Transportation Research Part C 20 (2012) 62–78

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances

Johanna Törnquist Krasemann *

School of Computing, Blekinge Institute of Technology, Box 214, 374 24 Karlshamn, Sweden

A R T I C L E I N F O

Article history:
Received 24 September 2009
Received in revised form 15 December 2010
Accepted 16 December 2010

Keywords:
Railway traffic
Scheduling
Timetabling
Slot allocation
Disturbance management
Optimization

A B S T R A C T

An attractive and sustainable railway traffic system is characterized by having a high security, high accessibility, high energy performance and offering reliable services with sufficient punctuality. At the same time, the network is to be utilized to a large extent in a cost-effective way. This requires a continuous balance between maintaining a high utilization and sufficiently high robustness to minimize the sensitivity to disturbances. The occurrence of some disturbances can be prevented to some extent but the occurrence of unpredictable events are unavoidable and their consequences then need to be analyzed, minimized and communicated to the affected users. Valuable information necessary to perform a complete consequence analysis of a disturbance and the re-scheduling is however not always available for the traffic managers. With current conditions, it is also not always possible for the traffic managers to take this information into account since he or she needs to act fast without any decision-support assisting in computing an effective re-scheduling solution. In previous research we have designed an optimization-based approach for re-scheduling which seems promising. However, for certain scenarios it is difficult to find good solutions within seconds. Therefore, we have developed a greedy algorithm which effectively delivers good solutions within the permitted time as a complement to the previous approach. 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.

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. (European Commission, 2001). 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 (Jansson and Jonsson, 2003). 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, 34% of the entire Swedish railway network is considered saturated (capacity utility of 81–100%) with a low average speed and high sensitivity to disturbances (Banverket, 2009). 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

* Tel.: +46 702 667683; fax: +46 457 271 25.
E-mail address: Johanna.Tornquist@bth.se

0968-090X/$ - see front matter
doi:10.1016/j.trc.2010.12.004

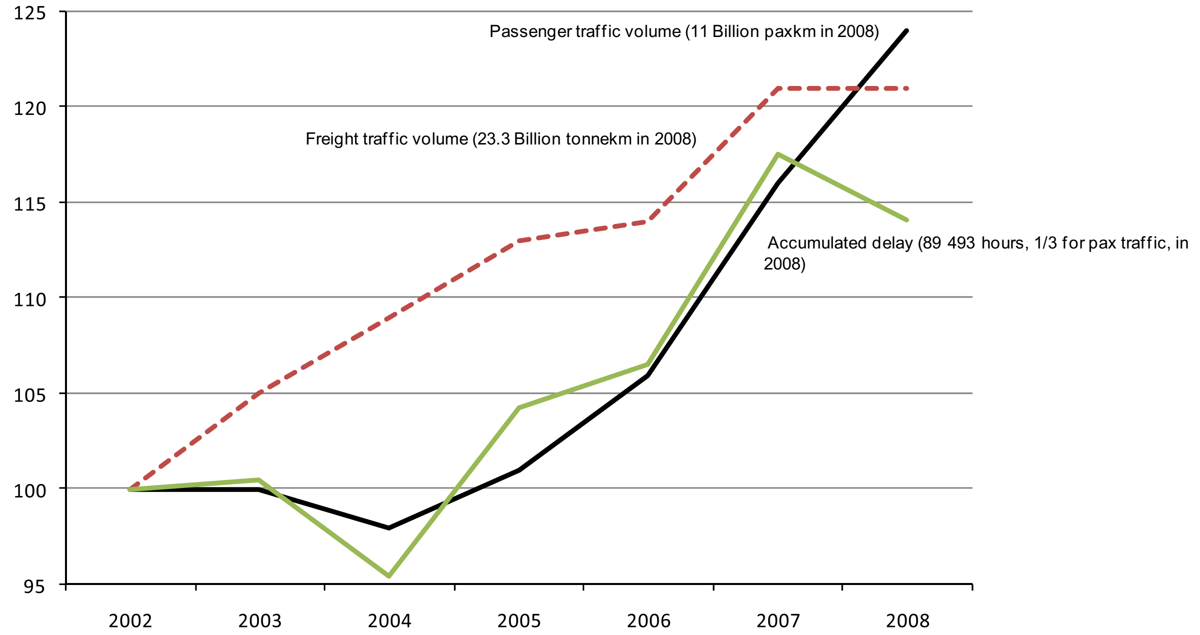

Fig. 1. The growth of the Swedish railway traffic volumes and accumulated delay between year 2002 and 2008 (2002 has index 100). *Source:* (Banverket, 2005–2009).

transport companies) also show that 41% of the cargo trains in the analysed major freight transport corridors arrived 30 min later than scheduled, or more (UIRR, 2008) (see Fig. 1).

The reduction of accumulated delay during 2008 is composed of a delay reduction for the freight traffic from 65,829 h in 2007 to 61,181 h while the delay instead increased for the passenger traffic from 26,377 h to 28,312 h. The corresponding numbers for 2009 are 57,601 and 31,002 h of delay (Banverket, 2009). During 2008 and 2009, several measures have been taken to prevent disturbances from occurring in the first place rather than improving the strategies to limit knock-on delays.

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 Trafikverket (formerly named 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.

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 causes the 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, analyze the consequences of the disturbance as well as the effects of their decision-making. This limitation hampers the possibilities to achieve 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 robust and sufficiently good solutions within reasonable time. For illustrative purposes, let us consider the small-scale example of re-scheduling the traffic in Fig. 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

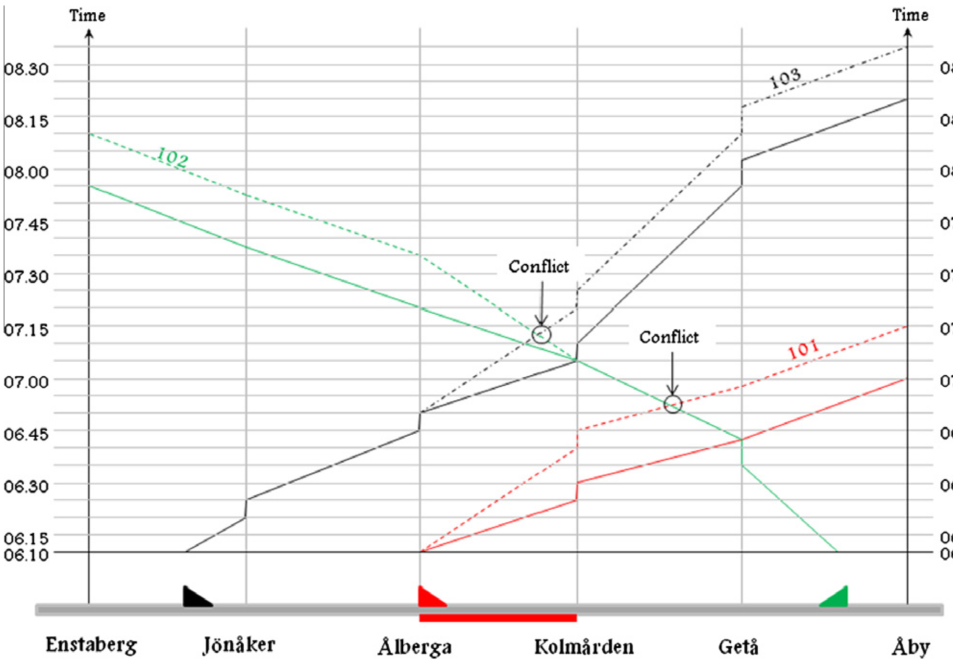

Fig. 2. The initial schedule of the railway traffic and the consequential change of arrival times when a signaling problem occurs generating a revised schedule (indicated by the dotted lines) with two conflicts which need to be resolved into a conflict-free revised schedule.

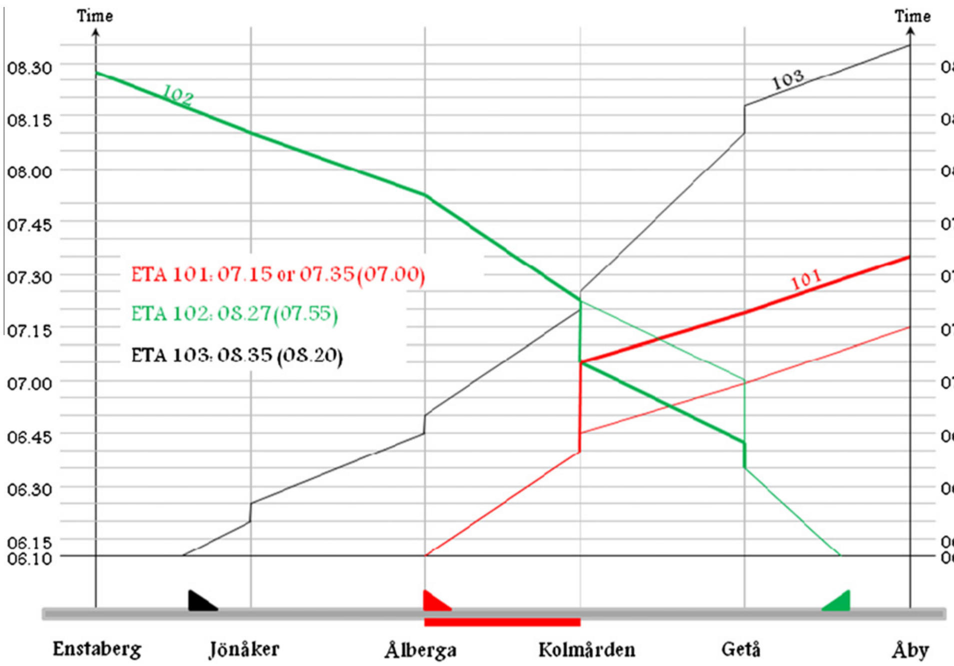

Fig. 3. Two alternative solutions to the disturbance problem illustrated in Fig. 2.

15 min and the revised timetable for each train (the dotted lines) is thus postponed 15 min 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 be 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 Fig. 3, two solutions are illustrated. In the first solution, train 102 and 101 meet at Getå (as in the initial timetable) but that forces train 102 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 in favour of a cargo train. Instead, the operator wants the traffic manager to arrange for the trains to meet in Kolmården, thereby reducing the waiting time for 102 at Getå significantly. When the traffic manager considers that alternative he discovers that in the end, the waiting time for train 102 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 102 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 Fig. 3.

The small example illustrated by Figs. 2 and 3 describes how difficult it is to achieve robust re-scheduling solutions and to communicate the motivation behind each decision to the operators so that they understand the benefits of co-operating and following the plans made by the traffic managers. Today, the safe strategy is often to keep the important trains rolling and prioritize them to reduce the risk of them becoming further delayed, but it does not always mean that it is the best solution.

3. Related work

The research analyzing 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, i.e. approaches for scheduling and re-scheduling railway traffic. Surveys by Assad (1980), Cordeau et al. (1998) and Törnquist (2005) provide an extensive overview of these. More recent re-scheduling approaches are presented by Zhou and Zhong (2007), who propose a branch-and-bound algorithm for railway traffic timetabling in a single-tracked network, and D'Ariano et al. (2007, 2009), who present an iterative algorithm that in real-time resolves conflicts in a pertubated railway traffic timetable. Since the approach suggested by Zhou and Zhong (2007) is developed for a single-tracked network it is not applicable for the problem setting we are facing in Sweden where certain lines have up to four parallel and bi-directional tracks. Furthermore, the approach suggested by D'Ariano et al. is applied to a network where the tracks on the double-tracked lines are devoted to traffic in one special direction (i.e. one track in each direction), while the Swedish double-tracked lines can be used by two trains in the same direction and run in parallel. If and how their approach could be extended to fit the setting this paper deals with, is not clarified in their paper. Another important aspect of the disturbance management problem, which is not explicitly addressed by the previous approaches, is the synchronization of the train services which is a major issue in public transportation; see e.g. Schöbel (2009).

Depending on the focus, the models used to describe the scheduling and re-scheduling problem differ and have different levels of detail. Radtke (2008) and Gely et al. (2008) 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. Kroon et al. (1997) and Billionnet (2003)). For further information on models and methods for railway traffic scheduling and re-scheduling, we refer to Jacobs (2008) and Kroon et al. (2008).

In our previous research, primarily outlined in Törnquist and Persson (2007) and Törnquist (2007), we have designed an optimization-based approach for re-scheduling which seems promising. However, for some disturbance scenarios and a time horizon longer than 60 min it is difficult to find good solutions within seconds. Consequently, the MILP needs to be complemented by a mechanism that independent of disturbance scenario provides a feasible, good-enough solution fast.

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. However, our most recent analysis of the infrastructure and train dependencies has indicated that it may be necessary to include more detailed characteristics of the infrastructure in the problem formulation. For example, some stations in the network have a more complex structure because they serve as a junction point for different lines (e.g. Åby). Furthermore, some stations (e.g. Strångsjö) have a more complicated safety system which requires different minimum separation times (from zero to 90 s) between the trains, depending on the sequence of incoming and outgoing trains and which path they take through the station. When the traffic is dense, such as during rush hour, these details have a significant impact even if it may not be clear yet exactly how. 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 as a complement to the previous approach. To quickly retrieve a feasible solution the algorithm performs a depth-first search using an evaluation function to prioritize 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 s) 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 optimization 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 Törnquist and Persson (2007), 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 simultaneously 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 Fig. 4 below.

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 s 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 min. 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 in opposite direction do not 'meet' between any two consecutive line sections. The separation time is then instead a very small number, e.g. 1 second. See constraint (D.12)–(D.20) in Appendix D. The constraints are only relevant to consider when two trains in opposite direction interact at any of these line sections.

The greedy algorithm iteratively searches for the best of all waiting train events to execute next and builds up a tree, referred to as **DT**, of consecutive active or terminated (the event has ended and the train has left the assigned track) events which become nodes in the tree. Each node holds an estimation of the disturbance consequences for the solution so far (a lower bound, LB).

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. See Appendix A and Figs. 5a, 5b and 5c) for an outline of the algorithm.

In Phase 1, the algorithm first activates the events that were already started when the disturbance occurred and it allocates a start time and a track as initially intended. These events correspond to the green nodes in Fig. 5a) below.

The first waiting event in each train's event list is then put into a *candidate list*, ***NC***, sorted after the maximum value of the minimum starting time of each event $k, b_k^{\min}$ and its planned starting time, $b_k^{initial}$. d_k refers to the minimum running time for event k.

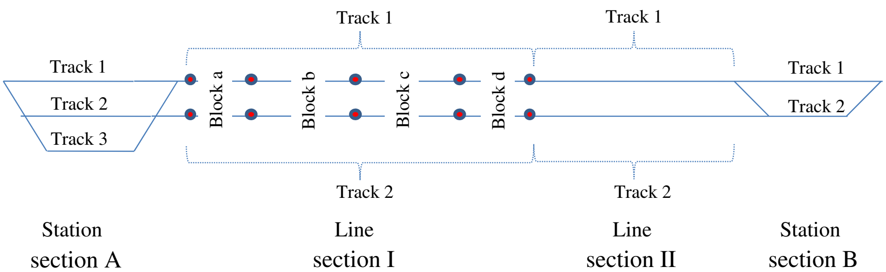

Fig. 4. The network model.

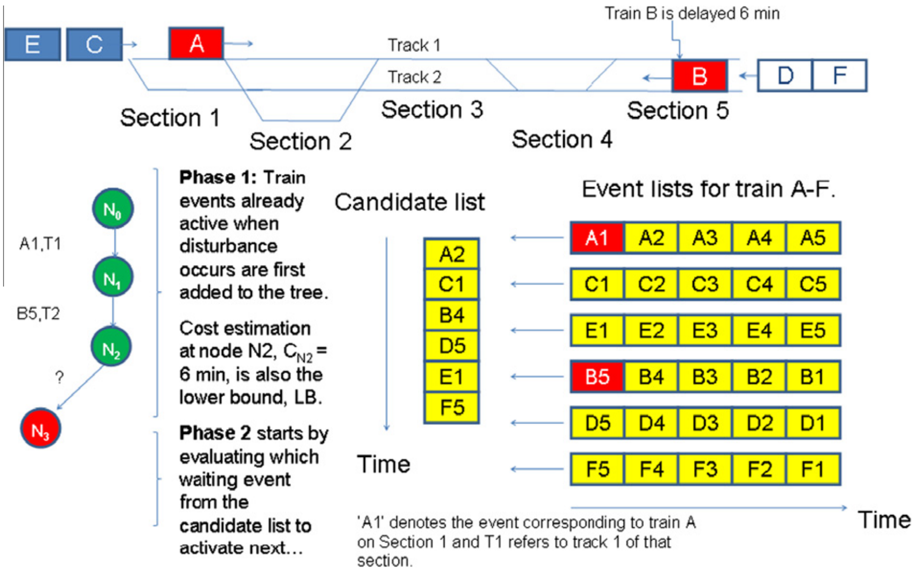

Fig. 5a. Illustration of the algorithmic procedure in Phase 1 and the start-up of Phase 2. The sequence of three green circles (i.e. nodes N_0 to N_2) and the first red circle N_3 represents the start of the decision tree (DT). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

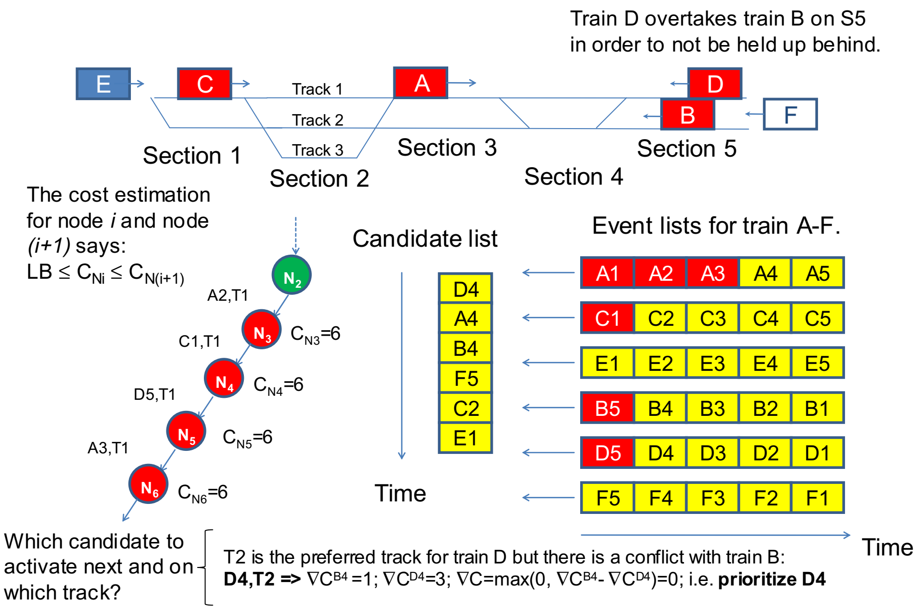

Fig. 5b. Illustration of Phase 2. A conflict of interest between foremost event D4 and B4 is discovered and the consequences of the two alternatives are computed and it shows that D4 should have priority.

In each iteration to select the next event in the candidate list to activate, the candidate with the earliest possible start time, $b_k^{\min}$, is then evaluated first (i.e. event A2 in Fig. 5a). In the evaluation process, the algorithm first checks whether there is any available track, and if the optimal track, t' is free – from the perspective of the event – it is selected. For example, let us assume that train A has a planned stop at track 1 on Section 2 to let passengers off and consequently this track is selected if possible. However, if necessary train A may go to track 2 or 3 just as well.

If the analysis shows that there are tracks possible to use but they are all occupied, the train will have to wait until next round and the algorithm continues to evaluate the next candidate in the list.

If there are no tracks available for other reasons, the train which the event belongs to is considered being deadlocked. That is, if the tracks are;

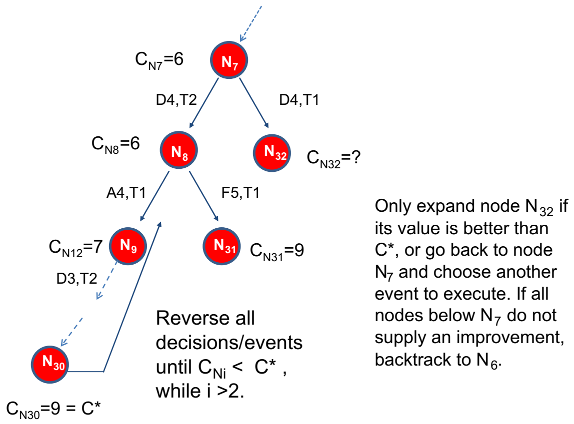

Fig. 5c. Illustration of Phase 3.

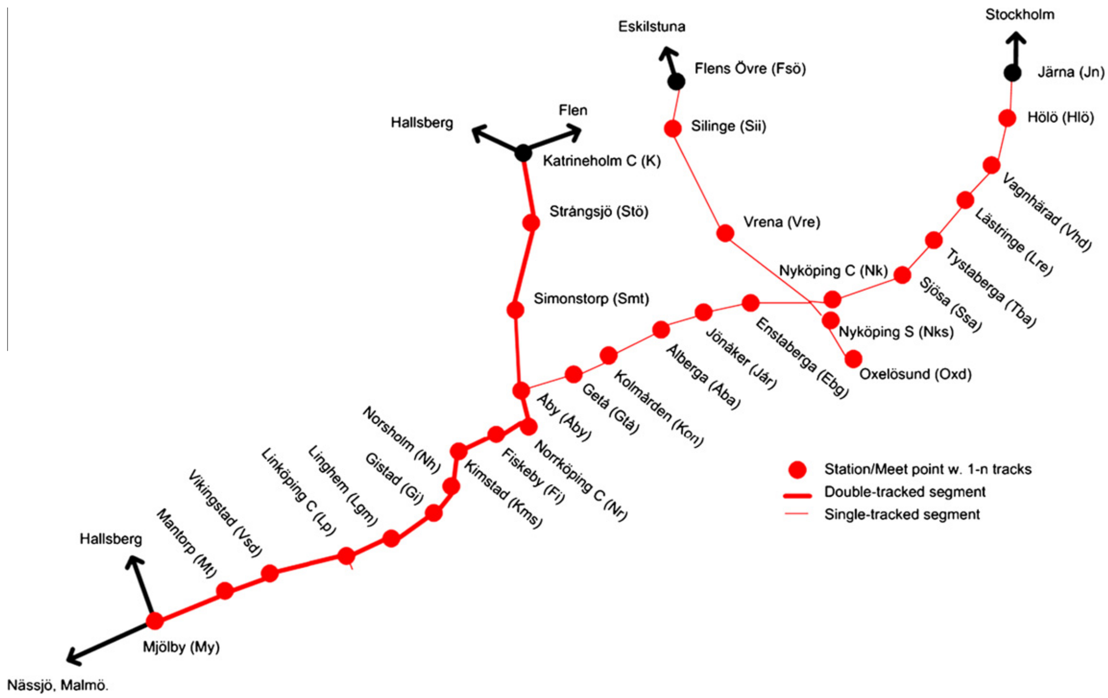

Fig. 6. The railway (sub-)network used for the scenarios and corresponding experiments. Between Ålberga and Kolmården as well as between Åby and Norrköping, there are two consecutive line sections and with no interconnection between the parallel tracks on the double-tracked sections. The fast passenger trains to/from Stockholm run via Katrineholm and Flen and the slower passenger trains run via Nyköping.

(i) physically impossible to reach, or
(ii) blocked, or
(iii) already used in a neighbouring child node (only relevant in Phase 3).

The search for next event to be activated is then interrupted in favour of backtracking to resolve the deadlock.

If there is an available track selected, a deadlock analysis is performed as well as a conflict of interest analysis. The deadlock analysis checks;

(i) whether there are any additional unoccupied tracks on the same section and if not,
(ii) whether the train has any way out of its potential new location if assigned the selected track. The analysis distinguishes between where the other trains are going in relation to that specific train. That is, a track occupied by a train running in the same direction is considered a possible way out.

If the investigated candidate event, k, is scheduled at line section j and this section has only one track which is functioning (i.e. it can be a single-tracked line section or a double-tracked line section where one of the tracks is blocked) the occurrence of possible conflicts of interest is investigated. This is done by checking if there are any events k' also scheduled at section j and who has an earliest possible start time which is smaller than the predicted termination time for event k. That is; $b_{k'}^{\min} \geqslant b_k^{\min}$ and $b_{k'}^{\min} < \left(d_k + b_k^{\min}\right)$

If a possible conflict of interest between two or more events is discovered, the cost (i.e. the consequential additional delay) for the conflict alternatives is computed. That is, let train D in Fig. 5b) be the current evaluated candidate event. If train D is to be allocated a track before train B, it may mean that train B becomes delayed and the *net minimum expected cost increase*, ∇C, 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 D would gain and that prioritizing train B potentially would be wiser. Below is an example:

∇C^k is the predicted resulting delay for event k (event D4 in the example) if it has to wait in favour of event k', while $\nabla C^{k'}$ is the reversed and takes into account that event k' may be delayed already. We know that $b_k^{\min} \leqslant b_{k'}^{\min}$, and if $\left(b_k^{\min} + d_k\right) \leqslant b_{k'}^{initial}$ we let

$\nabla C = 0$. If not, we compute:

$$\nabla C^k = \max(0, (b_{k'}^{\min} + d_{k'})) - \max\left(b_k^{\min}, b_k^{initial}\right)$$
$$\nabla C^{k'} = \max\left(0, \left(b_k^{\min} + d_k - b_{k'}^{initial}\right)\right) - \max\left(0, \left(b_{k'}^{\min} - b_{k'}^{initial}\right)\right)$$
$$\nabla C = \max(0, (\nabla C^{k'} - \nabla C^k))$$

In the example, event D4 is able to start before B4 and would become at least 3 min delayed if B4 was to be prioritized, while the reversed situation would cause B4 a delay of only one minute. Therefore, D4 is prioritized. 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 *activated* and allocated a track and a start time in line with the traffic conditions. The previous active event of the same train (if it has one) is at the same time closed (i.e. *terminated*) and its allocated resource hence released. That is, when D4 is activated, D5 is terminated and the corresponding track is denoted available.

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 each train minus any future buffer available.

When all events have been terminated 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 enters Phase 3 and 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. See example in Fig. 5c). Note that the backtracking can never go further back than to the last node of the ones activated during Phase 1 (i.e. the green nodes N_0–N_2 can not be revoked). 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.

In the deadlock analysis, the algorithm backtracks and reverses the most recent event of the deadlocked train. If this also leads to that this train becomes deadlocked, the algorithm backtracks one node higher than the previous backtracking.

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 Trafikverket, we have chosen to focus on the Norrköping traffic district in our study (see Fig. 6). The sub-network is composed of 28 stations and 15 double-tracked sections and 17 single-tracked sections. All tracks are bidirectional 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 B) 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 practice 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 C and Table 1, with a maximum tolerated computation time of 30 s and a time horizon of 90 min. One reason for choosing a time horizon as long as 90 min, is that the time to run the main double-tracked stretch takes about 75 min for a faster passenger train and for certain disturbances of e.g. 20 min for a single delayed train, the delayed train would interact with the relevant trains at the end of its trip instead of the tracks being totally empty.

In attempt to evaluate the optimality of the algorithm, we have also solved the scenarios using solver Cplex version 10 minimizing the total final delay and using the complete formulation proposed in Törnquist and Persson (2007) with some necessary modifications corresponding to the same restrictions as implemented in the algorithm (see Appendix D for the complete model). That is, adding (a) headway conditions for the line sections with more than one block and trains running in the same direction, (b) conditions forbidding changing tracks between two consecutive line sections and (c) routing restrictions in to/out from the stations Strångsjö, Simonstorp and Åby.

Table 1
A selection of evaluation results for a time horizon of 90 min. In the fifth column the maximum difference between the solution values provided by the algorithm and Cplex is presented (since the optimum solution could not be found within 24 h). In some cases is also the minimum {minimum;maximum} value given, when a feasible solution has been found.

Nr	Scenario	# trains/events/ binary variables	Found solutions (min)	Max difference (min)	Optimum (min)/LB (min)
1	Long-distance pax train 538, north-bound, delay 12 min Linköping–Linghem	48/547/8148	24.82; 19.58	9.8	–/9.77
2	Long-distance pax train 538, north-bound, delay 6 min Linköping–Linghem	48/547/8148	12.52; 7.28	3.5	–/3.77
3	Paxtrain 2138, south-bound, delay 12 min Katrineholm–Strångsjö	48/551/8260	19.17; 13.02	3.5	–/9.5
4	Paxtrain 2138, south-bound, delay 6 min Katrineholm–Strångsjö	48/551/8260	13.17; 7.02	3.5	–/3.5
5	Paxtrain 80866 (north-bound), delayed 12 min Linköping–Linghem	49/562/8369	16.9; 15.50	10.3	–/5.25
6	Paxtrain 80866 (north-bound), delayed 6 min Linköping–Linghem	49/562/8369	l.13; 0.88	0.9	0/0
7	Paxtrain 8764 (north-bound), delayed 12 min Mjölby–Mantorp	50/553/8317	9.47; 8.32	0.2	–/8.1
8	Paxtrain 8764 (north-bound), delayed 6 min Mjölby–Mantorp	50/553/8317	4.6; 3.45	1.4	–/2.1
9	Paxtrain 539 (south-bound), delayed 12 min Katrineholm–Strångsjö	50/555/8295	14.48; 13.33	5.7	–/7.63
10	Paxtrain 539 (south-bound), delayed 6 min Katrineholm–Strångsjö	50/555/8295	5.63; 4.48	2.9	–/1.63
11	Paxtrain 538 w. permanent speed reduction causing 50% increased run times on line sections starting at Linköping–Linghem	48/547/8148	25.78; 29.55	3.5	–/17.03
12	Paxtrain 2138 w. permanent speed reduction causing 50% increased run times on line sections starting at Katrineholm–Strångsjö	48/551/8260	17.48; 11.33	3.5	–/7.82
13	Paxtrain 80866 w. permanent speed reduction causing 50% increased run times on line sections starting at Linköping–Linghem	48/563/8321	38.82; 37.42	{1.6; 5.3}	35.8/32.17
14	Paxtrain 8764 w. permanent speed reduction causing 50% increased run times on line sections starting at Mjölby–Mantorp	50/553/8317	26.95; 24.25	{5.7; 6.6}	18.54/17.7
15	Paxtrain 539 w. permanent speed reduction causing 50% increased run times on line sections starting at Katrineholm–Strångsjö	50/555/8295	28.8; 27.65	1.0	26.64/26.64
16	Speed reduction for all trains between Strångsjö and Simonstorp (all trains get a runtime of 27 min, cf. 5–10 min planned runtime) starting w. freight train 43533	46/507/7001	230.83	0.0	–/230.83
17	Speed reduction for all trains between Åby and Simonstorp (all trains get a runtime of 20 min) starting w. train 2138	51/556/8442	129.77; 129.45; 129.42; 129.1	8.9	–/115.17
18	Speed reduction for all trains between Åby and Norrkoping (all trains get a runtime of 8 min) starting w. train 2138	49/551/8154	90.95; 85.42	{6.2; 42.2}	79.27/43.23
19	Speed reduction for all trains between Mjölby and Mantorp (all trains get a runtime of 20 min) starting w. train 8764	50/553/8317	481.38	318.0	–/163.25
20	Speed reduction for all trains between Linköping and Linghem (all trains get a runtime of 15 min) starting w. train 538	48/547/8148	393.48; 382.78; 382.57	266.0	–/116.65

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 and memory consuming to solve certain scenarios using the formulation in Appendix D and the solver Cplex version 10, and sometimes not even possible to find a feasible solution, it has not been possible to perform an optimality check for all scenarios. In Table 1, an overview of the simulated scenarios can be seen. The second column presents the scenarios which also can be seen in the graph in Appendix C. The third column presents information on the size of the scenarios and the number of binary variables is computed below where m_j refers to the number of events on section j that remains to be scheduled. The first term refers to the variables $\gamma_{k\hat{k}}$ while the second term refers to $\lambda_{k\hat{k}}$ and the last, q_{kt} (see notation in Appendix D). The last two only concerns non-single-tracked sections.

$$\sum_{j=1}^{|B|} m_j(m_j-1)/2 + \sum_{j=1:|P_j|>1}^{|B|} m_j(m_j-1)/2 + \sum_{j=1:|P_j|>1}^{|B|} m_j * |P_j| : |P_j| > 1 \tag{6.1}$$

The fourth column presents the cost value of all the solutions found by the algorithm within 30 s and where the highest value refers to the solution found first. It takes less than 1 s to find the first solution for the algorithm and very often it also finds the other solutions (if there are any) within the first few seconds.

The sixth column presents two values, where the first is the best solution found by the solver (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 the solver and (b) the lower bound found by the algorithm. If the solver 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 the solver 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 a few scenarios, there is also a minimum difference given, i.e. {minimum, maximum} since Cplex has found a feasible solution which is better than the one provided by the algorithm.

Table 2
A selection of evaluation results for a time horizon of 60 min. * Refers to that an optimum could not be found within 24 h by Cplex and hence the lower bound (LB) is given as well as the minimum and maximum difference between the solution values provided by the algorithm and Cplex.

Nr	Scenario	# trains/ events/binary variables	Found solutions (min)	Difference (min)	Optimum (min)
1	Long-distance pax train 538, north-bound, delay 12 min Linköping–Linghem	36/356/3795	16.33	2.1	14.23
2	Long-distance pax train 538, north-bound, delay 6 min Linköping–Linghem	36/356/3795	3.77	0.0	3.77
3	Paxtrain 2138, south-bound, delay 12 min Katrineholm–Strångsjö	34/357/3842	9.50	0.0	9.50
4	Paxtrain 2138, south-bound, delay 6 min Katrineholm–Strångsjö	34/357/3842	3.50	0.0	3.50
5	Paxtrain 80866 (north-bound), delayed 12 min Linköping–Linghem	37/364/3841	19.58	4.1	15.47
6	Paxtrain 80866 (north-bound), delayed 6 min Linköping–Linghem	37/364/3841	4.63	4.1	0.50
7	Paxtrain 8764 (north-bound), delayed 12 min Mjölby–Mantorp	39/357/3813	8.10	0.0	8.10
8	Paxtrain 8764 (north-bound), delayed 6 min Mjölby–Mantorp	39/357/3813	3.23; 2.98	{0.3; 0.9}	**2.93***(**2.1**)
9	Paxtrain 539 (south-bound), delayed 12 min Katrineholm–Strångsjö	41/365/3921	14.92	3.8	11.13
10	Paxtrain 539 (south-bound), delayed 6 min Katrineholm–Strångsjö	41/365/3921	4.27	0.0	4.27
11	Paxtrain 538 w. permanent speed reduction causing 50% increased run times on line sections starting at Linköping–Linghem	36/356/3795	17.03	0.0	17.03
12	Paxtrain 2138 w. permanent speed reduction causing 50% increased run times on line sections starting at Katrineholm–Strångsjö	34/357/3842	7.82	0.0	7.82
13	Paxtrain 80866 w. permanent speed reduction causing 50% increased run times on line sections starting at Linköping–Linghem	38/369/3858	32.08	4.1	27.98
14	Paxtrain 8764 w. permanent speed reduction causing 50% increased run times on line sections starting at Mjölby–Mantorp	39/357/3813	27.28	7.0	20.26
15	Paxtrain 539 w. permanent speed reduction causing 50% increased run times on line sections starting at Katrineholm–Strångsjö	41/365/3921	27.43	0.8	26.64
16	Speed reduction for all trains between Strångsjö and Simonstorp (all trains get a runtime of 27 min, cf. 5–10 min planned runtime) starting w. freight train 43533	34/321/3102	156.80	0.0	156.80
17	Speed reduction for all trains between Åby and Simonstorp (all trains get a runtime of 20 min) starting w. train 2138	40/363/3976	93.02	12.7	80.32
18	Speed reduction for all trains between Åby and Norrkoping (all trains get a runtime of 8 min) starting w. train 2138	39/364/3854	40.87	2.8	38.12
19	Speed reduction for all trains between Mjölby and Mantorp (all trains get a runtime of 20 min) starting w. train 8764	39/357/3813	278.87	{0; 126.2}	**2 78.8 7***(**152.7**)
20	Speed reduction for all trains between Linköping and Linghem (all trains get a runtime of 15 min) starting w. train 538	36/356/3795	155.40	{5.13; 34.8}	**136.8***(**120.62**)

Since it is so difficult to evaluate the level of optimality of the algorithm without optimal solutions or better lower bounds, we have also simulated the same disturbance scenarios but with a shorter time horizon of only 60 min. The results can be seen in Table 2.

In an evaluation of the performance of the algorithm and comparison to the solver, we can first of all see that in most scenarios the solver cannot provide a feasible solution despite a relatively high value on the time limit, i.e. 24 h. When comparing the solutions, we have found that the algorithm and the solver, not surprisingly, have 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 the solver 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. A visual analysis of some of the solutions provided by the algorithm showed that the solutions seem sound and no obvious, large improvement could be seen. Note also that it is important to understand, from a practical point of view, that the running times and separation times between trains are approximations of what they actually will be in real-time, so that if the solutions differ with a few minutes this is acceptable.

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 *may* have some difficulties finding good-enough solutions, see e.g. scenario 19 and 20. However, as described earlier the maximum difference presented in the fifth column in Tables 1 and 2 is really only relevant if there is a feasible solution found by Cplex (i.e. the sixth column contains two values). In the case of scenario 16–17 and 19–20 in Table 1, Cplex can not provide a feasible solution within the time limit and thus only provides a lower bound of the LP relaxation of the problem. The lower bound means that there may exist a solution which is as good as 163.25 (for scenario 19), but probably not in this case. How the disturbances are affecting the trains with respect to e.g. increased running times, can be seen in Appendix C. For scenario 19, we can see that the average running time between Mjölby and Mantorp in the timetable is approximately 5 min and because of the disturbance it is increased to 20 min for all trains passing through during those 90 min (approximately 13 trains). Consequently, each of these trains gets a delay of approximately minimum 15 min and the sum of these delays reaches 195 min and does not include any of the knock-on delays (cf. the lower bound of 163.25).

Since the algorithm focuses on preventing deadlocks from occurring (but is also able to unlock deadlocks) it has not created a deadlock before finding its first solution in our scenarios (i.e. within Phase 2), but later during Phase 3. The reason is that the algorithm is quite conservative when it searches for the first solution, which probably results in that the optimal solution is sometimes not found. That is, in Phase 2 it rather avoids making a decision which may lead to a deadlock than to take the chance and select a candidate event which may generate a lower delay cost estimation.

The ability of the algorithm to handle deadlocks at stations where a line is splitting up, e.g. Åby, could be improved. The reason why deadlocks are more likely to occur for this type of station is that the appropriate track needs to be selected earlier in time than for other, simpler stations.

7. Discussion and future work

In this paper we have presented a greedy algorithm which in short time delivers good re-scheduling solutions to the railway traffic disturbance management problem. We have performed an experimental evaluation and analysis of the performance of the algorithm. The evaluation shows that the algorithm provides good-enough solutions very fast but there are also several possible improvements which can be made and foremost related to its branching strategy.

Apart from making the algorithm select potential branching nodes more effectively and minimizing the amount of computations, 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 explored simultaneously by use of parallel computing, see e.g. (Emer et al., 2007), which would speed up the search for improvements.

Furthermore, during discussions with the traffic managers we have seen that even further modelling extensions may need to be considered. This includes e.g. modelling the separation of train movements in to and out from certain stations in more detail as well as 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 in more depth.

Finally, the main purpose of this algorithm is to act as a complement to the earlier optimization approach presented in Törnquist and Persson (2007) since for some scenarios and time horizons the previous approach was failing to quickly enough provide a feasible solution. For some of the scenarios considered in this paper, we have also applied the previous approach as well as Strategies 1 and 2 presented in Törnquist and Persson (2007) with the use of Cplex. Even Strategy 1, which simplifies and restricts the solution space significantly, failed to solve some of the scenarios within the time limit of 30 s. What could be interesting to investigate in future work is the possibility and effect of giving the initial solution provided by the greedy heuristic as a start solution to Cplex using our previous approach (if the starting solution is feasible with the restricted formulation) or the full model presented in Appendix D. This could potentially lead to improved solutions or at least generate better estimations of the lower bounds.

Acknowledgements

The research presented has been financially supported by Trafikverket (the Swedish Transport Administration). Furthermore, support and guidance have been provided by several people at Trafikverket – especially Jörgen Hwargård, Thomas Franzén, Hans Dahlberg and Tommy Jonsson – and by Prof. Paul Davidsson, Prof. Håkan Grahn and Dr. Jan A. Persson at Blekinge Institute of Technology, Sweden. Finally, I would like to thank the anonymous reviewers for their effort and significant comments on how to improve this paper.

Appendix A. Outline of the greedy algorithm

A.1. Notation

Overall, the same notation as in Törnquist and Persson (2007) is used.

Let T_0 be the time when the disturbance was inflicted. Let **DT** be the decision tree vector which stores the current branch, i.e. sequence, of active and terminated events and all those events are represented as nodes in **DT**. $\mathbf{DT}^*$ is the vector which stores the best complete solution found so far. s is used to denote the index of node N_s in **DT** and $s = |\mathrm{DT}|$ gives the index of the last activated node (i.e. event). **DT** starts with the nodes corresponding to all the events that were active at T_0 and s^* represents the index of the last of those nodes in **DT**. *LB* refers to the lower bound and C_s refers to the *minimum delay cost* to be expected after the previous events generating nodes N_1–N_s in **DT** have been activated (and possibly terminated). Consequently C_{s^*} provides the LB. Let C^* denote the best solution value with default value -1. Let $\mathrm{DT}_{\max}$ be the max size of **DT**, i.e. the total number of events including the prestarted ones. Let $s_{\mathrm{backtrack}}$ denotes the index of the height in **DT** to which the last backtracking was carried out and $s_{\mathrm{backtrack}} \geqslant s^*$.

Let $Stat_i = \{W = Waiting,\ A = Active,\ T = Terminated\}$ specify the actual status of train i at a certain point while $E_i = \{0; |\mathbf{K}_i| + 1\}$ denote the index of its current active event. $E_i = 0$ refers to $Stat_i = W$ and $E_i = |\mathbf{K}_i| + 1$ refers to $Stat_i = E$. Let $b_k^{\min}$ denote the earliest possible start time of event k. Define **NC** (the candidate list) as the set of the first unstarted event k of each train i. **NC** is also by definition sorted in chronological order w.r.t. $b_k^{\min}$. Let k^* denote the index of the current most high-ranked candidate event found in **NC**, while k^{**} denotes the second most high-ranked event. t^* denotes the chosen track for k^*, and t^{**} the track for $k^{**} \cdot i^*$ denotes the index of an identified deadlocked train.

Let $\boldsymbol{L}'_j$ be the list of the remaining non-active/-terminated events of section j, where $k \in \boldsymbol{L}_j$ and it is by definition sorted in chronological order w.r.t. $b_k^{\min}$. Let P'_k be a subset of tracks $P_{j,}$, where P'_k only includes tracks which event k is permitted to use (i.e. the event may already have been assigned track t before as a child node from the last active node in **DT**). Let nrT_j represent the actual number of unoccupied tracks at section j excluding any temporarily blocked tracks so that $nrT_j = \{0, 1, \ldots, |P_j|\}$. S_k specifies if event k takes place at a line section ($S_k = 1$) or a station section ($S_k = 0$).

Let $\mathbf{TL}_{j,t}$ be the vector of track t on section j which lists all the executed and active events which have been assigned track t. Let $T_{j,t}^{\min}$ denote the earliest time track t is available for a new event to be activated. The event with index $s = |\mathbf{TL}_{j,t}|$ is the last one which entered the track. Let $e_k^{\min}$ denote the earliest time the active event k is permitted to leave its assigned track. If the corresponding section only contains one block, $e_k^{\min}$ is dependent on the minimum running time and the earliest time the corresponding train can activate its next event. However, on sections with more than one block and where two trains run in the same direction, they can use the same track but separated by some minimum headway (here 3 min) and this needs to be considered both when entering and leaving the track. $e_k^{\min}$ may need to be associated with the closure of another event.

Let *deadlockRisk* be a Boolean variable (takes on the values true/false) specifying if the deadlock analysis resulted in an identified risk of running into a deadlock if the investigated candidate event k is assigned the investigated track t'.

Let t' denote the 'optimal' track at section j selected for the candidate k in focus so that $t' = \{-1, 0, 1, \ldots |P'_k|\}$ which means that if t' is negative there are no tracks possible to choose from and thus it refers to a deadlock. The value '0' refers to that all the tracks which event k would be permitted to use are occupied and the first one becomes available at time $T_j^{\min}$. $T_{j,t}^{\min}$ specifies the earliest time event k may enter the specific track t which becomes relevant when $t' > 0$.

Let *possibleConflictOfInterest* be a Boolean variable specifying if there are any trains which may compete with train i over a track. That is, if there are any trains which would want to start before train i could leave the requested track and no other track would be available either before the other train(s) planned to depart.

Even though the first element in a vector implemented in Java has the index '0', we assume here that the first element in any vector or list has index = 1 and the last element has index = |vector/list size|.

A.2. Phase 1

(1.1) Start event $k \in \boldsymbol{K}_i$, $i \in \boldsymbol{T}$, if it was ongoing at T_0 and assign the values of the corresponding variables; the initial starting time, $x_k^{begin} = b_k^{initial}$, and initial track, $q_{k,r_k^{track}} = 1$. Add event k as a node at the end of **DT**.
(1.2) Update the running times, d_k, of all events k which are directly affected by the inflicted disturbance.
(1.3) For $\forall$ trains $i \in \boldsymbol{T}$, update $Stat_i$.
(1.4) Compute $b_k^{\min}$ for all the non-active/-terminated events $k \in K_i$ of all trains i, LB and C_s: For $\forall\ i \in \boldsymbol{T}$:
If ($Stat_i$=W): Let $b_k^{\min} = \max\left(h_k * b_k^{init}, T_0\right)$, where k = 1 **then**
For $\forall k \in \boldsymbol{K}_{\mathrm{i}} : k > 1\left\{b_{k+1}^{\min} = \max\left(h_{k+1} * b_{k+1}^{init}, \left(b_k^{\min} + d_k\right)\right)\right\}$
Let $C_{s^*} = C_{s^*} + \max\left(0, \left(b_{|t_{\mathrm{i}}|}^{\min} + d_{t_{\mathrm{i}}|} - e_{t_{\mathrm{i}}|}^{init}\right)\right)$
End if
Else if ($Stat_i$ = A): Let $b_k^{\min} = \max\left(h_k * b_k^{init}, x_{k-1}^{start} + d_{k-1}\right)$ where $k = E_{i.} + 1$. **then**
For $\forall k \in \boldsymbol{K}_{\mathrm{i}} : k > (E_{\mathrm{i}} + 1)\left\{b_{k+1}^{\min} = \max\left(h_{k+1} * b_{k+1}^{init}, \left(b_k^{\min} + d_k\right)\right)\right\}$
Let $C_{s^*} = C_{s^*} + \max\left(0, \left(b_{|t_{\mathrm{i}}|}^{\min} + d_{t_{\mathrm{i}}|} - e_{t_{\mathrm{i}}|}^{init}\right)\right)$
End else if
Else (i.e. $Stat_i$ = T) **then** Let $C_{s^*} = C_{s^*} + \max\left(0, \left(x_{|t_{\mathrm{i}}|}^{end} - e_{t_{\mathrm{i}}|}^{init}\right)\right)$ **End else**

Let LB $= C_{s^*}$.
(1.5) Initiate **NC**.

A.3. Phases 2 & 3

While (|**NC**|> 0 and LB != C^* and the time limit is not reached yet) **do**
Let k = 1. Let $k^* = k^{**} = i^* = t^* = t^{**} = -1$.
While ($k^* < 0$ and $i^* < 0$ and for $\forall$ k $\in$ **NC**, where k $\in \mathbf{L}_j$ and $\mathbf{K}_i$) **do** Let $t' = -1$.
If ($|P'_k| > 0$ and ($Stat_i = W$ or ($Stat_i = A$ and event $k = E_i$ can be closed))) **then** Select track t' from P'_k .
If($t' < 0$) **then** Let $i^* = i$. **end if**
Else if(t' = 0) **then** Let $b_k^{\min} = T_j^{\min}$. For $\forall$ k $\in \boldsymbol{K}_i$: k > (E_i + 1), update $b_k^{\min}$ as in 1.4) **End else if**
Else if $\left(|\boldsymbol{L}'_j \leqslant nrT_j\right)$ **then** Let $t^* = t'$ and $k^* = k$. **End else if**
Else if (deadlockRisk=false) **then**
If (S_k = 1 and $nrT_j < 2$ and possibleConflictOfInterest=true) **then**
Compute ∇C. **If** ($\nabla C = 0$) **then** Let $t^* = t$ and $k^* = k$. **End if**
Else if ($k^{**} < 0$ or weight > ∇C) **then** Let $t^{**} = t$ and $k^{**} = k$. Let weight=∇C. **End else if**
End if
End else if
End if
End while

If($i^* > 0$) **then** Handle deadlock: Backtrack in **DT**reversing all actions (and deleting corresponding nodes) until the parent node of event $k = |E_{i^*}| - 1$. Update all relevant parameters and variables including adding the reversed events to the corresponding lists $\mathbf{L}'_j$ and removing them from $\mathbf{TL}_{j,t^*.}$ **End if**

Else if($k^* < 0$ and $k^{**} < 0$) **then** Handle deadlock: Backtrack in **DT**reversing the last action (and deleting the corresponding node). Update all relevant parameters and variables including adding the reversed event to the corresponding list $\mathbf{L}'_j$ and removing it from $\mathbf{TL}_{j,t^*.}$ **End else if**

Else then
If ($k^* > 0$) **then**
Let $b_{k^*}^{\min} = \max\left(b_{k^*}^{\min}, T_{j,t^*}^{\min}, e_{|E_i|}^{\min}\right) : k^* \in \boldsymbol{K}_i/\boldsymbol{L}_j$. Let $x_{|E_i|}^{end} = x_{k^*}^{begin} = b_{k^*}^{\min}$ and $q_{k^*,t^*} = 1 : t^* \in P'_k$.
Add event k^* to **DT** and $\mathbf{TL}_{j,t^*}$ and remove from $\boldsymbol{L}_j^*$.
If((|Ei| + 1) = |K_i|) **then** $x_{|E_i|+1}^{end} = \max\left(\left(b_{k*}^{\min} + d_{k*}\right), e_{|E_i|}^{\min}\right)$ **End if**
Update $Stat_i$, E_i and any relevant $b_k^{\min}$.
End if

Else if ($k^{**} > 0$) **then**
Let $b_{k^{**}}^{\min} = \max\left(b_{k^{**}}^{\min}, T_{j,t^{**}}^{\min}, e_{|E_i|}^{\min}\right) : k^{**} \in \boldsymbol{K}_i/\boldsymbol{L}_j$. Let $x_{|E_i|}^{end} = x_{k^{**}}^{begin} = b_{k^{**}}^{\min}$ and $q_{k^{**},t^{**}} = 1 : t^{**} \in P'_k$.
Add event k^{**} to **DT** and $\mathbf{TL}_{j,t^{**}}$ and remove from $\boldsymbol{L}'_j$.
If $((|Ei| + 1) = |K_i|)$ **then** $x_{|E_i|+1}^{end} = \max\left(\left(b_{k^{**}}^{\min} + d_{k^{**}}\right), e_{|E_i|}^{\min}\right)$ **End if**
Update $Stat_i$, E_i and any relevant $b_k^{\min}$.
End else if
Compute $C_{|DT|}$ as in 1.4).
If ($|\mathbf{DT}| = DT_{\max}$ and $C^* < 0$) **then** Let $C^* = C_{|DT|}$. Backtrack in **DT** to the first encountered node (counting from the end), where $C_s < C^*$ and according to already defined backtracking scheme above. Let $s_{\text{backtrack}} = |DT|$ after backtracking. **End if**
Else if($|\mathbf{DT}| = DT_{\max}$ and $C^* > 0$ and $C^* > C_{|DT|}$) **then** Let $C^* = C_{|DT|}$. Backtrack in **DT** to the node at index ($s_{\text{backtrack}} - 1$) and according to already defined backtracking scheme above.
Let $s_{\text{backtrack}} = s_{\text{backtrack}} - 1$. **End else if**
Else if ($|\mathbf{DT}| < DT_{max}$ and $C^* > 0$ and $C^* \leqslant C_{|DT|}$) **then** Backtrack in **DT** to the node at index ($s_{\text{backtrack}} - 1$) and according to already defined backtracking scheme above. **End else if**
End else
Reinitiate **NC**.
End while

Appendix B. Station configurations

See Fig. 7.

Appendix C. Timetable for the traffic scenarios

See Fig. 8.

Appendix D. Optimization model

D.1. Objective function

$$\text{Minimize} \quad \sum_{i \in T} z_i \tag{D.1}$$

D.2. Train restrictions

$$x_k^{end} = x_{k+1}^{begin} \qquad i \in T, k \in K_i : k \neq n_i \tag{D.2}$$
$$x_k^{end} \geqslant x_k^{begin} + d_k \qquad k \in E \tag{D.3}$$
$$x_k^{begin} \geqslant b_k^{initial} \qquad k \in E : h_k = 1 \tag{D.4}$$
$$x_k^{begin} = b_k^{static} \qquad k \in E : b_k^{static} > 0 \tag{D.5}$$
$$x_k^{end} = e_k^{static} \qquad k \in E : e_k^{static} > 0 \tag{D.6}$$
$$z_i \geqslant x_{n_i}^{begin} - b_{n_i}^{initial} \qquad i \in T \tag{D.7}$$

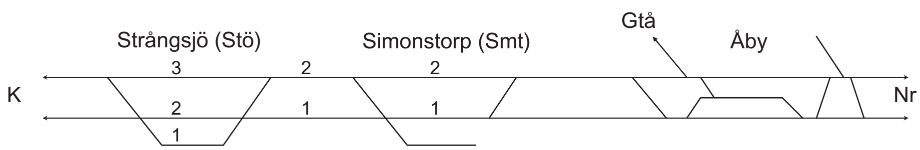

Fig. 7. Illustration of the double-tracked stretch between Katrineholm (K) and Norrköping (Nr) and connects via Åby to the single-tracked stretch to Nyköping and Stockholm. Cf. Fig. 6.

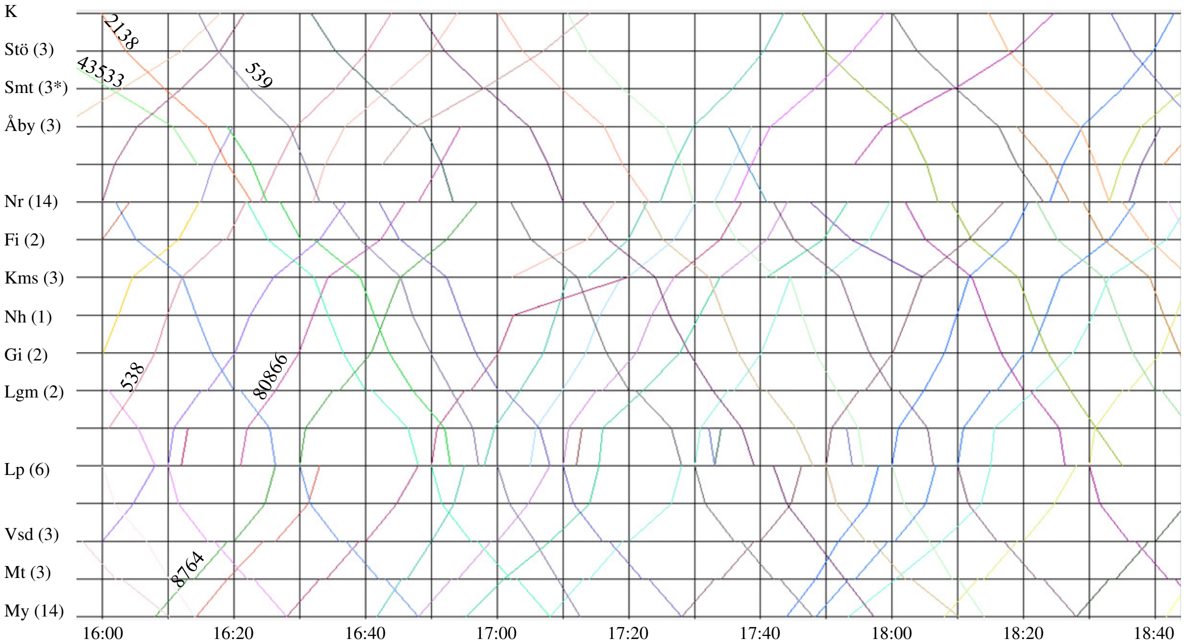

Fig. 8. The timetable for the double-tracked section Katrineholm–Mjölby which was used in the simulation experiments. The traffic on the single-tracked stretch is not as intense. In the parenthesis, after each station acronym on the y-axis, is the number of tracks within the station presented. Smt* refers to that the station Simonstorp has three tracks but the third one is not used. See Fig. 6 for an overview of the line.

D.3. Infrastructure restrictions

$$\sum_{t\in P_j} q_{kt} = 1 \qquad j \in B, k \in L_j : |P_j| > 1 \tag{D.8}$$

$$\sum_{t\in P_j} t^* q_{kt} = r_k^{track} \qquad j \in B, k \in L_j : |P_j| > 1 \& r_k^{fixed} = 1 \text{ (i.e. when } b_k^{static} > 0 \tag{D.9}$$

Constraint (D.10) is used when a train has two consecutive events where both are scheduled on a line section with no meeting possibility in between.

$$q_{kt} = q_{k+1,t} \qquad i \in T, j \in B, k \in L_j, k\&(k+1) \in K_i, t \in P_j : |P_j| > 1 \& k \neq n_i \& S_k = S_{k+1} = 1 \tag{D.10}$$

$$q_{\hat{k}t} + q_{kt} - 1 \leqslant \lambda_{k\hat{k}} + \gamma_{k\hat{k}} \qquad j \in B, k, \hat{k} \in L_j, t \in P_j : k < \hat{k} \& |P_j| > 1 \tag{D.11}$$

For line sections containing only one block section. Constraint (D.13) is for single-tracked sections and (D.14) for sections with multiple tracks:

$$x_{\hat{k}}^{begin} - x_k^{end} \geqslant \Delta_j \gamma_{k\hat{k}} - M(1 - \gamma_{k\hat{k}}) \qquad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& (dir_k \neq dir_{\hat{k}} \| nrB_j = 1) \tag{D.12}$$

$$x_k^{begin} - x_{\hat{k}}^{end} \geqslant \Delta_j (1 - \gamma_{k\hat{k}}) - M\gamma_{k\hat{k}} \qquad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& |P_j| = 1 \& (dir_k \neq dir_{\hat{k}} \| nrB_j = 1) \tag{D.13}$$

$$x_k^{begin} - x_{\hat{k}}^{end} \geqslant \Delta_j \lambda_{k\hat{k}} - M(1 - \lambda_{k\hat{k}}) \quad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& |P_j| > 1 \& (dir_k \neq dir_{\hat{k}} \| nrB_j = 1) \tag{D.14}$$

tive blocks within the section and hence where trains in same direction can follow each other if separated by at least H time units:

$$x_{\hat{k}}^{begin} - x_k^{begin} \geqslant H_j \gamma_{k\hat{k}} - M(1 - \gamma_{k\hat{k}}) \quad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& (dir_k = dir_{\hat{k}} \& nrB_j > 1) \tag{D.15}$$

$$x_{\hat{k}}^{end} - x_k^{end} \geqslant H_j \gamma_{k\hat{k}} - M(1 - \gamma_{k\hat{k}}) \qquad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& (dir_k = dir_{\hat{k}} \& nrB_j > 1) \tag{D.16}$$

$$x_k^{begin} - x_{\hat{k}}^{begin} \geqslant H_j (1 - \gamma_{k\hat{k}}) - M\gamma_{k\hat{k}} \quad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& |P| = 1 \& (dir_k = dir_{\hat{k}} \& nrB_j > 1) \tag{D.17}$$

$$x_k^{end} - x_{\hat{k}}^{end} \geqslant H_j (1 - \gamma_{k\hat{k}}) - M\gamma_{k\hat{k}} \qquad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& |P| = 1 \& (dir_k = dir_{\hat{k}} \& nrB_j > 1) \tag{D.18}$$

$$x_k^{begin} - x_{\hat{k}}^{begin} \geqslant H_j \lambda_{k\hat{k}} - M(1 - \lambda_{k\hat{k}}) \quad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& |P_j| > 1 \& (dir_k = dir_{\hat{k}} \& nrB_j > 1) \tag{D.19}$$

$$x_k^{end} - x_{\hat{k}}^{end} \geqslant H_j \lambda_{k\hat{k}} - M(1 - \lambda_{k\hat{k}}) \qquad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& |P_j| > 1 \& (dir_k = dir_{\hat{k}} \& nrB_j > 1) \tag{D.20}$$

$$\lambda_{k\hat{k}} + \gamma_{k\hat{k}} \leqslant 1 \quad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& |P_j| > 1 \tag{D.21}$$

$$x_k^{begin}, x_k^{end} \geqslant 0 \qquad k \in E \tag{D.22}$$

$$z_i \geqslant 0 \qquad i \in T \tag{D.23}$$

$$\gamma_{k\hat{k}} \in \{0,1\} \qquad j \in B, k, \hat{k} \in L_j : k < \hat{k} \tag{D.24}$$

$$\lambda_{k\hat{k}} \in \{0,1\} \qquad j \in B, k, \hat{k} \in L_j : k < \hat{k} \& |P_j| > 1 \tag{D.25}$$

$$q_{kt} \in \{0,1\} \qquad j \in B, k \in L_j, t \in P_j : |P_j| > 1 \tag{D.26}$$

The notation is based on the one presented in Törnquist and Persson (2007) but some new constraints, variables and parameters have been added.

T is defined as the set of trains, B is the set of sections, and E is the set of events where an event is a resource request by a certain train for a specific section. We let index i be associated with a train, j with a section and index k with an event. Each event is connected to both a train and a section. Let $K_i \subseteq E$ be the ordered set of events for train i ($i \in T$) and $L_j \subseteq E$ be the ordered set of events of section j ($j \in B$). Events in K_i and L_j are ordered according to the original timetable. We use $(k + 1)$ to denote the first proceeding event of event k (in K_i and L_j) and $k < \hat{k}$ to denote that $\hat{k}$ is any event proceeding event k with respect to the order in the sets. Furthermore, let n_i and m_j denote the last event of K_i and L_j, respectively. Each section has a set of parallel tracks $P_j = \{1, \ldots, p_j\}$.

The formulation contains six types of decision variables. The variables x_k^{begin} and x_k^{end} represent the start and end time of event k. z_i represents the delay train i experiences at its final destination within the problem formulation. In addition we have the binary variables:

$$q_{kt} = \begin{cases} 1, \text{if event } k \text{ uses track } t, \text{where } k \in L_j, , t \in P_j.), j \in B. \\ 0, \textit{otherwise}. \end{cases}$$

$$\gamma_{k\hat{k}} = \begin{cases} 1, \text{if event } k \text{ occurs } \textit{before} \text{ event } \hat{k} (\text{as in the initial timetable}), \\ \text{where } k, \hat{k} \in L_j, j \in B : k < \hat{k}. \\ 0, \text{otherwise}. \end{cases}$$

$$\lambda_{k\hat{k}} = \begin{cases} 1, \text{if event } k \text{ is re–scheduled to occur } \textit{after} \text{ event } \hat{k}, \\ \text{where } k, \hat{k} \in L_j, j \in B : k < \hat{k}. \\ 0, \text{otherwise}. \end{cases}$$

S_k specifies if event k takes place at a line section ($S_k = 1$) or a station section ($S_k = 0$).

M is a large constant and H is the headway parameter. The parameter dir_k gives the direction of the train during event k and nrB_j gives the number of consecutive block sections within one and the same line section j.

References

Assad, A., 1980. Models for rail transportation. Transportation Research Part A 14A, 205–220.

Banverket, 2009. Årsredovisning 2008 (in Swedish), Borlänge, Sweden.

Banverket, 2008. Årsredovisning 2007 (in Swedish), Borlänge, Sweden.

Banverket, 2007. Årsredovisning 2006 (in Swedish), Borlänge, Sweden.

Banverket, 2006. Årsredovisning 2005 (in Swedish), Borlänge, Sweden.

Banverket, 2005. Årsredovisning 2004 (in Swedish), Borlänge, Sweden.

Billionnet, A., 2003. Using integer programming to solve the train-platforming problem. Transportation Science 37 (2), 213–222.

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

D'Ariano, A., Pacciarelli, P., Pranzo, M., 2007. A branch and bound algorithm for scheduling trains on a railway network. European Journal of Operational Research 32 (2), 643–657.

D'Ariano, A., Pranzo, M., 2009. An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. Networks and Spatial Economics 9, 63–84.

Emer, J., Hill, M., Patt, Y., Yi, J., Chiou, D., Sendag, R., 2007. Single-threaded vs. multithreaded: Where should we focus? IEEE Micro 27 (6), 14–24.

European Commission, 2001. European transport policy: time to decide 2010. White paper.

Gely, L., Dessagne, G., Pesneau, P., Vanderbeck, F., 2008. A multi scalable model based on a connexity graph representation. In: Proceedings of Comprail 2008, Toledo, Spain.

Jansson, G., Jonsson, O., 2003. En pilotstudie av punktligheten i Europeiska godstransporter på järnväg. TFK Rapport 2003:7, Stockholm, Sweden.

Jacobs, J., 2008. Rescheduling. In: Hansen, Pachl (Eds.), Railway Timetable & Traffic, Analysis, Modelling, Simulation. Eurailpress, Hamburg, Germany, pp. 182–191.

Kroon, L., Huisman, D., Maróti, G., 2008. Optimization models for railway timetabling. In: Hansen, Pachl (Eds.), Railway Timetable & Traffic, Analysis, Modelling, Simulation. Eurailpress, Hamburg, Germany, pp. 135–154.

Kroon, L., Romeijn, E., Zwaneveld, P., 1997. Routing trains through railway stations: complexity issues. European Journal of Operational Research 98, 485–498.

Radtke, A., 2008. Infrastructure modelling. In: Hansen, Pachl (Eds.), Railway Timetable & Traffic, Analysis, Modelling, Simulation. Eurailpress, Hamburg, Germany, pp. 43–57.

Schöbel, A., 2009. Capacity constraints in delay management. Journal of Public Transport 1 (2), 135–154.

Törnquist, J., 2005. Computer-based decision support for railway traffic scheduling and dispatching: a review of models and algorithms. In: Proceedings of ATMOS2005, Palma de Mallorca, Spain, October 2005, Published within the Dagstuhl Research Online Publication Server (DROPS) <http://drops.dagstuhl.de/portals/ATMOS/>.

Törnquist, J., Persson, J., 2007. N-tracked railway traffic re-scheduling during disturbances. Transportation Research Part B 41 (3), 342–362.

Törnquist, J., 2007. Railway traffic disturbance management: an experimental analysis of disturbance complexity, management objectives and limitations in planning horizon. Transportation Research Part A 41 (3), 249–266.

UIRR (International Union of combined Road-Rail transport companies), 2008. Punctuality statistics for 1999–2007. <http://www.uirr.com/?action=page&page=52&title=Quality> (accessed 01.09.09).

Zhou, X., Zhong, M., 2007. Single-track train timetabling with guaranteed optimality: branch-and-bound algorithms with enhanced lower bounds. Transportation Research Part B 41 (3), 320–342.