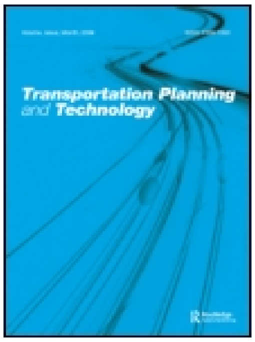

Routledge
Taylor & Francis Group

Transportation Planning and Technology

ISSN: 0308-1060 (Print) 1029-0354 (Online) Journal homepage: https://www.tandfonline.com/loi/gtpt20

Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area

F. Corman , A. D'Ariano , M. Pranzo & I.A. Hansen

Transportation Planning and Technology
Vol. 34, No. 4, June 2011, 341–362

Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area

F. Corman[a]*, A. D'Ariano[a,b], M. Pranzo[c] and I.A. Hansen[a]

[a]*Transport & Planning Department, Delft University of Technology, Delft, The Netherlands;* [b]*Dipartimento di Informatica e Automazione, Università degli Studi Roma Tre, Roma, Italy;* [c]*Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Siena, Siena, Italy*

(*Received 29 January 2010; final version received 29 March 2011*)

Railway traffic experiences disturbances during operations that cause conflicts between train paths or even deadlock situations. Dispatchers need actions to restore feasibility and limit spreading of delays through the network. To help them in such a task, the dispatching support tool ROMA (Railway traffic Optimization by Means of Alternative graphs) has been implemented in a laboratory environment. This paper reports on enhancements to the underlying train dispatching model as well as to the solution algorithms studied in order to tackle the increased complexity of busy stations with multiple conflicting paths and high service frequencies. Advanced train reordering and rerouting techniques are compared with straightforward rules and the current approach in the Netherlands. Extensive computational studies based on accepted statistical distributions of train delays for Utrecht Central Station assess the effectiveness of the ROMA tool in terms of solution quality and computation time.

Keywords: dispatching support tool; rescheduling; rerouting; alternative graph incompatibility graph; Utrecht

1. Introduction

Railway traffic management is mainly directed towards the implementation of an existing plan of operations (off-line timetabling) and its adjustment due to disruptive events as quickly as possible (real-time dispatching). The timetable is characterised by departure and arrival times of each train at station platforms and/or at relevant merging and crossing points. The assignment of routes, platforms and passing times may require months, during which several variants are analysed in depth under economical and operational constraints. In real-time, unforeseen events may disrupt the timetable and thus the resolution of route conflicts and other infeasibilities is required.

The real-time dispatching process is to determine feasible (conflict-free and deadlock-free) train movements minimizing timetable deviations. An accurate prediction of the effects of delays and other disturbances requires modelling the evolution of train traffic in sufficient detail and considering the actual state of the network, both the dynamic behaviour of circulating trains and the dispatching measures used to control traffic. Hence, the precise delay propagation cannot be

*Corresponding author. Email: f.corman@tudelft.nl

ISSN 0308-1060 print/ISSN 1029-0354 online

DOI: 10.1080/03081060.2011.577152
http://www.informaworld.com

predicted by dispatchers, especially in the case of complicated station areas, high density traffic and severe disturbances. Furthermore, railway managers are looking for decision support systems that enable their operators to determine implementable control actions as quick as possible. For these reasons, there is a need for developing more sophisticated and efficient decision support tools to forecast network delay propagation for individual dispatching measures.

In this context, we implemented a laboratory dispatching support tool, called ROMA (D'Ariano 2008, D'Ariano *et al*. 2008, D'Ariano and Pranzo 2008, Corman, *et al.* 2010). This tool is designed to support dispatchers in railway traffic monitoring, control and reaction to various types of disturbances (such as multiple delayed trains, dwell time perturbations, block sections unavailability, and others) within a short computation time. The dynamic traffic control may coordinate the speed of successive trains on open track (retiming), avoid expected route conflicts (reordering) and allow for dynamic use of platform tracks or alternative paths along a line (local rerouting).

The innovative scientific contribution of our approach is characterised by a unique combination of blocking time theory (Hansen and Pachl 2008) for the recognition of route conflicts in case of disturbances and a general discrete optimization model, based on the alternative graph formulation (Mascis and Pacciarelli 2002), for the real-time evaluation of train reordering and rerouting in railway networks. The feasibility of the rescheduling options is verified in a very limited computation time by dynamically updating the corresponding headway distances, train speeds and blocking times, while the costs of the different options are measured in terms of maximum and average delays at stations and other relevant points within the investigated network.

In this paper, we extend the laboratory dispatching support tool ROMA to cope with more saturated railway networks and complex infrastructure configurations. This further level of detailed information enables accurate computation of train trajectories and routes in complicated and busy station areas with several intersecting paths to lines and platforms. To this end, the following additional features have been implemented in ROMA:

- Automatic generation of timetable and infrastructure data (using an interchange format similar to RAILML, based on the Dutch InfraAtlas infrastructure format and DONS timetable database) in order to enable modularity in data handling and improve the information flow.
- Alternative graph formulation of the real-time train dispatching process in presence of many block sections and routes within interlocking areas.
- Modification of the algorithms to detect and solve possible conflicts on short block sections by taking into account incompatibilities between train routes.

We also discuss the interactions between dispatchers and decision support tool and the insertion of the tool in a dynamic setting with actual operations (a discussion on practical issues concerning the implementation, applicability and accuracy of the rescheduling actions has been presented elsewhere (Lüthi *et al.* 2007, Mazzarello and Ottaviani 2007).

Our dispatching support tool is applied to the management of a complex and busy station area of the Dutch railway network, Utrecht Central Station, and to evaluate its performance. Disturbed traffic conditions are simulated, according to a statistical description based on realization data of train traffic. For each disturbance the tool provides a series of dispatching actions for a traffic control period of one hour. The experiments aim at evaluating the impact of the dynamic traffic control strategies by assuming that times, orders, routes and connections of all trains at all stations could be adjusted to produce a better feasible schedule. The computational results intend to demonstrate the computational effort of the proposed test cases and to assess the effectiveness of the advanced dispatching support tool. The dispatching solutions are presented in terms of train delays. Furthermore, the solutions obtained by the optimization are compared with those computed by using straightforward (local and on the spot) dispatching rules or no dispatching action (timetable solution).

The paper is organised as follows. Section 2 gives a brief literature review of models and algorithms for railway traffic management. Section 3 presents the real-time train dispatching process, a detailed model developed for the resolution of possible conflicts between trains with extensions to the management of complex station areas and the train dispatching support tool based on that model. A discussion on the real-time application of this tool is also proposed, along with a description of the possible interactions with the human operator (dispatcher). Section 4 reports the computational results obtained for the Utrecht station area. Section 5 concludes the paper with a discussion on the current state of development of the dispatching support system and further research directions.

2. Related literature

Existing models for solving routing and scheduling problems can be classified according to two levels of approximation: macroscopic models and microscopic models. Off-line timetabling usually relies on macroscopic models, while microscopic approaches are mandatory when dispatching train traffic in real-time.

To keep complexity low at the planning stage, macroscopic approaches model a railway network as a simplified series of links connecting stations. A fixed running time is required to travel between two stations, and a fixed headway time is imposed between consecutive trains on the same link or platform at stations. The time variables are normally bounded to full minutes. Several works on timetabling use a formulation based on the periodic event scheduling problem by Serafini and Ukovich (1989) which assumes infinite capacity at stations and a rough model of headway times and safety system.

For the Dutch railways, DONS is the macroscopic tool adopted to design timetables. A network scheduler module, called CADANS (Schrijver and Steenbeek 1994), determines a feasible cyclic network timetable, while having fixed running and minimum headway times, and neglecting capacity at stations. A second level module, called STATIONS (Zwaneveld *et al.* 1996), manages the routing of trains in complex station areas. The model takes into account incompatibility between routes according to predefined safety constraints and builds a graph of incompatibilities in which a maximum weight node packing corresponds to a feasible routing solution, while neglecting the impact of signalling and train length on blocking times.

Carey and Crawford (2007) present a sophisticated model and a novel heuristic procedure to assess the benefits of an existing draft timetable for a network of busy stations. The precise track layout of stations and incompatibility of conflicting routes and platform tracks occupations are taken into account in the model, while train separation is formulated as a fixed minimum headway distance. The heuristic solves the route conflicts along the tracks by selecting the least immediate delay cost.

Caimi *et al.* (2009) solve ordering and routing problems in station areas simultaneously, by building a large conflict graph that takes into account multiple scheduling possibilities for each train. A fixed point iteration algorithm has been implemented to compute a feasible solution in a reasonable computation time. The procedure assumes that trains have fixed running and headway times in interlocking areas, while acceleration, braking, dwell time extensions, as well as variations of train length, are not discussed.

A greater level of detail is needed to properly control railway traffic during operations. Accurate train positions, speeds, and acceleration and braking time losses have to be computed for a reliable prediction of train trajectories including blocking times. The speed profile of trains has to be computed according to the actual speed limits and the corresponding acceleration and deceleration rates (Wegele *et al.* 2008). The signalling system with actual signal aspects needs to be modelled along corridors and in station areas, while a precise layout of interlocking areas is required to take into account incompatibilities between routes.

The resolution of the real-time dispatching process is a demanding task, especially in a network of stations with multiple merging and crossing lines, and experienced dispatchers usually limit the degree of freedom by looking for simple solutions, that often may be sub-optimal. Among the recent contributions, Jacobs (2004) has developed a train rescheduling model based on detection of route conflicts with high precision, by means of blocking time theory, with the objective of reducing the running time extensions. A heuristic procedure based on train priorities is applied to build up incrementally a dispatching plan, solving infeasibilities in an asynchronous and locally optimal way.

Rodriguez (2007) studies rerouting and reordering possibilities for a small network with up to 12 trains. A job shop scheduling model with additional state resources constraints is proposed to detect and solve route conflicts. Synchronization constraints are used to keep train running with sufficient headway distances, even in the case of yellow (caution) or red (stop) signal aspects. However, variability of train speed profiles is not considered. Dispatching solutions are computed by constraint-based programming in a short computation time.

Törnquist and Persson (2007) propose a mixed integer linear programming formulation to manage disturbed traffic conditions by means of train reordering and rerouting. They assume a fixed headway time between trains and fixed running times along segments between stations and relevant interlocking areas. The objective is to minimise a cost function based on train delays. Results on a real network with few delayed trains show that there are still good margins for improving punctuality of trains.

D'Ariano (2008) uses an alternative graph formulation to perform train reordering and rerouting. Blocking time theory is adopted to check whether the required minimum headway distances between trains are respected. The possibilities of optimizing the routing of trains are explored by means of metaheuristics while the problem of scheduling trains in complex areas is solved by a branch and bound algorithm within a given time of computation. Numerous results on two practical dispatching areas with small stations and several railway corridors are reported to assess the effectiveness of the proposed approach.

Most of the previous work on automated rescheduling has investigated simplified railway networks and implicit representation of incompatibilities in interlocking areas. The complexity of managing station areas with dense traffic and multiple conflicting routes is acknowledged by dispatchers, especially in the case of large disturbances. Hence, the development of a decision support system for rescheduling in complex interlocking areas is of great relevance to improve quality of railway operations. In order to proper manage traffic in complicated interlocking areas and provide accurate solutions to the dispatchers, detailed microscopic information is to be taken into account.

Therefore, there is a need to develop methods that use a rather detailed description of the rail network that is able to model the possible reordering and rerouting possibilities along tracks and inside main station areas. The solution of such instances of increased complexity should be found without losing necessary details. Realistic train separation rules based on signal spacing, interlocking of routes and blocking time theory should be used, as the use of fixed headway times is too rough in case of dense traffic and multiple interactions between several inbound and outbound routes at stations. Moreover, the trade-off between solution quality of the rescheduling process and time required to compute dispatching solutions is of crucial importance in the practical setting of a decision support tool and must not be underestimated.

3. Real-time train dispatching

3.1. Problem description

A railway network is composed of stations connected by lines and tracks that are operationally divided in block sections. The passage of a train through a block section is called an *operation* and a sequence of operations to be traversed by a train is called a *route*. At any time a route is passable if all its block sections are available, i.e. there is no blocked track along the route. The timing of a route specifies the starting time t_i of each operation in the route. Each operation requires a travelling time, called *running time*, that is computed according to the dynamics of each rolling stock, namely the parameters about maximum speed and acceleration ratios, speed restrictions on the infrastructure, and the driver behaviour according to signalling system constraints (e.g. in case of a red (stop) signal the train must brake to a complete stop and then re-accelerate).

According to standard railway safety regulations (Bailey 1995), no more than one train at a time is allowed to occupy any block section (conflict-free condition); the three-aspect fixed block signalling system and the Automatic Train Protection system are used to ensure a safe headway between successive trains and to generate an automatic brake in case of accidents or technical failures errors. This headway translates to a *setup time*, required to modify the signalling aspect and to change the route after the tail of a train has released the track segment. The minimum headway time between two successive trains depends on their speeds, the braking rate of the following train, the length of the preceding train, the distance between signals and the time needed to setup a new route. A train speed profile is acceptable when the acceleration rate, deceleration rate, maximum speed of the actual rolling stock and the minimum headway time between trains are respected.

We consider a cyclic timetable which describes the movement of all trains circulating in the network during subsequent time periods, specifying, for each train, the planned arrival/passing times at a set of relevant points along its route (e.g. stations, junctions, and the exit point of the network). At stations, a train is not allowed to depart from a platform stop before its scheduled departure time and is considered late if arriving at the platform later than its scheduled arrival time. At a platform stop, the scheduled stopping time of each train is called *dwell time*. Additional practical constraints related to passenger satisfaction should be taken into account, such as minimum transfer times between connected train services. This is the time required to allow passengers alight from one train, move to another platform track and board the other train.

Constraints due to rolling stock circulation must also be taken into account. During the service of a line rolling stocks complete a number of round-trips and may change their composition. In the Netherlands, train services may be coupled and uncoupled during operations, by combining one train with another train or by splitting into two distinct train units. Railway timetables include a technical service time at terminal stations, which is the time between the arrival of a train and the start of a new service using the same rolling stock. Various types of perturbations to the timetable occur during operations. Examples of disturbances may include infrastructure breakdowns, delay of trains entering the dispatching area from the previous one, speed limitations due to technical failure or extended dwell times at stops. Such disturbances may lead to potential conflicts or even deadlocks. A potential *conflict* is when two or more trains claim the same available block section simultaneously, and a decision on the train ordering has to be taken. In that case the movement authority is given to only one of the trains involved at a time. In case of short headway distances, the other trains are forced by the signalling and train protection system to decrease their speed or to even completely stop on the open track or within the interlocking area. The conflict resolution may therefore cause some train delays. A set of trains cause a *deadlock* when each train in the set claims a block section ahead which is not available and cannot be made available, due either to an infrastructure breakdown or to the occupation/reservation for another train in the set.

Following the notation of D'Ariano (2008) and D'Ariano, Pranzo, *et al.* (2007), the *total delay* is the positive difference between the estimated train arrival time and the scheduled time at a relevant point in the network, and can be divided into two parts. An *initial* (or *primary*) *delay* is directly caused by late departures, failures or disturbances and exceeds available running time margins until the next potential conflict or timetable reference point. A *consecutive* (or *knock-on*) *delay* is caused by the interdependence between trains running in the network and in principle can be minimised by proactively managing railway traffic.

Figure 1 shows two traffic situations that illustrate the problem addressed in this paper. In both situations, the railway network is composed of 10 block sections and a station with 2 platform stops on block sections 3 and 4 respectively.

In Figure 1(a), three trains (*TA*, *TB* and *TC*) are running in this simple network. *TA* is a fast train with no platform stop and three alternative routes: $R_A^1 = (A1, A2, A3, A5, A7, A9)$, $R_A^2 = (A1, A2, A3, A5, A7, A8, A10)$ and $R_A^3 = (A1, A2, A4, A6, A8, A10)$. *TB* is a fast train with a platform stop on block section 3 and two alternative routes:

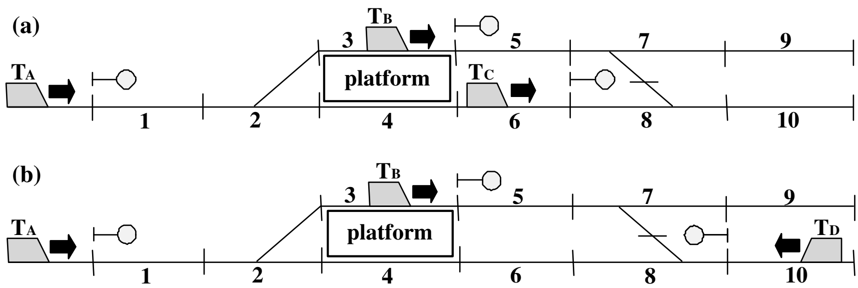

Figure 1. Examples: (a) situation with no deadlock; (b) situation with potential deadlocks.

$R_B^1 = (B3,\ B5,\ B7,\ B9)$ and $R_B^2 = (B3,\ B5,\ B7,\ B8,\ B10)$. Train *TC* is a slow train with only one route: $R_C = (C6,\ C8,\ C10)$. In this traffic situation there is no deadlock in the network for all possible route combinations. However, the train routes share a number of block sections (e.g. R_A^3 and R_C share block Sections 6, 8 and 10).

During disturbed operations, trains face initial delays and the conflict-free condition requires train ordering and routing decisions in order to compute a feasible schedule and to limit the propagation of consecutive delays. For the situation in Figure 1(a), let the dwell time of *TB* be strongly delayed and the running time of *TC* considerably larger than that scheduled. A critical routing decision is to select a route to *TA*. In fact, R_A^1 and R_A^2 face at least a potential conflict with R_B^1 and R_B^2 on the stopping point (i.e. block section 3) while R_A^3 face at least a potential conflict with R_C on non-stopping points (i.e. block sections 6, 8 and 10). A critical ordering decision is to select a precedence between R_B^2 and R_C on block section 8.

In Figure 1(b), three trains (*TA*, *TB* and *TD*) are running on the network. *TA* and *TB* have the same routes as for the previous situation. *TD* is a fast train with no platform stop and two alternative routes: $R_D^1 = (D10,\ D8,\ D6,\ D4,\ D2,\ D1)$ and $R_D^2 = (D10,\ D8,\ D7,\ D5,\ D3,\ D2,\ D1)$. In this traffic situation there are various sets of routes that lead to deadlocks. For instance, R_A^3 is always infeasible while R_A^2 and R_B^2 are only feasible if *TA* and *TB* are scheduled after *TD* on block sections 8 and 10.

3.2. Problem formulation

The real-time dispatching process can be approached by retiming trains, i.e. by using running time supplements that are included in the timetable. The train sequence at junctions and merging points may also be adjusted (reordering) to the actual delay situation; for example, the trains can be rescheduled in the order they arrive. A further degree of freedom is to change locally the route used by a train (rerouting); for example, an empty platform can be used instead of causing a delay while waiting for a still occupied track.

This section introduces a detailed formulation of the real-time process of changing dwell times as well as train orders and routes in order to solve potential conflicts and to avoid deadlocks in case of disturbed operations. This is called Conflict Detection and Resolution (CDR) and can be partitioned into two sub problems: (1) a scheduling problem, for which the starting time of each operation is determined, and (2) a rerouting problem, for which a route is associated to each train among a set of rerouting possibilities. In what follows, we refer to Conflict Detection and Resolution with Fixed Routes (CDRFR) to denote the scheduling problem with a single route and a fixed speed profile assigned to each train, with the objective of minimizing the consecutive delays.

It can be observed that the combinatorial structure of the CDR problem is similar to that of a job shop scheduling problem with several additional constraints. In job shop scheduling (Pinedo 1995), a job must be processed by a prescribed sequence of machines and each machine can process one job at a time. The processing of a job on a machine is an operation. The job shop scheduling problem therefore consists of defining starting times for all operations such that each operation of a job starts after the completion of its predecessor and no machine processes two operations simultaneously. In terms of the CDR problem, jobs correspond to running trains and machines to block sections of the signalling and train control system. The processing time of an operation can be used to represent the running time of the train on the corresponding block section. Since a block section cannot host two trains simultaneously, there is a potential conflict if two or more trains claim the same block section at the same time and therefore would be in conflict with the minimum setup time required for that block section. Solving the potential conflict corresponds to defining a processing order and time between incompatible operations (i.e. claims of infrastructure capacity by different trains). In a CDR solution, a set of routes and timings are feasible if, for each pair of operations associated to the same block section, the minimum setup time constraints are satisfied and there is no deadlock in the network.

Mascis and Pacciarelli (2002) introduce alternative graphs to model variants of job shop scheduling problems. An alternative graph is a triple $G = (N; F; A)$ being N a set of nodes, F a set of fixed directed arcs, and A is a set of pairs of alternative directed arcs. A graph selection S is a set of alternative arcs chosen from A such that at most one arc is selected for each pair. A feasible solution to the scheduling problem is graph selection that is complete (exactly one arc from each alternative pair is chosen) and consistent (there are no positive length cycles in the graph).

In the alternative graph formulation, the CDRFR problem is to find a feasible starting time ti to each operation oi (i.e. the exact time in which each train will enter each block section) such that all fixed precedence relations are satisfied, exactly one of each pair of alternative precedence relations is selected and the resulting alternative graph has no positive length cycles. A positive length cycle represents a deadlock situation, i.e. an operation preceding itself. In general, negative length cycles allow more general scheduling situations to be modelled (Mascis and Pacciarelli 2002). To summarise, a conflict-free and deadlock-free schedule for the CDRFR problem is associated with a complete consistent selection in the alternative graph $G(S)$.

An alternative graph gives all possible schedules once train routes have been fixed. When addressing the CDR problem, the set of routes can be changed by selecting a different set of fixed arcs (i.e. a different train route). The set F will be the decision variable taking care of the route chosen and $A = A(F)$ is a consequence of the choice for F. For a chosen route set Fi the solution to the associated CDRFR problem $S(Fi)$ is a complete consistent selection in $A(Fi)$. The solution to the CDR problem will be $(F; S(F))$. All the relevant times associated with the running of a train in a track segment can be modelled in the alternative graph formulation by using blocking time theory (D'Ariano, Pranzo et al. 2007). The blocking time is the time interval in which a block section is exclusively reserved to a train and blocked for other traffic. A virtual blocking time overlap arises if the scheduled minimum headway distance between two train paths is not respected. Several additional railway constraints may be easily included in the alternative graph model by a suitable choice of arcs and weights. Minimal connection time between different train services can be represented by an arc expressing the minimal required time distance between two operations. Coupling and decoupling of rolling stock can be represented by additional time relations between the services that are going to be split or combined.

3.3. Aggregation of block sections in complex station areas

A large amount of operations needs to be considered in the interlocking areas of complex stations. As a result, many variables and constraints affect the starting time of each operation. Due to the computational complexity of the problem, it is very difficult to compute suitable solutions to large scheduling instances in a short execution time. We next present an efficient procedure to aggregate information with the goal of reducing the number of decision variables and constraints.

An *aggregated block* is a sequence of consecutive block sections along the route that a train may traverse one after each other (see Figure 2). The sequence of corresponding operations *oi,. . ., om* can be modelled as an operation of the aggregated block. The aggregated block operation is completely defined by the set of operations performed over the involved block sections. Precisely, the running time over an aggregated block is determined by the sum of running times associated to the involved block sections. The setup time over an aggregated block operation is computed as follows. At the time a given train running in the interlocking area clears the last block section of its aggregated block, all block sections of the aggregated

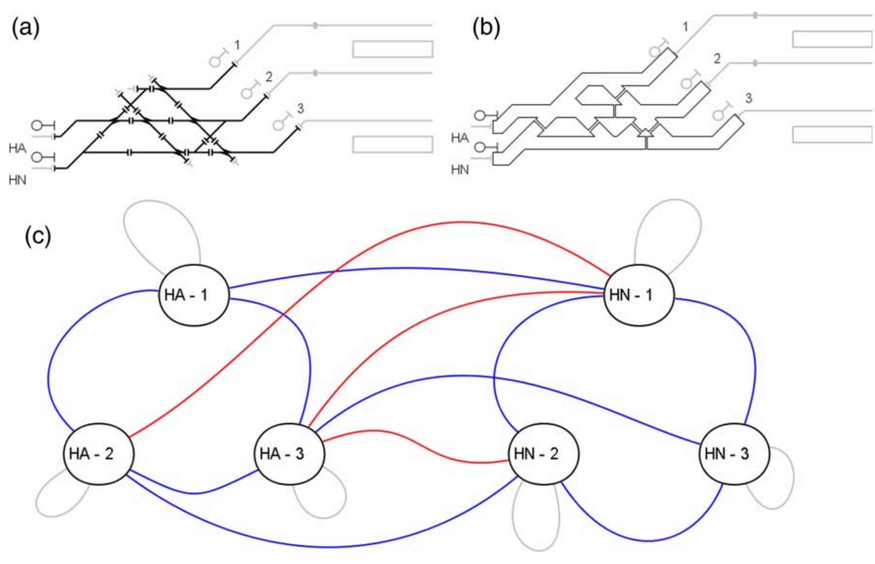

Figure 2. Block section and aggregated blocks: (a) block sections; (b) virtual machines associated to the aggregated blocks; (c) incompatibility graph between the aggregated blocks.

block are released simultaneously, and thus become available to other trains. In this way, each train can start running over its aggregated block without being hampered by other trains.

When dealing with complicated interlocking areas the use of aggregated block operations allows the size of the problem to be reduced. In fact, all the decision variables on the starting times of the individual operations in an aggregated block are taken into account at once. However, since too much aggregation of information leads to oversimplification, the procedure is to be restricted to the case when physical layout characteristics and operational rules constrain the individual operations to be related to each other. For instance, aggregating the whole path of a train into a single operation would incorrectly restrict the capacity of a railway corridor to be used by a single train at a time.

A convenient aggregation approach is to start and end aggregated blocks in correspondence to the main signals. In fact, when dealing with disturbances and short headway distances between trains, a red signal aspect may be shown, due to interlocking rules, to all the following trains having the same route or any other route with some shared block section. In this situation, the main signal would give a movement authority to each train only when all the block sections of its claimed route are free. In case of sectional release, block sections are released one at a time as the tail of the current train has cleared it.

This paper presents a more conservative approach that is based on the common dispatching practice for which a route is released only when the current train has left the last block section of its route. So doing, only one train at a time is effectively constrained to be scheduled in an aggregated block and complementary information about incompatibilities between routes should be associated with the derived aggregated blocks. (For a quantitative comparison of aggregated and disaggregated approaches we refer the reader to Corman *et al.* 2009)

A similar aggregation procedure is performed when modelling the track between two stations as a simple link (this is usually applied for timetabling purposes). In this case, compatibility between different paths is enforced by a minimum fixed headway time between trains in conflicting routes. However, we consider a more accurate model of the interactions between trains in case of disturbed operations; that is, we have to compute the speed trajectories of all the trains involved in route conflicts and their actual blocking times.

An example of aggregated blocks is described in Figure 2. We refer to an interlocking area connecting three platform tracks (1, 2, 3) to two open tracks (HA and HN). In Figure 2(a), there are nine block sections that are grouped into six aggregated blocks (HA-1, HA-2, HA-3, HN-1, HN-2 and HN-3), connecting each platform with each open track. The graph of Figure 2(c) depicts the incompatibilities between the six aggregated blocks, corresponding to the six nodes of the graph. Each arc connects two nodes that are incompatible. As only one train at a time is allowed to be scheduled in an aggregated block, each aggregated block is thus incompatible with itself. The other incompatibilities are given by the track layout. In total, there are 18 incompatibilities out of all the possible relations, i.e. only three pairs of blocks are not mutually excluding each other (<HA-1, HN-2 >, <HA-2, HN-3 >, <HA-1, HN-3 >).

As shown in the example of Figure 2, we propose an incompatibility graph (Corman *et al.* 2009) that is able to model incompatibilities between aggregated blocks that are due to the complex track topology of interlocking areas. A more compact representation relies on the introduction of *virtual machines* and on their association to aggregated blocks (see Figure 2). A virtual machine is a machine of the job shop problem that is used to keep track of the incompatibilities, such that any two aggregated blocks are compatible if there is no virtual machine associated to both of them. In other words, virtual machines represent all the incompatibilities between conflicting routes in a complicated interlocking area.

The number of virtual machines is given by the number of aggregated blocks that happen to be compatible with at least another aggregated block – that is, the number of nodes of the incompatibility graph that are not connected with all the other nodes. The procedure adopted in this paper is to scan the incompatibility graph and search for the virtual machines needed to model all the incompatibilities between the aggregated blocks. The virtual machines are then introduced in the alternative graph formulation of the CDR problem, allowing the characteristics of the non-aggregated model to be translated into the aggregated one.

In the illustrative example of Figure 2, all the incompatibilities between the six aggregated blocks can be expressed by means of four virtual machines only, since the aggregated blocks compatible with each other are HA-1, HN-2, HA-2 and HN-3. In Figure 2(b), the virtual machines are represented as composition of block sections.

3.4. Decision support tool

The ROMA software is a decision support tool designed to assist traffic controllers in the evaluation of real-time dispatching solutions. ROMA is implemented in C++ and uses the AGLibrary developed by the 'Aut.Or.I.' Research Group of Roma Tre University.

For any situation of timetable deviation, such as multiple delayed trains, ROMA performs the following main procedures:

- Assigning a feasible route to each train such that there are no blocked tracks;
- Defining optimal train orders, routes, and times such as the exact arrival and departure times at stations, and the passing times at a set of relevant points (e.g. stations, junctions and the exit point of the network).
- Ensuring a minimum required time headway between the exit of a train from a block section and the entrance of the subsequent train into the same block section while maintaining acceptable speed profiles.

The three procedures are solved automatically by ROMA by means of the modules described in Figure 3. So far, the tool is a laboratory version and the field layer is only simulated. We next give a brief description of each module.

The *automatic data loading* module is in charge of collecting information regarding on-line positions and speeds of trains, infrastructure availability status, timetable, and rolling stocks. Accurate running and setup times are also computed for all the trains running in the network. The off-line information about the infrastructure layout is loaded directly from the InfraAtlas database in use at ProRail (the Dutch infrastructure manager). The timetable is translated from the DONS format by a dedicated automatic procedure. The data is stored in a format compatible with the current RAILML specifications that are a standard interface for railway data in Europe.

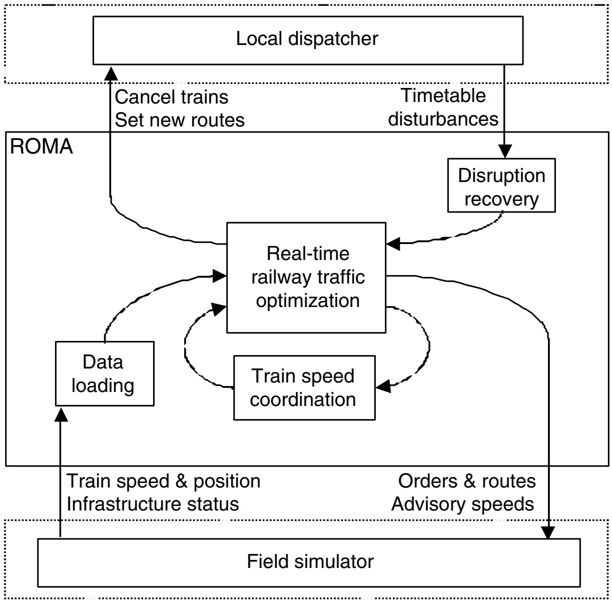

Figure 3. ROMA dispatching support tool architecture.

Regarding the on-line information, we suppose there are GPS sensors on board of trains such that information flows over a GSM-R channel to the dispatching control centre. A more comprehensive method could be to combine GSM-R for actual train speed data and an automatic data mining tool, such as TNV-Conflict (Daamen *et al.* 2009), for track occupancy and clearance data.

The *disruption recovery* module is in charge of providing a feasible route for each train when one or more block sections are unavailable to traffic, e.g. due to infrastructure disruptions. A list of train routes is given to this module, that selects the alternative most similar to the scheduled one. If no predefined route is available for a given train, the local dispatcher is asked to introduce new routes or cancel train services manually.

The *real-time railway traffic optimization* module is the optimization core of the ROMA tool. The module is able to exploit retiming, reordering and rerouting strategies. The first two degrees of freedom are tackled by solving a given CDRFR problem, while the third is addressed by the CDR problem. We next introduce the CDRFR and CDR algorithms used in this paper (a detailed description can be found in (Carey and Crawford 2007, D'Ariano 2008, D'Ariano *et al.* 2007, 2008).

Several heuristic methods can be used to solve the CDRFR problem. A straightforward method is to impose the order prescribed by the timetable; another is to follow the simple rule of First Come First Served (FCFS), i.e. to assign priority to the train that claims the concerned block section first. In the Dutch dispatching practice an automatic route setting system, called ARI (Berends and Ouburg 2005), is usually adopted so long as the delay is less than a pre-defined threshold. Specifically, the movement authority is given on the basis of the FCFS rule for tracks crossing each other while for merging tracks the orders specified in the timetable are followed. In case of larger delays, expert dispatchers take train ordering decisions directly with the support of a list of what-if scenarios. In order to implement the ARI system in an automatic way and for any kind of delay, we set up a list of priority rules for the latter case.

In order to perform an exhaustive search in the space of CDRFR solutions, a truncated Branch and Bound (BB) algorithm has been implemented (D'Ariano *et al.* 2007). This algorithm computes near-optimal solutions within a time limit of computation compatible with operations (a discussion about relevant times will be presented in Section 3.5). A good starting solution, or upper bound, is found by a set of heuristics for solving the CDRFR problem, such as the FCFS rule and other simple algorithms based on alternative graph properties.

A tight lower bound for the CDRFR problem based on the Jackson Pre-emptive Schedule (Jackson 1955) is computed for each (possibly virtual) machine. Using a level of description lower than the aggregated blocks improves the result by taking into account all the incompatibilities; on the other hand, the virtual machines have been introduced as the minimal set describing the incompatibilities, and hence are the best possible choice of job shop machines on which to compute the lower bound. The lower bound on each machine is computed in $O(z \log z)$ steps, where z is the number of trains scheduled on the machine.

After computing a good solution for the CDRFR problem, a Tabu Search (TS) algorithm has been developed to search for alternative routes potentially leading to better schedules (Carey and Crawford 2007). Given a route-set F, we evaluate the quality of a solution by computing a new solution $S(F)$ to the CDRFR problem. If no feasible solution $S(F)$ can be computed for a route-set F then the move is not allowed, which occurs e.g. when changing a train route leads to a deadlock situation. Whenever a better schedule is found, the new route is set as default route and the search is repeated until a time limit of computation is reached.

Since the alternative graph model assumes deterministic blocking and waiting times, train trajectories are only feasible in absence of disturbances. In fact, the impact of deceleration and acceleration for hindered trains is not taken into account in the formulation of the CDR problem. The *train speed coordination* module is needed to ascertain whether a safe space headway between trains is respected and to update the speed profiles of trains according to the actual signal aspects. Speed coordination among consecutive trains is achieved by iteratively adapting the speed trajectory of trains and by updating the blocking and waiting times in the corresponding alternative graph, such that the resulting train schedules comply with the constraints of the signalling and safety system (D'Ariano, Pranzo *et al.* 2007).

3.5. Interaction between decision support tool and dispatcher

The ROMA dispatching support tool computes train routes, orders and advisory speeds. However, before the implementation of the proposed actions the dispatcher receives a detailed forecast of the future traffic flow and train delays, and should recognise and acknowledge the changes in the timetable. The real-time interaction between the decision support tool and the dispatcher needs to be simple, clear and fast since the dispatcher has to decide which solution should be implemented among a set of possible solutions. In the case of small perturbations, a dispatching solution could be presented in terms of the only relevant actions that differ from the scheduled ones, avoiding unnecessary corrections of the paths of on-time trains.

In other more disturbed traffic conditions with multiple delayed trains, several timetable modifications may be needed to recover from delays and infeasible traffic situations. In this case, the dispatcher needs to be informed of the reasons for a particular modification of advisory speeds, arrival/departure times, and train routes and sequences.

Figure 4 depicts an interface between ROMA and the dispatcher. Blocking time graphs are useful to represent, visually, the future evolution of the train traffic. The main limitation is that only one corridor at a time can be investigated in sufficient level of detail (for example, Figure 4 shows four virtual overlaps of blocking times for trains C15931 and A3031 that are running on the same line) while the visualization of route conflicts and route booking actions is quite complex in station areas. So, specific points of interest, where a train would experience knock-on effects, and the suggested rescheduling solutions are highlighted with a sufficient amount of detailed information (for example, location of potential conflicts, suggested orders and routes, expected delay, etc) to let the dispatcher understand the proposed dispatching actions and their impact on railway operations.

3.6. Real-time dynamic setting of the tool

This subsection discusses how our decision support tool could be used in a dynamic setting with operations. We observe that the applicability of the proposed dispatching

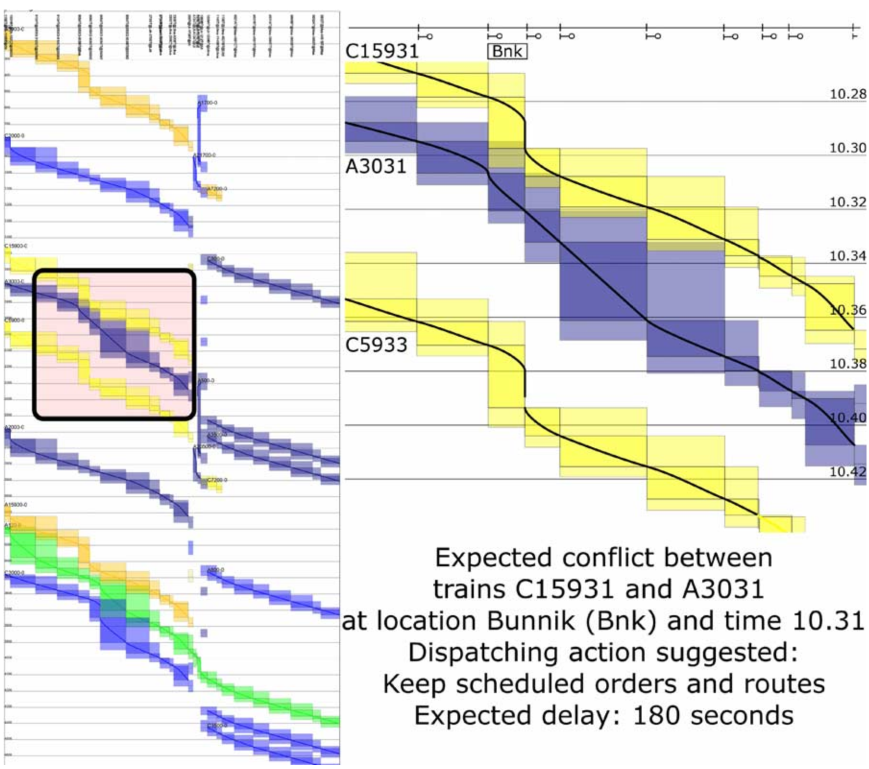

Figure 4. An interface based on the blocking time plot of proposed control actions.

solutions is influenced not only by the accuracy of the dispatching support tool in representing the actual and future traffic situations, but also by the reliability of the control actions (that is, the ability to forecast their outcome). For these reasons, we next focus our attention on the following time intervals necessary to close the loop with operations (see Figure 5):

*t*0: time at which the current position and speed of each train are updated; *t*1: time at which ROMA starts computing a dispatching solution;
*t*2: time at which ROMA returns a dispatching solution;
*t*3: time at which the ROMA solution is accepted by the dispatcher;
*t*4: time at which the dispatching actions are implemented;
*t*0': next time *t*0.

The above times can be interrelated as follows. The time between *t*0 and *t*1 is needed to record actual train positions and speeds and communicate these to the traffic control centre. At time *t*0, we make use of the best prediction of all train positions and speeds obtained before that time. This can be transmitted via GSM-R on-board units of trains or by interpolating real-time train detection and signalling data. The time step between *t*1 and *t*2 is needed by ROMA to reconstruct the current traffic conditions, simulate the future evolution, detect possible conflicts and provide solutions. The time between *t*2 and *t*3 is used by the dispatcher to check the dispatching solutions given by ROMA and, eventually, to compare those with other dispatching options. The time between *t*3 and *t*4 considers the delay due to the transmission of the control actions as well as the time needed to implement the dispatching actions in practice, such as switching signals and setting up routes.

We now introduce the starting time Π of the traffic prediction, the time horizon length T and the time τ to compute a dispatching solution, and describe how to set them up with respect to the other times. ROMA provides control actions in the time interval between Π and Π + T. It is assumed that no relevant unplanned action will occur from *t*0 to Π and the traffic flow in the network is determined exactly. However, an error between the simulated traffic and the actual traffic always exists due to the dynamic nature of the real-time operation and the inherent inertia of the dispatching process. If the dispatching support tool is not able to model the current status of the network with sufficient precision, the control actions suggested by the tool after Π might be sub-optimal, obsolete or even infeasible. Moreover, the suggested actions would be physically applicable only if Π is larger than *t*4. From a practical point of view, the longer the interval *t*4–*t*0 is, the larger is the error between the simulated and actual network status. To limit this error, the dispatching tool must achieve a sufficient precision in simulating the status of the system within the required computation and communication times.

Since the available time to compute a dispatching solution is rather limited, the time horizon of traffic prediction T may be also limited. So, the computation of

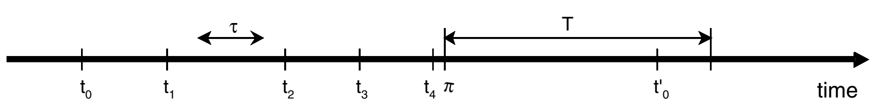

Figure 5. Time intervals to apply the dispatching support tool during operations.

downstream possible conflicts too far away in time should be avoided since the prediction uncertainty increases. However, the computation of optimal dispatching solutions requires global information regarding train traffic to be taken into account. To this end, the dispatching tool should be able to manage traffic in a large interconnected area with dense traffic.

Another important issue to study is the frequency of rescheduling. In fact, $t0$0 is a variable time since the dispatching support tool may be run periodically or event-based (for discussions on rescheduling under uncertainty see Vieira *et al.* 2003, Aytug *et al.* 2005, Hozak and Hill 2008). In general, important parameters for choosing the frequency of rescheduling are the traffic prediction horizon T, the accuracy of the simulation procedure and the robustness against random variations in the dynamic traffic flow evolution. The more often the dispatching support tool is used, the less is the divergence between the train operations simulated by the tool and the real traffic conditions. On the other hand, the dispatching support tool could be adopted when a particular condition triggers, i.e. when the error for an observed variable exceeds a given threshold or when an unplanned disruption has occurred and the current solution is infeasible.

A deeper analysis on the choice of the relevant times $t0; t1; t2; t3; t4$ and their link with the time horizon T and the starting time of traffic prediction Π would be necessary. To this end, experimental verification must still prove the applicability of the support tool in a real-world setting and the effectiveness and promptness of advanced dispatching measures compared to the current dispatching process

4. Test case

In this section we present a comprehensive computational study to evaluate the potential of employing ROMA as support tool for traffic management in the dispatching area around Utrecht Central Station. We next describe the instances and present the obtained results.

4.1. Description of the instances

The topology of the railway network around the main station of Utrecht is similar to a star with five main directions crossing each other (Figure 6). The main lines from the north and south of the Netherlands are connected to the lines to the west and east. The network considered is delimited by the following stations: Utrecht Overvecht on the line to Amersfoort, Driebergen-Zeist on the line to Arnhem, Culemborg on the line towards Den Bosch, Vleuten on the line to Rotterdam and The Hague plus Maarssen on the line towards Amsterdam. In total, the diameter of the area is around 20 km.

Utrecht Central Station is the most complex station area in the Netherlands and the interlocking area of the main station has around 200 block sections and more than 100 switches, leading to a large amount of inbound and outbound routes. In total, the considered area includes more than 600 block sections. In order to speed up the dispatching process, the whole network is transformed into around 200 aggregated blocks, with a total amount of around 250 virtual machines necessary to detect the incompatibilities between aggregated blocks.

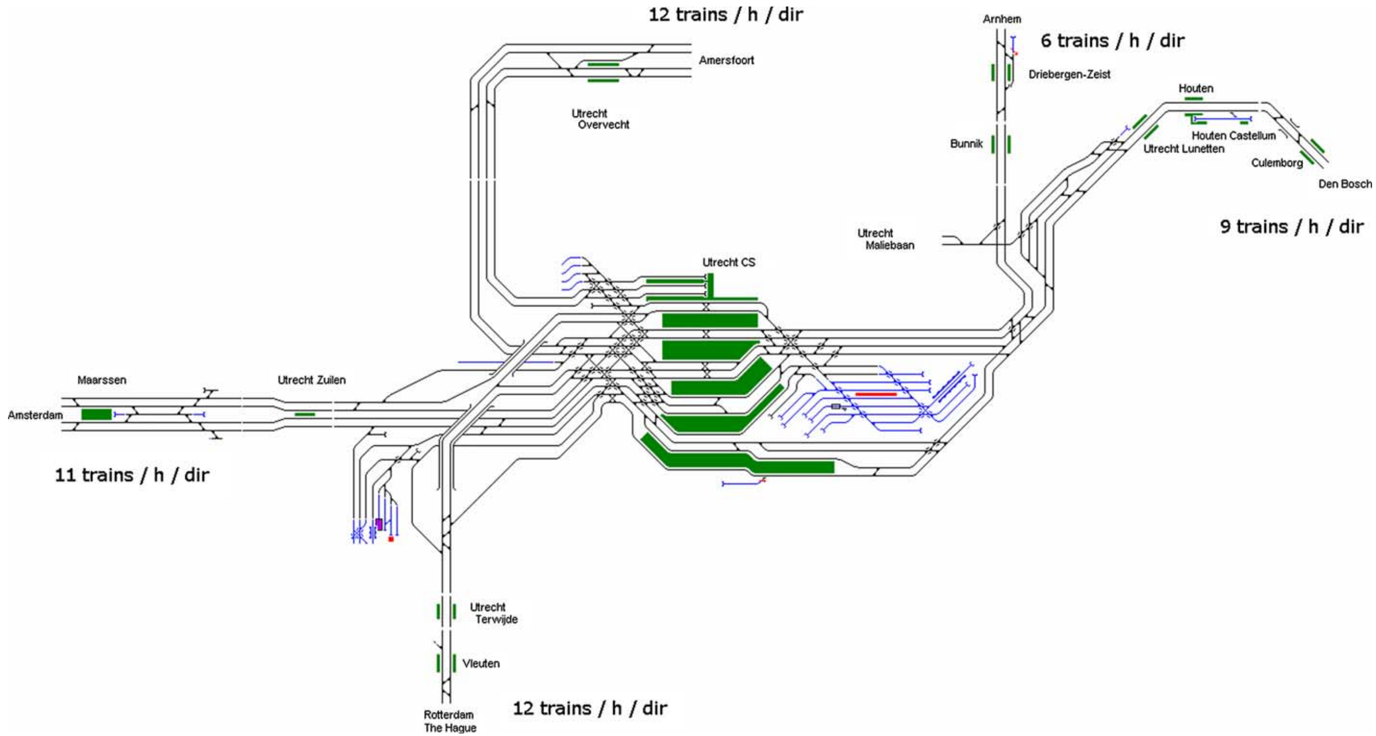

Figure 6. Utrecht Central Station showing scheduled hourly traffic per direction.

Utrecht Central Station provides 20 platform tracks, most of which may be reserved for two trains at a time, for instance when coupling or splitting. Most of the platform tracks are used by through traffic, that is, trains running in the opposite direction, even if some trains change direction after their stop at a platform. There are also three dead-end platforms.

For the computational experiments, we use the 2008 timetable that is periodic with a cycle length of one hour. The trains are mostly for passenger services, operated by NS (Nederlandse Spoorwegen), except for a few freight trains. The timetable schedules up to 80 trains in a peak hour and provides connections between passenger services, coupling and splitting of rolling stock for intercity and local services coming from/going to Rotterdam, the Hague or Amersfoort, as well as re-use of rolling stock for commuter services towards Utrecht Overvecht and Culemborg. The total amount of travellers at Utrecht Central Station is around 150,000 per day.

In order to limit the number of routings, we only consider two common routes for each train. The total number of routes is 160. Most of the alternative routes consider adjacent platforms since passengers' discomfort and extra time for transfer of passengers from one platform to another are to be limited.

We generate timetable perturbations that cause initial delays in the network. To this end, we analyse a total of more than 33,000 train events (arrivals, departures, dwell processes and passing times) that have been recorded at Utrecht Central Station in April 2008 by ProRail. We consider a statistical fitting procedure (Yuan 2006) to set the three parameters of the Weibull distributions, which are adopted to characterise the delays of different trains and the variation in the dwell time process (Table 1).

Based on the calculated statistical parameters, we generate 40 random instances with disturbed entrance times and 15 random instances with dwell time extensions at Utrecht Central Station. The average disturbance is around 30 seconds per train while the maximum disturbance is 685 seconds. These instances represent an average level of perturbed operations over a whole month. We combine each instance with the three infrastructure scenarios: (1) all infrastructure available; (2) platform 2 in Utrecht unavailable for traffic; (3) switch 1423A in Utrecht unavailable for traffic (so that platform 15 is blocked). In the given scenarios, respectively, 0%, 3% and 6% of the routes loaded into the dispatching support tool are unavailable, while 0%, 2% and 5% of the running trains have to be rerouted through the interlocking area, being still able to perform their scheduled trip with one of the available alternative routes. In total 1800 disturbance instances are tested with one hour of time horizon of traffic

Table 1. Parameters of the Weibull distributions.

Train type	Disturbed entrance times			Dwell time extensions		
	Scale	Shape	Shift	Scale	Shape	Shift
Intercity	395	2.2	−315	252	2.1	4
Stoptrein	227	2.4	−198	252	2.1	4
Sprinter	235	3.0	−186	252	2.1	4
International	560	1.4	−205	252	2.1	4
Freight	1099	2.6	−885	252	2.1	4

prediction (T) and with one minute of maximum time to compute a dispatching solution (τ).

5. Computational results

The main aim of the study is to assess how much train delays could be minimised by choosing suitable dispatching actions. We compare the dispatching solutions obtained by simple rules and advanced algorithms. In particular, we test the following dispatching rules: the one that keeps the timetable order fixed, the local heuristic based on the FCFS rule and the ARI-like procedure. To achieve optimal traffic control actions by ROMA, we use the following two configurations: ROMA-reordering that adopts the BB algorithm to find near-optimal solutions to the CDRFR problem, and ROMA-rerouting that improves the ROMA-reordering solutions by the TS algorithm (that is, it computes better solutions to the CDR problem).

The average behaviour of the proposed dispatching procedures is shown in Table 2 in terms of computation time (in seconds), maximum and average consecutive delays (in seconds), average total delays (in seconds) and percentage of on-time trains (named 'punctuality', since in the Netherlands a train is considered late if this has a total delay larger than three minutes). Each row of the table corresponds to the average results over the 1800 instances.

We now discuss the performance of the studied dispatching procedures on the basis of the results of Table 2. The solution found by keeping the train orders and routes as scheduled in the timetable has a poor quality. This is due to long waiting times for solving the route conflicts, causing a domino effect of delay propagation. The ARI-like procedure performs considerably better than the original schedule. However, the simple heuristic FCFS outperforms ARI. The use of the BB algorithm enables ROMA-reordering to achieve the best performance in terms of maximum consecutive delays. Specifically, the BB algorithm found the optimal solution to the CDRFR problem in 89% of the cases within the given time limit of computation.

When comparing the performance of ROMA-reordering to ARI, the average total delay experienced is reduced by around 18%, while a bigger reduction up to 48% is achieved in terms of average consecutive delays. With regards to the ROMA-rerouting procedure, we only note a small improvement compared to ROMA-reordering since few rerouting alternatives are explored in the given computation time (on average 54 CDRFR solutions are computed for each perturbed situation).

Table 2. Average behaviour of the analysed dispatching procedures.

Dispatching procedure	Comp. time (s)	Max. cons. delay (s)	Avg. cons. delay (s)	Avg. total delay (s)	Punctuality (%)
Timetable	5.8	622	50.1	94.5	83
ARI-like	5.7	446	28.2	74.3	84
FCFS	4.4	397	19	65.5	89
ROMA-reordering	5.7	296	15.1	61.2	91
ROMA-rerouting	52.3	299	14.6	60.8	92

On the other hand, significant results are obtained by ROMA-rerouting. Figure 7 shows the remarkable improvement of ROMA-rerouting versus FCFS in terms of average consecutive delays. The curve shows that ROMA-rerouting is, on average, around 20% better than FCFS, independently from the given initial delay.

Another important factor to consider is the number of reordering and rerouting actions needed to implement the dispatching solutions. We next compare the advanced procedures with the timetable solution. ROMA-rerouting changes around 2% of the train orders and 4% of the train routes. The other procedures also change at most 2% of the train orders. The limited number of modifications is due to the large number of instances with relatively small initial delay.

6. Conclusions

This paper has described a laboratory train dispatching support tool that has been developed for the management of complicated and densely occupied station areas, that is, many trains per hour dwelling at a large set of platform tracks. We used automatic scripts to convert the railway data (infrastructure, timetable and rolling stock information) from existing data formats supplied by the Dutch network infrastructure manager to the ROMA tool format. We also discussed the real-time use of the dispatching support tool, and generalised the applicability of earlier developed models and algorithms to manage distributed railway traffic in complex interlocking areas.

In order to assess the performance of the proposed algorithms, we proposed a comprehensive set of experiments based on statistical distributions of train delays fitted to the realization data of a monthly period in the area of Utrecht Central

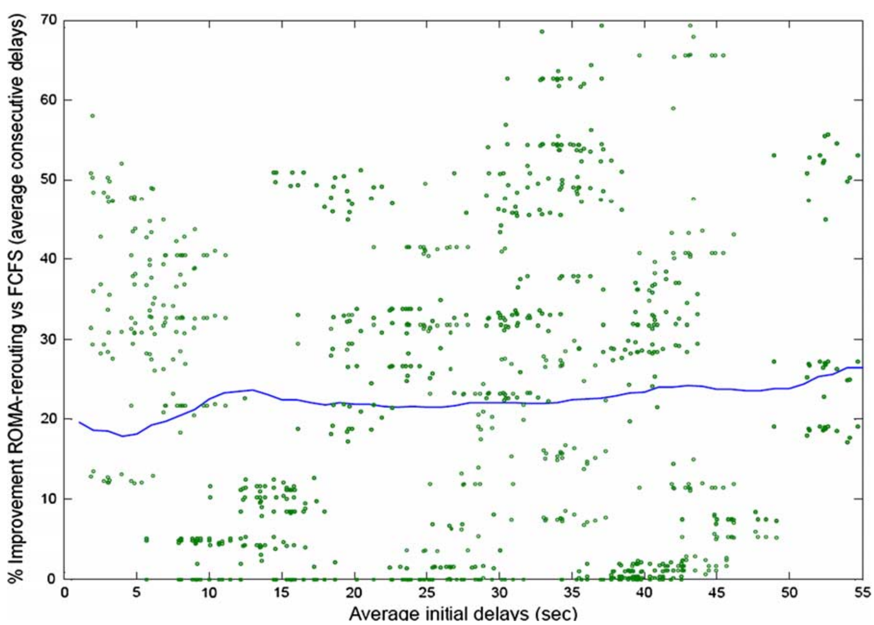

Figure 7. Scatter diagram improvement of ROMA-rerouting over FCFS, in terms of average consecutive delays, as a function of tested average initial delays.

Station. A large set of disturbances was generated and multiple control actions were required to restore feasibility of train operations and minimise the delay propagation. For the given set of instances, we investigated the computational effort and efficiency of advanced reordering and rerouting algorithms compared to simple dispatching procedures taken from traffic management practice. The tested instances aimed at simulating the average behaviour in case of perturbed traffic for one month of operations. The reordering solutions computed by ROMA were significantly better than the ones obtained by keeping the scheduled orders, by the ARI-like procedure or by the simple FCFS heuristic. The benefit of rerouting traffic has not yet been fully explored as the complexity of the scheduling problem still limits the computation time given to the rerouting task and only a limited number of alternative routes have been considered.

Future research should focus on the implementation of a closed-loop traffic monitoring and control system under the current safety regulations, and on studying the full benefits of dispatching trains in larger networks and for heavily disturbed operations.

Acknowledgements

We thank the Dutch infrastructure manager ProRail for providing the instances and D. Pacciarelli for fruitful discussions. This work is partially supported by the research program TRANSUMO 'Reliable Transport Chains' and by the Italian Ministry of Research, Grant number RBIP06BZW8, project FIRB 'Advanced tracking system in intermodal freight transportation'.

References

Aytug, H., *et al.*, 2005. Executing production schedules in the face of uncertainties: a review and some future directions. *European Journal of Operational Research*, 161 (1), 86–110.

Bailey, C., 1995. *European railway signalling*. London: A & C Black.

Berends, N. and Ouburg, N., 2005. *Beschrijving ARI-functionaliteit* (in Dutch). Utrecht: ProRail Internal Specification, Technical Report 20.

Caimi, G., *et al.*, 2009. Design of a railway scheduling model for dense services. *Networks and Spatial Economics*, 9 (1), 25–46.

Carey, M. and Crawford, I., 2007. Scheduling trains on a network of busy complex stations. *Transportation Research, Part B*, 41 (2), 159–178.

Corman, F., Goverde, R.M.P., and D'Ariano, A. 2009. Rescheduling dense train traffic over complex station interlocking areas. *In:* R.K. Ahuja, R. Moehring, and C. Zaroliagis, eds. *Robust and online large-scale optimization, lecture notes in computer science*, vol. 5868. Berlin: Springer-Verlag, 369–386.

Corman, F., *et al.*, 2010. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research, Part B*, 44 (1), 175–192.

Daamen, W., Goverde, R.M.P., and Hansen, I.A., 2009. Non-discriminatory automatic and distinct registration of primary and secondary train delays. *Networks and Spatial Economics*, 9 (1), 47–61.

D'Ariano, A., 2008. *Improving real-time train dispatching: models, algorithms and applications.* Thesis (PhD). Delft University of Technology.

D'Ariano, A., Pacciarelli, D., and Pranzo, M., 2007. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183 (2), 643–657.

D'Ariano, A., Pranzo, M., and Hansen, I.A., 2007. Conflict resolution and train speed coordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8 (2), 208–222.

D'Ariano, A. and Pranzo, M, 2008. 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 (1), 63–84.

D'Ariano, A., *et al*., 2008. Reordering and local rerouting strategies to manage train traffic in real-time. *Transportation Science*, 42 (4), 405–419.

Hansen, I.A. and Pachl, J., 2008. *Railway timetable and traffic: analysis, modelling and simulation*. Hamburg: Eurailpress.

Hozak, K. and Hill, J.A. Issues and opportunities regarding replanning and rescheduling frequencies. *International Journal of Production Economics*, 47 (18), 4955–4970.

Jackson, J.R., 1955. *Scheduling a production line to minimize maximum tardiness.* University of California, LA: Management Science Research Project, Technical Report 43.

Jacobs, J., 2004. Reducing delays by means of computer-aided 'on-the-spot' rescheduling. *In:* J. Allan, C.A. Brebbia, R.J. Hill, G. Sciutto, and S. Sone, eds. *Computers in railways IX*. Southampton: WIT Press, 603–612.

Lüthi, M., Medeossi, G., and Nash, A., 2007. Evaluation of an Integrated Real-Time Rescheduling and Train Control System for Heavily Used Areas. *In:* I.A. Hansen, A. Radtke, J. Pachl, and E. Wendler, eds. *Proceedings of the 2nd international seminar on railway operations modelling and analysis* [CD-ROM], 28–30 March 2007, Hannover, 1–14.

Mascis, A. and Pacciarelli, D., 2002. Job shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143, 498–517.

Mazzarello, M. and Ottaviani, E., 2007. A traffic management system for real-time traffic optimisation in railways. *Transportation Research, Part B*, 41 (2), 246–274.

Pinedo, M., 1995. *Scheduling: theory, algorithms and system*. Upper Saddle River, NJ: Prentice Hall.

Rodriguez, J., 2007. A study of the use of state resources in a constraint-based model for routing and scheduling trains. *In:* I.A. Hansen, A. Radtke, J. Pachl, and E. Wendler, eds. *Proceedings of the 2nd international seminar on railway operations modelling and analysis* [CD-ROM], 28–30 March 2007, Hannover, 1–14.

Schrijver, A. and Steenbeek, A., 1994. *Dienstregelingontwikkeling voor railned: rapport CADANS 1.0* (In Dutch). Amsterdam: Centrum voorWiskunde en Informatica, Technical report.

Serafini, P. and Ukovich, W., 1989. A mathematical model for periodic event scheduling problems. *SIAM Journal on Discrete Mathematics*, 2 (4), 550–581.

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

Vieira, G.H., Herrmann, J.W., and Lin, E., 2003. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling*, 6 (1), 39–62.

Wegele, S., Corman, F., and D'Ariano, A., 2008. Comparing the effectiveness of two realtime train rescheduling systems in case of perturbed traffic conditions. *In:* J. Allan, E. Arias, C.A. Brebbia, C. Goodman, A.F. Rumsey, G. Sciutto, and N. Tomii, eds. *Computers in railways XI*. Southampton: WIT Press, 535–544.

Yuan, J., 2006. *Stochastic modelling of train delays and delay propagation in stations.* Thesis (PhD). Delft University of Technology.

Zwaneveld, P.J., *et al*., 1996. Routing trains through railway stations: model formulation and algorithms. *Transportation Science*, 30, 181–194.