Electronic Research Archive of Blekinge Institute of Technology
<http://www.bth.se/fou/>

This is an author produced version of a journal paper. The paper has been peer-reviewed but may not include the final publisher proof-corrections or journal pagination.

Citation for the published Journal paper:

Title:

Author:

Journal:

Year:

Vol.

Issue:

Pagination:

URL/DOI to the paper:

Access to the published version may require subscription.

Published with permission from:

A GREEDY ALGORITHM FOR RAILWAY TRAFFIC RE-SCHEDULING DURING DISTURBANCES: A SWEDISH CASE

Johanna Törnquist Krasemann

Address: Department of Software Engineering and Computer Science,
Blekinge Institute of Technology, Sweden
E-mail: Johanna.Tornquist@bth.se

Abstract

The positive trend of increased use of railway transportation in Europe has resulted in an increased sensitivity to and occurrence of traffic disturbances. In addition to the need for extensions of the infrastructure, the need to effectively limit and predict the effects of disturbances becomes apparent. The kernel of the disturbance management problem is to revise the original timetable in line with the new conditions and decide where, when and how trains should overtake or meet to minimise the negative effect of the disturbance. In previous research we have designed an optimisation-based approach for re-scheduling which seems promising but for some scenarios it is difficult to find good solutions within seconds. Also more detailed constraints will have to be included, which makes the problem even more complex and difficult to solve. Therefore, we have developed a greedy algorithm which effectively delivers good solutions within the permitted time. To quickly retrieve a feasible solution the algorithm performs a depth-first search using an evaluation function to prioritise when conflicts arise and then branches according to a set of criteria. A performance analysis of the algorithm was carried out using simulated experiments showing its strengths and weaknesses.

1 Introduction

Freight as well as public transportation is a large and important part of our economy and daily life, and railway transportation has a significant share. As an effect of the increasing environmental awareness and desire to decrease emissions, noise pollution and accidents, political aims of increasing the market share of railway freight transportation have been stated, see e.g. [1]. However, in line with the increasing traffic flow and density, the railway networks are facing difficulties with congestion and insufficient reliability. The railway traffic networks in several European countries and regions are partly oversaturated, highly sensitive to even small disturbances and have low average punctuality [2]. For instance, some parts of the Swedish railway network have such a high capacity usage that even a minor incident can propagate and cause large disturbances. During the two most traffic-intensive hours per day, 41 % of the entire Swedish railway network is considered saturated (capacity utility of 81-100%) with a low average speed and high sensitivity to disturbances. Statistics shows that 31 % of the reported delays in the Swedish network during 2006 were knock-on delays. Furthermore, recent statistics from the UIRR (the International Union of combined Road-Rail transport companies) also show that 41% of the cargo trains in the analysed major freight transport corridors arrived 30 minutes later than scheduled, or more [3].

Figure 1. The growth of the Swedish railway traffic volumes and accumulated delay between year 2002 and 2008 (2002 has index 100). Source: Banverket annual reports 2004-2008 [4]-[8].

In order for the railway to become and remain an attractive means of transportation, the occurrence of disturbances needs to be limited as well as the consequences of the disturbances that do occur. While the most straightforward way to decrease the effects of disturbances would be to eliminate the risk of primary disturbances arising, it is simply not feasible. Some of the causes can be predicted and prevented from happening, while others cannot. Therefore the ability to deal with the disturbances that do occur can be argued to be

as important as eliminating potential causes of primary disturbances. This paper focuses on how to handle disturbances in railway traffic by effective re-scheduling rather than limiting the occurrence of primary disturbances (i.e. initial disturbances).

We will present previous results from a research project at Banverket (the Swedish Rail Administration) where focus is on the design of an algorithm for re-scheduling the traffic during or after a disturbance. The driving force is to achieve sufficiently good solutions within a short time. The performance of the algorithm has been evaluated for different types of disturbances in simulated experiments. The simulation experiments have been carried out on one of the densest and central parts of the Swedish railway traffic network and have been based on real data provided by Trafikverket (formerly known as Banverket).

2 Railway Traffic Disturbance Management

In this paper, a disturbance is considered to be a situation where the master schedule, or established timetable, has become invalid because one train (or several) is deviating from its schedule and the prerequisites consequently change. The event triggering a disturbance could be a signal malfunction on a track section which temporarily decreases the maximum allowed speed and causing trains to have an increased running time on that section. It could also be e.g. a no-show of staff resulting in a delayed train departure, or reduced speed of a train set due to partial engine failure or an unannounced increase in train set length and weight, etc.

Handling a disturbance in a railway network and re-scheduling the traffic is typically handled manually by traffic managers that only have very limited access to support systems to make use of relevant information, analyse the consequences of the disturbance as well as the effects of their decision-making. This limitation hampers the possibilities to achieve sustainable and system-optimal decision-making and to provide the stakeholders (e.g. train operators, cargo owners, commuters) with reliable prognoses about the situation. The time available for decision-making and consequence analysis is also limited and depends on the situation. The aim is thus to find sustainable and sufficiently good solutions within reasonable time. For illustrative purposes, let us consider the small-scale example of re-scheduling the traffic in Figure 2. It shows a graph of the planned railway traffic (train 101, 102 and 103) on the single-tracked line between the stations Enstaberg and Åby with several intermediate stations, and where the time is on the y-axis. The trains belong to three different private transport companies; 101 is a heavy block train, 102 is a long-distance high-speed passenger train while 103 is an express cargo train. Shortly before train 101 leaves Ålberga for Kolmården the track section suffers from a signalling problem (indicated by the thick line below the section) which decreases the maximum speed to 70 km/h. Thus, all trains passing through get an increased running time of 15 minutes and the revised timetable for each train (the dotted lines) is thus postponed 15 minutes each. This results in conflicts when the paths of the trains cross on the single-tracked line between stations, and that is forbidden due to safety restrictions. When and how the trains now should meet in order to resolve the conflicts need to be decided by the traffic managers. One policy could be to let the first train that can enter a section are given priority to that section, i.e. First

Come, First Served (FCFS) while another could be for example to give highest priority to all long-distance trains.

In Figure 3, two solutions are illustrated. In the first solution, train 103 and 101 meet at Getå (as in the initial timetable) but that forces train 103 to wait a long time at the station and since the train is a long-distance high-speed train, the corresponding operator does not accept that it will be delayed that much and in favour of a cargo train. Instead, the operator wants the traffic manager to arrange for the trains to meet in Kolmården instead and thereby reducing the waiting time for 103 at Getå significantly. When the traffic manager considers that alternative he discovers that in the end, the waiting time for train 103 over all stations will be the same in both solutions, while in the second solution train 101 will get an additional delay. So, at first it seems like train 103 is disfavoured by the first solution while that solution turns out to be the best one. The initial and updated ETA (Estimated Time of Arrival) for each train is given in Figure 3.

The small example illustrated by Figure 2 and 3 describes how difficult it is to achieve sustainable re-scheduling solutions and to communicate the motivation behind each decision to the operators so that they realise their incentive to co-operate and follow the plans made by the traffic managers. Today, the safe strategy is often to keep the important trains rolling and prioritise them to reduce the risk of them becoming further delayed, but it does not always mean that it is the best solution.

Figure 2. The initial schedule of the railway traffic and the consequential change of arrival times when a signalling problem occurs generating a revised schedule (indicated by the dotted lines) with two conflicts which need to be resolved into a conflict-free revised schedule.

Figure 3. Two alternative solutions to the disturbance problem illustrated in Figure 2.

3 Related Work

The research analysing effects of disturbances in railway traffic and approaches for disturbance management is extensive. Some studies focus on the primary (i.e. initial) reasons behind the disturbances and preventive counter measures, others focus on the approximation of disturbance propagation effects, while some address the topic of this paper and present approaches for scheduling and re-scheduling of a railway traffic timetable. Surveys in [9]-[11] provide an extensive overview of these. More recent re-scheduling approaches are presented by e.g. Zhou and Zhong [12] who propose a branch-and-bound algorithm for railway traffic timetabling in a single-tracked network and D'Ariano et. al. [13,14] who present an iterative algorithm that in real-time resolves conflicts in a perturbed railway traffic timetable. Those approaches focus primarily on re-scheduling the different traffic activities by allocating network resources to the different train slots. Another aspect of the disturbance management problem is the synchronisation of the train services which is a major issue in public transportation; see e.g. Schöbel [15].

Depending on the focus of related approaches, the models used to describe the scheduling and re-scheduling problem differ and have different levels of detail. Radtke [16] and Gely et. al. [17] distinguish between the different levels of detail in the infrastructure models used for railway traffic network representation, where macroscopic models contain least details and have a more aggregated representation of some resources (e.g. a station is composed of a number of parallel tracks with one platform each), while microscopic includes a lot more detail (e.g. a station is composed of a complex set-up of pieces of tracks

separated by switches and signals). When to use which level of detail is not obvious but often decided based on time available for computations and access to input data with sufficient accuracy. The more details included in the model, the more data and computations required and for a real-time scheduling problem there may not be enough time available to collect and process data and to compute the many alternatives that arise as the network is divided into more fine-grained resources. Most re-scheduling approaches do not use a very detailed representation of the station routings as it easily becomes rather complex, see e.g. [18] and [19]. For further information on models and methods for railway traffic scheduling and re-scheduling, we refer to [20] and [21].

In our previous research, primarily outlined in Törnquist and Persson [22], we have designed an optimisation-based approach for re-scheduling which seems promising. However, for some disturbance scenarios and a time horizon longer than 60 minutes it is difficult to find good solutions within seconds. Furthermore, the developed MILP formulation and solution method was not initially intended to consider the explicit routing of trains within stations since for most stations the choice of platforms and the routing is straightforward and the dependencies between different routes are more or less negligible. Meanwhile, our most recent studies have indicated that a few stations in the network have a more complex structure and serve as a junction point for different lines may require a more detailed description. When the traffic is dense, such as during rush hour, these details matter. Hence, we need to extend our previous model which will result in an increased number of constraints and variables and consequently the computation time required to solve the problem. For those two reasons, we have developed a greedy algorithm which effectively delivers good solutions within the permitted time. To quickly retrieve a feasible solution the algorithm performs a depth-first search using an evaluation function to prioritise when conflicts arise and then branches according to a set of criteria.

4 The Greedy Algorithm

As mentioned, the main motivation for developing an algorithm is to ensure that we quickly (within 30 seconds) can receive a good-enough feasible solution independent of type of disturbance scenario. Furthermore, to use a tailored scheduling algorithm, in contrast to formulating the problem as a formal optimisation model and using commercial solution software, is more flexible and dynamic in supporting the implementation of extended modelling requirements as discussed in Section 3. A disadvantage may, on the other hand, be risking not retrieving the most beneficial solutions.

The problem formulation used here is following the same structure as the one presented in [22] which divides the railway network resources into *sections*, where a section is either a *line section* or a *station section*. Each section has one to n parallel tracks. Each line section is a sequence of one or several consecutive blocks although these are not explicitly modelled but indicate that trains running in the same direction can use the tracks of a section simultaneous as long as they are separated by sufficient headway (i.e. not occupying the same block). Each section and track allows bi-directional traffic. See illustration in Figure 4 below.

Figure 4. Our network representation.

We denote a train slot, i.e. when a train is planned to occupy a certain section, as an *event* so that the timetable of a train i is a sequence of consecutive events, i.e. an *event list* K_i . Each event k has in its simplest form the parameters b_k^{initial} and e_k^{initial} to indicate its planned begin and end time, d_k to specify the minimum occupation time and h_k to specify if the train has a stop planned thus forcing the train to not leave a station before its planned end time. For trains which occupy the same track t within a station section j , a minimum of Δ_j time units separation time is required. We have used 30 seconds in our experiments. Trains running in the same direction and occupying the same track of a line section with more than one block are required to be separated by a minimum headway, H , of 3 minutes. Trains running in the opposite direction, or using the same track of a line section with only one block, need to be separated by zero time units, i.e. the first train needs to leave before the other one enters the section. If one line section j is directly followed by another line section, the trains cannot switch tracks unless it is explicitly stated to be permitted (in our scenarios, the infrastructure does not permit changing tracks between two consecutive line sections). In the case of two or more consecutive line sections, the time separation must be larger than zero to ensure that trains do not ‘meet’ between any two consecutive line sections. The separation time is then instead a very small number, e.g. 1 s.

The greedy algorithm iteratively searches for the best of all waiting train events to execute next and builds up a tree of consecutive executed events which become nodes in the tree. Each node holds a cost estimation of the solution so far (a lower bound).

The algorithm has three phases. Phase 1 is the pre-processing phase while phase 2 is a depth-first search to quickly find a feasible and good-enough solution by building up a first complete branch of the tree. During phase 3, the algorithm uses the remaining permitted computation time to improve the existing solution by backtracking to potential nodes (where the cost estimation is decreased) in the existing tree and branches to find better solutions.

In the pre-processing phase, the algorithm first executes the events that were active when the disturbance occurred and allocates a start time and a track as intended. The first waiting event in each train’s event list is then put into a *candidate list* sorted after the max value of the minimum start time of the event and its planned start time. The candidate with the earliest possible start time is then evaluated first. First, the algorithm checks whether there is any available track, if not it continues to evaluate the next candidate in the list. If there is

an available track, a deadlock analysis is performed as well as a conflict of interest analysis. If a possible conflict of interest between two or more events scheduled for that section (not necessarily all listed in the current candidate list) is discovered, the cost for the conflict alternatives is computed. That is, if train A is under evaluation to be allocated a track before train B and possibly making train B delayed, the *net minimum expected cost increase* is computed taking into account relevant buffer times and current delays. If the value is positive, it indicates that train B short term would suffer more than train A would gain and that prioritising train B potentially would be wiser. Consecutive effects on a third train are, however, not considered in the analysis.

When the best candidate in the candidate list has been chosen it is executed (i.e. activated) and allocated a track and a start time in line with the restrictions. The previous active event of the same train (if it has one) is at the same time closed and its allocated resource released. The new active event is also put as a node onto the tree and the parent node keeps in memory that it has had this particular event allocated to the actual track as a child node. A cost estimation of the solution so far is then computed by summing up the minimum expected delay for each train. That is, the delay so far experienced by the train minus any future buffer available. See Figure 5 for an illustration of the work process of the algorithm.

When all events have been executed (and closed) and stored as nodes in the tree, the first complete solution is found and stored with the corresponding cost for it (the sum of the final delay of all trains reaching their destination within the problem definition). The algorithm then uses the remaining computation time to expand the tree by backtracking to a node with a cost estimate which is lower than the value for the best complete solution. The execution of the nodes below the branching node are reversed, the branching node index is saved and the search continues as before until the algorithm reaches a feasible, improved complete solution, or stops and backtracks one node up again if it encounters a complete or incomplete solution with the same or worse value than the best one found. In order to avoid cycling and repeating to do the same move over and over again, any parent node can only have one specific event allocated to a certain track as a child one time. If the same event is allocated a different track, it is considered to be a different child node.

The application of this algorithm has been implemented in Java using a vector to store the tree (i.e. only the active branch is stored along with the best solution found and other necessary data).

5 Simulation Study Set-up

In discussion with Banverket, we have chosen to focus on the Norrköping traffic district in our study (see Figure 6). All tracks are bi-directional and the backbone of the region is double-tracked (the stretch Katrineholm-Åby-Mjölby) while the connecting lines are single-tracked. The stations have between two and 14 tracks apart from Norsholm which only has one. The line sections between the stations Katrineholm and Simonstorp contain seven consecutive blocks each. The stretch between Åby and Norrköping contains two consecutive line sections as does the stretches between Ålberga and Kolmården and Linköping and Linghem. We have also used a more detailed model of the stations Åby, Simonstorp and Strångsjö (see Appendix A) where only the available routes through the stations can be used whereas for all other stations we assume that all incoming and outgoing line section tracks connect with all station tracks.

We have applied the algorithm on three different categories of disturbances:

Category 1 refers to that a train is coming into the traffic management district with a certain delay or that it suffers from a temporary delay at one section within the district. This happens in practise quite frequently with the long-distance trains running through this traffic district.

Category 2 refers to that a train has a ‘permanent’ malfunction resulting in increased running times on all line sections it is planned to occupy. This is also a quite frequent problem.

Category 3 refers to an infrastructure failure causing e.g. a speed reduction on a certain section, which results in increased running times for all trains running through that section. This is a serious problem which often has a significant impact on large parts of the surrounding traffic but it is not as frequent as the other disturbance types.

We have used the traffic for a typical weekday (Thursday the 23rd of April, 2009), which contains the regular amount of passenger and freight traffic and we have induced the disturbances in the afternoon rush hour at dense sections. We have applied the algorithm on a number of scenarios, see Appendix B, with a maximum tolerated computation time of 30 seconds and a time horizon of 90 minutes. In attempt to evaluate the optimality of the algorithm, we have also solved a selection of the scenarios using CPLEX 8.0 minimising the total final delay and using the complete formulation proposed in [22] with some necessary modifications corresponding to the same restrictions as implemented in the algorithm (see Appendix C for the complete model). That is, adding a) headway conditions for the line sections with more than one block, b) conditions forbidding changing tracks between two consecutive line sections and c) routing restrictions in to/out from the stations Strångsjö, Simonstorp and Åby.

Figure 6. The railway (sub-)network used for the scenarios and corresponding experiments.

6 Results

The algorithm finds very fast in all scenarios a first feasible, complete solution but is then not always so effective in branching and finding an improvement. Since it is very time consuming to solve certain scenarios to optimality using the formulation in Appendix C and CPLEX 8.0 and sometimes not even possible to find a feasible solution, we have not yet managed to perform an optimality check for all scenarios. In Table 1 and overview of the simulated scenarios can be seen. The fourth column presents the cost value of the solutions found by the algorithm within 30 s and where the highest value is the solution found first. The sixth column presents two values, where the first is the best solution found by CPLEX (if no value is given, a feasible solution was not found within 24 h). The second value is the maximum of a) the lower bound provided by CPLEX and b) the lower bound found by the algorithm. If CPLEX has found an optimal solution, that is also the lower bound. In the fifth column, the largest possible difference between an optimum and the best solution value provided by the algorithm is presented. This value, however, gives very little information in the cases where CPLEX has found a solution with a large gap (i.e. large difference between its best found solution and its lower bound) or not found a feasible solution.

In an evaluation of the performance of the algorithm and comparison to CPLEX, we can first of all see that in most scenarios CPLEX cannot provide a feasible solution despite a relatively high value on the time limit, i.e. 24 hours. When comparing the solutions, we have found that the algorithm and CPLEX, not surprisingly, has different strategies to find a good solution. Even when they achieve solutions with the same objective value, they have

produced different solutions. The solutions provided by CPLEX contain quite some intermediate delays, i.e. several trains suffer from smaller delays of only a few minutes during their trip but arrive at final destination on time. The strategy used by the algorithm avoids delaying additional trains unless it really pays off and therefore it often delays a smaller number of trains but then delays those more. Which solution that is the best from a practical perspective depends but it shows that formulating a representative objective function is tricky and using a myopic approach has certain advantages.

When it comes to the performance of the algorithm with respect to the different categories, we can see that for the first two categories the algorithm performs fairly well. For category three it was difficult to find good values for comparison, but still we can see that the algorithm has some difficulties finding good-enough solutions, see e.g. scenario 18. This indicates, again, that the search strategy and objective function for such algorithms may need to be tailored to the disturbance scenario to some extent.

Nr	Scenario	# trains/events/binary variables	Found solutions (s)	Max difference (min)	Best solution (s)/LB(s)
1	Long-distance pax train 538, north-bound, delay 12 minutes Linköping- Lingham.	48/560/7304	1489,1175	9,8	-/586
2	Long-distance pax train 538, north-bound, delay 6 minutes Linköping- Lingham.	48/560/7304	751,437	3,5	-/226
3	Pax train 2138, south-bound, delay 12 minutes Katrineholm-Strängsjö.	48/558/7245	1150,781	3,5	-/570
4	Pax train 2138, south-bound, delay 6 minutes Katrineholm-Strängsjö.	48/558/7245	790,421	3,5	-/210
5	Pax train 80866 (north-bound), delayed 12 minutes Linköping- Lingham.	49/575/7521	1014, 930	10,3	-/315
6	Pax train 80866 (north-bound), delayed 6 minutes Linköping- Lingham.	49/575/7521	68, 53	0,9	0/0
7	Pax train 8764 (north-bound), delayed 12 minutes Mjölby-Mantorp.	50/567/7534	568, 499	0,2	-/486
8	Pax train 8764 (north-bound), delayed 6 minutes Mjölby-Mantorp.	50/567/7534	276, 207	1,4	-/126
9	Pax train 539 (south-bound), delayed 12 minutes Katrineholm-Strängsjö.	50/570/7490	869, 800	5,7	-/458
10	Pax train 539 (south-bound), delayed 6 minutes Katrineholm-Strängsjö.	50/570/7490	338, 269	2,9	-/98
11	Pax train 538 w. permanent speed reduction causing 50% increased run times on line sections starting at Linköping- Lingham	48/560/7304	1547,1233	3,5	-/1022
12	Pax train 2138 w. permanent speed reduction causing 50% increased run times on line sections starting at Katrineholm-Strängsjö.	48/558/7245	1049 , 680	3,5	-/469
13	Pax train 80866 w. permanent speed reduction causing 50% increased run times on line sections starting at Linköping- Lingham.	49/575/7521	2329,2245	5,3	2148/1930
14	Pax train 8764 w. permanent speed reduction causing 50% increased run times on line sections starting at Mjölby- Mantorp.	50/567/7534	1617,1455	6,6	1112,5/1062
15	Pax train 539 w. permanent speed reduction causing 50% increased run times on line sections starting at Katrineholm-Strängsjö.	50/570/7490	1728,1659	1,0	1598,5/1598,5
16	Speed reduction for all trains between Strängsjö and Simonstorp (all trains get a runtime of 27 min, cf. 5-10 min planned runtime) starting w. freight train 43533.	46/508/5999	13850	0,0	-/13850
17	Speed reduction for all trains between Åby and Simonstorp (all trains get a runtime of 20 min) starting w. train 2138.	51/570/7646	7786,7767,7765,7746	8,9	-/6910
18	Speed reduction for all trains between Åby and Norrköping (all trains get a runtime of 8 min) starting w. train 2138.	49/565/7364	5457,5125	42,2	4756/2594
19	Speed reduction for all trains between Mjölby and Mantorp (all trains get a runtime of 20 min) starting w. train 8764.	50/567/7534	28883	318,1	-/9795
20	Speed reduction for all trains between Linköping and Lingham (all trains get a runtime of 15 min) starting w. train 538.	48/560/7304	23609,22967,22954	265,9	-/6999

Table 1. A selection of evaluation results.

7 Discussion and Future Work

In this paper we have presented a first outline of a greedy algorithm which in short time delivers good re-scheduling solutions to the railway traffic disturbance management problem. We have performed a first experimental evaluation study and analysis of the performance of the algorithm. The evaluation shows that the algorithm provides satisfactory solutions but there are several possible improvements which can be made; foremost related to its branching strategy. Apart from making the algorithm select potential branching nodes more effectively, we are also working on an implementation and evaluation of the effect of multi-threading. The use of multi-threading means that several branches can be exploited simultaneously by use of parallel computing, see e.g. [23], which would speed up the search for improvements. We are also planning to solve additional scenarios to optimality with CPLEX for extended comparisons.

Furthermore, during discussions with the traffic managers we have seen that even further modelling extensions may need to be considered such as modelling the separation of train movements in to and out from certain stations in more detail and using a more dynamic way of specifying minimum train movement times depending on how trains enter and leave certain stations. How these extension, if relevant, can be incorporated then need to be considered.

8 Acknowledgements

The research presented has been financially supported by Banverket (the Swedish Rail Administration). Furthermore, support and guidance have been provided by several people at Banverket – especially Jörgen Hwargård and Thomas Franzén – and by Prof. Paul Davidsson, Prof. Håkan Grahn and Dr. Jan A. Persson at Blekinge Institute of Technology, Sweden.

9 References

1. European Commission, *European transport policy: time to decide 2010*, White paper, 2001.
2. Jansson, G., Jonsson, O., "En pilotstudie av punktligheten i europeiska godstransporter på järnväg", *TFK Rapport 2003:7*, TFK, 2003, Stockholm, Sweden.
3. UIRR (International Union of combined Road-Rail transport companies), *Punctuality statistics for 1999-2007*, 2008.
<http://www.uirr.com/?action=page&page=52&title=Quality> (last accessed 2009-09-01).
4. Banverket, *Årsredovisning 2004* (in Swedish), 2005, Borlänge, Sweden.
5. Banverket, *Årsredovisning 2005* (in Swedish), 2006, Borlänge, Sweden.
6. Banverket, *Årsredovisning 2006* (in Swedish), 2007, Borlänge, Sweden.
7. Banverket, *Årsredovisning 2007* (in Swedish), 2008, Borlänge, Sweden.
8. Banverket, *Årsredovisning 2008* (in Swedish), 2009, Borlänge, Sweden.
9. Assad, A., "Models for rail transportation", *Transportation Research Part A*, Vol.14A, Elsevier Science Ltd, 1980, p. 205- 220.
10. Cordeau, J-F., Toth, P., Vigo, D., "A survey of optimization models for train routing and scheduling", *Transportation Science*, Vol. 32, No. 4, INFORMS, 1998, p. 380-404.
11. Törnquist, J., "Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms", *Proceedings of ATMOS2005*, Palma de Mallorca, Spain, October 2005, Published within the Dagstuhl Research Online Publication Server (DROPS) <http://drops.dagstuhl.de/portals/ATMOS/>, 2005.
12. Zhou, X., Zhong, M., "Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds", *Transportation Research Part B*, Vol. 41, No. 3, Elsevier Science Ltd, 2007, p. 320-342.
13. D'Ariano, A., Pacciarelli, P., Pranzo, M., "A branch and bound algorithm for scheduling trains on a railway network", *European Journal of Operational Research*, Vol. 32, No. 2, Elsevier Science Ltd, 2007, p. 643-657.
14. D'Ariano, Pranzo, M., "An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances", *Network Spatial Economy* 9, p. 63-84, 2009.
15. Schöbel, A., "Capacity constraints in delay management", *Public Transport*, SpringerLink, 2009.
16. Radtke, A. "Infrastructure modelling", in: Hansen and Pachl (Eds.): *Railway Timetable & Traffic. Analysis, Modelling, Simulation*, Hamburg: Eurailpress, 2008.
17. Gely, L., Dessagne, G., Pesneau, P., Vanderbeck, F., "A multi scalable model based on a connexity graph representation", *Proceedings of Comrail 2008*, Zurich, Switzerland, 2008.
18. Kroon, L., Romeijn, E., Zwaneveld, P., "Routing trains through railway stations: complexity issues", *European Journal of Operational Research*, No. 98, pp. 485-498, 1997.
19. Billionnet, A., "Using integer programming to solve the train-platforming problem", *Transportation Science*, Vol. 37, No. 2, pp. 213-222, 2003.
20. Jacobs, J. (2008). "Rescheduling", in: Hansen and Pachl (Eds.): *Railway Timetable & Traffic. Analysis, Modelling, Simulation*, Hamburg: Eurailpress, 2008.

21. Kroon, L., Huisman, D., Maróti, G., "Optimization models for railway timetabling", in: Hansen and Pachl (Eds.): *Railway Timetable & Traffic. Analysis, Modelling, Simulation*, Hamburg: Eurailpress, 2008.
22. Törnquist, J., Persson, J.A., "N-tracked railway traffic re-scheduling during disturbances", *Transportation Research Part B*, Vol. 41, No. 3, Elsevier Science Ltd, 2007, p. 342-362.
23. Emer, J., Hill, M., Patt, Y., Yi, J., Chiou, D., Sendag, R., "Single-threaded vs. multithreaded: Where should we focus?" *IEEE Micro*, 27(6):14–24, Nov/Dec 2007.

10 Appendix A

Figure 7. Illustration of the station Strångsjö.

Figure 8. Illustration of the station Simonstorp.

Figure 9. Illustration of the station Åby.

11 Appendix B

Figure 10. The timetable for the double-tracked section Katrineholm-Mjölby which was used in the simulation experiments. The traffic on the other sections is not as intensive.

12 Appendix C. Optimization model

Objective function

$$\text{Minimize } \sum_{i \in T} z_{n_i} \quad (12.1)$$

Train restrictions

$$x_k^{end} = x_{k+1}^{begin} \quad i \in T, k \in K_i : k \neq n_i \quad (12.2)$$

$$x_k^{end} \geq x_k^{begin} + d_k \quad k \in E \quad (12.3)$$

$$x_k^{begin} \geq b_k^{initial} \quad k \in E : h_k = 1 \quad (12.4)$$

$$x_k^{begin} = b_k^{static} \quad k \in E : b_k^{static} > 0 \quad (12.5)$$

$$x_k^{end} = e_k^{static} \quad k \in E : e_k^{static} > 0 \quad (12.6)$$

Infrastructure restrictions

$$\sum_{t \in P_j} q_{kt} = 1 \quad j \in B, k \in L_j \quad (12.7)$$

$$\sum_{t \in P_j} t * q_{kt} = r_k^{track} \quad j \in B, k \in L_j : r_k^{fixed} = 1 \text{ (dvs. } d_k^a b_k^{static} > 0) \quad (12.8)$$

$$q_{\hat{k}t} + q_{kt} - 1 \leq \lambda_{k\hat{k}} + \gamma_{k\hat{k}} \quad j \in B, k, \hat{k} \in L_j : k < \hat{k}, t \in P_j \quad (12.9)$$

$$x_{\hat{k}}^{begin} - x_k^{end} \geq \Delta_j \gamma_{k\hat{k}} - M(1 - \gamma_{k\hat{k}}) \quad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& (dir_k \neq dir_{\hat{k}} \parallel nrB_j = 1) \quad (12.10)$$

$$x_k^{begin} - x_{\hat{k}}^{end} \geq \Delta_j \lambda_{k\hat{k}} - M(1 - \lambda_{k\hat{k}}) \quad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& (dir_k \neq dir_{\hat{k}} \parallel nrB_j = 1) \quad (12.11)$$

$$\lambda_{k\hat{k}} + \gamma_{k\hat{k}} \leq 1 \quad j \in B, k, \hat{k} \in L_j : k < \hat{k} \quad (12.12)$$

$$x_k^{begin}, x_k^{end} \geq 0 \quad k \in E \quad (12.13)$$

$$\gamma_{k\hat{k}}, \lambda_{k\hat{k}} \in \{0,1\} \quad j \in B, k, \hat{k} \in L_j : k < \hat{k} \quad (12.14)$$

$$q_{kt} \in \{0,1\} \quad j \in B, k \in L_j, t \in P_j \quad (12.15)$$