

# 5th Workshop on Algorithmic Methods and Models for Optimization of Railways

ATMOS 2005, October 7, 2005, Palma de Mallorca, Spain

Edited by

Leo G. Kroon

Rolf H. Möhring



### *Editors*

Leo G. Kroon  
Erasmus University Rotterdam  
Rotterdam School of Management  
P.O. Box 1738  
3000 DR, Rotterdam, The Netherlands  
lkroon@rsm.nl

Rolf. H. Möhring  
Technische Universität Berlin  
Institut für Mathematik  
Straße des 17. Juni 136  
10623 Berlin, Germany  
rolf.moehring@tu-berlin.de

### *ACM Classification 1998*

F.2 Analysis of Algorithms and Problem Complexity, G.1.6 Optimization, G.2.2 Graph Theory, G.2.3 Applications

## **ISBN 978-3-939897-00-2**

### *Published online and open access by*

Schloss Dagstuhl – Leibniz-Center for Informatics GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany.

### *Publication date*

July, 2006.

### *Bibliographic information published by the Deutsche Nationalbibliothek*

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

### *License*

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works license: <http://creativecommons.org/licenses/by-nc-nd/3.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the author's moral rights:

- Attribution: The work must be attributed to its authors.
- Noncommercial: The work may not be used for commercial purposes.
- No derivation: It is not allowed to alter or transform this work.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/OASlcs.ATMOS.2005.i

**ISBN 978-3-939897-00-2**

**ISSN 2190-6807**

**<http://www.dagstuhl.de/oasics>**

## OASlcs – OpenAccess Series in Informatics

OASlcs aims at a suitable publication venue to publish peer-reviewed collections of papers emerging from a scientific event. OASlcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

**ISSN 2190-6807**

**[www.dagstuhl.de/oasics](http://www.dagstuhl.de/oasics)**

# ATMOS Preface: Algorithmic Methods and Models for Optimization of Railways

Leo Kroon<sup>1,2</sup> and Rolf H. Möhring<sup>3</sup>,

<sup>1</sup> Erasmus University Rotterdam, Rotterdam School of Management  
P.O. Box 1738, 3000 DR, Rotterdam, The Netherlands

`lkroon@rsm.nl`

<sup>2</sup> Netherlands Railways, Department of Logistics  
P.O. Box 2025, 3500 HA, Utrecht, The Netherlands

`lkroon@rsm.nl`

<sup>3</sup> Technische Universität Berlin, Institut für Mathematik  
Straße des 17. Juni 136, 10623 Berlin, Germany

`rolf.moehring@tu-berlin.de`

This issue contains six papers that were presented in preliminary form at the 5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2005), held at Palma de Mallorca, Spain, October 7, 2005 in conjunction with ALGO 2005.

The authors of the papers in this volume were invited to submit extended versions of their ATMOS 2005 papers. All papers were accepted after a review process performed by members of the ATMOS 2005 Program Committee. These papers are representative of several areas of research within the scope of ATMOS: rolling stock circulation and engine assignment, station location, line planning, railway traffic scheduling and dispatching, transfer optimization within network design, and fast traffic information systems.

The paper “Analysis of the Parameters of Transfers in Rapid Transit Network Design” by R. García, A. Garzón-Astolfi, A. Marín, J. A. Mesa, and F. A. Ortega considers the rapid transit network design problem that consists in the location of train alignments and stations in an urban traffic context. For the first time, they incorporate into the location model the decisions about the transportation mode and the route to be chosen for urban trips. These decisions include transfers between train lines. The objective of the model is to maximize the number of expected users in the transit network taking limited budgets into consideration, in addition to location and allocation constraints. Furthermore, the transfer costs are considered in the generalized public costs when the users change lines. Some computational experience is included in the paper.

In their paper “Combinatorial Optimization Model for Railway Engine Assignment Problem”, T. Illés, M. Makai, and Zs. Vaik present an experimental study for the Hungarian State Railway Company (MÁV). The timetable of passenger trains of a region of Hungary is given, and engines (locomotives) must be assigned to each passenger train under some operational policies including maintenance and connection times for linking trains together. The goal is to minimize the number of engines used. The authors develop an integer programming model for the full problem and use a minimum cost flow algorithm for the problem without maintenance which reduces to a circulation problem.

The paper “Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms”, by J. Törnquist provides an overview of the research in railway scheduling and dispatching. A distinction is made between tactical scheduling, operational scheduling and re-scheduling. Tactical scheduling refers to master scheduling, whereas operational scheduling concerns scheduling at a later stage. Re-scheduling focuses on the re-planning of an existing timetable when deviations from it have occurred. 48 approaches published between 1973 and 2005 have been reviewed according to a framework that classifies them with respect to problem type, solution mechanism, and type of evaluation.

In their paper “Line Planning with Minimal Traveling Time”, A. Schöbel and S. Scholl deal with an important strategic element in the planning process of public transportation, viz. the development of a line concept, i.e., to find a set of paths for operating lines on them. So far, most of the models in the literature aim at minimizing the costs or maximizing the number of direct travelers. The authors present a new approach minimizing the travel times over all customers including penalties for the transfers needed. This approach maximizes the comfort of the passengers and makes the resulting timetable more reliable. Their approach is based on integer programming models and uses Dantzig-Wolfe decomposition for solving the LP-relaxation. Numerical results of real-world instances are presented.

The paper “Paying Less for Train Connections with MOTIS” by M. Müller-Hannemann and M. Schnee reports on the development of a multi-objective traffic information system (MOTIS) which finds all attractive train connections with respect to travel time, number of interchanges, and ticket costs. In contrast, most servers for timetable information as well as the theoretical literature on this subject focus only on travel time as the primary objective, and secondary objectives like the number of interchanges are treated only heuristically. Finding cheap train connections for long-distance traffic is algorithmically a hard task due to very complex tariff regulations. Several new tariff options have been developed in recent years, partly to react on the stronger competition with low-cost airline carriers. In such an environment, it becomes more and more important that search engines for travel connections are able to find special offers efficiently. The authors show in their paper by means of a case study how several of the most common tariff rules (including special offers) can be embedded into a general multi-objective search tool. Computational results show that a multi-objective search with a mixture of tariff rules can be done almost as fast as just with one regular tariff. For the train schedule of Germany, a query can be answered within 1.9s on average on a standard PC.

The paper by S. Mecke, A. Schöbel, and D. Wagner on “Station Location – Algorithms and Complexity” investigates the question to add stations to an existing geometric transportation network so that each of a given set of settlements is not too far from a station. The problem is known to be  $\mathcal{NP}$ -hard in general. However, special cases with certain properties have been shown to be efficiently solvable in theory and in practice, especially if the covering matrix has

(almost) consecutive ones property. In their paper the authors are narrowing the gap between intractable and efficiently solvable cases of the problem and present an approximation algorithm for cases with almost consecutive ones property.

We would like to thank the referees for their conscientious and timely work, and the editors of the Dagstuhl Seminar Proceedings for the opportunity to publish this special issue in DROPS.

**ATMOS 2005 — Abstract Collection**  
**Selected Papers from the 5th Workshop on Algorithmic  
Methods and Models for Optimization of Railways**

Leo G. Kroon<sup>1</sup> and Rolf H. Möhring<sup>2</sup>

<sup>1</sup> Erasmus University, NL

[l.kroon@fbk.eur.nl](mailto:l.kroon@fbk.eur.nl)

<sup>2</sup> TU Berlin, D

[moehring@math.tu-berlin.de](mailto:moehring@math.tu-berlin.de)

**Abstract.** This issue contains six papers that were presented in preliminary form at the 5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2005), held at Palma de Mallorca, Spain, October 7, 2005 in conjunction with ALGO 2005. These papers are representative of several areas of research within the scope of ATMOS: rolling stock circulation and engine assignment, station location, line planning, railway traffic scheduling and dispatching, transfer optimization within network design, and fast traffic information systems.

**Keywords.** Railway traffic, networks, algorithms, optimization

## Combinatorial Optimization Model for Railway Engine Assignment Problem

*Illés, Tibor; Makai, Márton; Vaik, Zsuzsanna*

This paper presents an experimental study for the Hungarian State Railway Company (MV). The engine assignment problem was solved at MV by their experts without using any explicit operations research tool. Furthermore, the operations research model was not known at the company. The goal of our project was to introduce and solve an operations research model for the engine assignment problem on real data sets. For the engine assignment problem we are using a combinatorial optimization model. At this stage of research the single type train that is pulled by a single type engine is modeled and solved for real data. There are two regions in Hungary where the methodology described in this paper can be used and MÁV started to use it regularly. There is a need to generalize the model for multiple type trains and multiple type engines.

*Keywords:* Engine assignment, circulation

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2006/662>

## Station Location – Complexity and Approximation

*Mecke, Steffen; Schöbel, Anita; Wagner, Dorothea*

We consider a geometric set covering problem. In its original form it consists of adding stations to an existing geometric transportation network so that each of a given set of settlements is not too far from a station. The problem is known to be NP-hard in general. However, special cases with certain properties have been shown to be efficiently solvable in theory and in practice, especially if the covering matrix has (almost) consecutive ones property. In this paper we are narrowing the gap between intractable and efficiently solvable cases of the problem. We also present an approximation algorithm for cases with almost consecutive ones property.

*Keywords:* Station Location, facility location, complexity, approximation

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2006/661>

## Line Planning with Minimal Traveling Time

*Schöbel, Anita; Scholl, Susanne*

An important strategic element in the planning process of public transportation is the development of a line concept, i.e. to find a set of paths for operating lines on them. So far, most of the models in the literature aim to minimize the costs or to maximize the number of direct travelers. In this paper we present a new approach minimizing the travel times over all customers including penalties for the transfers needed. This approach maximizes the comfort of the passengers and will make the resulting timetable more reliable. To tackle our problem we present integer programming models and suggest a solution approach using Dantzig-Wolfe decomposition for solving the LP-relaxation. Numerical results of real-world instances are presented.

*Keywords:* Line planning, real-world problem, integer programming, Dantzig-Wolfe decomposition

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2006/660>



## Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms

*Törnquist, Johanna*

This paper provides an overview of the research in railway scheduling and dispatching. A distinction is made between tactical scheduling, operational scheduling and re-scheduling. Tactical scheduling refers to master scheduling, whereas operational scheduling concerns scheduling at a later stage. Re-scheduling focuses on the re-planning of an existing timetable when deviations from it have occurred. 48 approaches published between 1973 and 2005 have been reviewed according to a framework that classifies them with respect to problem type, solution mechanism, and type of evaluation. 26 of the approaches support the representation of a railway network rather than a railway line, but the majority has been experimentally evaluated for traffic on a line. 94 % of the approaches have been subject to some kind of experimental evaluation, while approximately 4 % have been implemented. The solutions proposed vary from myopic, priority-based algorithms, to traditional operations research techniques and the application of agent technology.

*Keywords:* Decision support, railway traffic scheduling, railway traffic dispatching, overview

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2006/659>

## Analysis of the Parameters of Transfers in Rapid Transit Network Design

*García, Ricardo; Garzón-Astolfi, Armando; Marín, Angel; Mesa, Juan A.; Ortega, Francisco A.*

The rapid transit network design problem consists of the location of train alignments and stations in an urban traffic context. The originality of our study is to incorporate into the location model the decisions about the transportation mode and the route, to be chosen for urban trips. This paper proposes a new design model which includes transfers between train lines. The objective of the model is to maximize the number of expected users in the transit network taking limited budgets into consideration, in addition to location and allocation constraints. Furthermore, the transfer costs are considered in the generalized public costs when the users change lines. Waiting time to take the metro and walking time to transfer is included in the formulation of the costs. The analysis of transfer parameters is carried out using a test network. Some computational experience is included in the paper.

*Keywords:* Parameter analysis, rapid transit network design, trip choice in urban traffic

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2006/658>

## **Paying Less for Train Connections with MOTIS**

*Müller-Hannemann, Matthias; Schnee, Mathias*

Finding cheap train connections for long-distance traffic is algorithmically a hard task due to very complex tariff regulations. Several new tariff options have been developed in recent years, partly to react on the stronger competition with low-cost airline carriers. In such an environment, it becomes more and more important that search engines for travel connections are able to find special offers efficiently.

We have developed a multi-objective traffic information system (MOTIS) which finds all attractive train connections with respect to travel time, number of interchanges, and ticket costs. In contrast, most servers for timetable information as well as the theoretical literature on this subject focus only on travel time as the primary objective, and secondary objectives like the number of interchanges are treated only heuristically.

The purpose of this paper is to show by means of a case study how several of the most common tariff rules (including special offers) can be embedded into a general multi-objective search tool.

Computational results show that a multi-objective search with a mixture of tariff rules can be done almost as fast as just with one regular tariff. For the train schedule of Germany, a query can be answered within 1.9s on average on a standard PC.

*Keywords:* Timetable information system, multi-criteria optimization, shortest paths, fares, special offers, long-distance traffic

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2006/657>

# Analysis of the Parameters of Transfers in Rapid Transit Network Design

Ricardo García<sup>1</sup>, Armando Garzón-Astolfi<sup>2</sup>, Angel Marín<sup>3</sup>, Juan A. Mesa<sup>4</sup>, and  
Francisco A. Ortega<sup>5</sup>

<sup>1</sup> Departamento de Matemática. Universidad de Castilla La Mancha. Ciudad Real,  
Spain

`ricardo.garcia@uclm.es`

<sup>2</sup> Departamento de Matemática Aplicada I. Universidad de Sevilla. Sevilla, Spain

`agastolfi77@yahoo.es`

<sup>3</sup> Departamento de Matemática Aplicada y Estadística. Universidad Politécnica de  
Madrid. Madrid, Spain

`amarin@dmae.upm.es`

<sup>4</sup> Departamento de Matemática Aplicada II. Universidad de Sevilla. Sevilla, Spain

`jmesa@us.es`

<sup>5</sup> Departamento de Matemática Aplicada I. Universidad de Sevilla. Sevilla, Spain

`riejos@us.es`

**Abstract.** The rapid transit network design problem consists of the location of train alignments and stations in an urban traffic context. The originality of our study is to incorporate into the location model the decisions about the transportation mode and the route, to be chosen for urban trips. This paper proposes a new design model which includes transfers between train lines. The objective of the model is to maximize the number of expected users in the transit network taking limited budgets into consideration, in addition to location and allocation constraints. Furthermore, the transfer costs are considered in the generalized public costs when the users change lines. Waiting time to take the metro and walking time to transfer is included in the formulation of the costs. The analysis of transfer parameters is carried out using a test network. Some computational experience is included in the paper.

## 1 Introduction

Increasing mobility caused by the growth of cities is the reason why new lines of rail transit have been constructed. A crucial part of the network design process consists of the location of stations and alignments between them. In the paper [1] an approach to the network design problem, based on the previous selection of the key nodes (those providing a high number of trips) was described. Therefore, the transit network is defined on the edges which connect the key node set.

The transit system involves the node and edge locations at upper level and considers the user traffic behavior at lower level. At upper level the main factor

to consider is maximum coverage of the demand using public mode, taking constraints of our model and the budget constraints into account. Traffic demand leads to alternative configurations of networks, comparing private trip cost with public trip cost, the latter depending on previous location decisions.

Customers choose the most convenient routes and modes in order to carry out their trips. A decisive factor for attracting passengers to the public mode is to offer direct trips without transfers. Transferring is annoying and it is undesirable for customers. In our approach, transfers are explicitly considered in accordance with the central role played by user mentality.

The previous references to the rapid transit network design (RTND), [2], [3], [4], [5], [6], [7] consider travel cost as the time spent in travelling without taking into account any transfer cost.

The layout of the paper is as follows. In Section 2 the RTND is formulated including transfers between lines. In Section 3 the transfer costs are consistently introduced into the previous model. In Section 4 a transfer parametric analysis is implemented. The paper finalizes with conclusions and further research.

## 2 Rapid Transit Network Design Model with transfers

We assume that a set  $N = \{i : i = 1, \dots, I\}$  of potential locations for the stations is given. Let  $E$  be the set of feasible edges linking the potential stations. Thus, we have an undirected graph  $\mathcal{G} = (N, E)$  from which the transit network is going to be designed. For each node  $i \in N$ , let  $N(i)$  denote the set of nodes adjacent to it. A matrix of distances  $D = (d_{ij})$  between pairs of points of  $N$  is also known. The travel patterns are given by the origin-destination matrix  $G = (g_p)$ , where  $g_p$  is the demand of the pair  $p = (q, r) \in P$  and  $P$  is the set of pairs of demand.

The cost structure is as follows. Let  $c_{ij}$  and  $c_i$  denote the cost of constructing a section of an alignment on edge  $ij$  and that of constructing a station at node  $i$ , respectively. According to the available budget the length of the public lines will be bounded; for this purpose, there are bounds  $length_{\min}^l, length_{\max}^l, l = 1, \dots, L$ , on the length of line  $l$  and bounds  $Tlength_{\min}$  and  $Tlength_{\max}$  on the total length of the lines of the network.

In regard to the demand, let  $uc_p^{PUB}$  be the user's generalized cost of travelling within the constructed transit network and let  $uc_p^{PRIV}$  be the user's generalized cost using the private mode. Observe that this cost does not depend on the final topology of the transit network.

The problem we are dealing with consists of choosing a number of lines  $\mathcal{L} = \{l : l = 1, \dots, L\}$  covering as much as possible travel demand between the points of  $N$ , subject to the line length constraints and other constraints.

The decision variables are defined as follows:

- The station selection variables:  $y_i^l = 1$ , if station  $i$  of line  $l$  is constructed; and  $y_i^l = 0$  otherwise.
- The edge selection variables:  $x_{ij}^l = 1$ , if edge  $ij$  of the line  $l$  is constructed; and  $x_{ij}^l = 0$  otherwise.

- The following variables are defined for indicating whether or not the demand would use the transit network in case edge  $ij$  is selected. Specifically,  $u_{ij}^p = 1$ , if the demand of pair  $p$  would use edge  $ij$  in the public network,  $u_{ij}^p = 0$  otherwise.

- Mode choice variables:  $z_p = 1$  if the generalized cost for the demand of pair  $p$  within the public network  $\mathcal{A}$  is less than that of the private mode;  $z_p = 0$  otherwise.

- Flow routing variables:  $w_{ij}^{pl} = 1$  if the demand  $p$  traverses the edge  $ij$  using line  $l$ , 0 otherwise.

- Transfer variables:  $v_i^{pl} = 1$  if demand  $p$  transfers to line  $l$  in station  $i$ .

The RTND model with transfers is stated in the following terms:

- Objective function: Trip coverage

$$\max \sum_{p \in P} g_p z_p$$

- Length constraints

$$\sum_{ij \in E} d_{ij} x_{ij}^l \in [\text{length}_{\min}^l, \text{length}_{\max}^l], \quad l \in \mathcal{L} \quad (1)$$

$$\sum_{l \in \mathcal{L}} \sum_{ij \in E} d_{ij} x_{ij}^l \in [\text{Tlength}_{\min}, \text{Tlength}_{\max}] \quad (2)$$

- Alignment location constraints

$$\sum_{j \in N(o_l)} x_{o_l j}^l = 1, \quad l \in \mathcal{L} \quad (3)$$

$$\sum_{i \in N(d_l)} x_{i d_l}^l = 1, \quad l \in \mathcal{L} \quad (4)$$

$$y_{o_l}^l = y_{d_l}^l = 1, \quad l \in \mathcal{L} \quad (5)$$

$$\sum_{j \in N(i)} x_{ij}^l = 2y_i^l, \quad i \in N \setminus \{o_l, d_l\}, \quad l \in \mathcal{L} \quad (6)$$

$$x_{ij}^l = x_{ji}^l \quad ij \in E, \quad l \in \mathcal{L} \quad (7)$$

- Routing demand constraints

$$\sum_{j \in N(q)} u_{qj}^p = 1, \quad p = (q, r) \in P \quad (8)$$

$$\sum_{i \in N(q)} u_{iq}^p = 0, \quad p = (q, r) \in P \quad (9)$$

$$\sum_{i \in N(r)} u_{ir}^p = 1, \quad p = (q, r) \in P \quad (10)$$

$$\sum_{j \in N(r)} u_{rj}^p = 0, \quad p = (q, r) \in P \quad (11)$$

$$\sum_{i \in N(j)} u_{ij}^p - \sum_{k \in N(j)} u_{jk}^p = 0, \quad j \in N \setminus \{q, r\}, p = (q, r) \in P \quad (12)$$

- Location-Allocation constraints

$$u_{ij}^p + z_p - 1 \leq \sum_{l=1}^L x_{ij}^l, \quad ij \in E, \quad p = (q, r) \in P \quad (13)$$

- Splitting demand constraints

$$uc_p^{PUB} - uc_p^{PRIV} - M(1 - z_p) \leq 0 \quad p = (q, r) \in P \quad (14)$$

where  $uc_p^{PRI}$  is a data and  $uc_p^{PUB}$  will be defined in the next subsection.  $M$  is an enough big number.

• Transfer constraints

$$w_{ij}^{pl} \leq x_{ij}^l, \quad ij \in E, p = (q, r) \in P, l = 1, \dots, L \quad (15)$$

$$u_{ij}^p + z_p - 1 \leq \sum_{l=1}^L w_{ij}^{pl}, \quad p = (q, r) \in P, \quad ij \in E \quad (16)$$

$$u_{ij}^p - z_p + 1 \geq \sum_{l=1}^L w_{ij}^{pl}, \quad p = (q, r) \in P, \quad ij \in E \quad (17)$$

$$\sum_{ij \in E, l \in \mathcal{L}} w_{ij}^{pl} \leq M z_p, \quad p = (q, r) \in P \quad (18)$$

$$\sum_{j \in N(i)} w_{ij}^{pl} - \sum_{j \in N(i)} w_{ji}^{pl} \geq 2v_i^{pl} - 1, \quad i \in N \setminus \{r\},$$

$$p = (q, r) \in P, \quad l \in \mathcal{L} \quad (19)$$

$$\sum_{j \in N(i)} w_{ij}^{pl} - \sum_{j \in N(i)} w_{ji}^{pl} \leq 2v_i^{pl}, \quad i \in N \setminus \{r\},$$

$$p = (q, r) \in P, \quad l \in \mathcal{L} \quad (20)$$

$$x_{ij}^l, y_i^l, u_{ij}^p, z_p, w_{ij}^{pl}, v_i^{pl} \in \{0, 1\}.$$

Constraints (1) and (2) impose lower and upper bounds on the individual and total line lengths.

Constraints (3) and (4) guarantee that each line starts and ends at its specified origin and destination. Constraints (5) ensure that all the origins and destinations belong to  $\mathcal{A}$ . Constraints (6) impose that each line must be a path between the corresponding origin and destination.

Constraints (7), (8) and (9) guarantee demand conservation. Constraints (10) and (11) were introduced to ensure that identity  $z_p = 1$  implies that the demand of the pair  $p$  goes through the public network and  $z_p = 0$  if it uses the private network. Constraints (12) guarantee that the demand is routed on an edge only if this edge belongs to the public system.

Constraints (13) ensure that the demand is routed on an edge if it has been previously constructed. Constraints (14) force the demand to be assigned to public mode if public cost is less than private cost.

Note that this formulation does not include the common sub-tour elimination constraints. Therefore, when a solution contains a cycle, additional constraints can be imposed in order to avoid the presence of cycles in the solution network. Note that well developed networks (e.g. Paris, London, Moscow, Tokyo and Madrid) often contain circular lines. It has also been proved by Laporte, Mesa and Ortega (1997, [?]) that the inclusion of a circle line increases the effectiveness of the network and thus the inclusion of cycles can be interesting.

Transfer constraints (15) permit the flow to use edge  $ij$  of line  $l$  only if edge  $ij$  have already been established for public mode. Constraints (16) guarantee that if the flow of edge  $ij$  is carried through any line, the public mode and its flow at edge  $ij$  must already be chosen. Constraints (17) establish that any flow for the pair  $p$  can use any edge of the public network. Constraints (18) impose that if a transfer is made at node  $i$  then the flow leaving from this node is bigger than the flow coming in. Constraints (19) and (20) impose that if the flow leaving out is less than the flow coming in at a node  $i$ , then a transfer is made at this node.

### 3 Transit costs in transfers

Time spent in transference between lines is characterized by certain cost parameters. We have studied different values of the parameters which define transfer cost in order to conclude how they influence the RTND solution.

The paper is focused on the study of the increase in the cost using the public mode, assuming that users may transfer from one line to another. The transfer process has been modeled at RTND through constraints (15) to (20), but now we will give details about how the public mode cost is influenced by transfers.

The public cost for each demand  $uc_p^{PUB}$  is the sum of two terms: one term relative to *travel time* spent moving in the transit vehicle on the rapid transit network and another term related to the *transfer time* spent in transferring from one line to another.

The first term has been considered in all the references, and it is computed by the traveling time, which is calculated by the sum of the travel distance divided by the average velocity of the lines  $\hat{\lambda}$ . Hence, non-transfer public costs (NTPUB) are defined as follows:

$$uc_p^{NTPUB} = \frac{1}{\hat{\lambda}} \sum_{ij \in E} d_{ij} u_{ij}^p, p = (q, r) \in P \quad (16)$$

In the concept of vehicle velocity, we include vehicle moving time and the time spent at the station required to permit boarding and alighting of the passengers. These values are average values for standard lines. The line velocity may be considered by taking an average value of 20 kilometers per hour. In that case, the average velocity,  $\hat{\lambda}$ , is 1/3 kilometers per minute.

As was pointed out below, the demand is very sensible to transfer time. Thus, the transfer cost of each demand and station is considered as the sum of two terms: 1) one value fixed for each station  $i$ ,  $uc_i$ , that represents the average walking time between line platforms and 2) another value on the waiting time for taking the next train of a different line.

In our approach we assume  $uc_i$  as a parameter which depends on the travel time spent for any demand that transfers at station  $i$  from the previous board platform line up to the board platform of the next line. This value is given in minutes and it can model the cumulative sum of walking times between platforms and the annoyance associated to the transfers, which depends on the station characteristics. This value would be in average about 3 to 5 minutes.

The transfer waiting cost depends on the frequency of the line of the train to take in order to continue the trip towards the destination. The line frequency is considered fixed and represents a parameter of the model. Considering that the planning period is of 1 hour, and that the time cost will be given in minutes, the wait time is assumed equal to one divided by twice the frequency of the line,  $2 * f_l$ . So if the average frequency is of 6 vehicle per hour then the average waiting time is of 5 minutes. For a line with double frequency, 12 vehicles per hour, the waiting time will be of 2.5 minutes.

With these considerations, the public cost expression with transfer of a pair  $p$  is:

- UserPublic cost

$$uc_p^{PUB} = \frac{1}{\lambda} \sum_{ij \in E} d_{ij} u_{ij}^p + \sum_{i \in N \setminus \{r\}, l \in L_i} \left( uc_i + \frac{1}{2l_i} \right) v_i^{pl} \quad p = (q, r) \in P \quad (17)$$

where  $L_i$  is the set of lines that use the node  $i$ .

In this context, for a distance of 3 kilometers and for an average velocity of 20 km/h, the travel time is 9 minutes. If the fixed transfer time is of 3 minutes more, and the wait time of 5 minutes, the total travel time is of 17 minutes.

#### 4 Transfer parametric analysis

The previous model has been tested on the 6-node network shown in Figure 1, where each node  $i$  has an associated cost (variable  $c_i$  in the model) and each edge has been weighted by means of a pair  $(c_{ij}, d_{ij})$  its components respectively representing the cost of constructing edge  $ij$  and the generalized public cost of using edge  $ij$  to connect both nodes.

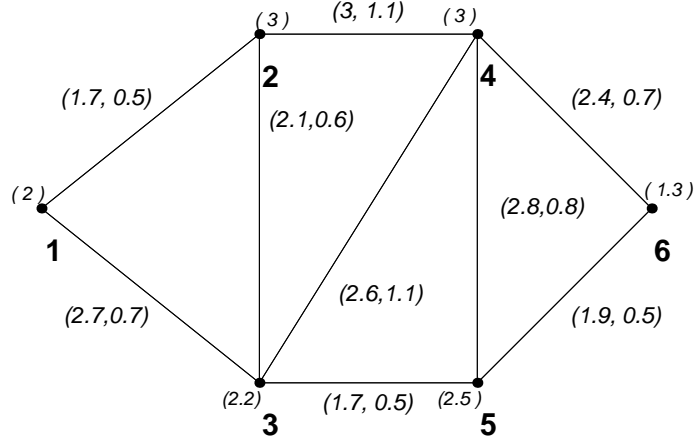


Fig. 1. Network considered.

The origin-destination demands  $g_p, p = (q, r) \in P$  and the private cost  $uc_p^{PRIV}$  for each demand pair  $p \in P$  are given in the following matrices:

$$G = \begin{pmatrix} - & 9 & 26 & 19 & 13 & 12 \\ 11 & - & 14 & 26 & 7 & 18 \\ 30 & 19 & - & 30 & 24 & 8 \\ 21 & 9 & 11 & - & 22 & 16 \\ 14 & 14 & 8 & 9 & - & 20 \\ 26 & 1 & 22 & 24 & 13 & - \end{pmatrix};$$



$$uc^{PRIV} = \begin{pmatrix} - & 1.6 & 0.8 & 2 & 2.6 & 2.5 \\ 2 & - & 0.9 & 1.2 & 1.5 & 2.5 \\ 1.5 & 1.4 & - & 1.3 & 0.9 & 2 \\ 1.9 & 2 & 1.9 & - & 1.8 & 2 \\ 3 & 1.5 & 2 & 2 & - & 1.5 \\ 2.1 & 2.7 & 2.2 & 1 & 1.5 & - \end{pmatrix}$$

Bounds on maximum and minimum lengths of the total public network have been established at 4 and 2, respectively. The solutions presented in the following scenarios have been obtained by using CPLEX 8.0 on a Pentium IV laptop computer at 2.56 Mghz, provided by 1 Gbyte of RAM.

Although different parametric analyses can be carried out for the model, we have emphasized the sensitivity of the solutions with respect to the dispersion of values associated to transfer costs.

Applying a calibration process to the public costs of the edges, the centralized value  $uc_{ij}^{PUB} = 0.685$  was obtained. Taking this result into account, the value 0.75 was considered a central value. Different decrements and increments of size 0.25 have been used to grade the dispersion with respect to the average 0.75. Therefore, a variation of modulus 0.25 will be identified as low dispersion and if the difference with respect to the central value is 0.50, then we will say that the dispersion is high.

#### 4.1 The effect of varying the transfer cost between lines on the configuration of the optimum network

An average speed of 20 km/h has been assumed for transit on all lines. The cost for transferring was established at 3 min. Private cost was six times higher than public cost; this proportion stimulates a desirable competition between transportation modes. The congestion level has been assumed equal to 1.5 remaining outside of this first parametric analysis.

The following tables show optimum configurations of the rapid transit networks for the constraints indicated; namely, the number of lines that will compose the final network and the range for the maximum length of each line. Each table deals with a different scenario:

Table 1 : All transfer costs coincide in a central value (0.75).

Table 2 : Low dispersion for the distribution of transfer costs in a 2-line network (0.5 and 1).

Table 3 : Low dispersion for the distribution of transfer costs in a 3-line network (0.5, 0.75 and 1).

Table 4 : High dispersion for the distribution of transfer costs in a 2-line network (0.25 and 1.25).

Table 5 : High dispersion for the distribution of transfer costs in a 3-line network (0.25, 0.75 and 1.25).

The analysis of the results lead us to the following conclusions:

Line Num.	Length Range	Optimum Lines	Obj. Func.	Line Lengths	Exec. Time(s)
2	[0.5, 2.5]	$n_1-n_2-n_3-n_4$	444	2.2	57.7
	[0.5, 2.5]	$n_3-n_5-n_6-n_4$		1.7	
3	[0.5, 1.5]	$n_3-n_5-n_6$	404	1	1666.72
	[0.5, 1.5]	$n_1-n_3-n_2$		1.3	
	[0.5, 1.5]	$n_3-n_4$		1.1	

**Table 1.** All transfer coefficients are equal to 0.75

Line Num.	Length Range	Optimum Lines	Obj. Func.	Line Lengths	Exec. Time(s)
2	[0.5, 2.5]	$n_2-n_1-n_3-n_5-n_4$	456	2.5	95.23
	[0.5, 2.5]	$n_5-n_6-n_4$		1.2	
2	[0.5, 3]	$n_1-n_3-n_5-n_4-n_6$	470	2.7	50.5
	[0.5, 2.5]	$n_2-n_3$		0.6	
2	[0.5, 3.5]	$n_2-n_1-n_3-n_5-n_4-n_6$	470	3.2	61.2
	[0.5, 2.5]	$n_2-n_3$		0.6	

**Table 2.** Low dispersion of transfer coefficients (2 lines)

1. A wide range for the line lengths produces better values of the objective function.
2. When the dispersion of transfer cost increases (i.e., waiting time is very heterogeneous for all lines), the flow shift is higher. Subsequently, the best results for a 2-line network are obtained when the dispersion is low and, on the other hand, the objective function reaches higher values for the 3-line network when the dispersion is high. This fact does not alter although the range of the line lengths varies.
3. The required execution time descends when the dispersion of transfer costs and the range for line lengths increase, which behaves even better on the 3-line network, as the tables show.

#### 4.2 The effect of varying the train frequency of the lines on the network configuration

The following tables show optimum configurations of the rapid transit networks for the train frequency indicated.

Table 6 : Low frequency of the train flow on the 2-line network.

Table 7 : High frequency of the train flow on the 2-line network.

Line Num.	Length Range	Optimum Lines	Obj. Func.	Line Lengths	Exec. Time(s)
3	[0.5, 1.5]	$n_1-n_3-n_2$	425	1.3	1252.69
	[0.5, 1.5]	$n_3-n_5-n_4$		1.3	
	[0.5, 1.5]	$n_3-n_5-n_6$		1	
3	[0.5, 2]	$n_1-n_3-n_5-n_4$	470	2	248.61
	[0.5, 2]	$n_2-n_3$		1.2	
	[0.5, 1.5]	$n_5-n_6-n_4$		0.6	
3	[0.5, 2.5]	$n_1-n_3-n_5-n_4$	470	2	248.61
	[0.5, 2.5]	$n_5-n_6-n_4$		1.2	
	[0.5, 1.5]	$n_2-n_3$		0.6	
3	[0.5, 3]	$n_1-n_3-n_5-n_4-n_6$	470	2.7	173.64
	[0.5, 2.5]	$n_2-n_3$		0.6	
	[0.5, 1.5]	$n_1-n_2$		0.5	

**Table 3.** Low dispersion of transfer coefficients (3 lines)

Line Num.	Length Range	Optimum Lines	Obj. Func.	Line Lengths	Exec. Time(s)
2	[0.5, 2.5]	$n_1-n_3-n_5-n_6-n_4$	447	2.4	64.63
	[0.5, 2.5]	$n_2-n_3$		0.6	
2	[0.5, 3]	$n_2-n_1-n_3-n_5-n_6-n_4$	456	2.9	46.81
	[0.5, 2.5]	$n_2-n_4$		1.1	
2	[0.5, 3.5]	$n_1-n_3-n_5-n_4-n_6$	456	2.7	38.81
	[0.5, 2.5]	$n_2-n_3$		0.6	

**Table 4.** High dispersion of transfer coefficients (2 lines)

### 4.3 The effect of varying the train frequency on the remainder parameter set

Table 8 contains the results obtained for a 3-line network whose congestion level is 5, range for line lengths is  $[0, 2.5]$  and total network length is less than 5. Moreover, the maximum value for the objective function is 496 (the total demand) and  $\lambda = 0.33$ .

The main aim of this subsection consists of showing how the solution varies when the line frequency increases. For this purpose, the first column of Table 8 collects an increasing sequence of values for the frequency while the remainder columns show the associated values in relation to the objective function, the total length and the execution time.

As can be noted, when frequency increases, the waiting time for riderships decreases and, subsequently, the use of public network increases (as shows the sequence of values corresponding to the objective function).

In relation to the total length of the lines, Table 8 shows, from a frequency greater than 6, how the system does not require a minimization of the location

Line Num.	Length Range	Optimum Lines	Obj. Func.	Line Lengths	Exec. Time(s)
3	[0.5, 1.5]	$n_1-n_3-n_2$	446	1.3	243.13
	[0.5, 1.5]	$n_3-n_5-n_4$		1.3	
	[0.5, 1.5]	$n_3-n_5-n_6$		1	
3	[0.5, 2]	$n_2-n_1-n_3-n_5$	456	1.7	432.17
	[0.5, 2]	$n_3-n_4-n_6$		1.8	
	[0.5, 1.5]	$n_5-n_6$		0.5	
3	[0.5, 2.5]	$n_1-n_3-n_5-n_6-n_4$	470	2.4	194.06
	[0.5, 2.5]	$n_2-n_3$		0.6	
	[0.5, 1.5]	$n_3-n_5$		0.5	
3	[0.5, 3]	$n_5-n_3-n_2-n_4-n_6$	470	2.9	142.63
	[0.5, 2.5]	$n_1-n_2$		0.5	
	[0.5, 1.5]	$n_5-n_6$		0.5	

**Table 5.** High dispersion of transfer coefficients (3 lines)

Congestion Level	Optimum Lines	Obj. Func.	Line Costs	Exec. Time(s)
1.2	$n_1-n_2-n_3-n_5-n_6-n_4$	331	23.8	494.47
	$n_5-n_6$		5.7	
1.5	$n_1-n_2-n_3-n_5-n_6-n_4$	444	23.8	97.76
	$n_5-n_6$		5.7	
2.2	$n_2-n_3-n_5-n_6-n_4$	496	20.1	22.99
	$n_1-n_3$		6.9	

**Table 6.** Frequency interval = (12 trains per hour, 6 trains per hour).

cost and then, takes advantage of all available resources although the improvement in terms of the objective function is minimum.

Calculation times considerably increase from frequencies greater than 10, due to the options of finding efficient routes inside the public network increase. Subsequently, since the size of possible solutions is greater, the time for exploring will become higher.

## 5 Comparative tests

This section is devoted to the comparison between results obtained in presence/absence of transfers. The general context for parameter values (network with 3 lines at most, congestion factor equals to 5 and range for total length equals to  $[0, 5]$ ) remains.

Congestion Level	Optimum Lines	Obj. Func.	Line Costs	Exec. Time(s)
1.2	$n_1-n_2-n_3-n_5-n_6-n_4$ $n_5-n_6$	414	23.8 5.7	253.14
1.5	$n_2-n_3-n_5-n_4-n_6$ $n_1-n_3$	470	21 6.9	297.95
2.2	$n_1-n_2-n_3-n_5$ $n_2-n_4-n_6$	496	15.2 12.7	0.63

**Table 7.** Frequency interval = (20 trains per hour, 10 trains per hour)

Frequency	Obj. Function	Total length	Exec. Time
4	14	4.7	3.51
5	27	3.3	4.87
6	47	4.9	11.63
7	115	4.5	9.95
8	132	4.8	3.75
9	172	4.8	9.4
10	174	4.5	36.52
12	207	4.8	262.07
14	282	4.8	152.81
16	282	4.8	282.51
18	301	4.8	311.2
20	301	4.6	297.96
30	341	4.9	336.03

**Table 8.** Influence of line frequency on the other parameters

### 5.1 The effect of varying the maximum lengths of the lines

Assuming a model without transfers where the speed in the public mode of transportation is 20 km/h, frequency is 10 trains per hour for all lines and an access time for boarding equals to 3 minutes, Tables 9 and 10 respectively show the optimum configurations, the objective function values, the line lengths and the execution times for the model with and without transfers.

As can be observed, when the line capacity in the model with transfers increases, the portion of demand which uses the public network suffers an increment but limited by a maximum value (174).

For the case without transfers, this upper boundary does not appear due to all the existing demand chooses the public mode of transportation, once the capacity of the lines can take values greater.

Computation time is higher in the model with transfers as consequence of its inherent complexity.

Max. Line Length	Lines	Obj. Function	Line Length	Exec. Time
[0.5,1]	$n_3-n_5-n_6$	89	1	9.1
	$n_4-n_5$		0.8	
	$n_1-n_2$		0.5	
[0.5,1.5]	$n_1-n_3-n_5$	112	1.2	276.2
	$n_3-n_5-n_6$		1.0	
	$n_4-n_5-n_6$		1.5	
[0.5,2]	$n_1-n_3-n_5-n_6$	153	1.7	271.69
	$n_5-n_4-n_6$		1.5	
	$n_1-n_2-n_4$		1.6	
[0.5,3]	$n_1-n_2-n_3-n_5-n_6$	174	2.1	36.59
	$n_4-n_5$		0.8	
	$n_3-n_4$		1.1	
[0.5,4]	$n_1-n_2-n_3-n_5-n_6$	174	2.1	22.23
	$n_4-n_5-n_6$		1.3	
	$n_3-n_4$		1.1	

**Table 9.** Model with Transfers

## 5.2 The effect of varying the maximum length of the network

The constraint of maximum length for the total network gives rise to different configurations (see Tables 11 and 12) in relation to the line number whose maximum was assumed 3. Results obtained suggest similar conclusions to the obtained in the previous test.

## 5.3 The effect of varying the congestion factor

In this section, models with and without transfers are compared from the point of view of the congestion influence. Both settings consider networks composed of three lines at maximum.

The values of the model parameters are the same: an average speed of 20 km/h assumed for the transit on all lines, cost for transferring 3 minutes and frequency of 10 trains per hour. Line lengths belong to range [0.5, 2.5] and total length of the network is in interval the [0, 4].

Table 13 states that when the congestion in the private mode with transfers increases then a higher portion of the demand uses the public network. Specifically, the maximum possible demand (496) is reached for a congestion factor equals to 15 and a total length of the network close to 4.

The model without transfers (Table 14) has a similar behaviour but the increasing in the preference of using the public mode is more significative.

Max. Line Length	Lines	Obj. Function	Line Length	Exec. Time
[0.5,1]	$n_3-n_5-n_6$	368	1.0	3.73
	$n_1-n_3$		0.7	
	$n_4-n_6$		0.7	
[0.5,1.5]	$n_4-n_5-n_6$	496	1.3	2.084
	$n_1-n_3-n_5$		1.2	
	$n_2-n_3$		0.6	
[0.5,2]	$n_2-n_1-n_3-n_5$	496	1.7	0.24
	$n_4-n_6-n_5$		1.2	
	$n_2-n_3-n_4$		1.7	
[0.5,3]	$n_1-n_3-n_4-n_6$	496	2.5	0.17
	$n_2-n_3-n_5-n_6$		1.6	
[0.5,4]	$n_1-n_3-n_4-n_6$	496	2.5	0.19
	$n_2-n_3-n_5-n_6$		1.6	

**Table 10.** Model without Transfers

Max Length	Lines	Obj. Function	Line Length	Exec. Time
[0,1]	$n_3-n_5-n_6$	38	1	3.78
[1,2]	$n_1-n_3-n_5-n_6$	77	1.7	64.05
[1,3]	$n_1-n_2-n_3-n_5-n_6$	147	2.1	271.69
	$n_4-n_5$		0.8	
[1,4]	$n_1-n_2-n_3-n_5-n_6$	163	2.1	75
	$n_4-n_5-n_6$		1.3	
	$n_1-n_2$		0.5	
[1,5]	$n_1-n_2-n_3-n_5-n_6$	174	2.1	29.36
	$n_3-n_4-n_5-n_6$		2.4	
	$n_3-n_5$		0.5	

**Table 11.** Model with Transfers

## 6 Conclusions and further research

The influence of transfer cost parameters have been studied for a test network in the context of the Rapid Transit Network Design problem. It has been established that an adequate penalization of the transfers has a big influence on travel behavior demands. Therefore, the public cost variable is sensitive to the transfer cost, and this variable is basic for network design decisions.

Another parameter analysis carried out in the paper deals with the influence of the line frequency, which has been assumed as a given fixed parameter which depends on design variables. Specifically, the line frequency is obtained as a function of the number of stations and the length of the lines. Again, it has been shown that these design variables have a direct influence on the time spent by a user moving along the lines. A more profound parametric analysis is our next

Max Length	Lines	Obj. Function	Line Length	Exec. Time
[0,1]	$n_3-n_5-n_6$	95	1	3.86
[1,2]	$n_1-n_3-n_5-n_4$	227	2	17.52
[1,3]	$n_4-n_5-n_6$	496	1.3	0.14
	$n_2-n_3-n_5$		1.1	
	$n_1-n_2$		0.5	
[1,4]	$n_5-n_3-n_4-n_6$	496	1.3	0.06
	$n_1-n_3-n_2$		2.3	
[1,5]	$n_4-n_3-n_5-n_6-n_4$	496	2.8	0.06
	$n_1-n_3-n_2$		1.3	

**Table 12.** Model without Transfers

Congestion Factor	Lines	Obj. Function	Line Length	Exec. Time
5	$n_1-n_2-n_3-n_5-n_6$	163	2.1	46.99
	$n_1-n_2$		0.5	
	$n_4-n_5-n_6$		1.3	
7	$n_1-n_3-n_5-n_6-n_4$	322	2.4	183.18
	$n_1-n_2$		0.5	
	$n_1-n_2-n_3$		1.1	
10	$n_1-n_2-n_3-n_5-n_4$	470	2.4	36.68
	$n_4-n_6-n_5$		1.2	
15	$n_4-n_3-n_5-n_6$	496	2.1	19.91
	$n_1-n_2-n_3$		1.1	
	$n_1-n_2$		0.5	

**Table 13.** Model with Transfers

objective in order to broaden the research thus gaining more insight into the rapid transit network design problem.

### Acknowledgements

This work has been supported in part by a grant from the Spanish Research Projects 2003/1360 (Ministerio de Fomento) and BFM2003-04062 (Ministerio de Ciencia y Tecnología).

### References

1. Laporte, G., Marín, A., Mesa, J.A., Ortega, F.A.: An Integrated Methodology for the Rapid Transit Network Design Problem. Presented to ATMOS'04 (Bergen, Norway) (2004) and accepted to be published in Lecture Notes in Computer Science
2. Laporte, G., Mesa, J.A., Ortega, F.A.: Assessing the Efficiency of Rapid Transit Configurations. *Top* **5** (1997) 95–104.



Congestion Factor	Lines	Obj. Function	Line Length	Exec. Time
5	$n_1-n_3-n_4-n_6$	496	2.5	0.11
	$n_3-n_5$		0.5	
	$n_2-n_3$		0.6	
7	$n_1-n_3-n_4-n_6$	496	2.4	0.24
	$n_3-n_5$		0.5	
	$n_2-n_3$		1.1	
10	$n_1-n_3-n_5-n_4-n_6$	496	2.4	1.24
	$n_1-n_2-n_4$		1.6	
15	$n_1-n_2-n_4-n_6$	496	2.3	0.66
	$n_4-n_3-n_5$		1.6	

**Table 14.** Model without Transfers

3. Bruno, G., Ghianini, G., Improta, G.: A multi-modal approach to the location of a rapid transit line. *European Journal of Operational Research* **104** (1998) 321–332.
4. García, R., Marín, A.: Urban multimodal interchange design methodology. In: M. Pursula and J. Niittymäki (eds.): *Mathematical Methods on Optimization in Transportation Systems*. Applied Optimization, Vol. 48. Kluwer Academic Publishers, Dordrecht (2001) 49–79.
5. García, R., Marín, A.: Parking capacity and pricing in park’n ride trips: A continuous equilibrium network design problem. *Annals of Operations Research* **116** (2002) 153–178.
6. Laporte, G., Mesa, J.A., Ortega, F.A.: Locating Stations on Rapid Transit Lines. *Computers & Operations Research* **29** (2002) 741–759.
7. Laporte, G., Mesa, J.A., Ortega, F.A., Sevillano, I.: Maximizing Trip Coverage in the Location of a Single Rapid Transit Alignment. *Annals of Operations Research* **136** (2005) 49–63.

# Combinatorial Optimization Model for Railway Engine Assignment Problem

Tibor Illés<sup>1</sup>, Márton Makai<sup>2</sup>, and Zsuzsanna Vaik<sup>3</sup> \*

<sup>1</sup> Eötvös Loránd University of Sciences, Department of Operations Research,  
H-1117, Budapest, Pázmány Péter sétány 1/C, Hungary

[illes@math.elte.hu](mailto:illes@math.elte.hu)

<sup>2</sup> Eötvös Loránd University of Sciences, Department of Operations Research,  
H-1117, Budapest, Pázmány Péter sétány 1/C, Hungary

[marci@cs.elte.hu](mailto:marci@cs.elte.hu)

<sup>3</sup> Szent István University, Ybl Miklós Architecture College, Department of  
Mathematics

H-1146, Budapest, Thököly út 74, Hungary

[zsuzska@cs.elte.hu](mailto:zsuzska@cs.elte.hu)

**Abstract.** This paper presents an experimental study for the Hungarian State Railway Company (MÁV). The engine assignment problem was solved at MÁV by their experts without using any explicit operations research tool. Furthermore, the operations research model was not known at the company. The goal of our project was to introduce and solve an operations research model for the engine assignment problem on real data sets. For the engine assignment problem we are using a combinatorial optimization model. At this stage of research the single type train that is pulled by a single type engine is modeled and solved for real data. There are two regions in Hungary where the methodology described in this paper can be used and MÁV started to use it regularly. There is a need to generalize the model for multiple type trains and multiple type engines.

**Keywords.** Engine assignment, circulation

## 1 Introduction

The area of railway operation involves a lot of deep optimization problems. The following real-life optimization problem was addressed by the Hungarian State Railway Co. Pl., (MÁV). The timetable of passenger trains of a region of Hungary is given, and engines (locomotives) should be assigned to each passenger train under some operational policies. The timetable contains all the necessary data related to the train like departure and arrival stations and times, railway lines where the trains are operated, etc. In the given region exactly one type of

---

\* Research is supported by OTKA grants T 037547 and TS 049788, by European MCRTN Adonet, Contract Grant No. 504438 and by the Egerváry Research Group of the Hungarian Academy of Sciences.

train is operated, with one type of locomotives. This information simplifies the problem. Our goal is to find an assignment of the engines to the trains that uses a minimum number of engines.

The operational policy of MÁV includes for instance that every engine between 48 to 60 hours of running must go for maintenance at one station of a prespecified subset of stations. Furthermore, it is allowed to define an artificial train with the goal to send a locomotive from a station to another station if this locomotive is needed to complete the assignment for a day (or a period) at that station. It can happen that by adding some artificial trains to the timetable we might find assignments that use less locomotives than the solution of the problem with the original timetable. Therefore from a practical point of view we might have different objectives like solving the problem with a minimum number of locomotives or minimizing the total energy consumption of the locomotives used in the assignment or minimizing the operational cost of the passenger trains over that region (with or without some artificial trains included). Currently, for the experts of the MÁV, solving this problem with the criteria of minimizing the number of necessary locomotives in the assignment, takes a few days for the given test region. Our task is to find a mathematical model for the problem and generate optimal solutions quickly.

After some attempts to collect the necessary data for our modeling purposes we agreed with the experts of MÁV that the goal of the optimization is to find the minimum number of necessary locomotives. However, we believe that if we had all necessary data we could have all the above mentioned objective functions for optimization.

Therefore the problem is the following: a passenger train timetable for a region of Hungary and a set of engines are given, and we have to assign an engine to each train so that each engine can be assigned to at most one train at a time. The connection of the engine from a train, which has just arrived at the station to the next train takes about half an hour. This time is known as *connection time* and its exact value influences the assignment of engines to the trains at a given station. The engines must go for maintenance at one station of a prespecified subset of the stations within a prescribed interval of working hours. After analyzing the problem it is clear that this maintenance condition makes the problem NP-complete. The task is again to minimize the number of used engines.

The MÁV has many different types of engines. For our purpose, we only have distinguished diesel-engines and electric-engines. As the lines of these two types are disjoint for technical reasons, we can assume that we have one type of engine, and each engine can pull each train. This assumption simplifies our model.

The problem refers to the classical engine assignment problem which is one of the most important problems of railway optimization (see e.g. Ahuja et al. [1]). Although this seems to be a widely studied problem, the instances with different extra conditions and specifications require different combinatorial optimization models and methods. These distinctions appear also in the size of the solvable instances. So there are locomotive-car assignment problems with different condi-

tions that can be written as (mixed)-integer programs, see for example Cordeau et al. [2], [3]. Most of these are solved by general optimization techniques as LP-relaxation, Branch-and-Bound method or Bender's decomposition.

We built up a 0-1 programming model, presented in [4], which has a lot of variables and conditions but it seems to describe the complete problem. This integer programming model was too large to be used for computations, (and unfortunately integer programming models and LP solvers are not widely used at MÁV by their experts). So we consider the engine assignment problem without the maintenance condition, that is called the weak engine assignment problem (*WEAP*). This problem can be modeled as a purely combinatorial optimization one, that is to find a minimum number of paths in a graph, which are disjoint and cover the node-set of the given graph. This model is presented in Section 2.

Our goal is not to introduce a model with full mathematical accuracy, rather to build a model that describes all the important requirements and that can be solved efficiently. Of course, those important constraints, that are not included, like maintenance, should be checked. For instance the prescribed maintenance needs to be placed in the task-list of all locomotives used in the transportation, by analyzing (and if it is necessary by modifying) the solution of the *WEAP*. The weak assignment problem is finally presented as a circulation model and we used the minimum cost flow algorithm of LEMON [5]. The computational experiments are provided for the Balassagyarmat region. The results are described in Section 3. Analyzing our computational results, we found out that we can place the maintenance property into the task list of locomotives. So in the region, where we have tested our method, this weak problem seemed to be good for practical purposes. During the computational study we found out that the model is very sensitive for some parameters like connection times, that we denoted by  $\mu$ . According to the operational policy of MÁV, a train can be pulled by more than one engine (usually by at most two). MÁV allows engines running without a train between two stations, these are said to be light-travels. Such light-travels depend on the given timetable of the region and the applied operational rules and policies. After dispensing with the opportunity of light-travels, other parameters are included in the circulation model of Section 2. In the practice of MÁV the task list of any engine should satisfy a return constraint, namely the engine starts its duty at a given station and after a few days period at the end of a day it arrives at the same station. Such a task list can be repeated periodically. (Of course there are some special days like weekends or holidays when these task lists can not be applied.) Using our model and solution method we can compute such kind of solutions, as well. Taking into consideration that the timetable is periodic we can solve our model on a week (period) base. Periodicity means that on each Monday, before the first train departs, we should have the same number of engines at the departure stations. This condition implies that we have to guarantee the necessary number of locomotives at the beginning of the period. This constraint can be included in the circulation model quite easily without destroying its nice combinatorial properties. In this way we gave a new

kind of solution to the engine assignment problem that had not been used at MÁV earlier.

## 2 Weak Engine Assignment Problem

The integer program presented in [4] describes precisely the engine assignment problem. But this is an NP-complete problem. The condition of maintenance makes the problem NP-complete. The goal of this section is to derive the Weak Engine Assignment Problem (*WEAP*). The *WEAP* does not contain the maintenance conditions, so it is a simplified model of the engine assignment problem. Due to these simplifications the *WEAP* can be solved in polynomial time, based on combinatorial optimization techniques. First we construct a directed acyclic graph, in this graph a path will correspond to a train-sequence, which can be pulled by the same engine. Then we reformulate the problem to a maximum matching problem in a bipartite graph, and finally to a circulation problem. The following notations will be used: the set of arcs leaving (resp. entering) the node  $v$  is denoted by  $\delta^{out}(v)$  (resp.  $\delta^{in}(v)$ ), the node-set of a graph  $G$  is denoted by  $V(G)$  and the arc-set is denoted by  $A(G)$ .

Let us construct a directed graph  $G = (V, A)$ , where the nodes one-to-one correspond to the trains listed in a given timetable. The train  $v$  and the corresponding node is referred also by  $v$ . There exists an arc from  $v_i$  to  $v_j$ , which is denoted by  $v_i v_j$  if and only if  $v_j$  can be pulled after  $v_i$  by the same engine, that is the arrival station of  $v_i$  equals the departure station of  $v_j$ , and the train  $v_i$  arrives at the station at least  $\mu$  minutes before the train  $v_j$  leaves it. Clearly the graph  $G = (V, A)$  is a directed acyclic graph.

We refer to the graph  $G = (V, A)$  as a *railway graph*, if it comes from *WEAP* by the above construction. A directed path of this graph corresponds to a train-sequence that can be pulled by one engine. (A single-node is also considered as a path.) Each node should be covered by one path. (If we allow only one engine to be coupled to a train, then each node should be covered by exactly one path, in the other case each node should be covered by at least one but at most 2, 3, ... etc. paths.) A set of path  $\mathcal{P} = \{P_1, P_2, \dots, P_k\}$  is said to be a *disjoint (double, triple) path-cover* if  $V(P_i) \cap V(P_j) = \emptyset$  for  $1 \leq i < j \leq k$  (or  $1 \leq |P_i : v \in P_j| \leq 2, 3 \forall v \in V$  resp.), and  $\bigcup_{1 \leq i \leq k} V(P_i) = V$ . It is easy to

see that a disjoint path-cover's composed of  $t$  paths corresponds to an engine assignment, which uses  $t$  engines. So the *WEAP* can be rephrased as follows:

There is a directed acyclic graph  $G = (V, A)$ , and we want to find a disjoint path-cover, which has a minimum number of paths. Such a disjoint path-cover gives the minimum number of engines needed, furthermore it gives the train-sequences. Let us construct a bipartite graph  $\bar{G} = (\bar{V}^1, \bar{V}^2; \bar{A})$  from the original graph  $G = (V, A)$ . The node-set  $\bar{V}^1$  consists of nodes  $v^1$  for every original node  $v \in V$ , and the node-set  $\bar{V}^2$  consists of nodes  $v^2$  for every original node  $v \in V$  likewise. Then the arc-set consists of arcs  $v_i^2 v_j^1$  for every original arc  $v_i v_j$ . A path-cover of the original graph  $G$  which has  $k$  disjoint paths corresponds to a matching of size  $|V| - k$  in the bipartite graph  $\bar{G}$ . So the minimum disjoint

path-cover problem is equivalent to a maximum matching problem in a bipartite graph, which is solvable in polynomial time, see König [6], or Hopcroft and Karp [7]. But if we want a double or triple path-cover or we have some more conditions, then there is a circulation model that gives us more features.

### 2.1 Circulation Model

The path-cover is a good model for the *WEAP*, but we have special cases when we need to modify the model. For example if we allow two engines to be coupled on a train. Then the paths, which correspond to the train-sequences, are not necessarily disjoint.

Let us construct the graph  $\tilde{G}$  from the railway graph  $G = (V, A)$ , with node-set  $\tilde{V} = V' \cup V'' \cup \{s, t\}$ , where  $V'$  and  $V''$  are two copies of the original node-set  $V$ . Nodes  $v', u'' \in V \setminus \{s, t\}$  mean that there exist trains  $v$  and  $u$  in the given timetable, namely every train is represented twice in  $\tilde{G}$ . The arc-set is  $\tilde{A} = \{v'v'' : v \in V\} \cup \{u''v' : uv \in A\} \cup \{v''t : v \in V\} \cup \{sv' : v \in V\} \cup \{ts\}$ , (see Figure 1 ).

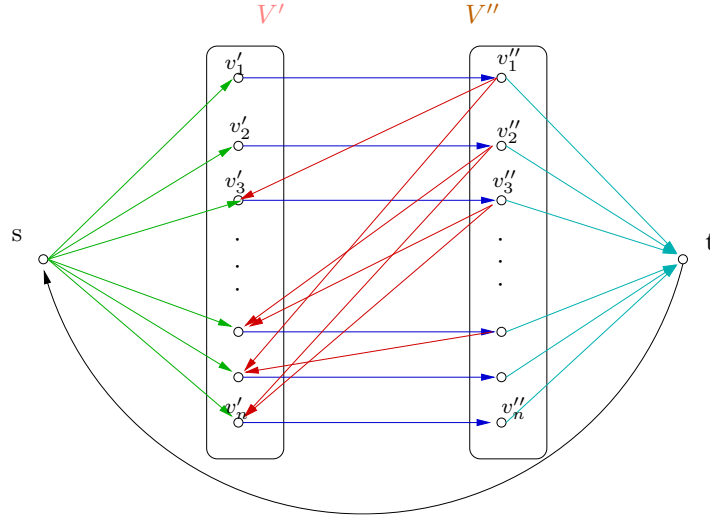


Fig. 1. Circulation

Define the lower bounds and the upper bounds  $f, g : \tilde{A} \rightarrow \mathbb{N}$  by

$$f(e) = \begin{cases} 1 & \text{if } e = v'v'' \text{ for some } v \in V, \\ 0 & \text{otherwise,} \end{cases} \quad g(e) = \begin{cases} 1 & \text{if } e = v'v'' \text{ for some } v \in V, \\ \infty & \text{otherwise.} \end{cases}$$

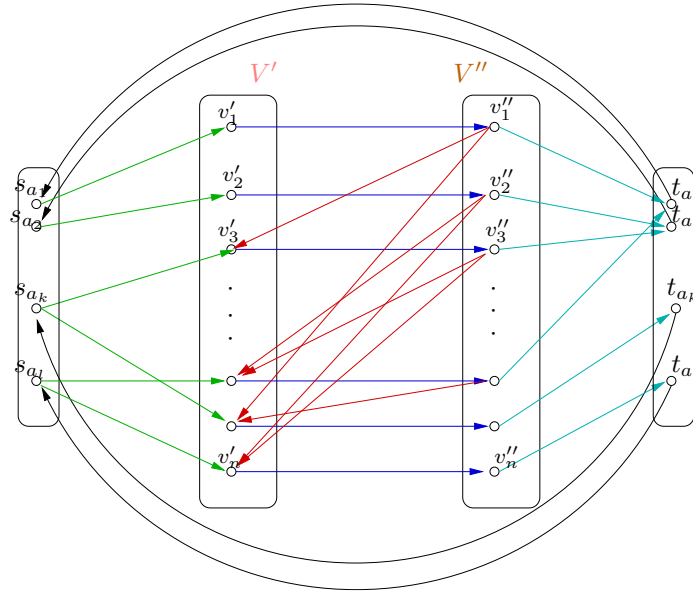
A function  $x : \tilde{A} \rightarrow \mathbb{R}$ , for which  $f(e) \leq x(e) \leq g(e)$  for every  $e \in \tilde{A}$  is said to be a circulation if  $\sum_{e \in \delta^{out}(v)} x(e) = \sum_{e \in \delta^{in}(v)} x(e)$  for each  $v \in \tilde{V}$ . If the upper and

lower bounds are integer then the existence of a circulation implies the existence of an integer circulation as in our case. Furthermore the integer circulations correspond to the solutions of the *WEAP*. Moreover, if we define a cost-function  $c : \tilde{A} \rightarrow \mathbb{R}$  by

$$c(e) = \begin{cases} 1 & \text{if } e = ts, \\ 0 & \text{otherwise.} \end{cases}$$

then the solutions of the *WEAP* correspond to minimum cost circulations.

The previous circulation model describes the *WEAP* completely. The situation when more than one engines are coupled to a train can be described by simple modifications. If for any train  $v$  it is allowed to be coupled to at most two engines then  $g(e) = 2$  for  $e = v'v''$ . Similarly we can change the upper bounds for 3 or more engines as well.



**Fig. 2.** Circulation for the Periodic Model

For periodic solutions further modification of the above model is needed. The timetable is periodic, so it is a natural aim to look for solutions such that for each station the number of engines is the same at the beginning and at the end of the period. Then let us modify  $\tilde{G}$  as follows: replace the nodes  $s$  and  $t$  by  $|N| = l$  pieces of node-pairs,  $s_a, t_a$  for each station  $a$ . Replace the arc  $v't$  by arc  $v't_a$  if the arrival-station of the train  $v'$  is  $a$ , and similarly replace the arc  $sv''$  by arc  $s_av''$  if the departure-station of the train  $v''$  is  $a$ , and finally let us put one arc  $t_as_a$  for each  $a \in N$ . (See Figure 2.)

If we define the lower bounds as zero and the upper bounds as infinity on the new arcs, then the solutions of the circulation problem in this new directed graph correspond to the solutions of the engine assignment problem with periodicity constraint. The periodicity constraint ensures that the same solution can be applied for the next period as well.

The minimum cost circulation problem can be solved in strongly polynomial time, see Goldberg and Tarjan [8], [9].

### 3 Computational results

We have got data for two regions of Hungary, one is the East-Hungarian Region, and the other is the region of Balassagyarmat. For the East-Hungarian Region we have got the freight-train timetable with 689 trains, 488 trains running with electric engines, and 201 trains running with diesel engines. While these data give two disjoint problems, we considered only the problem, where electric engines are used. The timetable of the freight trains is less dense, so the corresponding railway graph has many disjoint paths. The computations gave a very large number of engines, if we assume that passenger trains are pulled by different sets of engines. This computational experiment shows that there is a strong relation between the minimum number of necessary engines and the density of the railway graph. Certainly in real life the experts of MÁV solve the problem with fewer engines, but the engines pull freight trains and also passenger trains, so the corresponding railway graph is different. If it is necessary then the experts of MÁV define single-engine-runs (light-travels) to make the railway graph more connected. The current procedure of MÁV is purely heuristical.

Let us analyze the currently used timetable of the region of Balassagyarmat. (This timetable is available on <http://www.elvira.hu>.) In this region we have got a weekly timetable for the passenger trains, with 521 trains and 9 departure/arrival stations. We divided the problem into two parts, with and without the return-back constraint. The program that we have developed controls several parameters like the length of the modeling period (1, 2, 3, 4, 5, 6, 7, 14, 21 days), the connection time,  $\mu$ , and the maximal number of engines that can be coupled to a train. The starting day of the period can have unexpected effects, for instance the number of necessary engines is different for a one week model if we start the period on Monday or on Tuesday. The tables show the computational results.

Some computational experiences follow: according to the definition of the railway graph, the size of this graph's node-set depends on the length of the period. The size of the edge-set depends on the value of  $\mu$ , that is if we increase the value of the connection time the graph has less arcs, so the graph would be less dense. Certainly by the growth of the graph we will need more time both to build up the graph, and to solve the problem. The number of engines that can be coupled to a train influences the total number of needed engines. If we increase this number, the number of needed engines will decrease in our case.



The problem with periodicity constraint would become infeasible if we allowed only one engine to be assigned to a train.

**Table 1.** One-day period without return-condition for the region of Balassagyarmat, when we allow two engines to be assigned to a train

The value of $\mu$	1min	5min	10min	15min	20 min	30min
$ V(\tilde{G}) $	81	81	81	81	81	81
$ A(\tilde{G}) $	611	602	589	574	567	558
Time to build the graph (s)	0.187	0.141	0.14	0.14	0.141	0.203
Time to solve the problem (s)	0.406	0.437	0.484	0.546	0.547	0.516
Number of needed engines	11	13	14	15	18	19
Number of trains which use 2 engines	2	0	2	2	2	1

**Table 2.** One-day period with return-condition for the region of Balassagyarmat, when we allow two engines to be assigned to a train

The value of $\mu$	1min	5min	10min	15min	20 min	30min
$ V(\tilde{G}) $	81	81	81	81	81	81
$ A(\tilde{G}) $	611	602	589	574	567	558
Time to build the graph (s)	0.219	0.188	0.141	0.172	0.156	0.141
Time to solve the problem (s)	0.406	0.484	0.391	0.375	0.375	0.453
Number of needed engines	11	13	14	15	18	19
Number of trains which use 2 engines	2	7	5	5	11	9

**Table 3.** One-week period without return-condition for the region of Balassagyarmat, when we allow one engine to be assigned to a train

The value of $\mu$	1min	5min	10min	15min	20 min	30min
$ V(\tilde{G}) $	521	521	521	521	521	521
$ A(\tilde{G}) $	28824	28772	28695	28610	28563	28508
Time to build the graph (s)	6.266	6.281	6.296	5.141	6.296	6.281
Time to solve the problem (s)	10.828	10.25	10.36	12.141	10.297	10.235
Number of needed engines	12	19	21	21	24	25

**Table 4.** One-week period without return-condition for the region of Balassagyarmat, when we allow two engines to be assigned to a train

The value of $\mu$	1min	5min	10min	15min	20 min	30min
$ V(\tilde{G}) $	521	521	521	521	521	521
$ A(\tilde{G}) $	28824	28772	28695	28610	28563	28508
Time to build the graph (s)	6.328	6.234	6.25	6.281	6.109	6.203
Time to solve the problem (s)	11.422	10.765	10.844	10.953	10.938	11.016
Number of needed engines	12	13	14	15	18	19
Number of trains which use 2 engines	21	23	38	38	29	23

To solve the problem we use the circulation model (Figure 2) that we extended for different parameters of the problem. (For example the model includes the value of  $\mu$ , the number of engines that can be coupled to a train, and so on.) The computational model and solver use extended routines of LEMON. LEMON is an open source library written in C++, developed for combinatorial optimization algorithms by the Department of Operations Research, ELTE [5].

Summarizing our results we can say that a simplified model, *WEAP*, for the engine assignment problem has been introduced. The reason for simplification was that the maintenance constraints make the model NP-complete, therefore it becomes practically non-tractable. Among the advantages of *WEAP* is its pure combinatorial nature and that it is polynomially solvable. On the other hand several operational policies can be built into the circulation model (eg. coupling constraints of engines) keeping its nice optimization property. Our computations show that *WEAP* for small sizes of real life problems can be solved fastly. Further computational investigations are needed for larger problems.

*WEAP* can be extended for the case when we have several types of trains and several types of engines too. This is ongoing research now.

## Acknowledgments

The authors would like to thank Erzsébet Hamvas, Tibor Pólya and József Túri, who work at MÁV Institute for their fruitful discussions and help in collecting the necessary data and analyzing our results. Endre Németh was participating in building the earlier version of the 0-1 programming model. Tibor Takács's cooperation was very helpful in building the communication and data handling parts of our code. This research and the development of the software for *WEAP* was sponsored by the Hungarian Railway Company Pl. (MÁV).

## References

1. Ahuja, R., Liu, J., Orlin, J., Sharma, D., Shughart, L.: Solving real-life locomotive scheduling problems. Technical Report 4389-02, Mas-

- sachusetts Institute of Technology (MIT), Sloan School of Management (2003) <<http://ideas.repec.org/p/mit/sloanp/1799.html>> [date of citation: 2006-05-15].
2. Cordeau, J.F., Desaulniers, G., Lingaya, N., Soumis, F., Desrosiers, J.: Simultaneous locomotive and car assignment at via rail canada. *Transportation Research Part B* (2000) 767–787
  3. Cordeau, J.F., Soumis, F., Desrosiers, J.: Simultaneous assignment of locomotives and cars to passenger trains. *Operations Research* **49** (2001) 531–548
  4. Illés, T., Makai, M., Vaik, Z.: Railway engine assignment models based on combinatorial and integer programming. Technical Report 2005–04, Eötvös Loránd University of Sciences, Department of Operations Research (2005) <<http://www.cs.elte.hu/opres/orr/Reports.htm>> [date of citation: 2006-05-15].
  5. EGRES: Lemon library (2003) <<http://lemon.cs.elte.hu>> [date of citation: 2006-05-15].
  6. König, D.: Gráfok és mátrixok. *Matematikai és Fizikai Lapok* (1931) 116–119
  7. Hopcroft, J., Karp, R.: A  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing* (1973) 225–231
  8. Goldberg, A.V., Tarjan, R.E.: Finding minimum-cost circulations by canceling negative cycles. *Proceeding of the Twentieth Annual ACM Symposium on Theory of Computing, 20th STOC*. Chicago, Illinois, 1988 (1988) 388–397
  9. Goldberg, A.V., Tarjan, R.E.: Finding minimum-cost circulations by canceling negative cycles. *Journal of the Association for Computing Machinery* (1989) 873–886

# Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms

Johanna Törnquist

Department of Software Engineering and Computer Science,  
Blekinge Institute of Technology, Soft Center  
Box 520, 372 25 Ronneby, Sweden  
Johanna.Tornquist@bth.se

**Abstract.** This paper provides an overview of the research in railway scheduling and dispatching. A distinction is made between tactical scheduling, operational scheduling and re-scheduling. Tactical scheduling refers to master scheduling, whereas operational scheduling concerns scheduling at a later stage. Re-scheduling focuses on the re-planning of an existing timetable when deviations from it have occurred. 48 approaches published between 1973 and 2005 have been reviewed according to a framework that classifies them with respect to problem type, solution mechanism, and type of evaluation. 26 of the approaches support the representation of a railway network rather than a railway line, but the majority has been experimentally evaluated for traffic on a line. 94 % of the approaches have been subject to some kind of experimental evaluation, while approximately 4 % have been implemented. The solutions proposed vary from myopic, priority-based algorithms, to traditional operations research techniques and the application of agent technology.

## 1 Introduction

In most countries, the railway traffic system is a significant part of the backbone transport system as it is a major service provider for passenger traffic and freight transportation. Traffic and transport policies are striving towards decreasing road traffic pollution by e.g. increasing railway usage when appropriate. At the same time, the available railway systems are partly oversaturated creating bottlenecks on major links. An important issue is thus how to best use the existing capacity while ensuring sustainability and attractiveness.

Railway traffic scheduling is often considered a difficult problem primarily due to its complexity regarding size and the significant interdependencies between the trains. A railway network is generally far from as fine-grained as a road traffic network. The options to overtake and meet are very limited and depend on e.g. available side-tracks, switches, signalling facilities and the characteristics of the trains. Furthermore, in many countries the traffic is heterogeneous with trains carrying different types of cargo (commuters, long-distance passengers with connections, express freight, bulk

goods, etc) with different preferences, destinations and speed functions. All these specific attributes make the trains highly interdependent and their interplay complex to plan, overview and execute. In addition, the organisation of the railway traffic management differs between countries. In some, the operator and traffic manager are one and the same company while in some European countries the railway market is partly or fully deregulated with a separate authority governing the infrastructure and traffic management while several privatised and competing operators are using the tracks. The challenge is thus to comply with relevant preferences based on the available capacity to achieve and execute a robust and attractive timetable. This review surveys the research carried out within the area of railway scheduling and dispatching. Even though this is a rather well-known and studied problem domain, the number of reviews dealing with this topic is limited. In 1980, Assad [1] presented a survey of different models for rail transportation including optimisation, queuing, simulation approaches, etc. Later, a survey by Cordeau et. al. [17] was published and with a specific focus on various optimisation models for the most commonly studied railway problems.

The aim of this paper is to classify and compare the various approaches for railway traffic scheduling in more detail than previous surveys which instead have had a wider scope. Furthermore, new methodologies such as agent technology have appeared during the last years and these need to be taken into account and be compared to more traditional approaches. The next chapter will present the scope of this paper, followed by a description of the problem domain. The classification and review framework that has been applied is then presented. A discussion of the results from the review and some observations are later provided, followed by conclusions and directions for future research.

## 2 Scope

The focus here is railway traffic scheduling with an emphasis on slot allocation (i.e. the assignment of entry and exit times for trains on track sections) but also to some extent route allocation (i.e. which track sections to use to get from origin to final destination). That is, if we have a set of trains with individual and possibly competing requests for track capacity, how should the trains be scheduled to reach the scheduling objective(s)? Thus, primarily the perspective of an infrastructure provider that may schedule trains of several train traffic operators (rather than an operator scheduling its services exclusively on its own tracks) is in focus. Hence, rail transport scheduling, i.e. primarily scheduling of the available resources such as fleets of vehicles and staff for specific railway services, is not explicitly considered even if there are some common aspects. For these types of problems we refer to [1], [18], [6], [19], and [17]. Furthermore, approaches which focus on periodic timetabling, timetable synchronisation and sensitivity and robustness analysis of timetables are not reviewed explicitly either and we refer instead to e.g. [50], [58] and [48]. Even though the task of analysing and predicting the effects of a disturbance is a part of solving disturbances, research specifically focusing on that is not included, but can instead be found in e.g. [28].

A distinction is here made between tactical scheduling, operational scheduling and re-scheduling of railway traffic. Scheduling (or timetabling) is the process of constructing a schedule from scratch, while re-scheduling (or dispatching) indicates that a schedule already exists and will be modified. The scheduling can also be carried out with different time perspectives, i.e. on a tactical or operational (real-time) level. In Europe, there is a tradition of creating master schedules that specify a strict route and timetable for each train on a tactical level with the intention to execute it in real-time. The scheduling may thus involve both route choice and slot allocation, where a slot is the time window a certain train is planned to use a specific track section. For obvious reasons, scheduling of passenger traffic is often carried out on a tactical basis.

Operational scheduling is commonly used for example in North America (and for freight transport scheduling). Instead of creating a master schedule a long time before it is actually put into action, the operational scheduling takes place not long before departure. The routes are then generally already fixed but not the slots. Re-scheduling is related to disturbance handling, i.e. assigning new slots to the trains to minimise their deviations from the established timetable.

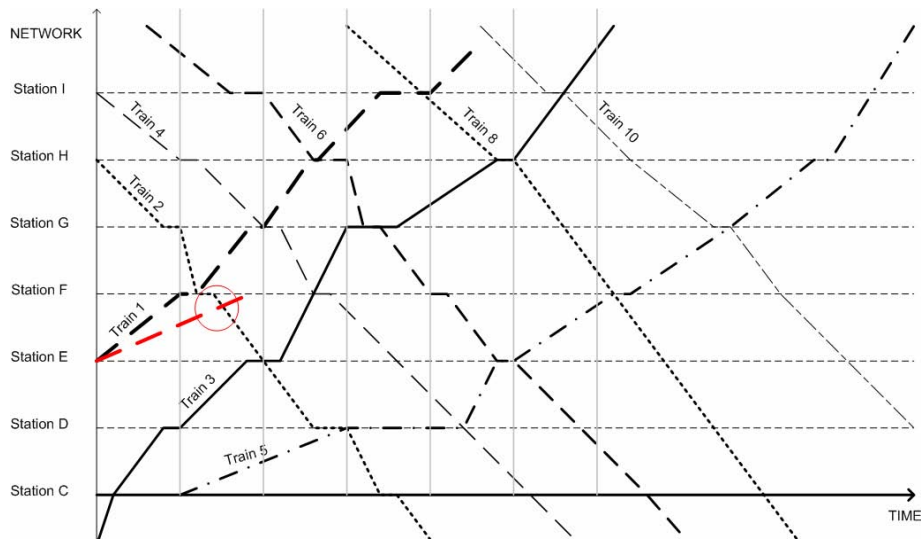
This review does not include an explicit survey of the tools used by the railway authorities or other stakeholders. Included are 48 approaches that have been published during the time period 1973-2005. Some approaches have been described in several publications, but only the references to the most recent and detailed descriptions are included here.

### 3 Domain description

Tactical scheduling, operational scheduling and re-scheduling have the basic problem and limitations in common. The kernel of the problem is the conflicts that arise when two or more trains want to occupy the same part of the network simultaneously. The railway network is usually divided into blocks (i.e. separate track sections) where each block can normally hold only one train at a time in order to maintain the required safety level (referred to as *line blocking*). Conflicts could appear when a train is too close behind another train travelling in the same direction, or when two trains are travelling in opposite directions and would meet within the same block. Due to the line blocking, trains are not allowed to get too close and not to meet within a block. The conflicts need to be solved not only taking into consideration one isolated conflict, but also the effect it will have on the surrounding traffic later on in time. Conflicts may thus be interdependent and nested. Solving one may consequently create additional conflicts or resolve others. The number of possible solutions can become very large depending on e.g. the network structure, the amount of traffic and type of trains.

Fig. 1 provides an illustration of a bi-directional (two-way traffic) single-tracked railway line with line blocking, and where a conflict has emerged due to a deviating train (i.e. Train 1). When Train 1 departs from Station E, it malfunctions temporarily and becomes significantly delayed. Since the schedule of Train 1 interferes with foremost Train 2, Train 2 becomes delayed as well due to the restriction of not

allowing two trains to use a block (i.e. between Station E and F) simultaneously. The circle indicates the violation of the restriction that would take place if the initial schedule of the trains was to be followed. Instead, Train 2 must wait for Train 1 which causes additional conflicts and possibly delays Train 3 and 4 as well depending on how the situation is resolved.



**Fig. 1.** A time-distance graph describing the railway traffic network between Station I and Station C and the scheduled traffic.

Even though the three types of scheduling problems have the main kernel in common, there are some significant differences regarding context, time frame and objective(s). Tactical scheduling usually involves scheduling for a large traffic network for a long time horizon (sometimes up to a year, but on a day-to-day basis) and the time available for creating the timetable may be several months. Operational scheduling has a shorter time frame and is initiated closer in time to the departure of the trains. The objective of tactical scheduling may be more complex reflecting the demand of several stakeholders and taking into account infrastructure maintenance. Operational scheduling balances also competing requests, but time is more of an issue and some new constraints such as definite time windows and connections may have been introduced.

Re-scheduling is initiated when a deviation from an initial schedule occurs with the aim to minimise the overall delays. The re-scheduling may need to be carried out within a short time frame (minutes or seconds) and not be able to or have time to explicitly consider the interests of all stakeholders. However, connections and the consequential importance of pairing slots, platforms and tracks are introduced; see e.g. [76], [44] and [12]. Those considerations are partly also taken into account when creating the

initial timetable but the liberties are fewer during timetable execution and re-scheduling since some parameters cannot be changed (i.e. rolling stock is already allocated, timetables for passengers are published and platforms announced, track maintenance is planned or have already started, etc).

In practice, tactical and operational scheduling are often carried out using a combination of computational tools and human expertise while for re-scheduling, human expertise and rules of thumb often is the dominating procedure.

## 4 Classification framework

The framework applied classifies the approaches according to the scheme in [Table 1](#).

**Table 1.** Classification and review framework.

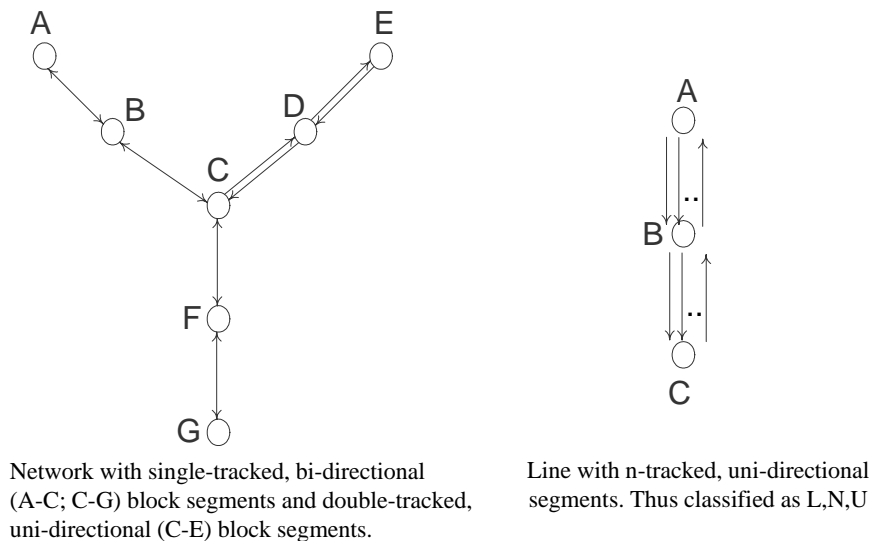
<b>PROBLEM TYPE</b>  PLANNING PERSPECTIVE: Tactical scheduling Operational scheduling Re-scheduling  INFRASTRUCTURE REPRESENTATION: Line (L), Network (N) Single- (S), Double- (D), (N)-tracked, Uni- (U), or Bi-(B)directional  OBJECTIVE(S)  SPECIAL CONSIDERATION(S)	<b>CONTROL</b> Centralised (C) Hierarchically distributed (H) Distributed (D) Localised (L)
	<b>PROBLEM FORMULATION</b>
	<b>SOLUTION MECHANISM</b>
	<b>EVALUATION LEVEL</b> 1. Conceptual approach 2. Simulated experiments with artificial data 3. Simulated experiments with real data 4. Field experiments 5. Implemented (deployed)
	<b>PROBLEM INSTANCE AND SIZE</b>

*Problem type* specifies which problem the reviewed approach is assigned to handle regarding the *planning perspective*, *infrastructure representation*, *objective(s)*, and *special considerations* in mind. As previously described, tactical scheduling is the most long-term planning perspective, whereas operational scheduling concerns scheduling close in time to departure. Re-scheduling focuses on the real-time re-planning of an existing timetable when deviations from it have occurred. *Infrastructure representation* describes what kind of railway infrastructure that the approach can be applied to. A *line* is a sequence of segments between two major stations with possibly several intermediate stations, while a *network* is composed of one or several junctions of lines. The classification of whether an approach can represent a line or also a network is based upon its problem formulation. E.g. if the problem formulation assumes that the segments and/or stations are sequenced into a line and that the traffic traverses them in that certain order, a network can not be represented by that approach.

Each segment is composed of one or several parallel track sections (i.e. blocks). The maximum number of tracks within a segment that an approach can represent is referred to as *single*, *double* or *N*. If an approach can handle tracks permitting traffic



in one direction, it is denoted ‘U’ (uni-directional), while if also (or instead) two-way traffic is accounted for it is denoted ‘B’ (bi-directional). Fig. 2 provides an illustration of the terminology used. Double-tracked segments are often in practice uni-directional, where one side of the segment is allocated to traffic in one direction and the other allows traffic in the other direction. The reason behind this restriction is that it facilitates the traffic management, or the signalling infrastructure is limited to show signals in only one direction per track section. However, in dense traffic areas, the tracks may need to be used for traffic in either direction (if the signalling infrastructure permits) since there may be an imbalance in the traffic volume during some parts of the day or some express trains may need to overtake slower trains. Allowing trains to run in both directions obviously increases capacity and flexibility but also increases the complexity.



**Fig. 2.** Illustration of terminology used for types of infrastructure representation.

*Objective(s)* state the purpose and goal of the solution mechanism (e.g. minimising travel time, operating costs or maximising utility). *Special considerations* (e.g. connecting trains, platform assignment, and train preferences) specify if the approach account for other characteristics and constraints beside line blocking and logical relations.

Besides classifying the problem type, we consider the *problem formulation*, the *control strategy* and *solution mechanism* applied. The formulation refers to the representation of the solution space. Most common are mathematical models such as MIP (Mixed Integer Programme), CSP (Constraint Satisfaction Problem), CP (Constraint Programme) and other models based on e.g. graph theory and network modelling. The control strategy represents how to search through the solution space

defined by the problem formulation. Four main control strategies for solving the problem can be found; centralised (C), hierarchically distributed (H), distributed (D) and localised (L). A centralised approach refers to when the problem is solved as one instance. That is, the full problem is considered simultaneously such as during some form of enumeration as in classical Branch and Bound, see e.g. [57]. A distributed (or decentralised) approach divides the main problem into sub-problems with the aim of solving them partly in parallel. The relation between the sub-problems (i.e. how they together form the main problem) needs then to be formulated and the solution processes need to be synchronised. If there is a hierarchy and some kind of central and synchronising control of the sub-problem solving, this is referred to as hierarchically distributed (e.g. classic Lagrangian relaxations, see [23]). If the sub-problems instead are solved independently, this is referred to as a distributed strategy. Sub-problems are usually solved in either a cooperative or competitive environment. In the cooperative environment, the sub-problems have a common goal and adjust to the overall best actions. In a competitive environment, all or some of the sub-problems are solved with individual and sometimes competing interests. A commonly used competitive environment is auctions, which often is referred to as a market-based mechanism. For more information see e.g. [74]. The localised strategy is very similar to the distributed; the problem is divided and its parts allocated to e.g. the stations, but the stations do not synchronise their behaviour in any way.

Examples on solution mechanisms are different types of heuristics such as Local Search (LS), Tabu Search (TS, see [27]) or Simulated Annealing (SA, see [39]). Branch and Bound (B&B), Lagrangian relaxations, expert systems and more straightforward tailored methods such as full or partial enumeration or priority-based conflict resolution are other examples. For further information on related terminology, we refer to [62] and [57].

The evaluation level of an approach refers to how developed and evaluated it is with regard to what is stated in the publication(s). That is, if it is a conceptual description, has been experimentally applied to a problem instance of a real or fictional setting, been evaluated in a real setting (field experiments), or has been implemented. By implemented, we mean that the approach has been, or is, a deployed system. The problem instance and size specifies the maximum size (number of stations, segments and trains) of the problem instance that the approach has been applied to (while the size of the practical problem in mind may be larger but not considered experimentally).

Finally, we have also tried to compare the advantages and disadvantages of the suggested modelling and solution approaches, considering the varying set of prerequisites during the publication year and the context. Generally, it would be interesting to have a quantitative benchmark that compares e.g. the speed and optimality measure of the approaches reviewed. However, due to lack of information on those attributes and the overall dominating use of individual data instances, such an analysis has not been possible.

## 5 Discussion of review results

The publications reviewed were published during the time period 1973-2005 and a summary of the approaches is presented in the [Appendix](#). The terminology used differs between the publications reviewed. When discussing the problem size by means of number of stations, segments and trains in the tables in the [Appendix](#), we have taken the liberty to translate the given settings into number of stations and segments, when possible. [Table 2](#) and [Table 3](#) present the number of approaches that considers the different types of infrastructure. ‘Unclassified’ means that the publication(s) did not provide enough information for a complete classification. Since the objectives and premises for tactical and operation scheduling and re-scheduling vary, different special side-constraints are applied. As can be seen in [Table 4](#), more details of the infrastructure are considered during scheduling while preferences related to trains and operators are more commonly considered during re-scheduling. The vast majority of the approaches adopts a quite simplified representation of stations and do not consider the potential crossing of train paths and allocation of tracks within stations.

**Table 2.** Frequency of infrastructure representation per problem type, where U = uni-directional, B = bi-directional, S=single-tracked, D=double-tracked, and N=n-tracked refer to the segment structure (the non-station segments).

Infrastructure	Tactical scheduling	Operational scheduling	Re-scheduling	Total
US Line	2	0	2	4
US Network	1	0	0	1
UD Line	1	0	2	3
UN Network	0	0	1	1
BS Line	4	4	3	11
BS Network	1	2	1	4
BS,UD Line	3	0	0	3
BN Line	1	0	0	1
BN Network	7	1	8	16
(Unclassified) Network	0	0	4	4
Total	20	7	21	48

**Table 3.** Frequency of infrastructure representation per problem type referring to the segment structure (the non-station segments).

Infrastructure	Tactical scheduling	Operational scheduling	Re-scheduling	Total
Line	11	4	7	22
Network	9	3	14	26
Uni-directional	4	0	5	9
Bi-directional	16	7	12	35
Undefined	0	0	4	4
Single-tracked	8	6	6	20
Double-tracked	4	0	2	6
N-tracked	8	1	9	18
Unclassified	0	0	4	4

**Table 4.** Special side-constraints and the number of approaches that considers them.

Special consideration	Tactical scheduling	Operational scheduling	Re-scheduling	Total
Switches, track connections	1	-	1	2
Station and platform characteristics	4	-	-	4
Time windows	1	-	1	2
Rolling stock/ Crew schedules	-	1	1	2
Train connections	-	-	3	3
Platform allocation	-	-	3	3

Regarding the problem formulations adopted, the infrastructure and traffic are modelled in a few main ways. It is common to formulate an explicit MIP using binary variables to represent the sequence of trains on the segments, and continuous variables for the entry and exit times of each train on each segment or specific track. The line or network is then explicitly composed of segments, while the nodes between the segments (intersections, meet points, stations, etc.) are implicitly modelled. A second formulation models instead the stations explicitly and the segments between

implicitly. The binary variables and their values specify then in which order the trains enter and exit the stations (i.e. their tracks) and according to that continuous variables specify when a train arrives at and leaves the corresponding stations and tracks.

It is difficult to assess what the advantages and disadvantages of each alternative are. The second formulation (i.e. modelling stations and meet-points explicitly) seems to be less flexible to extend and use for a network since a station may be connecting several segments (e.g. main stations that serve as junctions for several lines) while a segment only has two end points. The first formulation seems to handle such an increased complexity better than the second formulation, but the advantage of the second formulation is that constraints related to station attributes (e.g. usage of platforms and switches) are easier to handle. A combination of the two formulations is to model both stations and non-station segments explicitly. That facilitates the specification of detailed restrictions for all elements, but the number of variables will consequently increase.

Another common formulation is to have a graph model of arcs and nodes representing the binary variables that specify the order of trains on the segments in the MIP. A sequence of arcs then needs to be created while considering a set of constraints. Using an object-oriented or a discrete-event formulation of the problem is another common representation.

The formulations previously described use variables to represent the start and end times of the slots. The majority uses continuous variables for the times, while a few discretize the time into time units of one or several minutes. Each time unit per train and block is then represented by a binary variable where the value '1' specifies that the time unit for that block is used by the specific train. This way, the sequence of trains on the blocks does not have to be explicitly modelled but is implicitly considered already. On the other hand, discretizing time may result in a significant amount of binary variables if small time units are used. For re-scheduling and scheduling dense traffic, it may be necessary to use small time units in order to utilise the infrastructure to the full extent. Five approaches have used discrete time units where four of them address tactical scheduling, i.e. [5], [51], [8] and [35], and one re-scheduling, i.e. [64].

The slots can also be discretized into a set of fixed slots (block- and time-dependent) where the objective then is to create the optimal and feasible combination of slots for each and all trains. This formulation can be seen foremost in combination with the use of MAS and auctions. Auctioning is becoming more commonly used within scheduling and the use of agent technology is more commonly adopted in the traffic and transport domain [21]. There are several other mechanisms of allocating track capacity and a detailed discussion about the different principles can be found in [26]. One of the problems that hamper the use of auctions and its applicability in the railway domain is the need to have a discrete set of subjects to bid for. Railway slots are to some extent an infinite and continuous set of options and are thereby difficult to effectively translate into a discrete set. The main challenges for these approaches are the formulation of the bid generation (including handling multiple interdependencies) and the set-up for negotiation and communication within the auctions. Since several of the publications do not outline these parts of their approach (only the general bidding procedure and objective) and apply the proposals on relatively small data sets, it is difficult to assess the general applicability.

Difficulties in handling large problems and scalability issues are sometimes used as arguments to apply distributed (including hierarchically distributed) methods such as auctions instead of centralised ones. Even though the vast majority of the publications reviewed use a centralised approach, there is a significant usage of distributed problem solving (see [Table 5](#)). Tactical scheduling has a comparably less time restriction and favours solution quality rather than algorithmic speed. Consequently centralised solution methods are dominating while five of the 20 approaches reviewed apply a distributed solution mechanism. Three of them ([\[4\]](#), [\[2\]](#), [\[3\]](#) and [\[35\]](#)) use agent technology and MAS to solve the problem and two approaches ([\[4\]](#) and [\[35\]](#)) apply a market-based strategy. Three approaches apply Lagrangian relaxations.

Only two approaches for re-scheduling consider a distributed mechanism and four adopt a localised strategy. The main difference between having a distributed (and hierarchically distributed) and a localised strategy is that the synchronisation of the distributed approach may require significant computational effort for the overhead communication and is (like the centralised approach) sensitive to an increase in problem size and set-up of the problem structure while the more localised strategy is (time-wise) not as dependent on the problem size. However, the localised strategy may result in a sub-optimisation and less robust and reliable solutions. There is thus an obvious trade-off that needs to be made.

**Table 5.** Frequency of control strategy used per scheduling problem. \*One approach for tactical scheduling evaluates both a hierarchically distributed control strategy and a centralised one.

Planning perspective	Centralised	Hierarchically distributed*	Distributed	Localised
Tactical scheduling	16	3	2	0
Operational scheduling	7	0	0	0
Re-scheduling	15	0	2	4
Total	38	3	4	4

The use of context-dependent and tailored solution methods are more common for operational scheduling and re-scheduling purposes than for tactical scheduling. Several approaches apply myopic mechanisms that do not consider the secondary effects of a decision and thus this may make them less appropriate for the general scheduling problem. Some approaches propose enumeration techniques, which for small problem instances may be sufficient and successful but for a larger problem, interdependent conflicts and secondary effects will arise. It is also quite common, especially for the re-scheduling problem, to use expert systems and priority rules. Those approaches incorporate the current work process of the dispatchers in many ways by translating tacit knowledge and rules of thumb into computerised systematic reasoning. This differs from the all-human decision-making process as it has a larger capability to consider a longer time horizon with more complex and nested decisions.

In [Table 6](#), the number of approaches per evaluation level and scheduling problem is presented and [Table 7](#) presents the frequency of infrastructure type used in the

evaluations. As can be seen, many of the approaches reach the stage of being experimentally evaluated but several for rather modest problem instances. An increase in the railway traffic volume in several countries as well as the increase of computational capacity would make one expect a trend towards increasing size of the problem instances used in experiments. However, no significant relation between infrastructure type and problem size used for evaluation and publication year can be seen for tactical or operational scheduling. The focus on re-scheduling seems to have increased the past years and the size of the problem instances used to evaluate the approaches for tactical scheduling and re-scheduling are interesting enough similar in size and type.

**Table 6.** Overview of the number of approaches on the different evaluation levels.

Planning perspective	Conceptual approach	Simulated w. artificial data	Simulated w. real data	Field experiment	Implemented
Tactical scheduling	1	6	13	0	0
Operational scheduling	0	3	2	1	1
Re-scheduling	0	3	13	4	1
Total	1	12	28	5	2

In railway networks, the demand for slots is sometimes larger than the available capacity and the different trains have varying characteristics and use different parts of the network. Hence, the traffic interplay may be too complex to schedule operationally and needs to be scheduled on a tactical level. Despite the complexity of the tactical scheduling and that nine out of those 20 approaches are able to represent a network structure, only two of them have been evaluated for a network structure, see [Table 7](#).

**Table 7.** Overview of the number of approaches that has used a certain infrastructure representation in the evaluation.

Planning perspective	Line	Network	Not classified
Tactical scheduling	17	2	1
Operational scheduling	7	0	0
Re-scheduling	14	6	1
Total	38	8	2

The variety of solution methods applied is impressive, providing innovative ideas which often have been quantitatively evaluated (see a summary of the review in the [Appendix](#)). Unfortunately, the choice of method is rarely motivated. Some publications state that the problem in focus is NP-hard and too difficult to solve to

optimality and instead apply a heuristic approach. The reason is claimed to be the growing complexity of the problem due to an exponentially increasing number of solutions with the increase in problem size and binary variables. Theoretically, a problem with  $n$  binary variables could generate a search space of  $2^n$  possible solutions. That may very well be true for a certain problem size and formulation. However, most publications make no attempt to show this for their problem or try solve the problem instance to optimality, but just assume it is too difficult. Due to the interdependencies (infeasibility and transitivity relations) between the binary variables, a large number of constraints are present and reduce the solution space significantly. Additional trains and segments may add increased complexity due to an increase in number of variables, but they may also decrease the search space since the number of restrictions may increase as well. Therefore, general conclusions on the proportional relation between the number of binary variables, size of solution space and computation time are difficult to make. In addition, the complexity of the problem is also dependent on the input data and the objective function. For tactical scheduling six approaches have conducted an optimality check and one compares its results to the Nash Equilibrium. Three approaches for operational scheduling have been subject to an optimality check and five of the re-scheduling approaches. The presence of optimality checks is not strictly related to publication year, i.e. approaches in the early 1990's as well as recently published approaches have been evaluated, while several of the recently published are non-evaluated. Several of the approaches that have been subject to an optimality check have used comparably large problem instances.

It is difficult to assess the applicability of the different formulations and solution mechanisms. Obviously, it depends on the practical problem characteristics. Earlier models of the railway scheduling problems are to a great extent still applicable, since the structure of the railroad has not changed much. However, whether simplifications and assumptions made earlier are valid today with respect to changes in traffic flows and density is not clear. Moreover, the solution methods have been developed significantly since the access to computational capacity has increased dramatically along with the opportunity to solve larger problems than possible before. The trend of favouring standardised techniques gives an indication of this.

## **6 Conclusions and future research**

The variety of proposals is large, and many researchers have evaluated their approach with simulation experiments using real data. However, few incorporate previous work but instead create own mechanisms. That is, many publications mention related work while few seem to really consider whether it is relevant for their context. Furthermore, the choice of problem formulation and solution mechanism is often neither motivated nor compared to alternative approaches. However, a quantitative benchmark requires the researchers to have access to and use the same problem instances as previous researchers of earlier work. There is thus a need to have and to use publicly available and acknowledged problem instances for the railway scheduling problems as in several problem areas within the operations research community. To our knowledge there are currently none available. Furthermore, several publications do not provide



computational results related to speed or size of problem instance and possible scalability issues. An extended description of the size and characteristics of the practical problem in mind would also facilitate the comparison to other approaches and its applicability for a different setting. As mentioned earlier, it is common to assume that optimality is hard to achieve, while few attempts to do so are described. A comparison of computational results with results from an attempted optimisation (i.e. a lower bound or a gap) would be of interest whether it has been successful or not.

As we could see in the review, new techniques are arising, such as the use of auctions and agent technology. However, the challenges regarding synchronising the (partial) parallel solving of a distributed problem and how to generate and handle the selection of slots need to be presented further as does the impact on computational efficiency.

To conclude; researchers are encouraged to use well-known, common problem instances so that the research community can benchmark approaches. That assumes, however, that such are available. Furthermore, experiments should be carried out with respect to different problem sizes (and related to the practical problem size) and the corresponding computational-efficiency of the mechanism should be presented. Several approaches seem promising, and further experimentation and development would be of great interest. In addition, any attempts to achieve optimum solutions are recommended and the results should be presented. Finally, an extended discussion of the practical viability of the suggested approaches, motivation of the simplifications made and description of the real problems in mind would support conclusions and research results even further.

## 7 Acknowledgements

Prof. Peter Värbrand at Linköping University and Prof. Paul Davidsson and Dr. Jan A. Persson at Blekinge Institute of Technology have provided important comments and inspiring ideas while the Swedish National Rail Administration (Banverket), Blekinge Institute of Technology and the municipality of Karlshamn, Sweden have financed this work.

## References

1. Assad, A.: Models for rail transportation. *Transportation Research Part A*, Vol. 14A. (1980) 205- 220
2. Blum, J., Eskandarian, A.: Domain-specific genetic agents for flow optimization of freight railroad traffic. *Proceedings of the 8th International Conference on Computers in Railways*, June 12-14, Wessex Institute of Technology, Lemnos, Greece (2002a)
3. Blum, J., Eskandarian, A.: Enhancing Intelligent Agent Collaboration for Flow Optimization of Railroad Traffic. *Transportation Research Part A*, Vol. 36. (2002b) 919-930
4. Brewer, P.J., Plott, C.R.: A binary conflict ascending price (BICAP) mechanism for the decentralized allocation of the right to use railroad tracks. *International Journal of Industrial Organization* 14. (1996) 857–886

5. Brännlund, U., Lindberg, P.O., Nöu, A., Nilsson, J-E.: Railway timetabling using Lagrangian relaxation. *Transportation Science*, Vol. 32, No. 4, (1998) 358-369
6. Bussieck, M.R., Winter, T., Zimmermann, U.T.: Discrete optimization in public rail transport. *Mathematical Programming*, Vol. 79, (1997) 415-444
7. Cai, X., Goh, C.J., Mees, A.I.: Greedy heuristics for rapid scheduling of trains on single tracks. *IIE Transactions* 30. (1998) 481-493
8. Caprara, A., Fischetti, M., Toth, P.: Modeling and solving the timetabling problem. *Operations Research*, Vol. 50, No. 5. (2002) 851-861
9. Carey, M.: Extending a train pathing model from one-way to two-way track. *Transportation Research Part B*, Vol. 28B, No. 5. (1994a) 395-400
10. Carey, M.: A model and strategy for train pathing with choice of lines, platforms and routes. *Transportation Research Part B*, Vol. 28B, No. 5. (1994b) 333-353
11. Carey, M., Lockwood, D.: A model, algorithms and strategy for train pathing. *Journal of the Operational Research Society* 46. (1995) 998-1005
12. Carey, M., Carville, S.: Scheduling and platforming trains at busy complex stations. *Transportation Research Part A*, Vol. 37. (2003) 195 –224
13. Chang, S-C., Chung, Y-C.: From timetabling to train regulation-a new train operation model. *Information and Software Technology*, Vol. 47. (2005) 575-585
14. Cheng, Y.: Hybrid simulation for resolving resource conflicts in train traffic rescheduling. *Computers in Industry*, Vol. 35. (1998) 233–246
15. Chiang, T-W., Hau, H-Y., Chiang, H-M., Ko, S-Y., Hsieh, C-H.: Knowledge-based system for railway scheduling. *Data and Engineering* 27. (1998) 289-312
16. Chiu, C.K., Chou, C.M., Lee, J.H.M., Leung, H.F., Leung, Y.W.: A Constraint-Based Interactive Train Rescheduling Tool. *Constraints* 7. (1996) 167-198
17. Cordeau, J-F., Toth, P., Vigo, D.: A survey of optimization models for train routing and scheduling. *Transportation Science*, Vol. 32, No. 4. (1998) 380-404
18. Crainic, T.G., Laporte, G.: Planning models for freight transportation. *European Journal of Operational Research* 97. (1997) 409-438
19. Crainic T.G.: A survey of optimization models for long-haul freight transportation, Publication CRT-98-67. Centre de recherche sur les transports. Université de Montréal., Canada (1998)
20. D'Ariano, A., Pranzo, M.: A real time train dispatching system based on blocking time theory. *Proceedings of the 8th TRAIL 8th Annual Congress 2004 - A world of transport, infrastructure and logistics*. Delft, DUP Science. ISBN 90-407-2555-1. (2004) 129-152
21. Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J., Wernstedt, F.: An analysis of agent-based approaches to transport logistics. *Transportation Research Part C: Emerging Technologies*, Vol. 13, Issue 4. (2005) 255-271
22. Dorfman, M.J., Medanic, J.: Scheduling trains on a railway network using discrete-event model of railway traffic. *Transportation Research Part B*, Vol. 38. (2004) 81-98
23. Fischer, M.L.: The Lagrangian relaxation method for solving integer programming problems. *Management Science* 27. (1981) 1-18
24. Fukumori, K.: Fundamental scheme for train scheduling: Application of range-constriction search. A.I. Memo No. 596. Massachusetts Institute of Technology Artificial Intelligence Laboratory, USA (1980)
25. Ghoseiri, K., Szidarovszky, F., Asgharpour, M.J.: A multi-objective train scheduling model and solution. *Transportation Research Part B*, Vol. 38. (2004) 927-952
26. Gibson, S.: Allocation of capacity in the rail industry. *Utilities Policy* 11. (2003) 39–42
27. Glover, F., Laguna, M.: *Tabu Search*, Kluwer Academic Publishers, Dordrecht, Netherlands (1997)
28. Hallowell, S., Harker, P.: Predicting on-time performance in scheduled railroad operations: Methodology and application to train scheduling. *Transportation Research Part A*, Vol. 32, No. 4. (1998) 279-296

29. Hellström, P.: Analysis and evaluation of systems and algorithms for computer-aided train dispatching. Licentiate thesis, Uppsala University, Sweden (1998)
30. Hellström, P., Sandblad, B., Frej, I., Gideon, A.: An evaluation of algorithms for Computer-Aided Train Dispatching. In: Hellström, P.: Analysis and evaluation of systems and algorithms for computer-aided train dispatching. Licentiate thesis, Uppsala University, Sweden (1998)
31. Higgins, A., Kozan, E.: Heuristic techniques for single line train scheduling. *Journal of Heuristics* 3. (1997) 43-62
32. Ho, T.K., Yeung, T.H.: Railway junction traffic control by heuristic methods. *IEEE Proceedings Electr. Power Appl.*, Vol. 148, No. 1. (2001) 77-84
33. Ingolitto, L., Barber, F., Tormos, P., Lova, A., Salido, M., Abril, M.: An Efficient Method to Schedule New Trains on a Heavily. In: (eds.) Reyes, C.A., Gonzales, J.A.): *Proceedings of Advances in Artificial Intelligence - IBERAMIA 2004, 9<sup>th</sup> Ibero-American Conference on AI*, Puebla, Mexico, November 22-26, 2004. *Lecture Notes in Computer Science*. Springer Verlag. (2004) 164-173
34. Isaai, M.T., Singh, M.G.: Hybrid applications of constraint satisfaction and meta-heuristics to railway timetabling: A comparative study. *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews*, Vol. 31, No. 1. (2001) 87-95
35. Isacsson, G., Nilsson, J-E.: An experimental comparison of track allocation mechanisms in the railway industry. *Journal of Transport Economics and Policy*, Vol. 37, No. 3. (2003) 353-381
36. Iyer, R.V., Gosh, S.: DARYN – A distributed decision-making algorithm for railway networks: Modeling and Simulation. *IEEE Transactions on Vehicular Technology*, Vol. 44, No. 1. (1995) 180-191
37. Jia, L-M., Zhang, X-D.: Distributed intelligent railway traffic control based on fuzzy decisionmaking. *Fuzzy Sets and Systems*, Vol. 62, Issue 3. (1993) 255-265
38. Jovanovic, D.: Improving railroad in-time performance: models, algorithms and applications. Ph. D. thesis, Decision Science Department, Wharton School, University of Pennsylvania, USA (1989)
39. Kirkpatrick, S.: Optimization by Simulated Annealing - Quantitative Studies. *Journal of Statistical Physics* 34. (1984) 975-986
40. Kitahara, F., Kera, K., Bekki, K.: Autonomous decentralized traffic management system. *Proceedings of International workshop of Autonomous Decentralized Systems 2000*. (2000) 87-91
41. Koch, W.: A low cost tool for rescheduling in regional public transport systems. *Proceedings of the 9<sup>th</sup> IFAC Symposium on Control in Transport Systems 2000*. June 13-15. Braunschweig, Germany (2000)
42. Komaya, K., Fukuda, T.: A knowledge-based approach for railway scheduling. *Proceedings for the Seventh IEEE Conference on Artificial Intelligence for Applications*. IEEE Comput. Soc. Press. (1991) 404-411
43. Kraay, D., Harker, P.: Real-time scheduling of freight railroads. *Transportation Research Part B*, Vol. 29B, No. 3. (1995) 213-229
44. Kroon, L.G., Romeijn, H.E., Zwaneveld, P.J.: Routing trains through railway stations: complexity issues. *European Journal of Operational Research* 98. (1997) 485-498
45. Lamma, E., Mello, P., Milano, M.: A distributed constraint-based scheduler. *Artificial Intelligence in Engineering* 11. (1997) 91-105
46. Larroche, Y., Moulin, R., Gauyacq, D.: SEPIA: a real-time expert system that automates train route management. *Control Engineering Practice*, Vol. 4, Issue 1. (1996) 27-34
47. Lee, T.S, Gosh, S.: Stability of RYNSORD – A decentralized algorithm for railway networks under perturbations. *IEEE Transactions on Vehicular Technology*, Vol. 50, No.1. (2001) 287-301

48. Liebchen, C., Moehring, R.: The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables - and Beyond. Proceedings of the 9<sup>th</sup> International Conference on Computer-Aided Scheduling of Public Transport (CASPT). San Diego, USA (2004)
49. Lin, H-C., Hsu, C-C.: An interactive train Scheduling workbench based on Artificial Intelligence. Proceedings from International Conference on Tools with Artificial Intelligence (ICTAI) 1994. (1994) 42-48
50. Lindner, T.: Train Schedule Optimization in Public Rail Transport. Ph.D. thesis. Technische Universität Braunschweig, Germany (2000)
51. Mackenzie S.: Train timetabling on complex networks. Proceedings from the Conference on Railway Engineering 2000 (CORE2000), Adelaide, Australia (2000)
52. Missikoff, M.: An object-oriented approach to an information and decision support system for railway traffic. Proceedings for the 1997 First International Conference on Knowledge-Based Intelligent Electronic Systems - KES '97, Vol. 2. (1997) 633-641
53. Nilsson, J-E.: Allocation of track capacity: Experimental evidence on the use of priority auctioning in the railway industry. International Journal of Industrial Organization 17. (1999) 1139-1162
54. Oliveira, E., Smith, B. M.: A hybrid constraint-based method for single-track railway scheduling problem. Report 2001:4, Research report series. School of Computing, University of Leeds, UK (2001)
55. Pacciarelli, D., Pranzo, M.: A Tabu Search algorithm for the railway scheduling problem. Proceedings of MIC '2001- 4th Metaheuristics International Conference, Porto, Portugal. (2001) 159-163
56. Pacciarelli, D., Mascis, A., Pranzo, M.: Scheduling Models for Short-Term Railway Traffic Optimization. Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport (CASPT). San Diego, USA (2004)
57. Pardalos, P.M., Resende, G.C.: Handbook of applied optimization. Oxford University Press, N.Y, USA (2002)
58. Peeters, L.: Cyclic Railway Timetable Optimization. Ph.D. thesis. ERIM, Rotterdam School of Management, Erasmus University Rotterdam, Netherlands (2003)
59. Petersen, E., Taylor, A.: A structured model for rail line simulation and optimization. Transportation Science, Vol. 16, No. 2. (1982) 192-206
60. Ping, L., Axin, N., Limin, J., Fuzhang, W.: Study on intelligent train dispatching. IEEE Intelligent Transportation Systems Conference Proceedings. Oakland, USA (2001)
61. Pudney, P., Wardrop, A.: Generating train plans with Problem Space Search. Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport (CASPT). San Diego, USA (2004)
62. Reeves, C.R. (ed.): Modern heuristic techniques for combinatorial problems. McGraw-Hill, London, UK (1995)
63. Şahin, İ.: Railway traffic control and train scheduling based on inter-train conflict management. Transportation Research Part B, Vol. 33. (1999) 511-534
64. Şahin, G., Ahuja, R.K., Cunha, C.B.: New approaches for the train dispatching problem. submitted to Transportation Science (2005)
65. Salim, V., Cai, X.: A genetic algorithm for railway scheduling with environmental considerations. Environmental Modeling & Software, Vol. 12, No. 4. (1997) 301-309
66. Sauder, R., Westerman, W.: Computer aided train dispatching: Decision support through optimization. Interfaces 13. (1983) 24-37
67. Schaefer, H., Pferdmenges, S.: An expert system for real-time train dispatching. Computers in Railway (COMPRAIL) 94. Madrid, Spain (1994)
68. Shoji, S., Igarashi, A.: New trends of train control and management systems with real-time and non real-time properties. Proceedings for ISADS 97 - the Third International Symposium on Autonomous Decentralized Systems 1997. IEEE Comput. Soc. Press. (1997) 319-326

69. Spzigel, B.: Optimal train scheduling on a single track railway. In: Ross, M. (ed.): OR'72. North Holland Publishing Co. (1973) 343-352
70. Törnquist, J., Persson, J.A.: N-tracked railway traffic re-scheduling during disturbances: theoretical and practical implications. Presented at INFORMS Annual Conference and RASIG session November 13, 2005, San Francisco, USA. Accepted for publication in Transportation Research Part B during 2006 (2005).
71. Vernazza, G., Zunino, R.: A distributed intelligence methodology for railway traffic control. IEEE Transactions on Vehicular Technology, Vol. 39, Issue 3. (1990) 263-270
72. Vieira, P., Neto, L.E., Bessa, E., Gomide, F.: Railway dispatch and control. Proceedings for the 18th International Conference of the North American Fuzzy Information Processing Society (NAFIPS99), New York, USA. (1999) 134-138
73. Wegele, S., Schnieder, E.: Dispatching of train operations using genetic algorithms. Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport (CASPT), San Diego, USA (2004)
74. Wooldridge, M.: An introduction to MultiAgent Systems. John Wiley and Sons Ltd, Chichester, England, UK (2002)
75. Zhou, X., Zhong, M.: Bicriteria train scheduling for high-speed passenger railroad planning applications. European Journal of Operational Research, Vol. 167. (2005) 752-771
76. Zwaneveld, P.J., Kroon, L.G., Romeijn, E.H., Salomon, M., Dauzère-Pérès, S., van Hoesel, S. P.M., Ambergen, H.W.: Routing Trains Through Railway Stations: Model Formulation and Algorithms. Transportation Science 30. (1996) 181-194

## Appendix

**Table 8.** Summary of approaches for tactical scheduling, where each line represents an approach. The first parenthesis in the second column specifies if the approach considers a line (L) or can represent a network (N). The second parenthesis specifies what type of non-station segments that can be handled, i.e. if the segments can have bi-directional (B) tracks or only uni-directional (U) and the maximum number of tracks that are possible for a segment to include; single (S), double (D) or an arbitrary number (N). The third parenthesis specifies in the same way how segments that represent stations may look like. ‘-’ means that information is missing and ‘∞’ means that the capacity (number of tracks) is unrestricted.

Approach	Infrastructure representation	Objective	Solution mechanism	Control	Evaluation level	Problem instance and size
Petersen, Taylor (1982)	(L)(B:BN)(S:BN)	Max performance	Heuristic solving one conflict at a time based on rules/priorities	C	2	Line: 20 segments, 51 trains
Salim, Cai (1997)	(L)(B:BS)(S:B∞)	Min waiting and stopping costs	Determine meets w. GA	C	3	Line: 12 segments, 14 trains
Nilsson (1999), Isacson, Nilsson (2003)	(L)(B:BS)(S:B∞)	Max profit	Select a feasible combinations of slots via auctions	D	2	Line: 2 stations, 1 block forming 11 slots (markets) and 6 bidders
Blum, Eskandarian (2002a, 2002b)	(L)(B:BS)(S:B∞)	Max profit	Reserve tracks for trains in order of highest profit using MAS and heuristics (GA, critical path analysis)	C	3	Line: 200 segments, 64 trains
Isaai, Singh (2001)	(L)(B:BS)(S:BN)	Min waiting times	Determine the visiting order of trains on stations based on e.g. the earliest time of resource release principle.	C	3	Line: 51 stations, 40 single-, 10 double-tracked segments, 22 trains
Ingolotti et al (2004)	(L)(B:BS,UD)(S:B∞)	Min average traversal time for each new scheduled train	Determine visiting order on segments using a CSP formulation where new trains are added to an existing timetable and each conflicting track request is solved according to priority values and a back-tracking algorithm.	C	3	Line: 65 segments, 81 trains
Lin, Hsu (1994)	(L)(B:BS, UD)(S:BN)	Min delay (of sacrificed train) when solving a local conflict	Start with infeasible schedule and apply a 5-rule-based conflict solver w. earliest-conflict first that shift the slots (i.e. arrival and departure to stations)	C	3	Line: 102 stations, 350 trains
Fukumori (1980)	(L)(B:UD)(S:UN)	Min total weighted delay penalty	Depth-first search branching on train priority to shift departure times from stations allowing overtaking and determine order of trains	C	1	-
Chiang et al. (1998)	(L)(B:BS,UD)(S:BN)	Valid timetable	Repair non-conflict free timetable by set of repair-methods and earliest-first principle to modify overtakes and meets	C	3	Line: 102 stations, 350 trains
Caprara et al (2002)	(L)(B:US)(S:B∞)	Min travel time exceeding ideal run time	Modify train order and overtakes by Lagrangian relaxations and subgradient optimization	H	3	Line: 16 or 48 stations, 221 or 54 trains

**Table 9.** Continued summary of approaches for tactical scheduling, where each line represents an approach. The first parenthesis in the second column specifies if the approach considers a line (L) or can represent a network (N). The second parenthesis specifies what type of non-station segments that can be handled, i.e. if the segments can have bi-directional (B) tracks or only uni-directional (U) and the maximum number of tracks that are possible for a segment to include; single (S), double (D) or an arbitrary number (N). The third parenthesis specifies in the same way how segments that represent stations may look like. ‘-’ means that information is missing and ‘∞’ means that the capacity (number of tracks) is unrestricted.

Approach	Infrastructure representation	Objective	Solution mechanism	Control	Evaluation level	Problem instance and size
Chang, Chung (2005)	(L)(B:US)(S:US)	Min total time in system, passenger travel times and deviation from initial schedule	Decide visiting order of trains on stations using GA	C	3	Line: 30 stations, 100 trains
Oliviera, Smith (2001)	(N)(B:BN)(S:B-∞)	Min total tardiness	Determine order of trains using B&B and hill climbing	C	3	Line: 14 segments, 49 trains
Brewer, Plott (1996)	(N)(B:BN)(S:B-∞)	Max profit	Select a feasible combinations of slots using auctions	D	2	Line: 2 blocks, 9 train slots, 10 agents
Brännlund et al. (1998)	(N)(B:BN)(S:BN)	Max profit	Allocate discrete time units of segment to trains using Lagrangian relaxations	H	3	Line: 17 stations, 16 BS segments, 26 trains
Ghoseiri et al. (2004)	(N)(B:BN)(S:BN)	Multiple objectives: 1) Min fuel consumption, 2) Min passenger time	Determine visiting order on segments and stations allowing meets and overtakes w. an e-constraint method and distance-based method	C	2	Line: 24 segments, 6 trains
Mackenzie (2000)	(N)(B:BN)(S:BN)	Min weighted tardiness penalty function	Allocate discrete time units of blocks to trains using (1) Lagrangian relaxations, (2) Problem Space Search local search heuristics	H, C	2	Line: 60 segments, 34 trains
Pudney, Wardop (2004)	(N)(B:BN)(S:BN)	Min total lateness cost	Allocate start times at segments by a sorting algorithm and Problem space search perturbing the data	C	3	Network: 35 meet points, 260 trains
Pacciarelli, Pranzo (2001)	(N)(B:BN)(S:BN)	Min total travel time	Decide visiting order of trains on segments and stations using TS	C	3	Network: -
Carey and Lockwood (1995), Carey (1994a-b)	(N)(B:BS)(S:B-∞)	Minimise travel and waiting time costs	Decide visiting order of trains on segments and branching on which train to next path	C	2	Line: 10 stations, 28 segments, 10 trains
Zhou, Zhong (2005)	(N)(B:US)(S:B-∞)	Min interdeparture time, total travel time	Modify overtakes by B&B and Beam search	C	3	Line: 17 segments, 36 trains

**Table 10.** Summary of approaches for operational scheduling, where each line represents an approach. The first parenthesis in the second column specifies if the approach considers a *line* (L) or can represent a *network* (N). The second parenthesis specifies what type of non-station segments that can be handled, i.e. if the segments can have bi-directional (B) tracks or only uni-directional (U) and the maximum number of tracks that are possible for a segment to include; *single* (S), *double* (D) or an arbitrary number (N). The third parenthesis specifies in the same way how segments that represent stations may look like. ‘-’ means that information is missing and ‘∞’ means that the capacity (number of tracks) is unrestricted.

Approach	Infrastructure representation	Objective	Solution mechanism	Control	Evaluation level	Problem instance and size
Jovanovic (1989)	(L)(B:BS)(S:BD)	Min tardiness costs	Fixating where trains overtake and order of trains in each direction, while deciding where trains in opposite direction meet using B&B incorporating heuristics	C	4	Line: 130 meet points, 200 trains
Cai et.al(1998)	(L)(B:BS)(S:BD)	Min stopping and waiting costs	Modify train order and times on tracks using a Greedy algorithm w. a set of conflict-dependent subroutines	C	2	Line: 10 segments, 99 trains
Sauder, Westerman (1983)	(L)(B:BS)(S:BS)	Trains reach destination within a time interval and min total delay cost	Meet-plan decisions tree constructed by a branching algorithm solving conflicts by arranging meets (one at a time)	C	5	Line:
Higgins et.al (1997)	(L)(B:BS)(S:UD)	Min total weighted travel time	Solve conflicts by set of routines (LS, GA,TS, and hybrids).	C	3	Line: 14 UD stations, 13 US segments, 49 trains.
Dorfman, Medanic (2004)	(N)(B:BN)(S:BN)	Min energy costs	Determine next slot to occupy track resource w. a Greedy heuristic	C	2	Line: 8 segments, 36 trains
Szpigel (1973)	(N)(B:BS)(S:B∞)	Min weighted travel times	Determine visiting order of trains on segments w. various branching procedures	C	2	Line: 5 segments, 10 trains
Kraay and Harker (1995)	(N)(B:BS)(S:B∞)	Min tardiness costs	Column generation to find a bound, Applying local search heuristics to an LP-model (fixating/ignoring) the binary variables specifying where trains meet and overtake.	C	3	Line: 11 meet points, 16 segments



**Table 11.** Summary of approaches for re-scheduling, where each line represents an approach. The first parenthesis in the second column specifies if the approach considers a *line* (L) or can represent a *network* (N). The second parenthesis specifies what type of non-station segments that can be handled, i.e. if the segments can have bi-directional (B) tracks or only uni-directional (U) and the maximum number of tracks that are possible for a segment to include; *single* (S), *double* (D) or an arbitrary number (N). The third parenthesis specifies in the same way how segments that represent stations may look like. ‘-’ means that information is missing and ‘∞’ means that the capacity (number of tracks) is unrestricted.

Approach	Infrastructure representation	Objective	Solution mechanism	Control	Evaluation level	Problem instance and size
Hellström et.al(1998)	(L)(B:BS)(S:BD)	Min tardiness costs	Fixating where trains overtake and order of trains in each directions, while deciding where trains in opposite direction meet using a B&B procedure	C	3	Line: 23 single-tracked segments, 20 trains
Sahin (1999)	(L)(B:BS)(S:BD)	Min delay of the two local conflict resolutions	Solve each conflict (pair of conflicting track requests) by applying an approximative look-ahead heuristic comparing the effectiveness of the two alternative solutions (delaying train i or train j)	C	2	Line: 19 stations/meet points, 20 trains
Cheng (1998)	(L)(B:UD)(S:BN)	Solve conflicts based on priority	Decide order of use of resources w. priority-based sorting and simulation	C	2	Line: 3 stations, 2 uni-directional double-tracked segments, 8 trains
Chiu et.al. (1996)	(L)(B:BS)(S:BS)	Min largest delay per train	With varying heuristic strategies such as "choose smallest delay change first" the order of trains on segments are modified.	C	3	Line: several stations, BS segments
Ping et.al (2001)	(L)(B:UD)(S:B∞)	Min total delay	Determine visiting orders on segments and start times using GA	C	3	Line: Double-tracked with 14 stations, 250 trains
Komaya, Fukuda (1991)	(L)(B:US)(S:-)	Min total accumulated delay for all trains	Decide order of trains on segment between two ordered stations using an expert system	C	4	Line: 14 stations, 40 trains
Jia, Zhang (1993)	(L)(B:US)(S:US)	Based on priorities	Decide order of trains on stations w. priority-based sorting	L	3	Line: 12 stations, 12 trains
Vernazza, Zunino (1990)	(N)(B:-)(S:-)	Most urgent conflicts dealt with first	Allocate tracks to trains by trains "bidding" the capacity to the local DCs that handles and allocates based on local urgency and priority rules	L	3	Network:
Shoji, Igarashi (1997), Kitahara et.al. (2000)	(N)(B:-)(S:-)	-	-	D	5	Network: 17 lines, 250 stations, 6200 trains
Iyer, Gosh (1995);Lee, Gosh (2001)	(N)(B:-)(S:-)	Each train minimises its total travel time	Each train requests for N tracks ahead and negotiates with resp. infrastructure owner (i.e. stations) to grant or refuse the request	L	3	A network: 50 stations, 84 segments.
Viera et. al. (1999)	(N)(B:B-)(S:B-)	Several objectives	Decide meets and overtakes based on priorities from a fuzzy rule-base	C	4	Line: Single-tracked segments w. 43 sidings.

**Table 12.** Continued summary of approaches for re-scheduling, where each line represents an approach. The first parenthesis in the second column specifies if the approach considers a *line* (L) or can represent a *network* (N). The second parenthesis specifies what type of non-station segments that can be handled, i.e. if the segments can have bi-directional (B) tracks or only uni-directional (U) and the maximum number of tracks that are possible for a segment to include; *single* (S), *double* (D) or an arbitrary number (N). The third parenthesis specifies in the same way how segments that represent stations may look like. ‘-’ means that information is missing and ‘∞’ means that the capacity (number of tracks) is unrestricted.

Approach	Infrastructure representation	Objective	Solution mechanism	Control	Evaluation level	Problem instance and size
Hellström et.al(1998)	(L)(B:BS)(S:BD)	Min tardiness costs	Fixating where trains overtake and order of trains in each directions, while deciding where trains in opposite direction meet using a B&B procedure	C	3	Line: 23 single-tracked segments, 20 trains
Missikoff (1997)	(N)(B:BN)(S:-)	Min local weighted delay costs	Heuristics (Hillclimbing, A-search) that finds a conflict, solves it locally with respect to the local delay cost and approximative cost for global costs	L	3	Line: double-tracked
Wegele, Schnieder (2004)	(N)(B:BN)(S:BN)	Min passenger annoyance for platforms changes and delays	Determine train visiting times at nodes (signals and switches) using B&B improved by GA	C	3	Network: 104 stations, 1000 trains
Törnquist, Persson (2005)	(N)(B:BN)(S:BN)	Min total delay, min total delay costs	Determine train visiting order on segments using B&B (of IP solver CPLEX)	C	3	Network: 130 stations, 136 segments, 93 trains
Ho, Yeung (2001)	(N)(B:BN)(S:BN)	Min total weighted delay	Decide order of track usage using TS, SA, GA.	C	2	-
Lamma et.al. (1997)	(N)(B:BN)(S:BN)	Min train delays	Local schedulers allocate resources to train by using priority rules	D	3	Line:
Schaefer, Pferdenges (1994)	(N)(B:BN)(S:BN)	Min weighted delays	An expert-system w. rule-based greedy algorithm building a decision-tree w. breadth-first search and primary conflicts on top level	C	3	Line: Single- and double-tracked segments for traffic between 3 and 24-hours
Sahin et.al(2005)	(N)(B:BN)(S:BN)	Min total delay	IP-based heuristics limiting max allowed delay/train, "local" rule-based simulation heuristic, Greedy enumeration heuristic	C	3	Line: 25 stations, 24 single-tracked segments, 25 trains over 24-hours
D'Ariano, Pranzo (2004), Pacchiarelli et.al (2004)	(N)(B:BN)(S:BN)	Min the maximum secondary delay	Create a non-valid timetable, apply a greedy conflict resolution algorithm that chooses high priority conflicts first and solves them according to "most affected train gets priority", finally a pre-processing phase takes over.	C	3	Line: 21 US segments, 4 trains
Koch (2000)	(N)(B:BS)(S:-)	Min total delay cost	A* search	C	4	Line: 23 stations, single-tracked segments, 23 trains
Larrouche et.al.(1996)	(N)(B:UN)(S:-)	Multiple, context-dependent, tacit and subjective objectives	Search for a resource for each train slot using an expert system	C	4	Network: 250 trains

# Line Planning with Minimal Traveling Time

Anita Schöbel and Susanne Scholl

Institut für Numerische und Angewandte Mathematik,  
Georg-August-Universität Göttingen  
Lotzestrasse 16-18, 37083 Göttingen, Germany  
{schoebel}{sscholl}@math.uni-goettingen.de

**Abstract.** An important strategic element in the planning process of public transportation is the development of a line concept, i.e. to find a set of paths for operating lines on them. So far, most of the models in the literature aim to minimize the costs or to maximize the number of direct travelers. In this paper we present a new approach minimizing the travel times over all customers including penalties for the transfers needed. This approach maximizes the comfort of the passengers and will make the resulting timetable more reliable. To tackle our problem we present integer programming models and suggest a solution approach using Dantzig-Wolfe decomposition for solving the LP-relaxation. Numerical results of real-world instances are presented.

**Keywords.** Line planning, real-world problem, integer programming, Dantzig-Wolfe decomposition

## 1 Motivation and related literature

In the strategic planning process of a public transportation company one important step is to find a suitable line concept, i.e. to define the routes of the bus or railway lines. Given a public transportation network  $PTN = (S, E)$  with its set of stations  $S$  and its set of direct connections  $E$ , a *line* is defined as a path in this network. The *line concept* is the set of all lines offered by the public transportation company, together with their *frequencies*, where the frequency  $f_l$  of a line  $l$  contains the number of vehicles serving line  $l$  within the planning period considered. The *frequency of an edge*  $e$ , on the other hand, is the number of vehicles running along the edge.

The line planning problem has been well studied in the literature. For an early contribution we refer to Dienst, see [1]. The many models given after this time can be roughly classified into the following two types. In a *cost-oriented approach* the goal is to find a line concept serving all customers and minimizing the costs for the public transportation company. The basic cost model has been suggested in Claessens et al., see [2], where a binary (linear) programming formulation has been given. A solution approach by branch and cut has been developed in [3]. In [4] an alternative formulation with integer variables has been proposed. In [5]

Bussieck et al. present a fast solution approach combining nonlinear techniques with integer programming.

In [6] and [7] the authors get rid of the assumption that the passengers are assigned a priori for example by modal split to different types of trains. This is done by assigning a certain *type* to every node in the PTN, representing for example the size of the station. Then the type of a line determines the stations they pass. For example a line of type 1 stops at every station it passes, a line of type 2 will not halt at a station of type 1 but at every station of type 2 or higher. Several models, correctness and equivalence proofs are presented.

Recently, a fast heuristic variable fixing procedure which combines nonlinear techniques with integer programming is proposed in [5].

[6] presents a model that reconsiders the stations at which the trains stop for a given line plan. This model is used to determine the halting stations in such a way that the total travel time of passengers is minimized. Lagrangian relaxation is used to find lower bounds on this problem. Preprocessing and tree search techniques augment the efficiency of the branch&bound framework.

A second class of models are the *customer-oriented approaches*. In the *direct travelers approach* by Bussieck et al. [8] (see also [4]) the goal is to maximize the number of direct travelers (i.e. customers that need not change the line to reach their destination). As constraint, the number of vehicles running along an edge is restricted for each edge in the PTN, i.e. upper and lower bounds on the allowed frequencies on each edge are taken into account. The model maximizes the amount of one group of customers but without considering the remaining ones which might have very many transfers during their trips. It also does not take into account the travel times for the customers: Sometimes it is preferable to have a transfer but reach the destination earlier instead of sitting in the same line for the whole trip but having a large detour. This is done in recent models by [9,10,11,12] in which the goal is to design lines in such a way that the traveling time of the customers is minimized. The special case of locating one single line so as to maximize the number of passengers is treated in [13]. None of these models includes the number of transfers of customers in the objective function, which will be the basic feature of the model presented in this paper.

Another approach is to take into account that the behavior of the customers depends on the design of the lines. A first cost-oriented model including such demand changes was treated with simulated annealing in two diploma theses in cooperation with *Deutsche Bahn*, see [14,15]. Finally, we want to mention the work by Quak [16] in which lines are not taken out of a given line pool as done by all other publications mentioned here, but constructed from the scratch.

In our work we develop a new model which allows to sum over all travel times over all customers including penalties for the transfers needed. The first ideas for this model have been presented in [17]. Here, we also show how different frequencies of the lines can be taken into account. The remainder of the paper is organized as follows. In Section 2 we introduce the new line planning model, discuss its complexity in Section 3 and then describe and discuss five integer pro-

gramming models in Section 4. We present two ways to solve the LP-relaxation, one based on Dantzig-Wolfe decomposition (see Section 5). Finally, we present numerical results based on a real-world application of German Rail (DB).

## 2 Basic definitions

A *public transportation network* is a finite, undirected graph  $PTN = (S, E)$  with a node set  $S$  representing stops or stations, and an edge set  $E$ , where each edge  $\{u, v\}$  indicates that there exists a direct ride from station  $u$  to station  $v$  (i.e., a ride that does not pass any other station in between). For each edge  $\{u, v\}$  we assume that the driving time  $t_{uv}$  is known.

We assume the PTN as given and fixed. We further assume that a *line pool*  $\mathcal{L}$  is given, consisting of a set of paths in the PTN. Each line  $l \in \mathcal{L}$  is specified by a sequence of stations, or, equivalently, by a sequence of edges. Let  $E(l)$  be the set of edges belonging to line  $l$ . Given a station  $u \in S$  we furthermore define  $\mathcal{L}(u) = \{l \in \mathcal{L} : u \in l\}$  as the set of all lines passing through  $u$ .

Moreover, let  $\mathcal{R} \subseteq S \times S$  denote the set of all origin-destination pairs  $(s, t)$  where  $w_{st}$  is the number of customers wishing to travel from station  $s$  to station  $t$ .

The line planning problem then is to choose a subset of lines  $L \in \mathcal{L}$ , together with their frequencies, which

- allows each customer to travel from its origin to its destination,
- is not too costly, and
- minimizes the “inconvenience” for the customers.

In the literature, the main customer-oriented approach dealing with the inconvenience of the customers is the approach of [4] (see also [8]) in which the number of direct travelers is maximized. In our paper, however, we deal with the sum of *all* transfers over all customers. On a first glance, the problem to minimize the number of transfers seems to be similar to maximizing the number of direct travelers, but it can easily be demonstrated that both models are in fact different.

Note that considering the number of transfers only would lead to solutions with very long lines, serving all origin-destination pairs directly but having large detours for the customers. To avoid this we determine not only a line concept, but also a path for each origin-destination pair and count the number of transfers **and** the length of the paths in the objective function. This is specified next.

Given a set of lines  $L \subseteq \mathcal{L}$ , a customer can travel from its origin  $s$  to its destination  $t$ , if there exists an  $s$ - $t$ -path  $P$  in the PTN only using edges in  $\{E(l) : l \in L\}$ . The “inconvenience” of such a path is then approximated by the weighted sum of the traveling time along the path and the number of transfers, i.e.

$$\text{inconvenience}(P) = k_1 \text{Time}_P + k_2 \text{Transfers}_P.$$

On the other hand, the cost of the line concept  $L \subseteq \mathcal{L}$  is calculated by adding the costs  $C_l$  for each line  $l \in L$ , assuming that such costs  $C_l$  are known beforehand.

The *line planning problem* hence is to find a feasible set of lines  $L \subseteq \mathcal{L}$  together with a path  $P$  for each origin-destination pair, such that the costs of the line concept do not exceed a given budget  $B$  and such that the sum of all inconveniences over all paths is minimized.

Since the capacity of a vehicle is not arbitrarily large, we have to extend the basic problem to include frequencies of the lines. This makes sure that there are enough vehicles along each edge to transport all passengers. If each origin-destination pair can be served, the line concept is called *feasible*. We remark that often the number of vehicles running along the same edge is also bounded from above, e.g., for safety reasons.

### 3 Complexity Results

In this section we first show that the line planning problem as defined above is NP-hard, even in a very simple case, corresponding to  $k_1 = 0$  in the above definition.

**Theorem 1.** *The line planning problem is NP-complete, even if*

- *we only count the number of transfers in the objective function,*
- *the PTN is a linear graph.*
- *all costs  $C_l$  are equal to one.*

*Proof.* In the decision version, the line planning problem in the above case can be written as follows:

Given a graph  $\text{PTN} = (S, E)$  with weights  $c_e$  for each  $e \in E$ , origin-destination pairs  $\mathcal{R}$ , and a budget  $B$ , does there exist a feasible set of  $B$  lines with less than  $K$  transfers?

We reduce the set covering problem to the line planning problem: Given a set covering problem in its integer programming formulation

$$\min\{1_n x : Ax \geq 1_m, x \in \{0, 1\}^n\}$$

with an 0-1  $m \times n$  matrix  $A$ , and  $1_k \in \mathbb{R}^k$  the vector with a 1 in each component, we construct a line planning problem as follows:

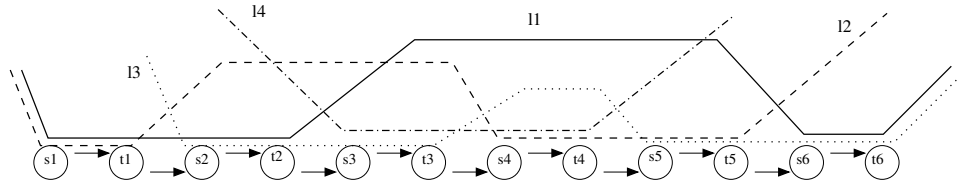
We define the PTN as a linear graph with  $2m$  nodes  $S = \{s_1, t_1, s_2, t_2, \dots, s_m, t_m\}$  and edges  $E = \{(s_1, t_1), (t_1, s_2), (s_2, t_2), (t_2, s_3), \dots, (s_m, t_m)\}$ . We define an origin-destination pair for each row of  $A$ ,

$$\mathcal{R} = \{(s_i, t_i) : i = 1, \dots, m\}.$$

For column  $j$  of  $A$  we construct a line  $l_j$  passing through nodes  $s_i$  and  $t_i$  if  $a_{ij} = 1$ .

As an example, Figure 1 shows the line planning problem obtained from a set covering problem with

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$



**Fig. 1.** Construction of the line planning problem in the proof of Theorem 1.

Setting  $K = 0$  we hence have to show that a cover with less than  $B$  elements exists if and only if the line planning problem has a solution in which all passengers can travel without changing lines. Due to our construction this is true and hence the theorem holds.

A question that might arise in this context, is what happens if the lines need not be chosen from a given line pool, but can be constructed as any path. Some of the basic cost models become very easy in this case, but unfortunately, the complexity status of the line planning problem treated in this paper does not change which can be shown by reduction to the Hamiltonian path problem (see [18]).

## 4 Models for the line planning problem

To model the line planning problem as integer program we use the PTN to construct a directed graph, the so-called *change&go network*  $G_{CG} = (\mathcal{V}, \mathcal{E})$  as follows:

We extend the set  $S$  of stations to a set  $\mathcal{V}$  of nodes with nodes representing either station-line-pairs (change&go nodes:  $\mathcal{V}_{CG}$ ) or the origins and destinations of the customers (origin-destination nodes:  $\mathcal{V}_{OD}$ ), i.e.  $\mathcal{V} := \mathcal{V}_{CG} \cup \mathcal{V}_{OD}$  with

- $\mathcal{V}_{CG} := \{(s, l) \in S \times \mathcal{L} : l \in \mathcal{L}(s)\}$  (set of all station-line-pairs)
- $\mathcal{V}_{OD} := \{(s, 0) : (s, t) \in \mathcal{R} \text{ or } (t, s) \in \mathcal{R}\}$  (origin-destination nodes)

The new set of edges  $\mathcal{E}$  consists of directed edges between nodes of the same stations (representing that customers board or unboard a vehicle or change lines) and edges between nodes of the same line (representing the driving activities):

$$\mathcal{E} := \mathcal{E}_{change} \cup \mathcal{E}_{OD} \cup \mathcal{E}_{go}$$

with

- $\mathcal{E}_{change} := \{((s, l_1), (s, l_2)) \in \mathcal{V}_{CG} \times \mathcal{V}_{CG}\}$  (changing edges)
- $\mathcal{E}_l := \{((s, l), (s', l)) \in \mathcal{V}_{CG} \times \mathcal{V}_{CG} : (s, s') \in E\}$  (driving edges of line  $l \in \mathcal{L}$ )
- $\mathcal{E}_{go} := \bigcup_{l \in \mathcal{L}} \mathcal{E}_l$  (driving edges)
- $\mathcal{E}_{OD} := \{((s, 0), (s, l)) \in \mathcal{V}_{OD} \times \mathcal{V}_{CG} \text{ and } ((t, l), (t, 0)) \in \mathcal{V}_{CG} \times \mathcal{V}_{OD} : (s, t) \in \mathcal{R}\}$  (origin-destination edges)

We define weights on all edges  $e \in \mathcal{E}$  of the change&go network representing the inconvenience customers have when using edge  $e$ . Given a set of lines  $L \subseteq \mathcal{L}$  we then can determine the lines the customers are likely to use by calculating a shortest path in the change&go network for each single origin-destination pair. Therefore the choice of the edge costs  $c_e$  is very important. We give two examples:

1. Customers only count transfers:

$$c_e = \begin{cases} 1 & : e \in \mathcal{E}_{change} \\ 0 & : \text{else} \end{cases}$$

Note that in this case, it is possible to shrink the change&go network to a network with  $|\mathcal{L}| + |S|$  nodes and  $|\mathcal{E}_{change}| + |\mathcal{E}_{OD}|$  edges.

2. Real travel time:

$$c_e = \begin{cases} 0 & : e \in \mathcal{E}_{OD} \\ \text{travel time in minutes} & : e \in \mathcal{E}_{go} \\ \text{time needed for changing platform} & : e \in \mathcal{E}_{change} \end{cases}$$

More specific, to model the line planning problem as defined in Section 2, we set

$$c_e = \begin{cases} 0 & \text{if } e \in \mathcal{E}_{OD} \\ k_1 t_{uv} & \text{if } e = ((u, l), (v, l)) \in \mathcal{E}_{go} \\ k_2 & \text{if } e \in \mathcal{E}_{change} \end{cases}$$

Since we assume that customers behave selfish we need an implicit calculation of shortest paths (with respect to the weights  $c_e$ ) within our model. This is obtained by solving the following network flow problem for each origin-destination pair  $(s, t) \in \mathcal{R}$ .

$$\theta x_{st} = b_{st},$$

where

- $\theta \in \mathbb{Z}^{|\mathcal{V}| \times |\mathcal{E}|}$  is the node-arc-incidence matrix of  $G_{CG}$ ,



–  $b_{st} \in \mathbb{Z}^{|\mathcal{V}|}$  is defined by

$$b_{st}^i = \begin{cases} 1 & : i = (s, 0) \\ -1 & : i = (t, 0) \\ 0 & : \text{else} \end{cases}$$

– and  $x_{st}^e \in \{0, 1\}$  are the variables, where  $x_{st}^e = 1$  if and only if edge  $e$  is used on a shortest dipath from node  $(s, 0)$  to  $(t, 0)$  in  $G_{CG}$ .

To specify the lines in the line concept we introduce variables  $y_l \in \{0, 1\}$  for each line  $l \in \mathcal{L}$ , which are set to 1 if and only if line  $l$  is chosen to be in the line concept. Our model, *Line Planning with Minimal Travel Times* (LPMT) can now be presented.

(LPMT1)

$$\min \sum_{(s,t) \in \mathcal{R}} \sum_{e \in \mathcal{E}} w_{st} c_e x_{st}^e \quad (1)$$

$$s.t. \quad \sum_{(s,t) \in \mathcal{R}} \sum_{e \in \mathcal{E}^l} x_{st}^e \leq |\mathcal{R}| |\mathcal{E}^l| y_l \quad \forall l \in \mathcal{L} \quad (2)$$

$$\theta x_{st} = b_{st} \quad \forall (s, t) \in \mathcal{R} \quad (3)$$

$$\sum_{l \in \mathcal{L}} C_l y_l \leq B \quad (4)$$

$$x_{st}^e, y_l \in \{0, 1\} \quad \forall (s, t) \in \mathcal{R}, e \in \mathcal{E}, l \in \mathcal{L} \quad (5)$$

Constraint (2) makes sure that a line must be included in the line concept if the line is used by some origin-destination pair. Constraint (3) models the selfish behavior of the customers, i.e., that customers use shortest paths according to the weights  $c_e$ .

Having only constraints (2) and (3), the best line concept from a customer-oriented point of view would be to introduce all lines of the line pool. This is certainly no option for a public transportation company, since running a line is costly. Let  $C_l$  be an estimation of the costs for running line  $l$  and let  $B$  be the budget the public transportation company is willing to spend. Then the budget constraint (4) takes the economic aspects into account.

The objective function we use is customer-oriented: We sum up the costs

$$\sum_{e \in \mathcal{E}} w_{st} c_e x_{st}^e$$

of a shortest path from  $s$  to  $t$  for each origin-destination pair  $(s, t) \in \mathcal{R}$ , i.e., we minimize the average costs of the customers.

We get three alternative formulations of this problem by substituting constraints (2) by one of the following constraints

$$\sum_{(s,t) \in \mathcal{R}} x_{st}^e \leq |\mathcal{R}| y_l \quad \forall l \in \mathcal{L}, e \in \mathcal{E}^l \quad (6)$$

$$\sum_{e \in \mathcal{E}^l} x_{st}^e \leq |\mathcal{E}^l| y_l \quad \forall l \in \mathcal{L}, (s,t) \in \mathcal{R} \quad (7)$$

$$x_{st}^e \leq y_l \quad \forall (s,t) \in \mathcal{R}, e \in \mathcal{E}^l : l \in \mathcal{L} \quad (8)$$

We denote the formulation using constraints (6) (LPMT2), using (7) (LPMT3), and using (8) (LPMT4). As shown in [19], these formulations are equivalent, i.e. they are valid IP formulations for the same integer set  $X$  of feasible solutions of the line planning problem. Nevertheless the bounds provided by the corresponding LP-relaxations differ. This will be analyzed next.

Let  $X \subseteq \mathbb{Z}^n$  be a set of feasible solutions, and let two polyhedrons  $P_A$  and  $P_B$  be valid formulations for  $X$ , i.e.,  $X = P_A \cap \mathbb{Z}^n = P_B \cap \mathbb{Z}^n$ . Then  $P_A$  is said to be a *stronger* formulation than  $P_B$  if  $P_A \subset P_B$ , see, e.g., [20]. In this case,

$$\min_{x \in X} cx \geq \min_{x \in P_A} cx \geq \min_{x \in P_B} cx,$$

i.e., the bound provided by the stronger formulation  $P_A$  is better than the bound provided by  $P_B$ .

We can use this theory to analyze the strengthness of the four formulations presented for the line planning problem.

**Theorem 2.** *The convex hull of the integer set described by formulation (LPMT1) is denoted by  $P_1$ . The corresponding polyhedra described by formulation (LPMT2), (LPMT3), and (LPMT4) are denoted by  $P_2$ ,  $P_3$ , and  $P_4$ , respectively. Then, the following holds:*

- $P_4$  is stronger than  $P_1$ ,  $P_2$ , and  $P_3$ .
- $P_3$  is stronger than  $P_1$ .
- $P_2$  is stronger than  $P_1$ .
- Comparing  $P_3$  and  $P_2$ , none of them is stronger than the other.

The proof can be found in [19]. Note that in real world instances (LPMT3) comes out to be in most cases stronger than (LPMT2), see Section 5.1.

In (LPMT) we implicitly assume that all customers traveling from station  $s$  to station  $t$  choose the same path in the change&go network, i.e., the same set of lines. This can be done if edge capacities are neglected in (LPMT). In practice, this is usually not the case, since each vehicle only can transport a limited number of customers and usually there is only a limited number of vehicles possible along each line (e.g. due to safety rules). In the following, we therefore present an extension of (LPMT) taking into account the number of vehicles on

each line in a given time period. Consequently, this formulation allows to split customers along different paths from  $s$  to  $t$  in the change&go network  $G_{CG}$ .

Let  $N$  denote the capacity of a vehicle and let the new variables  $f_l \in \mathbb{N}$  contain the frequency of line  $l$ , i.e., the number of vehicles running along line  $l$  within a given time period. Furthermore we choose variables  $x_{st}^e \in \mathbb{N}$  and change the vector  $b_{st}$  to

$$b_{st}^i = \begin{cases} w_{st} & \text{if } i = (s, 0) \\ -w_{st} & \text{if } i = (t, 0) \\ 0 & \text{else} \end{cases}$$

Then the *Line Planning Model with minimal transfers and frequencies* (LPMTF) is the following:

(LPMTF)

$$\min \sum_{(s,t) \in \mathcal{R}} \sum_{e \in \mathcal{E}} c_e x_{st}^e \quad (9)$$

$$s.t. \quad \frac{1}{N} \sum_{(s,t) \in \mathcal{R}} x_{st}^e \leq f_l \quad \forall l \in \mathcal{L}, e \in \mathcal{E}_l \quad (10)$$

$$\theta x_{st} = b_{st} \quad \forall (s, t) \in \mathcal{R} \quad (11)$$

$$\sum_{l \in \mathcal{L}} C_l f_l \leq B \quad (12)$$

$$\sum_{l \in \mathcal{L}: k \in \mathcal{E}_l} f_l \leq f_k^{max} \quad \forall k \in E \quad (13)$$

$$x_{st}^e, f_l \in \mathbb{N} \quad \forall (s, t) \in \mathcal{R}, e \in \mathcal{E}, l \in \mathcal{L} \quad (14)$$

Constraints (10) make sure that the frequency of a line is high enough to transport the passengers. If  $f_l = 0$ , the line  $l$  is not chosen in the line concept. Constraints (11) are flow conservation constraints routing the passengers on the shortest possible paths. Note that the  $x_{st}^e$  variables can take integer values, such that passengers may choose different paths for the same origin-destination pair. Constraint (12) is again the budget constraint but with costs for each vehicle of a line (which are multiplied by the frequency to get the costs of the line). The capacity constraint (13) may be included if upper bounds for the frequencies are present.

## 5 Solving the LP-relaxation

As we have shown in Section 3 the line planning problem is NP-hard, and, moreover in real-world instances, gets huge. But fortunately the formulations of (LPMT) and (LPMTF) have block diagonal structure with only few coupling constraints. Moreover, in both models, all blocks are totally unimodular since they represent network flow problems.

In Section 5.1 we identify cases in which the solution of the LP-relaxation can be found by solving shortest path problems. If this does not work we have to take advantage of the block diagonal structure by using a Dantzig-Wolfe decomposition, which is shown in Section 5.2.

### 5.1 Using the trivial solution

**Definition 1.** A trivial solution  $(\bar{x}, \bar{y}^1)$ ,  $(\bar{x}, \bar{y}^2)$ ,  $(\bar{x}, \bar{y}^3)$ ,  $(\bar{x}, \bar{y}^4)$  of (LPMT1), (LPMT2), (LPMT3), (LPMT4), respectively, is defined as the solution  $\bar{x}_{st}^e$  of the shortest path problems

$$\theta x_{st} = b_{st} \quad \forall (s, t) \in \mathcal{R}$$

on the change&go-network constructed of all lines of the line pool and

$$\begin{aligned} \bar{y}_l^1 &:= \frac{\sum_{(s,t) \in \mathcal{R}} \sum_{e \in \mathcal{E}^l} \bar{x}_{st}^e}{|\mathcal{E}^l| |\mathcal{R}|} \quad \forall l \in \mathcal{L} \quad (\text{for (LPMT1)}) \\ \bar{y}_l^2 &:= \frac{\max_{e \in \mathcal{E}^l} \sum_{(s,t) \in \mathcal{R}} \bar{x}_{st}^e}{|\mathcal{R}|} \quad \forall l \in \mathcal{L} \quad (\text{for (LPMT2)}) \\ \bar{y}_l^3 &:= \frac{\max_{(s,t) \in \mathcal{R}} \sum_{e \in \mathcal{E}^l} \bar{x}_{st}^e}{|\mathcal{E}^l|} \quad \forall l \in \mathcal{L} \quad (\text{for (LPMT3)}) \\ \bar{y}_l^4 &:= \max_{(s,t) \in \mathcal{R}} \max_{e \in \mathcal{E}^l} \bar{x}_{st}^e \quad \forall l \in \mathcal{L} \quad (\text{for (LPMT4)}) \end{aligned}$$

It is in general not unique and need not to be feasible in the sense that it fulfills the budget constraint.

In real world instances it appears quite often that a trivial solution is an optimal solution of the LP-relaxation of (LPMT1). This is clear since the right hand sides  $|\mathcal{R}| |\mathcal{E}^l|$  of the coupling constraints (2) are chosen such that all passengers could use all edges of all lines. In real world only few edges of the network are used and so  $K_l := \sum_{(s,t) \in \mathcal{R}} \sum_{e \in \mathcal{E}^l} x_{st}^e$  is much smaller than  $|\mathcal{R}| |\mathcal{E}^l|$ , hence

$$\sum_{l \in \mathcal{L}} C_l \bar{y}_l^1 = \sum_{l \in \mathcal{L}} C_l \frac{K_l}{|\mathcal{E}^l| |\mathcal{R}|} \leq B$$

is often satisfied.

The following Lemma generalizes this for the other formulations. The proof can be found in [19].

**Lemma 1.** Let  $i \in \{1, 2, 3, 4\}$  and let  $(\bar{x}, \bar{y}^i)$  be a trivial solution of (LPMT $i$ ), as defined in Definition 1. If

$$T_i := \sum_{l \in \mathcal{L}} C_l \bar{y}_l^i \leq B$$

is satisfied, the trivial solution  $(\bar{x}, \bar{y}^i)$  is an optimal solution of (LPMT $i$ ). Note that for  $i = 4$  the solution  $(\bar{x}, \bar{y}^4)$  of the LP-relaxation of (LPMT $4$ ) is integer and thus if  $T_4 \leq B$  holds, the trivial solution is an optimal solution to the original problem.

In Table 1 we see the  $T_i$ -values for different line pool sizes, where the line costs are set to one. Note that in this case a value below one means that the trivial solution is always the optimal solution independently of the choice of the budget. Only if the given budget is smaller than the  $T_i$  value, the trivial solution is not a feasible solution of the LP-relaxation of (LPMT $i$ ). Thus, table 1 demonstrates the difference of the strength of the formulations. The higher the  $T_i$ -value, the better the lower bound provided by the corresponding formulation.

We see that in real world instances the bound provided by (LPMT3) is much stronger than (LPMT2) even if we could not show this in general. This is due to the fact that there exists an instance in which (LPMT2) is stronger than (LPMT3) but in real world this hardly ever happens.

Regarding the  $T_4$ -values, we recall that in this formulation the  $\bar{y}_l^4$  are integer valued and since all  $C_l = 1$  this means that if we are allowed to choose more than  $T_4$  lines out of the line pool, every passenger can travel on shortest path. If our budget is smaller, some passengers have a detour. In this case we have to use other methods to solve the problem like the Dantzig-Wolfe approach explained in the next section.

No.	$ \mathcal{L} $	obj.val.	$T_1$	$T_2$	$T_3$	$T_4$
1	10	2271.3	0.69	0.99	9.53	10
2	50	9459.9	0.20	0.35	25.31	48
3	100	24780.0	0.13	0.29	41.83	96
4	132	31654.2	0.11	0.26	53.12	129
5	200	15128.9	0.07	0.19	54.89	197
6	250	19096.0	0.05	0.16	61.07	235
7	275	20118.2	0.04	0.15	63.47	252
8	300	26598.3	0.06	0.19	72.35	282
9	330	26817.7	0.04	0.16	74.44	302
10	350	26450.0	0.07	0.23	90.04	331
11	375	27517.8	0.06	0.20	90.75	345
12	400	34781.3	0.06	0.20	100.05	370
13	423	35135.5	0.06	0.20	102.19	389

**Table 1.** Minimal budgets such that trivial solution is an optimal solution of the LP-relaxation of the different formulations of the (LPMT), see Lemma 1.

## 5.2 Using Dantzig-Wolfe decomposition

In this section we present an approach for solving the LP-relaxation of the (LPMT) formulations using Dantzig-Wolfe decomposition. The method can also be applied for solving (LPMTF) since the model structure is very similar. However, the numerical results deal with (LPMT). We will present two different decompositions. Since the blocks in both decompositions are totally unimodu-

lar, we know that the bound provided by the Master formulations is as good as the bound of the LP-relaxation (see [20]).

**One block for each origin-destination pair (LPMT1(LP))**

$$\min \sum_{(s,t) \in \mathcal{R}} \sum_{e \in \mathcal{E}} c_{st}^e x_{st}^e \quad (15)$$

$$\left. \begin{array}{l} \boxed{\begin{array}{l} \sum_{(s,t) \in \mathcal{R}} \sum_{e \in \mathcal{E}^l} x_{st}^e \leq |\mathcal{R}| |\mathcal{E}^l| y_l \quad \forall l \in \mathcal{L} \\ \sum_{l \in \mathcal{L}} C_l y_l \leq B \end{array}} \right\} \text{coupling constraints} \\ \left. \begin{array}{l} \boxed{X^{s_1, t_1}} \\ \quad \boxed{X^{s_2, t_2}} \\ \quad \quad \dots \\ \quad \quad \quad \boxed{X^{s_r, t_r}} \end{array} \right\} |\mathcal{R}| \text{ blocks}$$

where  $X^{st} := \{x_{st} \in \mathbb{R}^{|\mathcal{E}|} : \theta x_{st} = b_{st}, 0 \leq x_{st}^e \leq 1, \forall e \in \mathcal{E}\}$   
The coupling constraints can be written as

$$\begin{array}{l} -A_Y y + \sum_{(s,t) \in \mathcal{R}} A_X x_{st} \leq 0 \\ C y \leq B \end{array}$$

where

- $A_X$  is an  $|\mathcal{L}| \times |\mathcal{E}|$  matrix given by elements  $a_{le} = 1$ , if  $e \in \mathcal{E}_l$ , zero otherwise. It is equal for each origin-destination pair.
- $A_Y$  is an  $|\mathcal{L}| \times |\mathcal{L}|$  diagonal matrix containing  $|\mathcal{R}| |\mathcal{E}^l|$  as its  $l$ th diagonal element.
- $C$  is the line cost vector  $(C_1, \dots, C_{|\mathcal{L}|})$ .

So, we get the following coefficient matrix of (LPMT1):

$$\begin{pmatrix} -A_Y & A_X & \dots & A_X \\ C & & & \\ & \theta & & \\ & & \ddots & \\ & & & \theta \end{pmatrix}$$

Defining the *weight-cost-parameters*  $c_{st}^e := w_{st} c_e$ , we get the following Master Problem corresponding to decomposition (15):

(Master 1)

$$\begin{aligned}
 z = \min & \sum_{(s,t) \in \mathcal{R}} \sum_i (c_{st} x_{st}^{(i)}) \alpha_{st}^i \\
 \text{s.t.} & \sum_{(s,t) \in \mathcal{R}} \sum_i (A_X x_{st}^{(i)}) \alpha_{st}^i - A_Y y + Iv = 0 \\
 & \sum_{l \in \mathcal{L}} C_l y_l \leq B \\
 & \sum_i \alpha_{st}^i = 1 \quad \forall (s,t) \in \mathcal{R} \\
 & y_l \geq 1 \quad \forall l \in \mathcal{L} \\
 & v_l, \alpha_{st}^i, y_l \geq 0
 \end{aligned}$$

where the  $|\mathcal{L}|$ -vector  $v$  are slack variables, and  $x_{st}^{(i)}$  are the extreme points of  $X^{st}$ . This problem has  $|\mathcal{L}| + 1$  coupling constraints and  $|\mathcal{R}|$  convexity constraints.

For each  $(s, t) \in \mathcal{R}$  we obtain the following subproblem:

$$\begin{aligned}
 z_{st} = \min & (c_{st} - \pi A_X) x_{st} - \mu_{st} \\
 \text{s.t.} & x_{st} \in X^{st}
 \end{aligned}$$

where  $\{\pi_i\}_{i \in \mathcal{L}}$  are the dual variables of the coupling constraints, and  $\{\mu_{st}\}_{(s,t) \in \mathcal{R}}$  are the dual variables of the convexity constraints.

The  $X^{st}$  blocks correspond to shortest path problems which are known to be totally unimodular, hence the  $x_{st}^{(i)}$ -values are in  $\{0, 1\}^{|\mathcal{E}|}$ . The formulations (LPMT2), (LPMT3), (LPMT4) as well as (LPMTF) can be reformulated analogously.

**One block for all origin-destination pairs** If we treat the  $X^{st}$ -blocks as one block we get the following reformulation:

(LPMT1(LP))

$$\min \sum_{e \in \mathcal{E}} c^e x^e \tag{16}$$

$$\boxed{\begin{aligned} \sum_{e \in \mathcal{E}^l} x^e &\leq |\mathcal{R}| |\mathcal{E}^l| y_l \quad \forall l \in \mathcal{L} \\ \sum_{l \in \mathcal{L}} C_l y_l &\leq B \end{aligned}} \quad \text{coupling constraints}$$

$$\boxed{X} \quad \text{1 block}$$

with  $X := \{x \in \mathbb{R}^{|\mathcal{E}|} : x^e = \sum_{(s,t) \in \mathcal{R}} x_{st}^e \quad \forall e \in \mathcal{E}, x_{st} \in X^{st}\}$  and  $c^e := \sum_{(s,t) \in \mathcal{R}} c_{st}^e$ .

The Master Program corresponding to decomposition (16) is

(Master 2)

$$\begin{aligned}
 z = \min & \sum_i (c x^{(i)}) \alpha^i \\
 \text{s.t.} & \sum_i (A_X x^{(i)}) \alpha^i - A_Y y + Iv = 0 \\
 & \sum_{l \in \mathcal{L}} C(l) y_l \leq B \\
 & \sum_i \alpha^i = 1 \\
 & v_l, \alpha^i, y_l \geq 0
 \end{aligned}$$

where the  $|\mathcal{L}|$ -vector  $v$  are slack variables, and  $x^{(i)}$  are the extreme points of  $X$ . This problem has  $|\mathcal{L}|+1$  coupling constraints and one convexity constraints.

The subproblem of the  $X$ -block is

$$z = \min \sum_{(s,t) \in \mathcal{R}} (c_{st} - \pi A_X) x_{st} - \mu$$

$$\text{s.t. } x_{st} \in X^{st}$$

where  $x^e := \sum_{(s,t) \in \mathcal{R}} x_{st}^e$  and  $\{\pi_i\}_{i \in \mathcal{L}}$  are the dual variables of the coupling constraints,  $\mu$  is the dual variable of the convexity constraint.

As in the previous formulation, the  $x^{(i)}$ -values are integer because they are the component wise sum over shortest path problem solution which are in  $\{0, 1\}$ . In this decomposition we loose the information of the exact paths of the customers which are needed in (LPMT3), (LPMT4) and (LPMTF) and thus this Master cannot be adapted to these formulations.

**Implementation** We implemented the Dantzig-Wolfe decomposition approach of (LPMT) using Xpress MP 2003 and Microsoft Visual C++ 6.0. The CPU times of this section are based on a 3.06 GHz Intel4 processor with 512 MB RAM. The subproblems were solved with Dijkstra’s shortest path algorithm.

In column ‘CPU1’ of table 2 we see the CPU times in seconds for solving the LP-relaxation of (LPMT1) using Dantzig-Wolfe approach with (Master2) for different line pool sizes of the network of German long distance trains. In column ‘CPU2’ we see the CPU times in seconds for solving the LP-relaxation of (LPMT3) using Dantzig-Wolfe approach with (Master1). We have mentioned that the lower bound provided by (LPMT3) is stronger than (LPMT1) and so the computation times increase in this case. We see, that using our approach it is possible to solve the LP-relaxation of (LPMT3) for medium sized networks within reasonable time. Note that the size of the problem not only depends on the size of the line pool but on the number of origin-destination pairs and the size of the PTN which may be much smaller e.g. in urban underground networks. Solving the LP-relaxation of the weaker (LPMT1) formulation is possible even for big real world instances like the long distance network of German railway within two and a half hours.

As we have seen, the main problem of our approach is the size of the change&go-network depending mainly on the size of the line pool. A wise choice of a possibly small line pool is therefore advisable. On the other hand it makes sense to analyze the underlying PTN. For example if two lines go parallel for a long time, it is sufficient to add changing edges only at the first and the last station. Also arcs between stations without changing possibility can be shrunken to decrease the size of the network.



No.	$ \mathcal{L} $	$ \mathcal{R} $	CPU1	CPU2
0	3	2	0.05	0.1
1	10	2602	1	228
2	50	4766	3	606
3	100	11219	16	8706
4	132	18238	48	M
5	200	10126	78	M
6	250	13246	329	M
7	275	14071	691	M
8	300	17507	1171	M
9	330	18433	1911	M
10	350	17095	1814	M
11	375	18350	2727	M
12	400	22191	4789	M
13	423	22756	8715	M

**Table 2.** CPU times of the LP-relaxation of (LPMT1) and (LPMT3) using Dantzig-Wolfe approach with (Master2) and (Master1), respectively, for different line pool sizes. M denotes "out of memory".

## 6 Conclusions

We developed integer programming models for the line planning problem with the goal to minimize the travel times over all customers including penalties for the transfers needed and proposed an extension that includes frequencies. We showed that the problem is NP-hard. Since the problem gets huge, a straightforward solution of the LP relaxation is not possible. We showed that in many real world cases the trivial solution is optimal or, if it is infeasible, it can be found by a solution approach based on Dantzig-Wolfe decomposition. Computational results for various real world instances and different decompositions were presented.

We are currently working on a branch&price algorithm and heuristics to get an integer solution.

## References

1. Dienst, H.: Linienplanung im spurgeführten Personenverkehr mit Hilfe eines heuristischen Verfahrens. PhD thesis, Technische Universität Braunschweig (1978)
2. Zwaneveld, M., Claessens, M., van Dijk, N.: A new method to determine the cost optimal allocation of passenger lines. In: Defence or Attack: Proceedings 2nd TRAIL Phd Congress 1996, part 2, TRAIL Research School (1996)
3. Goossens, J., van Hoesel, C., Kroon, L.: A branch and cut approach for solving line planning problems. *Transportation Science* **38** (2004) 379–393
4. Bussieck, M.: Optimal lines in public transport. PhD thesis, Technische Universität Braunschweig (1998)

5. Bussieck, M., Lindner, T., Lübbecke, M.: A fast algorithm for near cost optimal line plans. *Mathematical Methods of Operational Research* **59** (2004) 205–220
6. Goossens, J.: Models and algorithms for railway line planning problems. PhD thesis, University of Maastricht (2004)
7. Goossens, J., van Hoesel, C., Kroon, L.: On solving multi-type line planning problems. Technical Report RM/02/009, University of Maastricht (2002) METEOR Research Memorandum.
8. Bussieck, M., Kreuzer, P., Zimmermann, U.: Optimal lines for railway systems. *European Journal of Operational Research* **96** (1996) 54–63
9. Borndörfer, R., Grötschel, M., Pfetsch, M.E.: Models for line planning in public transport. Technical Report 04-10, ZIP Berlin (2004)
10. Borndörfer, R., Grötschel, M., Pfetsch, M.E.: A path-based model for line planning in public transport. Technical Report 05-18, ZIP Berlin (2004)
11. Schneider, R.S.: A new customer-oriented cost model for the line planning problem. Master’s thesis, Technische Universität Kaiserslautern (2005)
12. Laporte, G., Marin, A., Mesa, J., Ortega, F.: An integrated methodology for rapid transit network design. In: *Algorithmic Methods for Railway Optimization. Lecture Notes on Computer Science*. Springer (2004) presented at ATMOS 2004, to appear.
13. Laporte, G., Mesa, J., Ortega, F.: Maximizing trip coverage in the location of a single rapid transit alignment. *Annals of Operations Research* **136** (2005) 49–63
14. Klingele, H.: Verfahren zur Optimierung eines Linienkonzeptes der Deutschen Bahn AG. Master’s thesis, Universität Karlsruhe (2000)
15. Schmidt, M.: Modelle zur Linienoptimierung im Zugverkehr unter Berücksichtigung der Nachfrage. Master’s thesis, Universität Kaiserslautern (2001)
16. Quak, C.: Bus line planning. Master’s thesis, TU Delft (2003)
17. Schöbel, A., Scholl, S.: Planung von Linien mit minimalen Umsteigevorgängen. In Mattfeld, D., ed.: *Proceedings of the GOR-workshop on "Optimierung im öffentlichen Nahverkehr"*. (2003) 69–89 <http://server3.winforms.phil.tu-bs.de/gor/tagung34/>.
18. Schöbel, A., Scholl, S.: Line planning with minimal transfers. NAM Bericht 72, Georg-August Universität Göttingen (2004)
19. Scholl, S.: Customer-oriented line planning. PhD thesis, Technische Universität Kaiserslautern (2005)
20. Wolsey, L.: *Integer Programming*. John Wiley & Sons (1998)

# Paying Less for Train Connections with MOTIS

Matthias Müller–Hannemann and Mathias Schnee

Darmstadt University of Technology, Computer Science,  
64289 Darmstadt, Hochschulstraße 10, Germany  
muellerh,schnee@algo.informatik.tu-darmstadt.de,  
<http://www.algo.informatik.tu-darmstadt.de>

**Abstract.** Finding cheap train connections for long-distance traffic is algorithmically a hard task due to very complex tariff regulations. Several new tariff options have been developed in recent years, partly to react on the stronger competition with low-cost airline carriers. In such an environment, it becomes more and more important that search engines for travel connections are able to find special offers efficiently.

We have developed a multi-objective traffic information system (MOTIS) which finds all attractive train connections with respect to travel time, number of interchanges, and ticket costs. In contrast, most servers for timetable information as well as the theoretical literature on this subject focus only on travel time as the primary objective, and secondary objectives like the number of interchanges are treated only heuristically. The purpose of this paper is to show by means of a case study how several of the most common tariff rules (including special offers) can be embedded into a general multi-objective search tool.

Computational results show that a multi-objective search with a mixture of tariff rules can be done almost as fast as just with one regular tariff. For the train schedule of Germany, a query can be answered within 1.9s on average on a standard PC.

**Keywords:** timetable information system, multi-criteria optimization, shortest paths, fares, special offers, long-distance traffic

## 1 Introduction

In recent years, there has been strong interest in efficient algorithms for timetable information in public transportation systems (with emphasis on public railroad systems). For a given customer query, the problem is to find all attractive train connections with respect to several objectives. We concentrate on travel time, number of interchanges, and ticket costs.

Most work has considered optimization subject to a single criterion, namely to find the fastest connection. Such a problem can easily be modeled as a shortest path search in a graph where the edge lengths correspond to travel times. Likewise it is not difficult to extend these graph models so that also the minimum number of train interchanges can be solved as a shortest path problem with  $\{0, 1\}$ -lengths on the edges.

However, difficulties arise with fares as objectives. Pricing systems of railway companies are very complex and actual fares depend on many parameters. In recent years, railway companies faced higher competition caused by the strong increase of low-cost airlines. As a reaction on this development, marketing departments of railway companies answer with the introduction of different types of special offer tariffs. For origin-destination pairs with a low-cost competitor the relation-based prices are occasionally decreased.

For this and several other reasons, the fare of a connection cannot be modeled in an exact way as an additive function on the edges of a graph which can simultaneously be used for a fastest connection search.

**Previous work.** Two main approaches have been proposed for modeling time table information as a shortest path problem: the *time-expanded* [1,2,3,4,5,6,7], and the *time-dependent* approach [6,8,9,10,11,12,13,14]. The common characteristic of both approaches is that a query is answered by applying some shortest path algorithm to a suitably constructed graph. These models and algorithms are described in detail in a recent survey [15]. The time-expanded model is much more flexible than the time-dependent model. It is therefore preferred if all side constraints of a real-world scenario have to be respected.

As mentioned above, most of the cited papers consider fastest connections only. Multi-criteria search for train connections in a fully-realistic environment has been studied in [7]. The latter paper already used a simplified model to search for regular fares. Apart from initial work in [7], we are not aware of any previous work which takes fares as an optimization criterion into account.

**Contribution of this paper.** Usually, marketing experts design a new tariff with respect to expected sales but without considering how such an offer can be searched for in an efficient way. It seems that Germany has one of the most complicated tariff systems of the world, providing us with the most challenging task to find cheap connections systematically.

In this paper, we analyze the different tariff options with respect to searchability. We show that a systematic, simultaneous search for different tariffs can be integrated into a suitable graph model and a generalized version of Dijkstra’s algorithm.

In particular, we focus on tariff options which are based on the availability of contingents, yielding either a fixed price or a certain discount.

Currently we develop the information server MOTIS (multi objective traffic information system) in cooperation with datagon GmbH, Waldems, Germany. The main features of MOTIS are the following:

- It contains a Dijkstra-based multi-objective search algorithm (travel time, number of interchanges, ticket costs).
- It provably yields *exact* minimization of travel time and number of interchanges. In contrast, the electronic timetable information system HAFAS [16], which is used by many European railway companies provides only heuristic solutions.

- It delivers many attractive alternatives by using the concept of relaxed Pareto optimality [7].
- MOTIS is extensible to add further criteria like the possibility of seat reservation or to incorporate safety margins for train changes in case of delays.

An extensive computational study shows that the computational cost increases only very slightly when we combine the search with respect to regular fares and other tariff options. On average, a query can be answered within 1.9s on a standard PC.

**Overview.** The rest of the paper is organized as follows. In Section 2, we first give a brief description of MOTIS. Afterwards, in Section 3, we present a systematic overview on fare regulations. For each tariff class we analyze the algorithmic consequences for efficient searchability of connections which fall into this class. Thereafter we explain more details on the search algorithm of MOTIS in Section 4. Then, we provide computational results based on a large test set of real customer queries. Finally, we conclude with a summary and directions for future work.

## 2 The Information Server MOTIS

This section is intended to give a brief introduction to MOTIS and the main ideas behind it. In the following subsections we first explain what kind of queries can be handled, and define what we understand by “finding all attractive connections”. Then we briefly touch upon the graph model used and the general search algorithm.

### 2.1 Queries

A *query* to a timetable information system usually includes:

The (start or) *source station* of the connection, the *target station* and an *interval* in time in which either the departure or the arrival of the connection has to be, depending on the *search direction*, the user’s choice whether to provide the interval for departure (“forward search”) or arrival (“backward search”). Additional query options include:

*Vias and duration of stay.* A query may contain one (or more) so called *vias*, stations the connection has to visit and where at least the specified amount of time can be spent, e.g. from Cologne to Munich via Frankfurt with a stay of at least two hours for shopping in Frankfurt.

*Train class restrictions.* Each train has a specific *train class* assigned to it. These classes are high-speed trains such as the German ICE and French TGV; ICs and ECs; Interregios and the like; local trains, “S-Bahn” and subway; busses and trams. The *query* may be restricted to a subset of all *train classes*. Certain train

tariffs exclude some of the higher-valued train classes. Hence, by excluding high speed trains one can search for special tariffs.

*Attribute requirements.* Trains have *attributes* describing additional services they provide. Such attributes are for example: “bike transportation possible”, “sleeping car”, “board restaurant available”. A user can specify attributes a connection has to satisfy or is not allowed to have. We allow Boolean operators for specifying *attribute requirements* like: (a restaurant OR a bistro) AND bike transportation.

*Passenger related attributes.* Additional attributes are relevant for the fare calculation. One has to choose the desired *comfort class* (i.e. first or second class). In order to determine possibilities for discounts, the query has to provide the number of passengers, and for each passenger the type of discount card which is available (if any). Families with children also have to specify the age of each child.

## 2.2 Attractive Train Connections

A simple measurement for the “attractiveness” of a connection does not exist. Different kinds of costumers have differing (and possibly contrary) preferences. Key criteria for the quality of a connection are travel time, ticket cost and convenience (number of interchanges, comfort of the used trains, time for train changes). In order to build a traffic information system that can provide attractive connections we avoid the drawbacks of weighted target functions or “preference profiles”. Instead we want to serve each possible costumer by presenting him a selection of highly attractive alternatives with one single run of the algorithm.

When dealing with multiple criteria a standard approach is to look for the so-called Pareto set. For two given  $k$ -dimensional vectors  $x = (x_1, \dots, x_k)$  and  $y = (y_1, \dots, y_k)$ ,  $x$  *dominates*  $y$  if  $x_i \leq y_i$  for  $1 \leq i \leq k$  and  $x_i < y_i$  for at least one  $i \in \{1, \dots, k\}$ . Vector  $x$  is *Pareto optimal* in set  $X$  if there is no  $y \in X$  that dominates  $x$ . Here, we assume for simplicity that all cost criteria shall be minimized. In our scenario we compare 3-dimensional vectors (travel time, ticket costs, number of interchanges) for our connections.

We argued in [7] that the set of Pareto optima still does not contain all attractive connections and proposed to apply the concept of *relaxed Pareto optimality*. It provides more alternatives than Pareto optimality can give. Under relaxed Pareto dominance

- connections that are nearly equivalent but differ slightly do not dominate each other;
- the bigger the difference in time between start or end of two connections the less influence they have on each other;
- traveling longer needs to yield a fair *hourly wage* (i.e. the amount of money saved divided by the extra time in hours) to make a cheaper alternative attractive. The latter also excludes irrelevant Pareto optima.

We used the following rules to compare connections  $A$  and  $B$  which have departure times  $d_A, d_B$ , arrival times  $a_A, a_B$ , travel times  $t_A, t_B$  (all data given in minutes),  $i_A, i_B$  interchanges and associated costs  $c_A, c_B$  in Euros, respectively. Connection  $A$  *dominates* connection  $B$

- with respect to the criterion travel time if  $B$  does not overtake  $A$  and

$$t_A + \alpha(t_A) \cdot \min\{|d_A - d_B|, |a_A - a_B|\} + \beta(t_A) < t_B,$$

where,  $\alpha(t_A) := t_A/360$  and  $\beta(t_A) := 5 + \sqrt{t_A}/4$ ;

- with respect to the number of interchanges only if  $i_A < i_B$ ;
- with respect to the cost criterion only if

$$c_A + \frac{t_A - t_B}{60} \cdot \Delta < c_B,$$

where the required hourly wage  $\Delta$  is set to 5 Euros.

### 2.3 Time-Expanded Graph Model

The basic idea of a so-called *time-expanded graph model* is to introduce a directed search graph where every node corresponds to a specific event (departure, arrival, change of a train) at a station.

A connection served by a train from station  $A$  to station  $B$  is called *elementary*, if the train does not stop between  $A$  and  $B$ . Edges between nodes represent either elementary connections, waiting within a station, or changing between two trains. For each optimization criterion, a certain length is associated with each edge.

Traffic days, possible attribute requirements and train class restrictions with respect to a given query can be handled quite easily. We simply mark train edges as *invisible* for the search if they do not meet all requirements of the given query. With respect to this visibility of edges, there is a one-to-one correspondence between feasible connections and paths in the graph.

More details of the graph model can be found in [7].

### 2.4 The Search Algorithm in MOTIS

Our algorithm is a “Pareto-version” of Dijkstra’s algorithm using multi-dimensional labels. See Möhring [2] or Theune [17] for a general description and correctness proofs of the multi-criteria Pareto-search.

Each label is associated with a node  $v$  in the search graph. A label contains key values of a connection from a start node up to  $v$ . These key values include the travel time, the number of interchanges, a ticket cost estimation and some additional information. For every node in the graph we maintain a list of labels that are not dominated by any other label at this node. Every time a node is extracted from the priority queue, its outgoing edges are scanned and (if they are not infeasible due to traffic days, attributes and train class restrictions etc.)

labels for their head nodes are created. Such a new label is compared to all labels in the list at the head node. It is only inserted into that list and the priority queue if it is not dominated by any other label in the list. On the other hand, labels dominated by the new label are removed.

As a further means of dominance we keep a short list of Pareto-optimal labels at the terminal station and compare each new label to these labels. To compare labels at an intermediate node  $v$  with a node at the terminal, we use lower bounds on the key values of a shortest, a most convenient, and a cheapest path from  $v$  to the terminal station. We increase the criteria of the label at  $v$  by lower bounds on the according values. If the label with its increased values is dominated by any label at the terminal, it is excluded from further search.

To make lower bounds available, we determine a guaranteed fastest connection from source to target using a goal-directed single criterion search in an initialization phase before the actual multi-criteria search. This search is by orders of magnitude faster than the multi-criteria search and can be performed in less than 50ms on average.

## 2.5 Black-Box-Pricing Component

As noted in the introduction, the fare regulations are extremely complex. Furthermore, the system undergoes rapid change. Therefore, it is reasonable to have a *black-box pricing component (BPC)* that can be used to calculate the exact ticket cost for some connection. Unfortunately, one call to this black-box routine is very costly. Hence, it is impossible to calculate the correct price for every label and achieve a bearable running time.

As a consequence, we use fast to compute price estimates in the labels that are updated during the search. To this end, we associate an estimated base fare with each travel edge in our search graph. (How we derive these estimations will be described in more detail in Section 3.1.)

This simplified model provides helpful estimates for the search. In order not to lose low cost connections due to this approximation we need a safety margin which is incorporated into the corresponding relaxation function for the relaxed Pareto dominance. After a search is completed, all connections are correctly priced by the BPC and relaxed Pareto dominance can be applied to true fares.

## 3 Modeling Regular Fares and Special Offers

The purpose of this section is to provide an overview on the many different classes of tariffs commonly used by train companies.

As the number of different tariffs being in use is very large, tariffs differ considerably from country to country, and they are subject to frequent changes, this overview is far from being comprehensive. However, we try to group the most commonly used tariffs into certain classes. For each tariff class, we analyze how a search for connections which fall under this class can be modeled and incorporated into our general framework of MOTIS.



In some rare cases it might be profitable to partition the desired connection into smaller connections. To each partial connection a different tariff option may apply, yielding an overall saving if several tickets are bought. However, this is very impractical and potentially confusing for the customer. In this paper, we therefore restrict our discussion to a single tariff for each connection.<sup>1</sup>

### 3.1 Regular Fares

Regular fares apply at any time to everyone without any restrictions. To calculate regular fares, two main principles are in use: distance-based and relation-based fares.

**Distance-based fares.** For this type, regular fares are modeled by piecewise affine-linear functions which depend on the number of kilometers of the connection and the used train classes. These functions are encoded in tables and the calculation of fares is done with a table look-up. For example, regular fares in France (SNCF) follow this scheme.<sup>2</sup>

**Relation-based fares.** For long-distance travel in a highly connected network like that of Germany the regular fare is more often based on relations, i.e. origin-destination pairs associated with a regional corridor. The corridor of a relation describes what is considered as a common route. A relation can only be applied to a connection if the connection passes stations from a relation-specific set which specifies the corridor.

If a connection leaves the corridor of a relation, the fare has to be determined by partitioning the entire connection into smaller connections. The details of this procedure are beyond the scope of this paper.

Marketing considerations influence the price for each relation. In general, the fare of a relation is derived from the travel distance, but it may be changed for marketing reasons in either direction.

**Properties of regular fares.** In most cases, we can assume that regular fares are monotonously increasing and subadditive. That is, for a connection  $c$  from station  $s$  to station  $t$  via station  $v$ , the price  $p_c(s, t)$  satisfies

$$p_c(s, t) \leq p_c(s, v) + p_c(v, t).$$

Distance-based fares are degressive functions in the travel kilometers. Hence, they are always strictly subadditive.

In dominance tests, good lower bounds are of crucial importance for the efficiency of the search. Hence, we need a lower bound on the price of a connection. With distance-based fares, we get a lower bound on the distance of a connection

<sup>1</sup> Note that a combination of tariffs is necessary in multi-vendor systems.

<sup>2</sup> [http://www.voyages-sncf.com/info\\_rese/guide\\_du\\_voyageur/Calcul\\_PT.htm](http://www.voyages-sncf.com/info_rese/guide_du_voyageur/Calcul_PT.htm)

from the distance traveled from  $s$  to  $v$  plus a lower bound on the distance from  $v$  to  $t$ .

In sharp contrast, valid lower bounds are hard to obtain for relation-based fares as these may even violate our subadditivity assumption. But even if we assume subadditivity, it is not clear how to get a lower bound on the price of a connection from  $s$  to  $t$  given the prices from  $s$  to  $v$  and from  $v$  to  $t$ .

**Frequent user cards.** For holders of frequent user cards (like “BahnCard”) a general  $x\%$  discount applies to the regular fare. As this kind of discount yields the same reduction rate for all connections, our price estimation merely needs a flag indicating whether such a card is available or not. Such a flag is necessary for a comparison with other tariff options.

**Approximation of regular fares.** We use a very simple but efficiently computable model to approximate regular fares. Basically, we simulate a distance-based fare and associate a travel distance with each edge. The distance between the two stations of a train edge is taken as the straight line distance obtained from the coordinates of the stations. During the search, we add for each train edge the travel distance times a constant factor (in Euros/km) depending on the train class used. If true regular fares are based on relations, we have to incorporate relatively large safety margins in order not to lose too many attractive connections.

### 3.2 Surcharges

An additive surcharge applies to certain trains (night trains, ICE sprinter) or train classes (IC,EC). It has to be paid once, if such a train is used. If a connection uses several trains to which a surcharge applies, then usually only the highest surcharge has to be paid once.

During the search, the amount of the surcharge is added to the price estimation when a partial connection first enters a train with a surcharge. In order to guarantee that a surcharge is paid only once, the labels characterizing a partial connection store in flags which surcharges have already been applied.

### 3.3 Contingent Based Discount Fares

Contingent-based offers are intended to increase the average passenger load on high-speed trains. For each train in a connection for such an offer, a contingent of available seats must not be exceeded by previous bookings. For high-speed trains the contingent may be something like 10% of all seats. For local trains, there is typically no contingent restriction, i.e. the contingent is regarded as being unlimited. As a consequence, such offers are only valid for connections which contain at least one contingent-restricted train.

Many train companies offer discounted fares on long-distance travel under certain restrictions. These restrictions typically include that

- the ticket has to be bought a certain time in advance (for example, at least three days in advance);
- passengers restrict themselves to a particular day and a certain connection which has a contingent available;
- passengers make a return journey to and from the same station.

Discount rates may also be subject to weekend restrictions. For example, Deutsche Bahn AG offers “Savings Fare 50” (“Sparpreis 50”) only if the following restrictions apply: For trips starting from Monday to Friday, the return trip cannot be any sooner than the following Sunday. If you travel on Saturday or Sunday you may return that same day.

To incorporate such types of offers into the search, we add and maintain a contingent flag in our labels. The *contingent flag* is a Boolean flag which is set to true if and only if all previous train edges of this connection have a contingent available.

### 3.4 Fixed Price Offers

**Contingent-Based Restrictions.** Certain special tariffs offer fixed price tickets within a limited time period (of several weeks or even months, like “Summer Special”) subject to the availability of contingents.

A further restriction is that the itinerary of a connection from station  $A$  to  $B$  must use a “common route”. This rule is to prevent from possible misuse by making round-trips or stop-overs during the travel for which one usually would have to buy several tickets or at least to pay for the deviation.

The easiest way to model common routes is to impose the restriction that the length of an itinerary of a connection has to be at most a certain percentage, say 20%, longer than the shortest route from  $A$  to  $B$ . Alternatively, the travel time should not be more than a certain percentage longer than the fastest route from  $A$  to  $B$ .

The modification of our model for this kind of tariff is similar to the previous case. We also maintain a contingent flag in each label indicating whether a contingent has been available on all previous edges. As contingents for discounts and for fixed prices may be different, we use different kind of contingent flags. At each intermediate station, we also check whether the partial connection up to this station can still be extended in such a way that it stays on a “common route”. To this end, we use lower bounds for the remaining path from this intermediate station to the final destination.

**Time Interval Restrictions.** Tickets allowing unlimited travel may be available for a fixed price provided the time of the trip falls into a certain time interval.

For example, Deutsche Bahn AG offers a “Happy-Weekend-Ticket” which can be used on all trains except high-speed trains on Saturdays or Sundays between 12 a.m. until 3 a.m. of the following day for a fixed price. Another example would

be a fixed price ticket valid from 7 p.m. until the end of the same business day (“Guten-Abend-Ticket”).

Such offers can be handled in the following way. For a given query, we first check whether the given start interval falls into the interval of a special offer. If not, the corresponding tariff is definitely not applicable. If the offer has no train class restrictions, we can use the standard multi-objective search. For each alternative found by this search, we finally have to check whether the complete connection falls into the time interval. If this is the case, the price for this connections is the minimum of the regular fare and the fixed price.

If train class restrictions apply, we could use two independent searches, one with train class restriction and one without. However, it is more efficient to treat train class restrictions as a further criterion in the multi-criteria search and to run just a single simultaneous search for both cases.

**Rail Passes.** Many train companies also offer different kinds of so-called *rail passes* which allow unlimited travel. Prices depend on country and number of days. Rail passes may be restricted to special user groups (students, disabled, unemployed), restrictions may be based on the age (children, seniors), or restrictions on the place of permanent residence apply.

Further restrictions may be imposed on the set of allowed train classes. For example, a regional rail pass like “Hessenticket” offered by Deutsche Bahn AG is only valid for local trains.

Passengers with rail passes can use the standard multi-objective search on the basis of regular fares which delivers, in particular, all attractive connections with respect to travel time and convenience. The price information can simply be ignored. The search has only to make sure that the whole connection lies within the region where the rail pass is valid.

### 3.5 Discounts for Groups

Groups of 2 or more passengers either get an  $x\%$  discount on the regular tariff which can be applied to all trains, or they get an even larger discounts of  $y > x\%$  based on the availability of certain contingents. During the search, both options can be handled in the same way as for single passengers.

### 3.6 Further Possibilities for Discounts

Discounts for single passengers or groups may also be restricted to certain Boolean conditions which depend only on properties of the travelers but not on the particular trip they are going to make. For example, if the group is a family with children below a certain age, then special discounts apply. Another example would be discounts for employees of certain companies (corporate clients).

## 4 More Details on the Search Algorithm

### 4.1 Simultaneous Search

The aforementioned modeling of the various tariffs allows the search for combinations of tariffs simultaneously. This is preferable over having individual searches for each of the tariff rules that apply in a scenario and - as we will show in the subsequent section - can be done without sacrificing search speed.

However, as the number of tariff rules increases, more and more labels become mutual incomparable. For example, consider two labels representing partial connections that can gain a fixed price or discounted fare, respectively. Either connection might not be extendable to a connection from source to target with contingents available on all edges. So neither of them can dominate the other depending on an estimate of the special price. Furthermore, they cannot even be compared regarding the estimation for the regular price, as the final price may differ substantially if a special tariff is applicable.

The dominance test between a connection that has already reached the terminal station and a partial connection has to compare the lowest possible price reachable by extending the partial connection to the actual price of the complete connection. So it is even more important to have a fast and cheap connection at the terminal fairly early in the search process (compare Section 4.2).

### 4.2 Fast Search for the Fastest Fixed Price Connection

For several reasons we implemented a specialized version of our algorithm to search for fixed price connections. Our motivation was

1. to have a stand alone tool to find one fixed price connection, and
2. to strengthen our dominance with terminal labels, or
3. to have a certificate that no fixed price connection is available at all. In the latter case, we can turn off our fixed price search.

Our specialized algorithm for fixed price search (“fixed price Dijkstra”) is a single-criterion goal-directed search algorithm. It determines a fastest connection among all connections using only available contingent edges and edges without contingent restrictions.

### 4.3 Determining Lower Bounds in the Preprocessing Phase

The initialization phase now consists of up to two searches: First we use the standard single-criterion goal-directed search algorithm to determine a fastest connection from source to target. It keeps track of the contingent information and

- either finds a connection with a fixed price (it includes a high-speed train and contingents are available on all contingent edges),

number of stations	8,861
number of trains	45,370
number of high-speed trains	1,006
number of nodes	1,427,726
number of edges	2,395,703

**Table 1.** Size parameters of the time-expanded graph.

- or finds a connection without high-speed train (therefore no fixed price is possible for it). As it is the fastest connection, we may use it for dominance testing later on. It is also quite often cheaper than the fixed price (see Section 5.4).
- Otherwise, it triggers the specialized algorithm for fixed price search.

If triggered, the “fixed price Dijkstra” algorithm

- either finds a connection with a fixed price (it includes a high-speed train, contingents are available on all contingent edges, and it is within the allowed margin (here 20% more travel time) compared to the fastest connection),
- or finds a connection without high-speed train (therefore no fixed price is possible for it). If a fixed price connection exists, it must be slower than this connection. Such a connection is also quite often cheaper than the fixed price (see Section 5.4) and therefore very useful for later dominance testing.
- Otherwise it finds a connection with contingents available on all contingent edges but that does not stay within the allowed margin. In this case no fixed price connection exists (as all other connections with contingents available are even slower).

In the latter case the following multi-criteria search is performed with the option to search for fixed price connections turned off. Note, that the algorithm sometimes fails to compute a connection with a fixed price although one may exist. However, it delivers an alternative connection for dominance testing that is faster than any fixed price connection, if there are any, and in most cases cheaper than the fixed price (see Section 5.4).

## 5 Computational Results

### 5.1 Test Cases

We took the train schedule of trains within Germany from 2003. For our experiments, we used a snapshot of about 5000 real customer queries of Deutsche Bahn AG falling within the week January 13-19, 2003. For all queries, we searched for valid connections within a two-hours time interval. This schedule and the derived time-expanded graph have sizes as shown in Table 1.

Ticket contingents exist for high-speed trains (like ICE, Thalys, TGV, IC, EC) or night trains. Each train  $t$  has a certain capacity  $cap(t)$  (depending on the

scenario	average CPU time in msec.	average extract min operations	average # of Pareto optima	average # of relaxed Pareto optima
MOTIS	1,702	169,114	3.93	7.26
C10	1,889	176,861	4.22	8.05
C20	1,839	175,221	4.13	7.92
C40	1,776	170,976	3.90	7.73
C60	1,734	167,114	3.67	7.46
C80	1,676	161,446	3.43	7.32
C100	1,605	155,219	3.19	7.06

**Table 2.** Computational results for simultaneous search of several tariff types (minimum of regular fare, contingent-restricted special offer and contingent-restricted 50% discount.)

train type). We do not have access to real pre-booking data for trains. Therefore, we simulate the booking status for each train.

A random number of passengers uses each train with contingent restrictions. This number is based on the train class and some other criteria (number of stops, importance of the served stations, etc.). For each of the passengers a random station for entering and leaving the train is chosen evenly distributed from the stations the train visits. We then set thresholds  $x_A(t)$  for the number of passengers required to exhaust the contingent on a train edge of train  $t$  according to the desired level of availability  $A = x\%$ . A travel edge which may have a contingent restriction is called *contingent edge*. For two availabilities  $A, A'$  with  $A < A'$  we require  $x_A(t) \geq x_{A'}(t)$  for all trains  $t$ . So the contingent edges that are not available for some availability  $A$  are not available for every availability  $A^* < A$ .

We consider the following scenarios for the availability of contingents: C10, C20, C40, C60, C80 and C100, where Cx has an availability of  $A = x\%$  on the contingent edges. For comparison we also include the numbers for the search for regular fares (denoted by MOTIS).

For all queries, we assume the same type of passenger, namely a single adult booking early enough to get a 50% discount if a contingent is available. The fixed price for special offers is assumed to be 29 Euros.

## 5.2 Computational Environment

All computations are executed on a standard Intel P4 processor with 3.2 GHz and 4 GB main memory running under Suse Linux 9.2. Our C++ code has been compiled with g++ 3.x and compile option -O3.

## 5.3 Searching for Multiple Tariffs

In the following, we compare computational results for running our code with regular fares only (this version is called MOTIS in the following) and a simultaneous search of several tariff types for different scenarios of available contingents.

In the simultaneous search, we finally select the relaxed Pareto-optimal connections where the fare is taken as the minimum of the regular fare, a contingent-restricted special offer and a contingent-restricted 50% discount on the regular fare if the contingent is available. Table 2 summarizes the key figures obtained in our experiment. In the first column of numbers we present the average CPU time in milliseconds for a single query. The average CPU running time lies within the relatively small range of 1.6s and 1.9s for all scenarios.

As CPU times are very hardware-dependent, we prefer to add representative operations counts for the performance evaluation of algorithms. Previous studies [7] indicated that a suitable parameter for operation counts of a multi-criteria version of Dijkstra’s algorithm is the number of extract minimum operations from the priority queue. This parameter is highly correlated with the CPU running time for the corresponding query. Therefore, we display in the second column of numbers in Table 2 also the average number of these extract operations.

The computational effort increases with decreasing availability of contingents mainly due to two reasons: On the one hand, very few available contingent edges force the algorithm to take longer detours to find cheap contingent prices. On the other hand, a high availability of contingent edges leads to many cheap connections. These help in dominance. There are actually less connections to explore to find cheap alternatives. If about half or more of the contingent edges are available, the contingent version has less operations than the version MOTIS not considering different tariffs.

We note that dominance rules are faster to evaluate if only regular fares are considered (case MOTIS) as less connections are mutually incomparable, see Section 4.1. Therefore, the workload per extract minimum operation is smaller in this version. For all versions using contingent information the correlation between running time and number of extract min operations is plain to see.

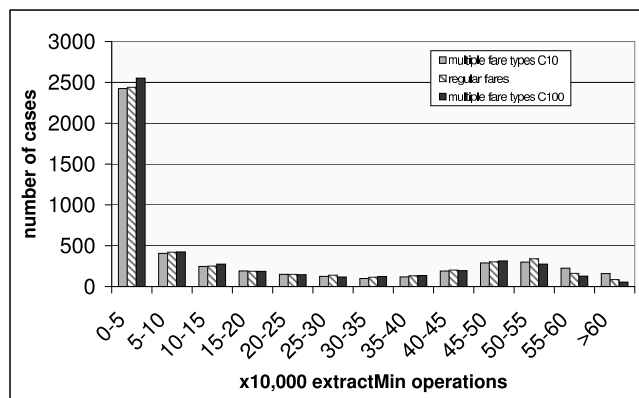
In Figure 1, we also show a histogram on the distribution of extract minimum operations. Case MOTIS mostly lies between the easiest (C100) and most difficult (C10) contingent scenario. The overall distribution looks very similar for all versions of our algorithm. It turns out that about half of all test cases require less than 50,000 extract operations. Such queries are very easy and take only a few milliseconds.

The two remaining columns of Table 2 display the average number of true Pareto optima and the number of relaxed Pareto optima, respectively. These numbers are visualized in Figure 2.

MOTIS offers about 7-8 attractive connections on average, i.e. about four additional connections in comparison to standard Pareto filtering. The more contingents are available, the smaller is the number of Pareto optima, since more fast connections have a cheaper price.

Figure 3 shows the distribution of the number of Pareto optima and relaxed Pareto optima over the test cases for MOTIS and the most difficult contingent version C10.





**Fig. 1.** Histogram showing the distribution of the number of extract min operations from the priority queue. We compare MOTIS (search only for regular fares) with a new version which simultaneously searches for a mixture of fare types.

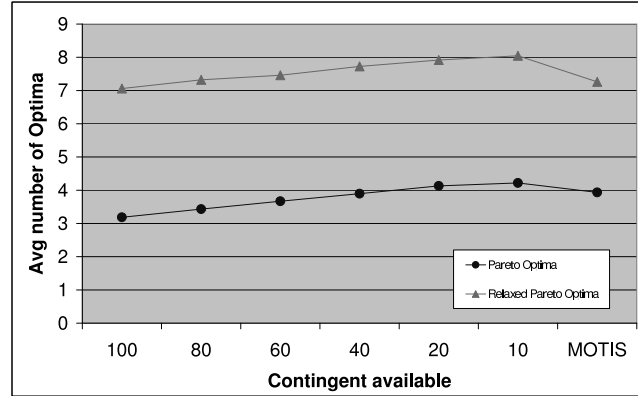
#### 5.4 Fast Search for Fixed Price Connections

We also evaluated the results of the preprocessing phase with our test set. In this experiment, we have run the subroutines “fastest travel time Dijkstra” (FTTD) and our “specialized fixed price Dijkstra” (SFPD). Recall that the purpose of these routines is to find either a fixed price connection, a suitable connection for dominance testing or a certificate, that no fixed price connection exists.

Table 3 shows the average running time, the number of calls to the SFPD, the number of different types of connections and the number of certificates that no fixed price connection exists. The connections are either fixed price connections

scenario	average	# calls to SFPC	# fixed price		# certificate no fixed price conn. exists	# non high-speed conn.	
	CPU time in msec.		conn. from FTTD	SFPD		total	too expensive
C10	204	3641	82	317	2790	1811	373
C20	153	3502	221	841	2224	1714	321
C40	111	3101	622	1490	288	2450	256
C60	90	2579	1144	1742	194	1920	216
C80	70	1534	2189	1275	59	1477	171
C100	45	0	3723	-	0	1277	152

**Table 3.** Results for the fast search for fixed price connections. Either a fixed price connection was found, a certificate that no fixed price connection exists was computed, or a non high-speed connection was found which is cheaper than the fixed price in most cases.

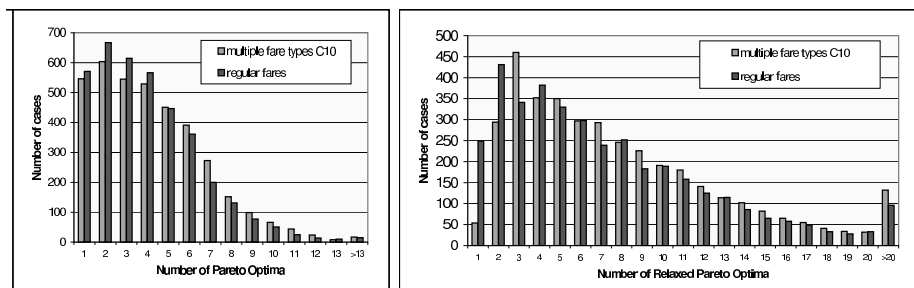


**Fig. 2.** Distribution of the average number of Pareto optima and relaxed Pareto optima for different scenarios of contingent availability.

found by either of the algorithms or non-high-speed connections. In the last column we give the number of cases where such a non-high-speed connection was more expensive than the fixed price. These cases are the only ones, where we have neither a connection to use in dominance testing (either a fixed price connection or a connection without high-speed train that is faster than any fixed price connection) nor the knowledge that no fixed price connection exists. This only happens in 152 to 373 cases, which is 3.04% to 7.5% of the cases, depending on the availability of contingent edges. This is acceptable for a heuristic that runs in at most a fifth of a second on average.

Not surprisingly the total number of fixed price connections increases with the availability of contingents. With decreasing availability the running time, the number of calls to the SFPD and the number of certificates that no fixed price connection exists increase. As the availability of contingent edges increases, the number of fixed price connections determined by the FTDD increases and the number of calls to the SFPD decreases, therefore the running time improves. The number of fixed price connections SFPD determines increases with the availability but decreases if many fixed price connections have already been found by FTDD.

Fixed price search in MOTIS becomes harder the less contingent edges are available (as more detours have to be investigated). Fortunately, with decreasing availability of contingents the number of queries increases significantly for which we can turn off the tariff option fixed price search in the multi-criteria search due to the preprocessing phase.



**Fig. 3.** Histogram showing the distribution of the number of Pareto optima and relaxed Pareto optima.

## 6 Conclusion and Outlook

The focus of this paper was to demonstrate how a large variety of different tariff classes can be incorporated into a multi-objective shortest path framework for travel information. We successively integrated a combined search for regular tariffs and contingent-based tariffs into MOTIS. In our computational experiments we observed that the computational cost of this advanced search increases only slightly over the regular fare search. Sometimes the contingent-restricted versions run even faster. The computational time for a query is less than 1.9s on average. This is significantly more than for a single-criteria search, and further speed-up is desirable.

We also observed that our simple model to represent regular fares within Germany is not as accurate as desired. Hence, future work should concentrate on improved approximations of regular fares. A tighter approximation would allow stricter dominance rules. We do expect significant savings of computational time from stricter dominance rules.

Within this paper we did not consider a specialized search for night trains. Night train search differs from ordinary search in one of the main objectives. A night train passenger typically does not wish to have the fastest connection. Instead, he wishes to have a long sleeping period without interruptions caused by train changes.

### Acknowledgments

This work was supported by the DFG Focus Program 1126 “Algorithmic Aspects of Large and Complex Networks”, grant Mu 1482/2-2. This work was done in cooperation with datagon GmbH, Waldems, Germany. In particular, we wish to thank Wolfgang Sprick for fruitful discussions and close collaboration in the development of MOTIS.

## References

1. Pallottino, S., Scutellà, M.G.: Shortest path algorithms in transportation models: Classical and innovative aspects. In: *Equilibrium and Advanced Transportation Modelling*. Kluwer Academic Publishers (1998)
2. Möhring, R.H.: Verteilte Verbindungssuche im öffentlichen Personenverkehr: Graphentheoretische Modelle und Algorithmen. In: *Angewandte Mathematik - insbesondere Informatik*, Vieweg (1999) 192–220
3. Schulz, F., Wagner, D., Weihe, K.: Dijkstra’s algorithm on-line: An empirical case study from public railroad transport. *ACM Journal of Experimental Algorithmics* **5** (2000) Article 12
4. Müller-Hannemann, M., Weihe, K.: Pareto shortest paths is often feasible in practice. In: *Algorithm Engineering – WAE 2001*. Volume 2141 of LNCS. Springer (2001) 185–198
5. Müller-Hannemann, M., Schnee, M., Weihe, K.: Getting train timetables into the main storage. In: *Proceedings of the 2nd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2002)*. Volume 66 of *Electronic Notes in Theoretical Computer Science*. Elsevier (2002)
6. Pyrga, E., Schulz, F., Wagner, D., Zaroliagis, C.: Experimental comparison of shortest path approaches for timetable information. In: *Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments, SIAM (2004)* 88–99
7. Müller-Hannemann, M., Schnee, M.: Finding all attractive train connections by multi-criteria Pareto search. In: *Proceedings of the 4th Workshop in Algorithmic Methods and Models for Optimization of Railways (ATMOS 2004)*. LNCS, Springer (2005), to appear.
8. Cooke, K.L., Halsey, E.: The shortest route through a network with time-dependent internodal transit times. *Journal of Mathematical Analysis and Applications* **14** (1966) 493–498
9. Orda, A., Rom, R.: Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM* **37** (1990)
10. Orda, A., Rom, R.: Minimum weight paths in time-dependent networks. *Networks* **21** (1991)
11. Kostreva, M.M., Wiecek, M.M.: Time dependency in multiple objective dynamic programming. *Journal of Mathematical Analysis and Applications* **173** (1993) 289–307
12. Nachtigal, K.: Time depending shortest-path problems with applications to railway networks. *European Journal of Operations Research* **83** (1995) 154–166
13. Brodal, G.S., Jacob, R.: Time-dependent networks as models to achieve fast exact time-table queries. In: *Proceedings of the 3rd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2003)*. Volume 92 of *Electronic Notes in Theoretical Computer Science*. Elsevier (2004)
14. Pyrga, E., Schulz, F., Wagner, D., Zaroliagis, C.: Towards realistic modeling of time-table information through the time-dependent approach. In: *Proceedings of the 3rd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2003)*. Volume 92 of *Electronic Notes in Theoretical Computer Science*. Elsevier (2004) 85–103
15. Müller-Hannemann, M., Schulz, F., Wagner, D., Zaroliagis, C.: Timetable information: Models and algorithms. In: *Algorithmic Methods for Railway Optimization*. LNCS, Springer (2005), to appear.

16. (HAFAS) A timetable information system by HaCon Ingenieurgesellschaft mbH, Hannover, Germany. <http://www.hacon.de/hafas/>.
17. Theune, D.: Robuste und effiziente Methoden zur Lösung von Wegproblemen. Teubner Verlag, Stuttgart (1995)

# Station Location — Complexity and Approximation

Steffen Mecke<sup>1</sup>, Anita Schöbel<sup>2</sup>, Dorothea Wagner<sup>1</sup>

<sup>1</sup> University of Karlsruhe, {mecke,dwagner}@ira.uka.de

<sup>2</sup> University of Göttingen, schoebel@math.uni-goettingen.de

**Abstract** We consider a geometric set covering problem. In its original form it consists of adding stations to an existing geometric transportation network so that each of a given set of settlements is not too far from a station. The problem is known to be  $\mathcal{NP}$ -hard in general. However, special cases with certain properties have been shown to be efficiently solvable in theory and in practice, especially if the covering matrix has (almost) consecutive ones property. In this paper we are narrowing the gap between intractable and efficiently solvable cases of the problem. We also present an approximation algorithm for cases with almost consecutive ones property.

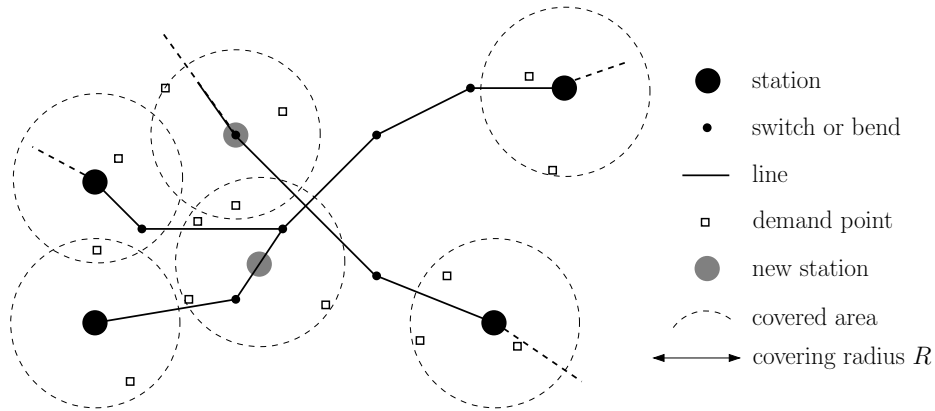
**Keywords.** Station Location, facility location, complexity approximation

## 1 Introduction

The STATION LOCATION problem consists of placing new stops along a given public transportation network in order to reach all potential customers. Building and maintaining new stops causes, of course, additional costs to the maintainers of the network. So the goal is to minimize the number of new stations to be built. The problem has arisen from a project with the largest German railway company with the goal of improving attractiveness of the public transportation network. A formal definition of the problem is the following:

**Problem 1** (STATION LOCATION, cf. Fig. 1). *Given a geometric graph  $G = (V, E)$ , i.e. a set  $V$  of vertices in the plane (stations, switches, bends) and a set  $E$  of edges (tracks or lines) represented as straight line segments, a set  $\mathcal{P}$  of points in the plane (settlements or demand points), and a fixed radius  $R$ . Find minimum set of vertices  $\mathcal{S}$  (new stops) on the edges that cover  $\mathcal{P}$ , i.e.  $\mathcal{P} \subseteq \text{cov}(\mathcal{S})$ , where  $\text{cov}(\mathcal{S}) = \{x \in \mathbb{R}^2 : d(x, \mathcal{S}) \leq R\}$ . In the weighted version there are costs associated with the edges, and the goal is to minimize the sum of costs.*

This  $\mathcal{NP}$ -hard problem has, in the form presented here, been brought up in [1] while similar problems have been studied already before. The authors of [2] consider a variant of station location with similar structural properties. Some of



**Figure 1.** Problem example

the special cases mentioned there are also of interest in our case. STATION LOCATION is closely related to several other hard optimization problems, especially  $k$ -median,  $k$ -center, and facility location problems, as well as the COVERING BY DISCS problem (see [3,4]). It can also be seen as a SET COVERING problem which has many applications and has been widely studied. For an overview of work on SET COVERING problem see [5].

Experimental studies have shown that most practical instances of the problem can be solved rather quickly ([6]). Many (sub-)instances of the SET COVERING problem occurring in practice have the so called consecutive ones property, that is, the ones in each row of the matrix of the set covering problem occur consecutively (see Def. 2). This property is often fulfilled due to the geometric setting of STATION LOCATION. It plays an important role in explaining the relatively well-behaved nature of real-world instances of the STATION LOCATION problem. In [7] and [8] this topic is further illuminated. In this paper we try to further converge with theoretical results to the practical findings by identifying specializations of the STATION LOCATION problem that are still hard on the one hand, and giving efficient algorithms for approximating and solving quite general variants of the problem on the other hand. Note that our “positive” algorithmic results are usually valid for the more general SET COVERING problem while the “negative” hardness results concern the more special STATION LOCATION.

Overview of the paper: Section 2 gives some basic definitions and summarizes the negative results, namely the  $\mathcal{NP}$ -completeness of special cases of the problem. Section 3 describes an approximation algorithm based on a block-based reformulation of the integer linear program of the SET COVERING problem. And Section 4 shortly describes an approach to get a grip on the hardness of the problem by means of parameterized complexity.

## 2 $\mathcal{NP}$ -Hardness Results

It was shown in [9] that there exists a finite set  $\mathcal{C}$  of *candidate* locations for new stops which contains an optimum solution and which can be computed by an algorithm which is polynomial in the sizes of  $G$  and  $P$ . In the following we will assume that such a set of candidates is given.

**Definition 1.** Let  $\mathcal{P} := \{1, \dots, M\}$  and  $\mathcal{C} := \{1, \dots, N\}$ . The matrix  $A^{\text{cov}} = (a_{pc})$  with  $a_{pc} := \begin{cases} 1 & \text{if } c \text{ covers } p \\ 0 & \text{otherwise} \end{cases}$  (for all  $p \in \mathcal{P}, c \in \mathcal{C}$ ) is called the covering matrix of an instance of STATION LOCATION.

We will sometimes use the terms “demand points” and “rows” (resp. “candidates” and “columns”) as synonyms.

Given the above, STATION LOCATION can be seen as a special case of the well-studied SET COVERING (aka. HITTING SET) problem. We use the following notation to describe it as a linear problem:

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & A^{\text{cov}} x \geq \mathbf{1}_M \\ & x \in \{0, 1\}^N, \end{aligned} \tag{SCP}$$

where  $\mathbf{1}_M \in \mathbb{R}^M$  denotes the vector consisting of  $M$  ones,  $c \in \mathbb{R}^N$  contains the costs of the columns, and  $A^{\text{cov}}$  is an  $M \times N$ -matrix with elements  $a_{mn} \in \{0, 1\}$ . We may assume, without loss of generality, that all rows and columns of  $A^{\text{cov}}$  have at least one non-zero entry and that the costs  $c_j$  are positive.

The goal is to find an optimal solution  $x^*$ , i.e. a solution with minimal costs, or equivalently, an optimal set  $\mathcal{C}^* \subseteq \mathcal{C} = \{1, \dots, N\}$  of columns of  $A^{\text{cov}}$ , where  $\mathcal{C}^* = \{n \in \mathcal{C} : x_n^* = 1\}$ .

**Theorem 2 ([1]).** STATION LOCATION is  $\mathcal{NP}$ -hard.

**Definition 2.** 1. A matrix  $A^{\text{cov}}$  over  $\{0, 1\}$  has the **strong consecutive ones property (strong C1P)** if all of its rows  $m \in \{1, \dots, M\}$  satisfy the following condition for all  $j_1, j_2 \in \{1, \dots, N\}$ :

$$a_{mj_1} = 1 \text{ and } a_{mj_2} = 1 \implies a_{mj} = 1 \text{ for all } j_1 \leq j \leq j_2.$$

2. A matrix has the **consecutive ones property (C1P)** if there exists a permutation of its columns such that the resulting matrix has the strong consecutive ones property
3. If  $A_m^{\text{cov}}$  is a row of  $A^{\text{cov}}$  let  $bl_m$  be its number of blocks of consecutive ones.

If a matrix has the consecutive ones property the permutation of the columns making the ones appear consecutively can be found by using the algorithm of [10,11]. This algorithm can be performed in  $O(MN)$ . Without loss of generality we can therefore assume that a matrix with consecutive ones property is already ordered, i.e. we assume that its ones already appear consecutively in all of its rows, i.e.  $bl_m = 1$  for  $m = 1, \dots, M$ . We say that a SET COVERING problem has C1P if its covering matrix  $A^{\text{cov}}$  has C1P.



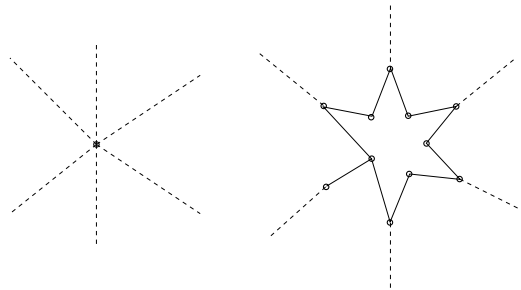
**Theorem 3** ([9]). STATION LOCATION is polynomially solvable if the covering matrix has C1P.

*Proof.* It can be shown that a covering matrix with C1P is totally unimodular. Now remember the well-known fact that a linear program with totally unimodular matrix has an integer optimal solution. Hence, the SET COVERING problem can be solved by relaxing the integer constraints and solving the (non-integer) linear program in polynomial time.  $\square$

More efficient approaches for solving SET COVERING problems with C1P can be found in [12,7]. As an example, it is easy to show that the covering matrix of a STATION LOCATION problem has C1P if no settlement can be covered by new stops on more than one line. On the negative side we have

**Theorem 4.** STATION LOCATION is  $\mathcal{NP}$ -hard in the strong sense (with unit costs), even for the case that no settlement can be covered from more than two lines.

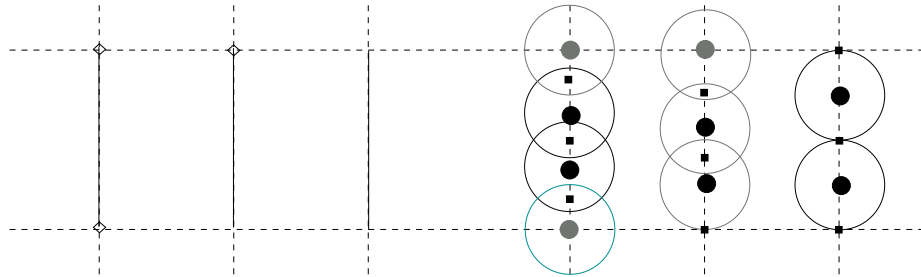
*Proof.* By reduction from PLANAR VERTEX COVER (i.e., given a planar graph  $G = (V, E)$ , find a minimum set of nodes  $V' \subset V$  such that for every edge, at least one of its end nodes is in  $V'$ ). In [13] it has been shown that this problem remains  $\mathcal{NP}$ -complete even for planar graphs with maximum degree 6. This can be further constrained to maximum degree 3 (which has been shown already in [14]). To this end we replace every vertex of degree 6 by the gadget of eleven vertices shown in Fig. 2. A very similar procedure works for vertices of degree 4 and 5. The resulting graph  $G'$  has  $|V| + 10v_6 + 8v_5 + 6v_4$  vertices, maximum degree 3, and it is still planar ( $v_6, v_5$ , and  $v_4$  are the numbers of vertices of degree 6, 5, and 4, resp., in the original graph  $G$ ). It follows almost immediately that a vertex cover of size  $K$  in  $G$  exists if and only if a vertex cover of size  $K' := K + 5v_6 + 4v_5 + 3v_4$  exists in  $G'$ . We will call this restricted problem PD3VC.



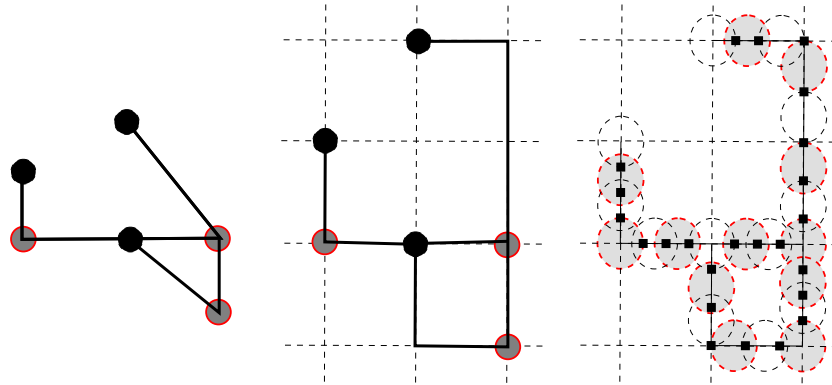
**Figure 2.** PLANAR DEG-6 VERTEX COVER  $\propto$  PLANAR DEG-3 VERTEX COVER

The next step is the reduction PD3VC  $\propto$  STATION LOCATION. There exists a planar orthogonal unit grid drawing of  $G'$  with  $O(n^2)$  area and at most

$2n + 4$  bends which is constructible in polynomial time (cf. [15], Theorem 4.16). We construct an instance of STATION LOCATION as follows: Let  $R = 1/4$  and construct the settlements and candidates as follows. Each edge consists of a sequence of segments of unit length in the grid. Each such segment has either two, one, or zero vertices of  $G'$  at its ends. First, replace every vertex in  $V$  by a candidate. Then, replace the segments by settlements and candidates according to the gadgets shown in Fig. 3. An example is sketched in Fig. 4.



**Figure 3.** Three gadgets (right) for the three different types of segments (left). Settlements are depicted by squares, candidates for stations as small discs; the big circles indicate the covering radius; the grid is dashed.



**Figure 4.**  $\text{PD3VC} \propto \text{STATION LOCATION}$  (with vertex cover resp. station cover in grey; candidates omitted)

Note that, after all segments have been replaced, there are exactly  $|V| + 2|S|$  candidates and  $3|S|$  settlements. Further note, that settlements are covered from candidates from different segments if and only if the corresponding segments are adjacent and no settlement is covered by more than two candidates. Finally, let

$K'' := K' + |S|$ . A vertex cover of cardinality  $K'$  in  $G'$  exists if and only if there is a solution with cardinality  $K''$  for the constructed instance of STATION LOCATION. As the construction works in polynomial time, this implies the  $\mathcal{NP}$ -hardness of this variant of STATION LOCATION.  $\square$

**Corollary 1.** *The (unweighted) STATION LOCATION problem remains  $\mathcal{NP}$ -complete even for the case that the covering matrix can be split into two submatrices  $A$  and  $B$  such that  $A^{\text{cov}} = (A|B)$ , and  $A$  and  $B$  have the consecutive ones property (even if  $(A|B)$  has exactly two 1s per row,  $A$  has no more than one 1 per row and  $B$  has no more than two 1s per row).*

*Proof.* Consider the instance of STATION LOCATION and the graph  $G'$  constructed in the above proof. There are two classes of candidates: Candidates corresponding to vertices of  $G'$  and candidates on edges of  $G'$ . Assign columns corresponding to candidates of the first class to  $A$  and columns corresponding to candidates of the second class to  $B$ . Order the columns of  $B$  in the natural way, namely corresponding to their order on the edges between two vertices of  $G'$  (cf. Fig. 4). Then the following properties hold:

1. No row of  $A$  has more than one entry, because every station can only be reached by one class- $A$  vertex. It follows that  $A$  has C1P.
2. An ordering of the columns of  $B$  following the above rule exists. For every row covered by columns of  $B$  the (up to two) columns covering it are consecutive.

It follows that  $A$  and  $B$  have C1P, and no row of  $(A|B)$  has more than two non-zero entries.  $\square$

Note, however, the following result:

**Definition 3.** *Let  $l_m$  ( $r_m$ ) be the index of the leftmost (rightmost) 1 in the  $m$ -th row of  $A^{\text{cov}}$ . A matrix  $A^{\text{cov}}$  with strong C1P is **strictly monotone** if the sequence  $(l_m)_{1 \leq m \leq M}$  and  $(r_m)_{1 \leq m \leq M}$  are strictly increasing.*

**Lemma 1 ([7]).** *Let  $A = (A_1|A_2)$  where  $A_1$  and  $A_2$  both are strictly monotone matrices. Then the SET COVERING problem with coefficient matrix  $A$  can be solved in polynomial time.*  $\square$

### 3 Approximation

It is shown in [16,17] that SET COVERING cannot be approximated within a factor of  $O(\log(n))$  unless some likely assumption on complexity classes holds. Using the so called *shifting technique* of [18], however, it was proven in [19] that a PTAS exists for COVERING BY DISCS, which is similar to STATION LOCATION except for the fact that the locations for new stations are not restricted but can be chosen anywhere in the plane. The authors of [20] find a PTAS for a version of STATION LOCATION where stations cannot be arbitrarily close to each other. However, it seems that their techniques cannot be adapted to our problem (although they would probably work well in practice).

We therefore followed a different approach for approximating more general instances, especially those having only a few blocks of consecutive ones per row.

### 3.1 A Block-based Reformulation

Since SET COVERING problems with C1P can be solved efficiently, our idea is to split each row  $m$  with more than one block of consecutive ones (i.e.  $bl_m > 1$ ) into  $bl_m$  rows, each of them satisfying C1P. We then require that at least one of these rows needs to be covered.

Now consider a zero-one matrix  $A^{\text{cov}}$  with  $M$  rows, such that  $bl_m = 1$  for  $m = 1, \dots, p$ , and  $bl_m > 1$  in the remaining rows  $p + 1, \dots, M$ .

For the  $i$ th block of consecutive ones in row  $m$  let

- $f_{m,i}$  be the column of the first 1 of block  $i$ , and
- $l_{m,i}$  be the column of its last 1.

This means that

$$a_{mn} = \begin{cases} 1 & \text{if there exists } i \in \{1, \dots, bl_m\} \text{ such that } f_{m,i} \leq n \leq l_{m,i} \\ 0 & \text{otherwise.} \end{cases}$$

Consider a row  $A_m^{\text{cov}}$  of  $A^{\text{cov}}$  with  $bl_m > 1$ . According to the transformation introduced in [8] we replace  $A_m^{\text{cov}}$  by  $bl_m$  rows  $B_{m,1}, B_{m,2}, \dots, B_{m,bl_m}$ , each of them containing only one single block of row  $A_m$ , i.e., we define the  $j$ th element of row  $B_{m,i}$  as

$$(B_{m,i})_j = \begin{cases} 1 & \text{if } f_{m,i} \leq j \leq l_{m,i} \\ 0 & \text{otherwise.} \end{cases}$$

Hence, due to [8], the SET COVERING problem (SCP) is equivalent to

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & A_m^{\text{cov}} x \geq 1 && \text{for } m = 1, \dots, p \\ & B_{m,i} x \geq y_{m,i} && \text{for } m = p + 1, \dots, M, i = 1, \dots, bl_m \\ & \sum_{i=1}^{bl_m} y_{m,i} \geq 1 && \text{for } m = p + 1, \dots, M \\ & y_{m,i} \in \{0, 1\} && \text{for } m = p + 1, \dots, M, i = 1, \dots, bl_m \\ & x \in \{0, 1\}^N. \end{aligned} \tag{SCP'}$$

It is more convenient to rewrite (SCP') in matrix form. To this end, we define

- the matrix  $A$  as the first  $p$  rows of  $A^{\text{cov}}$ ,
- $bl = \sum_{m=p+1}^M bl_m$  as the total number of blocks in the “bad” rows of  $A^{\text{cov}}$ , i.e., in rows of  $A^{\text{cov}}$  without C1P,
- $I$  as the  $bl \times bl$  identity matrix,
- $B$  as the matrix containing the  $bl$  rows  $B_{m,i}$ ,  $m = p+1, \dots, M$ ,  $i = 1, \dots, bl_m$ ,
- $C$  as a matrix with  $M - p$  rows and  $bl$  columns, with elements

$$c_{ij} = \begin{cases} 1 & \text{if } \sum_{m=p+1}^{p+i-1} bl_m < j \leq \sum_{m=p+1}^{p+i} bl_m \\ 0 & \text{otherwise.} \end{cases}$$

(SCP') can hence be reformulated as

$$\begin{aligned}
\min \quad & cx \\
\text{s.t.} \quad & Ax \geq \underline{1}_p \\
& Bx - Iy \geq \underline{0}_{bl} \\
& Cy \geq \underline{1}_{M-p} \\
& x \in \{0, 1\}^N, \\
& y \in \{0, 1\}^{bl}.
\end{aligned} \tag{SCP''}$$

The constraint  $Cy \geq \underline{1}_{M-p}$  makes sure that at least one block of each row  $A_m^{\text{cov}}$  with  $m \geq p + 1$  is covered.

Note that all three matrices  $A$ ,  $B$ , and  $C$  have C1P. Unfortunately, the coefficient matrix of (SCP'') does not have C1P and is in general even not totally unimodular.

### 3.2 Approximation

From Cor. 1 we know the complexity status of SET COVERING problems with at most  $k$  blocks of consecutive ones per row: Let  $k$  be an upper bound on the number of blocks in each row of  $A$ , i.e., such that  $bl_m \leq k$  for all  $m = 1, \dots, M$ .

**Corollary 2.** *For  $k = 1$  the SET COVERING problem is polynomially solvable, for all fixed  $k \geq 2$  the problem is NP-hard.*

*Proof.* For  $bl_m = 1$  the problem has the consecutive ones property and is thereby totally unimodular. For  $k = 2$  one can use a reduction to min *vertex cover* (see [3]) to obtain a set covering problem with exactly two nonzero elements in each row; hence a set covering problem with at most two blocks per row.

To solve (SCP) we suggest Alg. 1, for which we will show that it provides a  $k$ -approximation, if  $k$  is an upper bound on the number of blocks of consecutive ones per row.

Note that Alg. 1 can be solved by linear programming, since in line 5, the coefficient matrix has C1P.

**Theorem 5.** *Algorithm 1 is a  $k$ -approximation algorithm, where*

$$k = \max_{m=1, \dots, M} bl_m.$$

*Proof.* Let  $(x^*, y^*)$  be an optimal solution, and  $(x', y')$  be an optimal solution of the linear programming relaxation of (SCP''). This means that

$$cx' \leq cx^*. \tag{1}$$

Now note that

$$ky'_{m,i} \geq \tilde{y}_{m,i}. \tag{2}$$

---

**Algorithm 1:**  $k$ -approximation
 

---

**Input:**  $M \times N$  matrix  $A$   
**Output:** approximate solution  $\tilde{x}$

- 1 Solve LP-Relaxation of the reformulation (SCP'') to obtain a solution  $(x', y')$ ;
- 2 **for**  $m := 1, \dots, M$  **do**
- 3     Find an index  $i(m)$  with  $y'_{m,i(m)} \geq y'_{m,i}$  for all  $i = 1, \dots, bl_m$ ;
- 4     Define
 
$$\tilde{y}_{m,i} = \begin{cases} 1 & \text{if } i = i(m) \\ 0 & \text{otherwise.} \end{cases}$$
- 5 Solve  $\min\{cx : Bx \geq \tilde{y}, x \in \{0, 1\}^N\}$  to obtain  $\tilde{x}$ ;
- 6 **return**  $\tilde{x}$ ;

---

This trivially holds for  $\tilde{y}_{m,i} = 0$ , while for  $\tilde{y}_{m,i} = 1$  we know that

$$\begin{aligned}
 y'_{m,i} &= \max_{k=1, \dots, bl_m} y'_{m,k} \\
 &\geq \frac{1}{bl_m} \sum_{k=1, \dots, bl_m} y'_{m,k} \\
 &\geq \frac{1}{bl_m} \quad \text{since } Cy \geq \mathbf{1}_{M-p} \\
 &\geq \frac{1}{k}
 \end{aligned}$$

Moreover,  $\min\{cx : Bx \geq \tilde{y}\} = \min\{cx : Bx \geq \tilde{y}, x \in \{0, 1\}^N\}$ , since in any optimal solution of the latter,  $x \leq 1$ , and the integrality constraint  $x \in \mathbb{N}^N$  can be deleted since  $B$  has CIP and hence is totally unimodular. Now estimate  $B(kx')$  as

$$B(kx') = kBx' \geq ky' \geq \tilde{y},$$

where the last inequality is due to (2). In other words,  $kx'$  is feasible for  $\{x : Bx \geq \tilde{y}\}$ , and hence we get

$$\begin{aligned}
 kcx' &\geq \min\{cx : Bx \geq \tilde{y}\} \\
 &= \min\{cx : Bx \geq \tilde{y}, x \in \{0, 1\}^N\} = c\tilde{x}
 \end{aligned}$$

Combining the latter with (1) we finally obtain  $c\tilde{x} \leq kcx' \leq kcx^*$ . □

## 4 Further Issues

*Parameterized Complexity.* A further means of tackling STATION LOCATION is to apply parameterized complexity techniques. It does not make much sense in our context to take the “canonical” parameter, the number of steps of the solution. We want a parameter that is small but there are usually many steps in a solution. Instead, we choose the maximum distance  $k$  between the first and

**Algorithm 2:** FPT Algorithm

---

**Input:** Covering Matrix  $A^{\text{cov}}$   
**Output:** Optimal solution  $\text{OPT} \in S$   
**Data:**  $S$ , a collection of partial solutions covering rows  $1, \dots, r$

- 1 Sort columns of  $A^{\text{cov}}$  lexicographically;
- 2 Initialize  $S$  with the partial solution  $\emptyset$ , covering no row of  $A^{\text{cov}}$ ;
- 3 **forall** rows  $r$  of  $A^{\text{cov}}$  **do**
- 4     **forall** partial solutions  $s \in S$  that do not cover  $r$  **do**
- 5         **forall** columns  $g$  of  $A^{\text{cov}}$  that cover  $r$  **do**
- 6             Add a solution  $s \cup \{g\}$  to  $S$ ;
- 7             Remove  $s$  from  $S$ ;
- 8     Remove from  $S$  all duplicate partial solutions (covering the same set of rows) except the smallest such solution;

---

the last non-zero entry of the covering matrix in every column. We have found the following result which is especially useful for instances which have a fairly “linear” structure and therefore their covering matrix is almost a band diagonal matrix.

**Theorem 6.** STATION LOCATION is solvable in  $O(\text{poly}(m, n) \cdot 2^k)$  if the distance between the first and last non-zero entry is not greater than  $k$  for every column of  $A^{\text{cov}}$ . If  $k = \Omega(m)$  this leads to an exponential running time. But for small values of  $k$  it can be quite efficient.

$$\left( \begin{array}{ccccccc} \text{110001101} & & & & & & 0 \\ \text{-----} & & & & & & \\ & \text{1} & \text{-----} & \text{1} & & & \\ & & & \vdots & & & \\ & & & & \text{-----} & & \\ & & & & \text{1} & \text{-----} & \text{1} \end{array} \right) \Bigg\}^k$$

*Proof (sketch).* In each iteration, after step 8,  $S$  contains the optimal solution covering rows  $1, \dots, r$ . The correctness follows from this property. The running time follows from the fact that after each iteration all solutions in  $S$  cover rows  $1, \dots, r$ , and no solution covers any row after  $r + k$ , so  $|S| \leq 2^k$ .  $\square$

*Outlook.* We are not aware of a constant factor approximation algorithm for STATION LOCATION nor a PTAS. There is no FPTAS for STATION LOCATION since it is strongly  $\mathcal{NP}$ -complete (see [21] for a comprehensive introduction to approximation algorithms). The relatively nice behaviour of practical instances can be explained by several factors. First, the geometric nature of the problem along with the fact that most settlements can be reached by only a few lines results in covering matrices that are close to having C1P. This and reduction techniques described in [6] allow large portions of the instances to be solved efficiently, resulting in relatively small problem kernels. Secondly, the distribution of settlements allows to apply the shifting technique even if its effectiveness has not been proven theoretically for our version of the problem.

A further approach not mentioned so far could be to use techniques applied to unit disc graphs as many hard problems are easy if restricted to unit disc graphs.

## References

1. Hamacher, H.W., Liebers, A., Schöbel, A., Wagner, D., Wagner, F.: Locating new stops in a railway network. In: Proc. ATMOS 2001. Volume 50(1) of Electronic Notes in Theoretical Computer Science. (2001)
2. Kranakis, E., Penna, P., Schlude, K., Taylor, D.S., Widmayer, P.: Improving customer proximity to railway stations. In: Proc. CIAC 2003, Rome. Volume 2654 of LNCS. (2003) 265–276
3. Garey, M.R., Johnson, D.S.: Computers and Intractability. W. H. Freeman and Company (1979)
4. Johnson, D.S.: The NP-completeness column: An ongoing guide. *J. Algorithms* **3** (1982) 182–195
5. Caprara, A., Fischetti, M., Toth, P.: Algorithms for the set covering problem. *Annals of Operations Research* **98** (2000) 353–371
6. Mecke, S., Wagner, D.: Solving geometric covering problems by data reduction. In: Proc. ESA '04, Norway. Volume 3221 of LNCS. (2004) 760–771
7. Schöbel, A.: Set covering problems with consecutive ones property. Technical report, University of Kaiserslautern (2005)
8. Ruf, N., Schöbel, A.: Set covering problems with almost consecutive ones property. *Discrete Optimization* **1** (2004) 215–228
9. Schöbel, A., Hamacher, H.W., Liebers, A., Wagner, D.: The continuous stop location problem in public transportation networks. Technical Report 81, Wirtschaftsmathematik, University of Kaiserslautern (2002) Submitted to a journal.
10. Booth, K.S., Lueker, G.S.: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Science* **13** (1976) 335–379
11. Meidanis, J., Porto, O., Telles, G.P.: On the consecutive ones property. *Discrete Applied Mathematics* **88** (1998) 325–354
12. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley (1988)
13. Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified NP-complete problems. In: Proc. STOC '74, ACM Press (1974) 47–63
14. Uehara, R.: NP-complete problems on a 3-connected cubic planar graph and their application. Technical report, Tokyo Woman's Christian University (1996)
15. di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing. Prentice Hall, Inc. (1999)
16. Lund, C., Yannakakis, M.: On the hardness of approximating minimization problems. *J. of the ACM* **41** (1994) 960–981
17. Feige, U.: A threshold of  $\ln n$  for approximating set cover. *J. ACM* **45** (1998) 634–652
18. Hochbaum, D.S., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM* **32** (1985) 130–136
19. Franceschetti, M., Cook, M., Bruck, J.: A geometric theorem for approximate disk covering algorithms. Technical report, California Institute of Technology (2001)
20. Galota, M., Glaßer, C., Reith, S., Vollmer, H.: A polynomial-time approximation scheme for base station positioning in umts networks. In: Proc. DIALM '01, ACM Press (2001) 52–59
21. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M.: Complexity and Approximation. Springer (1999)