, j$ with positive travel demand, we select a fixed set $\mathcal{L}^\theta_{i j}$ of “suitable” lines connecting $i$ and $j$ which appear to be convenient for travelers between $i$ and $j$. For example, lines containing a shortest path between $i$ and $j$ will be suitable. For $i, j$ with $x_{i j} > 0$, the variable $y_{i j l} \in \mathbb{Z}_+$ denotes the number of direct travelers of the origin-destination pair $i, j$ using line $l$. Obviously, summation over all suitable lines

$$
\sum_{l \in \mathcal{L}^\theta_{i j}} y_{i j l} \leq t_{i j}^\theta
$$

(4)

gives an upper bound on the number of travelers from the pair $i, j$. We assume a fixed train capacity $\gamma^\theta$ for every $l \in \mathcal{L}^\theta$. For each edge $e \in \mathcal{L}$ with $l \in \mathcal{L}^\theta$, we obtain a capacity bound

$$
\sum_{i j} y_{i j l} \leq \gamma^\theta x_{i j}
$$

by summation over all pairs $i, j$ with $l \in \mathcal{L}^\theta_{i j}$ and $e$ between $i$ and $j$ on $l$. The objective is to maximize the total number of direct travelers

$$
\sum_{\theta \in \mathcal{S}} \sum_{i j, t_{i j}^\theta > 0} \sum_{l \in \mathcal{L}^\theta_{i j}} y_{i j l}.
$$

(6)

A solution $(x, y)$ maximizing (6) subject to (1), (4), and (5) will make full use of the upper bound $t_{i j}^\theta$ in (1). On the other hand, the total length of the lines in the line plan is limited, e.g. by the fleet size of the railroad company as assigned to each subsystem, hence we have to satisfy

$$
\sum_{l \in \mathcal{L}^\theta} \lambda_l x_{i j} \leq \Lambda^\theta.
$$

(7)

for certain constants $\Lambda^\theta$. The model allows the decomposition of the line planning problem into independent subproblems defined on the subsystems $\theta \in \mathcal{S}$. We can take the different importance of connections into account using corresponding positive weights on the $y_{i j l}$, e.g. the distance between $i$ and $j$. The model is simplified relaxing the integrality of the variables $y_{i j l}$. In fact, we use the number of direct travelers for the evaluation of line plans, and are not interested in the exact optimal number of direct travelers.
Computational experience shows that both models are too complex to be directly solvable for the size of real world instances. The minimum cost model includes integer variables in quadratic terms in the objective as well as in the constraints. No standard solution technique for large quadratic integer programs is available yet. The model for maximizing the number of direct travelers is a mixed integer linear program (MIP) of tremendous size, e.g. the IR subsystem of the Germany railroad network with $|E^u| = 398$, $|L^u| = 19701$, and $|\{(i,j)|r^u_{ij} > 0\}| = 9215$ leads to a MIP with more than $9 \times 10^9$ variables, $2.6 \times 10^9$ constraints, and $4.8 \times 10^9$ nonzeros. Bixby [Bix95] solved its LP relaxation within 20 hours on an SGI-Power-Challenge using CPLEX [CPL95]. Currently, the solution of the MIP is unknown.

The authors suggest different approaches to overcome these difficulties. In the minimum cost model all quadratic terms have the form $y_ix_i$. A straightforward linearization is obtained by introducing new binary variables $z_{l,\varphi,\gamma}$ with $z_{l,\varphi,\gamma} = 1$, if the line plan contains the line $l$ with frequency $\varphi$ and capacity (number of carriages) $\gamma$, and $z_{l,\varphi,\gamma} = 0$, otherwise. The term $y_ix_i$ is replaced by $\gamma \cdot \varphi \cdot z_{l,\varphi,\gamma}$ in the objective and in the constraints. With appropriate bounds $\varphi_{\text{min}}, \varphi_{\text{max}}, \gamma_{\text{min}}, \gamma_{\text{max}}$ for the frequencies and capacities, the linearized objective function is:

$$\sum_{\varphi \in \varphi} \sum_{\gamma \in \gamma_{\text{min}}} \sum_{\gamma \geq \gamma_{\text{min}}} \sum_{\gamma \leq \gamma_{\text{max}}}(\varphi \cdot \gamma \cdot z_{l,\varphi,\gamma} + \kappa \cdot \gamma \cdot [\mu_1(\varphi) \cdot z_{l,\varphi,\gamma}], 8)$$

The constraints are treated analogously. The resulting linear 0-1 program has 5629 binary variables, 194 constraints, and 111733 nonzeros for the north west part of the railroad network in the Netherlands with 28 stations. Claassens et al. propose a branch-and-bound (B&B) algorithm based on the LP relaxation of the binary program. The program was preprocessed with various strategies before applying the B&B algorithm which uses CPLEX to solve the LP relaxations. The first preprocessing phase is based on the idea that some variables can always be replaced by other variables in any line plan and this replacement results in a feasible line plan that is at least as cheap as the original. The authors give a dominance rule that fixes some variables $z_{l,\varphi,\gamma} = 0$. Another phase improves the model formulation by coefficient improvement, e.g., the right hand side of the linearized version of (2) can be increased to $[p_c/\gamma^u] \gamma^u$. Furthermore, the complete program was processed by CPLEX MIP presolve [CPL95]. A special node selection scheme, the use of special ordered sets, and the inclusion of some violated cover inequalities [HP85] in every B&B node completes the algorithm. After preprocessing the mentioned example reduces to 1547 variables, 139 constraints, and 18192 nonzeros. The value of the LP relaxation increases from 3920.43 to 7577.53 and gives a better lower bound for the B&B algorithm. This example was solved in 3989 seconds on a SUN LX workstation.

For the maximization of the number of direct travelers, Bussieck et al. relax (5) to

$$y_{ij} \leq \gamma_{ij} x_i$$

for all appropriate pairs $i, j$ and, by aggregation of (9) for all $l \in L^u_{ij}$, to

$$\sum_{l \in L^u_{ij}} y_{ij} \leq \gamma_{ij} \sum_{l \in L^u_{ij}} x_i.$$  

The left hand side sums the number of direct travelers between $i$ and $j$ over all suitable lines. Now the variables $y_{ij}$ only occur in sums $y_{ij} := \sum_{l \in L^u_{ij}} y_{ij,l} \in \mathbb{R}_+$. The resulting relaxation consists of the maximization of

$$\sum_{\# \in \#} \sum_{ij, \tau_{ij}^u > 0} y_{ij}$$

subject to (1), (7), (10), $x_i \in \mathbb{Z}_+$, and $0 \leq y_{ij} \leq \tau_{ij}^u$. The relaxation still allows the decomposition into subsystems $(V, E^u_\theta, \theta \in \#)$. For the IR subsystem of the German railroad network, the size of the mixed integer linear problem (MIP) reduces to 28915 variables, 9692 constraints, and about
1.2-10^6 nonzeros. After a strong reformulation using cuts and coefficient reduction the problem can be solved with a standard B&B code (CPLEX MIP) in 409 seconds on an HP 715/50 workstation.

Some of the cutting planes are quite simple but efficient, and may be of general interest. Let \( P = \{(x, y) \in \mathbb{Z} \times \mathbb{R} \mid y < a, y < bx\}, b > 0 \). Then

\[
y - \Delta x < |a/b|(b - \Delta)
\]

with \( \Delta = a - |a/b|b \) is a valid inequality for \( P \). Such cuts, with \( y = y^0_j \) and \( x = \sum_{i \in \mathcal{L}^b_j} x_i \), are successfully added to the relaxation.

Using the optimal solution \((x^*, y^*)\) of the relaxation and the optimal solution of the original model for fixed \( x = x^* \), the authors provide solutions within an optimality gap of less than 3.1\% for each of the five networks from Germany and the Netherlands.

The crucial part in both models is the size of the sets \( \mathcal{L}^a \), since all lines are explicitly included in the model. Therefore, \( \mathcal{L}^a \) is chosen to be the smallest reasonable subset of the set of all paths in \((V, E^a)\). In fact, in the minimum cost model at most two lines between the origin and destination stations are included. In the maximum direct traveler model all shortest paths between classification yards form the set of suitable lines.

A classification yard is a station with special equipment to start and end a line, e.g. sidings to compose trains. In [ZBK97, Bus97] Bussieck et. al. include all paths between classification yards in \( \mathcal{L}^a \). They succeed in solving the enlarged model instances of [BZK95] using a branch-and-price procedure and obtain line plans with slightly more direct travelers and a smaller number of lines in the optimal line plan.

A comparison of the cost and direct traveler approaches can be found in [CvZ95]. Claessens, van Dijk, and Zwaneveld use a heuristic procedure described in [Die78] to generate a line plan with large (“maximal”) number of direct travelers. In a minimum cost line plan, cost is reduced by 17.2\% and the number of direct travelers is reduced by only 6\%. The latter observation is quite surprising since no constraint is added in order to bound the number of direct travelers from below. A close look at the infrastructure of the Netherlands reveals an answer for this nice behaviour. The average travel distance in the Netherlands is only about 44 kilometers, and the average edge length in \( E^a \) is 48 kilometers (in \( E^w \) 19 kilometers) [Kri96]. Hence, in any line plan, most travelers have a direct connection. This justifies a cost approach for such dense networks. On the other hand, the results for the Netherlands cannot be generalized for other railroad networks, e.g. the average travel distance in the German railroad network is about 285 kilometers with average edge length of 60 (38) kilometers for \( E^a \) (\( E^w \)). Therefore, in a recent paper [ZCD96], the minimum cost model is enhanced by variables for direct travelers with artificial negative coefficients in the objective. However, based on computational experience with small networks, even without such additional variables, an exact solution for larger networks seems to be out of reach, at least for the time being.

We conclude this section with some critical remarks on the objectives and the resulting line plans. In the minimum cost model operation cost of a railroad system are investigated at a quite early stage of planning. The schedules of carriages and locomotives (cf. section 5) are fixed (cf. turnaround times) without consideration of maintenance, depot capacities, fleet size, etc. Hence it can provide only a rough estimate of the operation cost caused by a given line plan. Furthermore,
one should carefully discuss the cost decrease which is due to replacing expensive, fast lines, say LC-lines or LR-lines, by cheaper but slower and less attractive AR-lines. Corresponding cuts in cost may reduce the attractiveness of the railroad system.

For a line plan \( x^* \) with maximum number \( Z^* \) of direct travelers, we denote the number of train changes for all other travelers by \( C(x^*) \). Then the minimum number \( C^* \) of train changes necessary in any line plan satisfies

\[
\sum_{ij} T_{ij} - Z^* \leq C^* \leq C(x^*).
\]

(13)

Even a small gap does not imply small numbers of train changes for all travelers, there may well be unacceptable large numbers of train changes for minorities among the travelers. Furthermore, an optimal line plan consists of long lines with fixed train capacity. This leads to an irregular load of the line, i.e. the number of passengers in the line. The frequency of the line is determined by the maximum loaded track hence we obtain a large number of free seats on some tracks.

4 Train Schedule

A regular train schedule consists of suitable arrival and departure times for trains at each station. For a line \( l \) these events (arrival and departure) occur with a certain frequency within the basic period \( T \), as described in a line plan. Consecutive events are subject to several constraints, since trains have to share the resources of the network, in particular the stations. It will be necessary to allocate enough time to leave and enter a train, and to provide a safety distance between subsequent trains on the same track or platform. A usual objective for the evaluation of train schedules is the minimization of the waiting time of travelers. The waiting time of a traveler changing from line \( i \) to line \( j \) is the amount of time between the arrival of line \( i \) and the departure of line \( j \) minus the amount of time \( d_{ij} \) the traveler spends for changing platforms.

We will assume that all periodic events occur once per period \( T \), and denote the scheduled time for event \( i \) by \( \pi_i \), where \( 0 < \pi_i < T \). Then event \( i \) occurs at \( \ldots, \pi_i - T, \pi_i, \pi_i + T, \ldots \). Lower and upper bounds \( l_{ij} \) and \( u_{ij} \) on the distance of consecutive instances of two events \( i \) and \( j \), where \( i \) precedes \( j \), are modeled by periodic constraints (14). There is no conflict between \( i \) and \( j \) if and only if

\[
l_{ij} \leq \pi_i - \pi_j + zT \leq u_{ij}.
\]

(14)

for some \( z \in \mathbb{Z} \). We may shift all time data to the basic interval \([0, T]\) by \( \text{per}(x) := \min\{x + zT \mid x + zT > 0, z \in \mathbb{Z}\} \) and reformulate (14) by

\[
\text{per}(\pi_j - \pi_i - l_{ij}) \leq u_{ij} - l_{ij}.
\]

(15)

For example, the distance of events \( i \) and \( j \) in any order is guaranteed to be within interval [4, 8] for \( T = 60 \) if

\[
\text{per}(\pi_j - \pi_i - 4) < 52, \quad \text{per}(\pi_j - \pi_i + 8) < 16.
\]

Analogously, most of the relevant constraints and objectives can be modeled using periodic expressions of the form \( \text{per}(\pi_j - \pi_i - l_{ij}) \). In this survey we focus on models related to such expressions. For approaches with different objectives and models based on different assumptions about train schedules, we refer to the literature [BBH90, LCD92, Car94, Kri96, NV96].

Serafini and Ukovich [SU89] introduce periodic event scheduling for a directed graph \( G = (N, A) \), where node \( i \in N \) represents a periodic event at time \( \pi_i \) and an arc \( a = ij \in A \), labeled with an interval \([l_a, u_a]\), represents the periodic constraints (15). The decision problem consists in finding a node vector \( \pi \) satisfying (15) for all arcs. In particular, feasible train schedules are solutions of periodic event scheduling problems (PESP). Serafini and Ukovich propose a backtrack algorithm for the NP-complete PESP. At first, they calculate a feasible schedule for the arcs of some fixed spanning tree of \( G \) (easy). As long as possible, the schedule is adjusted to include
constraints from other arcs. Each step requires the solution of an easy tension problem [Roc84]. Odijk [Odi96] proposes a different algorithm for the PESP based on the original formulation (14). For fixed \( z \in \mathbb{Z}^{|A|} \), feasibility of the inequality system (14) for all \( ij \in A \) is called the feasible differential problem (FDP) [Roc84]. A feasible solution exists if and only if

\[
\sum_{a \in C_+} z_a - \sum_{a \in C_-} z_a \leq \frac{1}{T} \left( \sum_{a \in C_+} u_a - \sum_{a \in C_-} l_a \right) \tag{16}
\]

holds for all cycles \( C \) in \( G \), where \( C_+ (C_-) \) denotes arcs in \( C \) with positive (negative) orientation. Seralini and Ukovich prove that without loss of generality \( z_{ij} \) can be fixed to zero for a fixed spanning tree \((N,A_0)\) of \( G \). For the cycle generated by adding a non-tree arc \( ij \) to \( A_0 \), (16) implies lower and upper bounds \( \underline{z}_{ij}, \overline{z}_{ij} \) for \( z_{ij} \). By (16), \( z \in [-|V|, +|V|]|^{|A|} \). The PESP is feasible if and only if

\[ Q_0 := \{ z \in [-|V|, +|V|]|^{|A|} | \underline{z}_{ij} \leq z_{ij} \leq \overline{z}_{ij} , ij \in A; z_a = 0, a \in A_0 \} \]

contains an integral \( z \) satisfying (16) for all cycles \( C \) in \( G \). Odijk chooses \( z_0 \in Q_0 \cap \mathbb{Z}^{|A|} \) and solves the corresponding FDP. A feasible solution of the FDP solves the PESP, otherwise, some cycle \( C \) violates (16) and \( Q_1 \) is derived from \( Q_0 \) by adding the corresponding cut. For each \( Q_i \) of the resulting finite cutting plane approach an integral point of \( Q_i \) can be generated by a standard B&B procedure. Odijk [Odi96] applies this algorithm to real world instances that come from the Dutch railroad. The size of the resulting graph \( G \) is up to 24 nodes and 54 arcs. The solution time is about 2 seconds on a PC 486.

Nachtigal [Nac93, Nac96a, Nac96b] proposes minimizing the weighted synchronization time

\[
\min \sum_{ij \in A} w_{ij} | \pi_i - \pi_j - d_{ij} \tag{17}
\]

for nonnegative weights \( w_{ij} \) and time \( d_{ij} \) for changing corresponding platforms. Using suitably large penalty factors, all periodic constraints can be added to the objective to obtain an unconstrained periodic program that can be reformulated in terms of tensions \( x \in U \) with \( U := \{ u \in \mathbb{R}^{|A|} | u_{ij} = \pi_j - \pi_i \text{ for } \pi \in [\mathbb{R}^{|V|}] \} \). Tensions can be described by a corresponding network matrix \( \Gamma \) [Sch86, Nac96b], i.e. \( U = \{ x \mid \Gamma x = 0 \} \). Therefore, the unconstrained periodic problem is equivalent to

\[
\min \sum_{ij \in A} w_{ij} | x_{ij} - d_{ij} \mid \Gamma x = 0 \tag{18}
\]

or, using the substitution \( y := x + Tz - d \), to the MIP

\[
\min \sum_{ij \in A} w_{ij} y_{ij} \mid \Gamma y = TTz - \Gamma d; y \geq 0. \tag{19}
\]

Since \( z \) is unconstrained, \( \Gamma z \) can be replaced by \( Hz \) where \( H \) is the Hermite of \( \Gamma \) to improve the performance of B&B for solving (19). For random instances \((|V| \leq 80, |A| \leq 95)\) (19) can be solved in less than 6 minutes on a SUN IPSC workstation.

In a recent paper [Nac96a] Nachtigall considers the polyhedron \( P \) associated with the periodic constraints in (19):

\[
P := \text{conv} \{ x \mid \Gamma x = -\Gamma d \mod T, x \geq 0 \}. \tag{20}
\]

He also derives a class of facet defining change cycle inequalities with a polynomial separation algorithm and a class of chain inequalities that are facet defining. In a preliminary phase of the B&B algorithm, these cuts can be applied in a cutting plane procedure to improve the LP bound of (19). In this way a real world instance from the German railroad network with 26 nodes and 141 arcs can be solved in 4 minutes on a SUN Sparc 5 using CPLEX 4.0 for the linear programs.
For small period $T$, a completely different approach is proposed by Klemt and Stemme [KS88]. In the network $G$ of periodic events each node $i$ is replaced by $T$ nodes $i1, \ldots, iT$. Any two nodes $ih$ and $jk$, $i \neq j$, are connected by an edge. A vector $x$ of binary variables $x_{ih}$ satisfying
\[
\sum_{h=1}^{T} x_{ih} = 1 \tag{21}
\]
for all periodic events $i$ represents a periodic schedule. Edge cost $c_{ihjk}$ reflect an estimate of the demand of travelers between $i$ and $j$ and the waiting time per($k-h$) or define a penalty for combinations that violate a constraint. The minimization of
\[
\sum_{i,h} \sum_{j,k} c_{ihjk} x_{ih} x_{jk}, \tag{22}
\]
for all binary vectors $x$ subject to (21) is known as the quadratic semi-assignment problem (QSAP) [DFS92]. For solving this NP-hard problem, Klemt and Stemme compare complete enumeration and a method related to the nearest neighbor heuristic for the TSP. On average, the observed cost of the heuristic solutions stay within a 12% error bound above the optimum. Already for a small subway model in West-Berlin with 8 periodic events and 68 resulting edges complete enumeration fails. Domshke [Dom89] proposes a B&B algorithm for the QSAP using some relaxation of the problem for bound calculation. He solves instances with up to 16 periodic events in two days on an IBM AT 02. On the average, heuristics based on simulated annealing and local search produce solutions, the cost of which stay within a 2.5% bound above the optimum determined by the B&B algorithm. Daduna and Voß [Voß92, DV95] report on certain improvements when applying exhaustive local search strategies like tabu search in combination with reasonable initial solutions. Voß [Voß92] also presents some polynomial solvable cases of the QSAP.

Until now, we assumed that all events occur once within $T$, i.e. all events have the same period $T$. In principle, the concept of periodic event scheduling can be generalized for different periods. The periodic constraint
\[
l_{ij} \leq \pi_j + z_j T_j - \pi_i - z_i T_i \leq u_{ij} \tag{23}
\]
for events $i$ and $j$ with different periods $T_i$ and $T_j$ can be reformulated as
\[
l_{ij} \leq \pi_j - \pi_i - z_i T_i \leq u_{ij} \tag{24}
\]
with $T := \gcd(T_i, T_j)$. Nachtragall [Nac96b] generalizes his B&B approach for PESP and solves random instances of the generalized PESP with up to 40 nodes and 68 arcs in 8 minutes on a SUN IPC workstation. However, from the practical point of view, the inequalities in the generalized PESP are not very useful, since they only capture lower and upper bounds for some, not all, consecutive occurrences of events $i$ and $j$.

The departure and arrival times in an automatically generated regular tram schedule have to be adjusted to local requirements that cannot be included in an efficient global model. Though most of these changes will be done by hand, some occurring problems can be formulated and solved by mathematical programming methods.

### 4.1 Creation of Short-Turn Lines

In long passenger lines, the load (number of passengers) on different tracks varies considerably. Fitting frequency and capacity of the line to the track with maximum load results in a large number of empty seats on other tracks. A line $l$ connecting suburbs $a$ and $b$ via a centre $c$ has an extremely changing load. In the morning, the load is high on tracks from $a$ to $c$ but low on tracks from $c$ to $b$. If $n$ departures of $l$ from $a$ in some time interval $T$ are sufficient to carry the high load, $m < n$ departures of $l$ in $c$ are sufficient for the low load. Therefore, we replace line $l$ for $n - m$
departure times by a short-turn line $a \to c$. Which ordered subset $T' = \{\pi_1 < \pi_2 < \ldots < \pi_m\}$ of the ordered set of departure times $T = \{\pi_1 < \pi_2 < \ldots < \pi_n\}$ of line $l$ should stay in the schedule?

Ceder \cite{Ceder1991} suggests minimizing the headway in $T'$, i.e., the maximal distance of two consecutive departures. In the case of a regular schedule $\pi_t = \pi + (i-1)t$ with $t := T/n$, the optimal choice is $\pi_t = \pi + (i-1)[n/m]$. Due to previously mentioned necessary adjustments, $T$ may be irregular. If we take the headway $(\pi_1 + T) - \pi_m$ into account, we obtain identical departures for each period $T$. On the other hand, $\pi_1$ and $\pi_n$ may be fixed necessary departures, e.g., connections to the regular part of the train schedule. Then $\pi_1 = \pi_1$ and $\pi_m = \pi_n$ and we have to find $\pi_2, \ldots, \pi_{m-1} \in T$ such that the headway

$$\max\{\pi_i - \pi_{i-1} \mid 2 \leq i \leq m\}$$

is minimized. Ceder discusses only the latter case. Let $\tau := (\pi_n - \pi_1)/(m-1)$. If $h_{\max} := \max\{|\pi_i - \pi_{i-1}| \mid 2 \leq i \leq m\} \geq 2\tau$ then $h_{\max}$ is the minimum headway and $\pi_i = \max\{|\pi_j| \mid \pi_j \leq \pi_{i-1} + h_{\max}\}$. Otherwise, for $h_{\max} < 2\tau$, we consider the directed graph $G = (T, A)$ with $A := \{\pi_i \pi_j \mid \pi_i < \pi_j\}$ and nonnegative weights $w(\pi_i \pi_j) := \pi_j - \pi_i$ on $A$. The nodes of any path $P$ of length $m - 1$ from $\pi_1$ to $\pi_n$ define a feasible train schedule. In particular, a path $P^*$ minimizing

$$\max_{e \in P} w(e)$$

corresponds to the optimal schedule. Finding such a path with constrained length in arbitrary graphs is NP-hard. Without the length constraint, it can be solved by Dijkstra's shortest path algorithm in the monoid $(\mathbb{Z}_+, \max, <)$ \cite{Zimmerman1981}. Ceder \cite{Ceder1991} proposes a variation of $G$ to solve the constrained shortest path problem. In an optimal solution, consecutive departures will satisfy $\pi_i - \pi_{i-1} < 2\tau$. Therefore, we consider the layered graph $G' = (N_1 \cup N_2 \cup \ldots \cup N_m, A')$ where $N_1 = \{\pi_1\}, N_m = \{\pi_n\}$, and $N_i = \{\pi_j \mid (i-2)\tau \leq \pi_j \leq i\tau\}$ for $2 \leq i \leq m - 1$. In order to obtain disjoint node sets, we attach an upper index $i$ to the nodes in $N_i$. An arc $\pi_i^{j-1} \pi_k^i$ leads from layer $N_{i-1}$ to layer $N_i$ if and only if $\pi_i^{j-1} < \pi_k^i$. Any path from $\pi_1^1$ to $\pi_n^m$ has $m - 1$ arcs and defines a feasible train schedule. Since the shortest path $P^*$ from the constrained shortest path problem described above is contained in $G'$, any path from $\pi_1^1$ to $\pi_n^m$ in $G'$ minimizing (26) yields an optimal schedule. Ceder solves random examples with $n < 270$ and $m < 217$ in a few seconds on an IBM 370/380.

### 4.2 Routing Trains Through a Station

Global models generate train schedules that are feasible with respect to safety constraints between stations but do not consider the detailed structure of a station with its safety rules. A station consists of several platforms and a large number of sections that contain switches and crossings of tracks (cf. figure 3). The route reservation procedure for an incoming train allocates the complete route from the entry point to a suitable platform. A section of this inbound route is released when the train has left the section (adding some buffer time). The reservation of the complete inbound route reflects the de facto safety rules of European railroad companies and prevents deadlocks. An analogous procedure applies for the outbound route. The velocity, acceleration, and other train parameters define time intervals $[x_t, y_t]$, that correspond to the reservation time of train $t$ for each section $s$ independent of the chosen route. Route $r$ of train $t$ and route $r'$ of train $t'$ are compatible if $[x_t, y_t] \cap [x_{t'}, y_{t'}] = \emptyset$ for all sections $s$ concurrently used by $r$ and $r'$. The concept of compatible routes can be extended to model service constraints. If many travelers change from train $t$ to train $t'$, both trains should use opposite sides of the same platform. Route combinations $r$ and $r'$ of such trains $t$ and $t'$ will
be called \textit{compatible}. For possible inbound (outbound) routes $R^i_t$ ($R^o_t$) and reservation intervals $[x_t, y_t]_r$ for each train of a set $T$ of trains, one has to find a \textit{routing} through the station, i.e. a set of $2|T|$ compatible routes.

Kroon, Romeijn, and Zwanve H [KRZ96] prove that routing subject to safety rules is \textit{NP}-complete, if $|R^i_t| > 3$, and describe an $O(|T|^2)$ algorithm, if $|R^o_t| < 2, t \in \{i, o\}$. For a fixed layout of the station the routing problem transforms to a shortest path problem in a graph the size of which is polynomial in the number of trains (but exponential in the size of the station).

Zwanve H et. al. [ZDPvH+96] propose a node packing formulation of the problem using the graph $G = (N := \bigcup_{t \in T} R^i_t \cup R^o_t, E)$. Edges represent pairs of incompatible routes. An independent set of nodes corresponds to a set of compatible routes, hence we have to find an independent set (or a node packing) of size $2|T|$. For node incidence vectors $x$, the maximum node packing problem reads

$$\max \left\{ \sum_{r \in N} x_r \mid x_r + x_{r'} \leq 1 \quad \forall r, r' \in E, x_r \in \{0, 1\} \right\}.$$  \hspace{1cm} (27)

Due to the poor LP relaxation of (27), a standard B&B approach is not very promising. Zwanve H et. al. [ZDPvH+96] improve the linear formulation by adding valid inequalities, and by using exhaustive preprocessing. For example, a route variable $x_r$ for train $t$ can be fixed to 0 if another route $r'$ for the train allows at least the same routings for all other trains, i.e. $\{r | r \not\in E\} \subseteq \{r | r' \in E\}$. Using the special structure of the graph $G$ we can replace the inequalities in (27) for route $r$ of train $t$ and incompatible routes $r'$ of train $t'$ by the single inequality

$$x_r + \sum_{r' \in R^o_t, r \not\in E} x_{r'} \leq 1.$$ \hspace{1cm} (28)

These modifications reduce the size of the binary program, and tighten the LP relaxation of (27). Large cliques in $G$ are easily generated, and lead to strong cuts such as

$$\sum_{r \in R^i_t} x_r + \sum_{r' \in R^o_t, r \not\in E, r' \in E} x_{r'} \leq 1.$$ \hspace{1cm} (29)

Zwanve H et. al. develop a branch-and-cut (B&C) procedure for this tighter formulation using a heuristic for the generation of violated clique constraints in the B&B nodes. Several instances modeling the station of Zwolle can be solved in 85 seconds on a SUN LX workstation. A related decision support system used at NS Heizigers is reported to solve all practical instances within three minutes [Zwa97]. Surprisingly, for all instances the (truncated) optimal LP value coincided with the optimal value of a certain heuristic method. Thus, the B&C part of the algorithm was never used. In a recent paper [KZ95] Kroon et. al. extend their approach to include shunting decisions and deviations from the train schedule [ZDPvH+96].

5 \textbf{Circulation of Rolling Stock}

In order to realize the train schedule it is necessary to plan the circulation of the needed stock. Each train requires a locomotive and carriages. The length of a train may vary during service due to the coupling and decoupling of carriages. It is also possible to exchange locomotives at certain stations. In long distance rail traffic, the most important schedule required is that for dispatching locomotives to the trains of the schedule. Depending on the available stock, one has to decide at which stations to couple or decouple train units (cf. 5.2). For short distance rail traffic, the used stock mainly consists of single train units which may be of different type. In the following, a piece of the train’s route served by a particular train unit will be called a \textit{schedule trip}. The set of all schedule trips will be denoted by $T$. 

13
5.1 Dispatching Train Units to Schedule Trips

The problem of assigning train units from several depots to schedule trips, such that every trip is served and certain constraints are satisfied, is called multiple depot vehicle scheduling problem (MDVSP) and is known to be NP-hard for more than one depot [BCG87]. A trip \( i \in T \) can be served by certain train units (or vehicles) of different capacity and operating cost. Vehicles of the same type and the same cost are collected in a depot \( d \in D \). Depot \( d \) contains \( r_d \) vehicles. Two schedule trips \( i, j \in V \) are called compatible if they can be served by the same vehicle of depot \( d \) in consideration of the travel time from the end-point of trip \( i \) to the starting-point of trip \( j \), including some waiting time. A feasible solution of the MDVSP assigns suitable vehicles to compatible schedule trips such that each trip is served and the number of vehicles taken from a depot does not exceed \( r_d \). This problem can be formulated as a multi-commodity flow (MCF) problem in the following directed (multi-)graph \( G = (D \cup T, A = \bigcup_{d \in D} A_d) \). For two compatible trips \( i, j \in T \) there is an arc in \( A_d \) (cf. figure 4) if a vehicle of \( d \) is suitable to serve them. The cost on these arcs covers the operational cost for using such a vehicle for service.

For depot \( d \) and trip \( i \), the arcs \( di \) and \( id \) belong to \( A_d \), if \( i \) can be served by a vehicle of \( d \). The cost on \( id \) covers the operational cost, whereas cost on \( di \) include operating and capital cost for using such a vehicle. Using a minimum number of vehicles can be forced by prohibitively large capital cost (big-M) on \( di \). We remark that all arcs represent deadhead trips and each arc is assigned to a unique depot.

A feasible circulation consists of cycles in \( G \). Each cycle starts and ends in the same \( d \in D \) and passes through a chain of compatible schedule trips along arcs from \( A_d \). Such cycles are called compatible cycles. The objective is to minimize the sum of the cost of the deadhead trips used to serve all schedule trips.

Let \( \delta^-(d) = \{ij | ij \in A_d\} \) and \( \delta^+(i) = \{ij | ij \in A_d\} \) for all \( d \in D \). We describe solutions by an arc incidence vector \( x \) with \( x_a = 1 \) if and only if \( a \) is used in one of its cycles. In this notation, we derive the following MCF formulation:

\[
\begin{align*}
\min & \sum_{d \in D} \sum_{a \in A_d} c_a x_a \\
\text{s.t.} & \sum_{a \in \delta^-(d)} x_a \leq r_d \quad \text{for all } d \in D \quad (30) \\
& \sum_{a \in \delta^+(i)} x_a = 1 \quad \text{for all } i \in T \quad (31) \\
& \sum_{a \in \delta^-(i)} x_a - \sum_{a \in \delta^+(i)} x_a = 0 \quad \text{for all } i \in T, \quad d \in D \quad (32) \\
& r_a \in \{0, 1\} \quad \text{for all } a \in A \quad (33)
\end{align*}
\]

The MDVSP has been investigated in various papers. There are some surveys concerned with scheduling problems in public transport (cf. [DW88, DR92, DBP95, DDSS95]). We will briefly discuss some traditional approaches [Nen81, Rad95] and will describe some recent results and techniques to solve large scale MDVSPs arising in passenger transportation [CDFT89, FHW94, RS94, Löb96b].

At the beginning of the 1980s, the decomposition of the MDVSP into several single depot problems is considered. Neng [Nen81] investigates locomotive schedules for the German railroad company (DB AG), formerly called Deutsche Bundesbahn (DB). His objective consists in minimizing the total number of required locomotives and the total amount of deadhead trips. Neng defines which locomotive type (depot) may serve a schedule trip. This implies the decomposition into single depot problems. Then Neng uses a modified assignment algorithm for 700 schedule trips of DB and three different types of locomotives. A related approach is used for a larger subsystem of DB AG system by Radtke in 1995 [Rad95].
With the increasing power of computers and with improved techniques from combinatorial optimization, many depot problems became solvable within an acceptable time. One of the first papers on this subject is due to Carpaneto, Dell’Amico, Fischetti, and Toth [CDFT89]. They consider an MDVSP with identical vehicles but different operational cost for the pull-out and pull-in trips. They propose a path-oriented formulation of the MCF problem which they solve using a B&B scheme similar to that for the ATSP [CT80]. Ribeiro and Soumis [RS94] develop a set partitioning formulation with side constraints for the MDVSP, and solve its LP relaxation by column generation. Forbes, Holt, and Watts [FHW94] solve the MCF problem by a B&B algorithm, and apply it to real-world data. Grötschel, Lőbel, and Vöker [GLV97, Lő96b, Lő97] describe a B&C algorithm for the MDVSP, including a detailed and efficient solution technique for its LP relaxation. The developed algorithm is applied to large-scale instances from three German local transport companies.

In the B&B algorithm of Carpaneto et al. [CDFT89] at each branch node \( \alpha \) of the decision tree, a single depot relaxation is solved, and a simple interchange heuristic is applied to find some feasible solution that provides an upper bound. For the current node \( \alpha \), let \( X \) be the set of all nodes of \( G \) already connected by compatible cycles, and let \( D_X \) be the multi-set of all depots touched by these. Furthermore, \( F^{\alpha} \) contains arcs forbidden at some previous step.

Now, if the solution of the single depot relaxation in node \( \alpha \) consists of compatible cycles, we have found a feasible solution of the MVSDP, and can proceed with the next node of the decision tree. Otherwise, its cost is only a lower bound which can possibly be improved by adding the minimum cost of the connection of each trip \( t \in T \) to only one depot with respect to the reduced cost defined by the solution. Ribeiro and Soumis prove that this lower bound cannot be better then lower bounds obtained from Lagrangean relaxation. Let \( S^\alpha = \{(v_1, v_2), \ldots, (v_h, v_{h+1})\} \) be a path of \( h \) arcs connecting depot \( d_1(=v_1) \) with depot \( d_2(=v_{h+1}) \). We can split this path at each of the \( h \) arcs. This gives us \( h \) children of \( \alpha \). At child node \( \alpha_i \) we forbid the arc \((v_i, v_{i+1})\), add the first part of chain \( S^\alpha \) to \( X \), and update \( D_X \). Fathoming is improved by a certain dominance criterion.

The interchange heuristic chooses two paths \( S_1 \) and \( S_2 \), each starting and ending in different depots. Choose two nodes \( i \) and \( j \), one out of each path. Connect the node \( h \) preceding \( i \) in \( S_1 \) with the successor node \( k \) of \( j \) in \( S_2 \) and \( j \) with \( i \). Delete \( hi \) and \( jk \), and update the cost which provides some upper bound. At least one cycle results. This heuristic may connect incompatible nodes increasing the cost infinitely.

Ribeiro and Soumis [RS94] propose a set partitioning reformulation of the MDVSP based on compatible cycles. For each depot \( d \in D \), the set \( \Omega_d \) consists of all compatible cycles starting in \( d \). For every \( p \in \Omega \) \( = \bigcup_{d \in D} \Omega_d \), let \( c_p \) be the cost of its arcs, and let \( a_{ip} = 1 \) if and only if \( p \) visits schedule trip \( i \in \mathcal{T} \), and 0 otherwise. The binary decision variable \( y_p \) indicates whether \( p \) is used in the solution. The optimal solutions of the resulting set partitioning formulation

\[
\begin{align*}
\min & \quad \sum_{d \in D} \sum_{p \in \Omega_d} c_p y_p \\
\text{s.t.} & \quad \sum_{p \in \Omega_d} y_p \leq r_d \quad \text{for all } d \in D \\
& \quad \sum_{d \in D} \sum_{p \in \Omega_d} a_{ip} y_p = 1 \quad \text{for all } i \in \mathcal{T} \\
& \quad y_p \in \{0, 1\} \quad \text{for all } p \in \Omega
\end{align*}
\]

(35, 36, 37, 38)

 correspond to compatible minimum cost cycles. Ribeiro and Soumis solve the LP relaxation using a column generation technique (Dantzig-Wolfe decomposition). The subproblem generating a column with minimal reduced cost among those of \( \Omega_d \) consists in finding a shortest compatible cycle \( p \) visiting at least one schedule trip. Then the new column consists of the incidence vector \( A_p \) of \( p \) and \( c_p \) is the cost of cycle \( p \). The subproblem can be formulated as an unconstrained shortest path problem in an acyclic graph.

Using this column generation technique, Ribeiro and Soumis solve the LP relaxation. The column generation approach was implemented using GENCOL [SDD+90] as part of a B&B scheme with DFS branching strategy.
Forbes, Holt, and Watts [FHW94] solve the MDVSP using the following B&B algorithm. In a first step, they solve a single depot relaxation in which they contract all multiple arcs to a single arc. The resulting network flow problem is easily solved. Then they add all the flow conservation constraints (33) to the relaxed problem which is solved by a dual simplex algorithm starting from the solution achieved in step 1. Finally, a B&B scheme leads to integrality.

The branching strategy they apply is a two step procedure. In the first step, they branch on the sum of all variables \( x_a \) with \( a \in \delta^+_d(d) \), if not integral. Round-up and round-down define two child nodes. If all sums are integral, but the current solution \( x \) is not integral then there exists some arc \( a = id \) with \( x_a < 1 \). Branching on \( x_a \) yields two children with \( x_a = 0 \) and \( x_a = 1 \).

Grötschel, Löbel, and Völlker [Löb96b, GLV97, Löb97] propose a B&C algorithm for solving the MCF formulation of the MDVSP. They mainly discuss the efficient solution of the LP relaxation, and try to close the integrality gap.

Löbel et al. consider the LP relaxation of the MCF, possibly omitting certain inequalities. If all known inequalities are added, the algorithm may proceed to B&B [GLV97]. Löbel solves the LP relaxation as follows [Löb96b]. Firstly, he determines an initial, primal feasible solution of the MDVSP. Here, the schedule first cluster second heuristic (cf. [GLV97] for details) or the much faster nearest depot heuristic which assigns each schedule trip to a depot with the smallest deployment cost can be used. In spite of its theoretically worse behaviour the latter is more efficient in practice. Then Löbel starts column generation beginning with the initial solution. His algorithm splits into three parts which are iterated until the relaxation is solved.

In the first part, Löbel considers the Lagrangean relaxation with respect to equation (32) by adding the redundant constraint set \( \sum_{a \in \delta^+_d(i)} x_a \leq 1 \) for all \( i \in \mathcal{T}, \ d \in \mathcal{D} \). The problem decomposes into \( m \) independent minimum cost flow problems which are solved by a network simplex algorithm [Löb96a]. All columns corresponding to variables with a positive value are added to the LP relaxation.

In the second part, Löbel solves the Lagrangean relaxation with respect to the flow conservation constraints (33) and to the depot capacity constraints (31) by adding the redundant constraint set \( \sum_{a \in \delta^-_d(i)} x_a = -1 \) for all \( i \in \mathcal{T}, \ d \in \mathcal{D} \). This problem turns out to be a large scale single depot problem which is also solved by the network simplex algorithm. Again, all columns corresponding to variables with a positive value are added to the LP relaxation. The Lagrangean relaxation approaches are discussed in detail in [KL96] and [Löb96b].

After column generation, the updated LP is solved using the dual simplex method. If the obtained optimal value is significantly improved, the Lagrangean phase is iterated. Otherwise Löbel continues with a standard column generation technique to solve the LP relaxation. Then to derive an integral solution, he applies some rounding heuristic which does not violate feasibility. The value of the LP solution turns out to be very close to the optimal integer solution.

A motivation for the necessity of the multiple-phase method is that the LP relaxation is completely degenerated. Furthermore, standard column generation activates only variables with negative reduced cost whereas further improvement is achieved by activating some other variable which may locally promise some objective progress. The reader may consult [Löb96b, KL96] for details. Löbel determines the minimal number of vehicles needed for service by a big-M method. He also minimizes the operational cost for the required deadhead trips. Surprisingly, all instances solved do not require explicit use of the derived cutting planes [Löb97].

Computational Results

Carpaneto et al. implement their algorithm in FORTRAN 77, and solve problems with random data for \( |D| = 2, 3 \) and \( |T| = 30, \ldots, 60, 70 \) on an HP 9000/840 workstation in less than 4000 seconds.

Ribeiro and Soumis apply their algorithm to problems with random data for \( |D| = 2, \ldots, 6 \) and \( |T| \) up to 300. Some problem instances are solved with observed average computation time less than 6250 seconds on a Sun Sparc 2 (CPLEX). The average gap between their LP relaxation and the optimal solution is less than 0.004 percent.
Forbes, Holt, and Watts test their algorithm on the random data problems of Ribeiro et al. Furthermore, they apply it to subproblems from the weekday trip data set of an actual bus operator. These instances consists of three depots and over 6500 schedule trips. They solve problem instances of both types for $|D| = 2, 3$ and $|T| = 100, 200$ within less than 11000 seconds on a PC 80486/25Mhz. The largest gap between the solution of the relaxation and the integer solution was observed to be 0.005 percent. They report on solving problems with $|T| = 600$ and an average number of variables of approximately 90000.

Löbel et al. apply their algorithm to real world data of three German transport companies of Hamburg and Berlin. The largest problem consists of 44 depots, 24906 schedule trips, and up to 70000000 deadhead trips. The LP relaxation of this problem could be solved, concerning a fleet minimal solution, within 200 hours running CPLEX 4.0 on a SUN UltraSparc (Model 170) with 512 MByte main memory. For this large instance, the gap with respect to cost between the best known upper bound on the optimal integer solution and the solution of the relaxation is about 10 percent. Löbel can solve problem instances for up to 13 depots and up to 8500 schedule trips within acceptable time (less than 120000 seconds) [Löb96b].

Conclusion

Within the last seven years, several studies on the MDVSP show that large problem instances became solvable in acceptable time, or at least solvable. The described algorithms can optimize the dispatch of vehicle units in local transport as well as in railroad traffic systems.

5.2 Dispatching Stock for a Single Line

Schrijver [Sch93] develops a detailed model for minimizing the number of train units for an hourly train service of the Nederlandse Spoorwegen (NS) on the line Amsterdam - Vlissingen. At four stations (Amsterdam, Rotterdam, Roosendaal, and Vlissingen), train units can be coupled to and decoupled from a train, or stored overnight. Link demands are known, and require a minimum of available seats. The available train units differ with respect to length and number of first and second class seats. The length of a train, i.e. the number of its carriages, is bounded by a constant $K$. For identical train units the resulting minimum cost flow problem can easily be solved. For different types of train units a minimum cost multi-commodity flow problem results.

The underlying network $G = (N, A)$ contains a node $v = (X, t)$ for each arrival and departure at time $t$ of a train at station $X \in \{\text{Amsterdam, Rotterdam, Roosendaal, Vlissingen}\}$. An arc $uv$ links node $u = (X, t)$ to $v = (X, t')$ whenever $t'$ is an immediate successor of $t$ in the cyclically ordered set of arrival and departure times. $A_\Delta$ denotes the subset of the arcs $uv$ corresponding to nodes $u = (X, t_{\max})$ and $(X, t_{\min})$ in the linearly ordered set of arrival and departure times. Node $u$ is linked to a node $w = (Y, t')$, $X \neq Y$ if a schedule trip leaves $X$ at time $t$ and arrives in $Y$ at time $t'$ (cf. figure 5).

In particular, Schrijver considers the case of two different types of train units. For $D = \{d_1, d_2\}$, a train unit of type $d \in D$ consists of $k_d$ carriages, each containing a number of $f_{d}$ (resp. $s_{d}$) of first (resp. second) class seats, and requires cost $c_d$. $x_{da}$ denotes the number of train units of type $d$ on arc $a$. $p_{ja}$ denotes the traffic demand for seats of class $j$ on arc $a$.

![Figure 5: The network $G$](image)
We derive the following integer linear program (ILP)

\[
\begin{align*}
\min & \quad \sum_{d \in D} \sum_{a \in A^d} c_{d,a} x_{d,a} \\
\text{s.t.} & \quad \sum_{a \in \delta^+(v)} x_{d,a} - \sum_{a \in \delta^-(v)} x_{d,a} = 0 \quad \text{for all } d \in D, \quad v \in N \\
& \quad \sum_{d \in D} \sum_{a \in \delta^+(v)} f_{d,a} x_{d,a} \geq p_{1a} \quad \text{for all } a \in A \\
& \quad \sum_{d \in D} s_{d} x_{d,a} \geq p_{2a} \quad \text{for all } a \in A \\
& \quad \sum_{d \in D} k_{d} x_{d,a} \leq K \quad \text{for all } a \in A \\
& \quad x_{d,a} > 0, \text{ integer} \quad \text{for all } a \in A, \quad d \in D
\end{align*}
\]  

(39)

(40)

(41)

(42)

(43)

(44)

Constraint (40) represents the flow conservation in each node of \( G \), constraints (41) and (42) represent the demands for first and second class seats on each arc, and constraint (43) bounds the total flow on each arc.

Solving this problem by standard B&B code (CPLEX 2.1) requires several hours on an SGI R4400. Schrijver proposes tightening the constraints (41),(42), and (43) for each arc \( a \). In fact, the exact convex hull of the integral solutions of the two-dimensional polytope \( P_a \) defined by (41),(42), and (43) can be derived in a preprocessing step in 0.04 seconds. In this way, Schrijver solves the ILP in less than 2 seconds.

6 Real-time Problems while Executing the Schedules

Whenever planned schedules are executed one has to face real-time problems. Some external influences and some insufficiencies within the underlying model used to construct the schedules will cause trouble. Breakdowns of locomotives as well as unpredictable delays will disturb the real-time execution. To keep these influences on the traveler’s comfort at a low level, the dispatcher has to decide within a short time interval how to improve the situation.

Most of the concepts known to the authors consider real-time control techniques to keep the systems running at a satisfactory level [LRG95, Ada95]. Due to lack of time to react, complete reoptimization will not be possible. Alternatively, local improvement strategies are applied.

As an example, we consider internal difficulties appearing in storage yards which are usually neglected when preparing the schedule of rolling stock. In fact, the dispatch in storage yards is severely influenced by the real-time situation [KK96, Cen96].

For light rail traffic Winter and Zimmermann² consider shunting problems in storage yards with dead end tracks. Here, the dispatcher knows the arrival times of the trams as scheduled. The trams can be partitioned into several sets of identically equipped vehicles. To execute each trip of the schedule, a tram of a certain type has to depart at predetermined times. The objective is to find an assignment of the arriving stock to siding positions and to forthcoming departures such that the total amount of shunting movements needed for dispatching is minimized. Due to delays and other on-line effects the actual arrival order of the trams is unknown, so that the time interval

\[\text{Figure 6: shunting}\]

²Project “Assignments of local transport vehicles to routes and sidings minimizing shunting cost in storage yards”, cf. http://www.math.tu-bs.de/wo/projects/winter.html supported by the German Science Foundation (DFG)
between two arrivals or departures may be very narrow. Since the trams have to be dispatched on arrival, the dispatcher must decide in real-time which assignment is suitable for the incoming stock. The off-line problem can be modelled by the following quadratic program

\[
\min_{s.t.} \ y_{i,k} \ a_{i,k} b_{i,p} x_{i,p} x_{k,q} + y_{i,j} a_{i,j} b_{i,q} y_{j,i} y_{k,q} \tag{45}
\]

\[
\sum_{p} x_{i,p} = 1 \quad \text{for all} \quad p \in P \tag{46}
\]

\[
\sum_{i} x_{i,p} = 1 \quad \text{for all} \quad i \in A \tag{47}
\]

\[
x_{i,p} \in \{0, 1\} \quad \text{for all} \quad i \in A, \quad p \in P \tag{48}
\]

\[
\sum_{p} y_{i,p} = 1 \quad \text{for all} \quad p \in P \tag{49}
\]

\[
\sum_{j} y_{i,p} = 1 \quad \text{for all} \quad j \in D \tag{50}
\]

\[
y_{i,q} \in \{0, 1\} \quad \text{for all} \quad j \in D, \quad p \in P \tag{51}
\]

\[
x_{i,p} + y_{i,p} \leq 1 \quad \text{for all} \quad p \in P, \quad \text{type}(i) \neq \text{type}(j) \tag{52}
\]

where

\[
a_{i,k} = \begin{cases} 1 & \text{if } i < k \\ 0 & \text{otherwise} \end{cases} \quad b_{p,q} = \begin{cases} 1 & \text{if } p > q \text{ within the same siding} \\ 0 & \text{otherwise} \end{cases}
\]

A denotes the set of arriving vehicles, D the set of departures, and P the set of siding positions. Note that this model consists of two independent quadratic assignment problems connected by the constraint set (52) representing the assignment of stock of the right type to the departures. Assume a tram arriving at time \( k \) is assigned to siding position \( q \). Shunting is unavoidable if a tram arriving before time \( k \) is assigned to a position \( p > q \) (cf. figure 6), with an analogous statement holding for two departures.

Blasum et al. [BBH+96] attack the real-time problem by considering the subproblem concerning the more time sensitive departure part. In the real world instances considered, one observes that in most cases the optimal solution does not require any shunting. Winter et al. present a decision procedure to determine whether or not a shunting-free solution exists.

Winter and Zimmermann apply reactive tabu search developed by Battiti and Tecchiolli [BT94] to this NP-hard subproblem [BBH+96]. Computational results show that this heuristic performs well in practice, and delivers a solution close to an optimal value.

In fact, there are real-time problems in public transport disturbing the schedule. Delays of trains will have an influence on the schedule and on the traveling comfort as well as construction work on tracks. These days, due to the improvement of control techniques in public transport, e.g. new and more efficient satellite surveying systems, it is possible to detect disturbing real-time effects soon and to react faster and more precisely because of the increasing computational power. In the future, mathematical programming methods will help to reduce real-time influences using local reoptimization procedures.

7 Conclusion

In this survey, we reviewed several problems from public rail transport and their solution approaches. With the exception of resource sharing (cf. section 4, 4.2), most of the models can be transferred to other public transport systems with regular service schedules. In fact, most of the literature cited in section 5 deals with public bus transport.

During last decade, mathematical programming methods have been applied to real world instances of hard optimization problems arising in public rail transport. Improving the cooperation with transportation companies results in a deeper insight of the planning problems. This was and will be necessary to model operational restrictions, and to increase the acceptance of mathematics in practice. The continuous progress in mathematical programming methods may result
in an attempt to aggregate certain subtasks, like the creation of line plans and train schedules or the scheduling of staff and rolling stock, and may lead to improved overall solutions. Currently, however, the size and the complexity of such combined models provides the mathematical programming community with a real challenge.

References


21


