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

F. Corman[1], A. D'Ariano[2,1], M. Pranzo[3], I.A. Hansen[1]
(1) Department of Transport and Planning, Delft University of Technology,
Stevinweg 1, 2600 GA Delft, the Netherlands,
f.corman@tudelft.nl, a.dariano@tudelft.nl, i.a.hansen@tudelft.nl
(2) Dipartimento di Informatica e Automazione, Università degli Studi Roma Tre,
via della Vasca Navale 79, 00146 Roma, Italy,
a.dariano@dia.uniroma3.it
(3) Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Siena,
via Roma 56, 53100 Siena, Italy
pranzo@dii.unisi.it

Abstract
Railway traffic experiences disturbances during operations while dispatchers need actions to restore feasibility and limit spreading of delays through the network. To help the dispatcher in such task, the dispatching support tool ROMA (Railway traffic Optimization by Means of Alternative graphs) has been designed and implemented. We report on enhancements to the underlaying train dispatching model as well as to the solving algorithms studied in order to tackle the increased complexity of busy stations with multiple conflicting paths and high service frequency. The system is compared with straightforward rules and the current approach in the Netherlands. Extensive computational studies based on accepted statistical distributions of delays assess the effectiveness of the ROMA tool.

Keywords
Dispatching support, rescheduling, rerouting, alternative graph, incompatibility graph

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 characterized 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 analyzed in depth under economical and operational constraints. In real-time, unforeseen events may disrupt the timetable and thus the resolution of train 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 modeling the evolution of train traffic sufficiently in detail and considering the actual state of the network, both the dynamic behavior of circulating trains and the dispatching measures used to control traffic. Hence, the precise

delay propagation cannot be predicted by dispatchers, especially in 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 quickly enough. For these reasons, there is a need for developing more sophisticated and efficient decision support tools to forecast the network delay propagation for individual dispatching measures.

In this context, we developed a laboratory dispatching support tool, called ROMA (see D'Ariano [7]). This tool is designed to support dispatchers in the 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 co-ordinate 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 characterized by a unique combination of the blocking time theory for the recognition of timetable conflicts in case of disturbances and a general discrete optimization model, based on the alternative graph formulation, 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 dynamic updating of 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 formulation enables an accurate computation of train trajectories and routes in complicated and busy station areas with several intersecting paths to lines and platforms. Specifically, we have developed the additional innovative features:

- 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 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 can also be found e.g. in [14, 16]).

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 organized 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 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, as follows. Off-line timetabling usually relies on macroscopic models, while microscopic approaches might also be used. The latter approaches are required 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 [19], 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 (see e.g. [18]), 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 (see e.g. [24]), manages the routing of trains in complex station areas. The model takes into account incompatibility between routes according to predefined safety constraints and builds an incompatibility graph in which a maximum weight node packing corresponds to a feasible routing solution, while neglecting the impact of signaling and train length on blocking times.

Carey and Crawford [4] 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 conflicts along the tracks by selecting the least immediate delay cost.

Caimi et al. [3] 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 the trains 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 (see e.g. [22]). The signaling system with actual signal aspects needs to be modeled 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, specially 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 suboptimal. Among the recent contributions, Jacobs [13] 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 [17] 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 conflicts between trains. Synchronization constraints are used to keep train running with sufficient headway distances, even in case of yellow or red 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 [20] 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 minimize 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 [7] 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, specially in case of strong 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. For instance, realistic separation rules based on blocking times 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 should therefore 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 *operation* and a sequence of operations to be traversed by a train is called *route*. At any time a route is passable if all its block sections are available, i.e., there are no blocked tracks. The timing of a route specifies the starting time t_i of each operation in the route. Each operation requires a traveling 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 behavior according to signaling system constraints (e.g. in case of a red signal the train must brake to a complete stop and then re-accelerate).

According to safety regulations, no more than one train at any time is allowed in a block section; the signaling system and the Automatic Train Protection (ATP) system are used to restrict the availability of one track segment (one or more consecutive block sections) in order to ensure a safe headway between successive trains and to generate an automatic brake in case of incidents or human errors. This headway translates to a *setup time*, required to modify the signaling aspect and to change the switching layout 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 and the space between signals. 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 their composition. In the Netherlands, train services may be coupled and uncoupled during operations, by combining the rolling stock of a service with another train or by splitting a rolling stock into two distinct services. In these cases, 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.

When disturbances occur during operations, perturbations to the timetable are experienced that may lead to deadlocks or consecutive delays. Example of such perturbations may include infrastructure breakdowns, delays of train entering the dispatching area from the previous one, speed limitations due to technical failure or extended dwell times at stops.

Following the notation of [7], the *total delay* is defined as the difference between the actual arrival time and the planned time at a relevant point. The total delay can be interpreted as made up of 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 conflict or timetable reference point. A *consecutive (or secondary) delay* is caused by the hinder between trains running in the network and in principle could be minimized by pro-actively managing railway traffic. A *conflict* is when two or more trains claim the same available block section simultaneously; in that case the movement authority is given to only one of the trains involved, and the others are forced by the signaling system to decrease their speed or even to reach a complete stop on the open track or within the interlocking area.

The real-time train dispatching problem is to find a schedule for the trains in a given area and a given time horizon during operations, providing a solution that is compatible with the actual traffic condition and infrastructure constraints. This problem can be defined as follows: given a railway network, a set of train routes and passing/stopping times at relevant points in the network, the position and speed of each train at starting time, find a conflict-free and deadlock-free schedule with minimum required safe headway distances between trains, such that the selected train routes are not blocked, the speed profiles are feasible, no train leaves a relevant point before its scheduled departure time, additional constraints for rolling stock and connected services are respected, and trains arrive at the relevant points with the smallest consecutive delay.

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, e.g. 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), e.g. an empty platform can be used instead of causing a delay while waiting for a still occupied track.

This section introduces a model for the real-time process of changing dwell times as well as train orders and routes in case of conflicts while avoiding deadlocks. This is called Conflict Detection and Resolution (CDR) and can be partitioned into two subproblems: *(i)* a scheduling problem, for which the starting time of each operation is determined, and *(ii)* 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, 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 signaling 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 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 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 [15] introduce alternative graphs to model variants of job shop scheduling problems. An alternative graph is a triple $\mathcal{G} = (N, F, A)$ 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 pair is chosen) and consistent (there are no positive length cycles).

In the alternative graph formulation, the CDRFR problem is to find a feasible starting time t_i to each operation o_i (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 to model more general scheduling situations [15]. To summarize, a conflict-free and deadlock-free schedule for the CDRFR problem is associated with a complete consistent selection in the alternative graph $\mathcal{G}(S)$.

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 F_i the solution to the associated CDRFR problem $S(F_i)$ is a complete consistent selection in $A(F_i)$. 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 modeled in the alternative graph formulation by using blocking time theory (see e.g. [10]). 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 minimum headway distance between two trains 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.

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 oper-

ation. 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 that a train may traverse one after each other. The sequence of corresponding operations $o_i, \ldots, o_m$ can be modeled as an *aggregated block operation*. 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 the setup time associated with the last block section of the aggregate block.

When dealing with complicated interlocking areas the use of aggregated block operations allows to reduce the size of the problem. 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 rail 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 actual 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.

A similar aggregation procedure is performed when modeling 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 conflicts during operations, i.e., we have to compute the speed trajectories of all the trains involved in the conflicts and their actual blocking times.

An example of aggregated blocks is described in Figure 1. We refer to an interlocking area connecting three platform tracks (1, 2, 3) to two open tracks (HA and HN). In Figure 1(a), there are nine block sections that are grouped into 6 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 1(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 eighteen 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>).

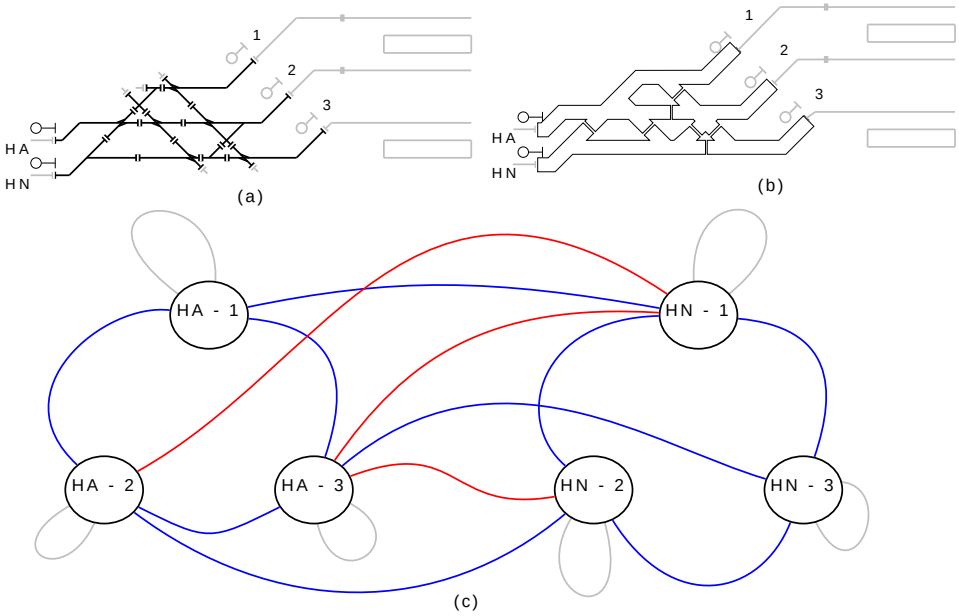

Figure 1: Block section and aggregated blocks: (a) block sections; (b) virtual machines associated to the aggregated blocks; (c) incompatibility graph between the aggregated blocks.

As shown in the proposed example, the incompatibility graph is able to model incompatibilities between different aggregated blocks that are due to the track topology of complex interlocking areas. A more compact representation relies on the introduction of *virtual machines* and on their association to aggregated blocks. 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 complex 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, i.e., 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 incompatibilites between the aggregated blocks. The virtual machines are then introduced in the alternative graph formulation of the CDR problem, allowing to translate the characteristics of the non-aggregated model into the aggregated one.

In the illustrative example of Figure 1, 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 1(b), the virtual machines are represented as composition of block sections.

3.3 Decision support tool

The ROMA software is designed as a decision support tool to assist traffic controllers in the evaluation of real-time dispatching solutions. ROMA is implemented in C++ language 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 2. So far, the tool is a laboratory version and the field layer is only simulated. We next give a brief description of each module.

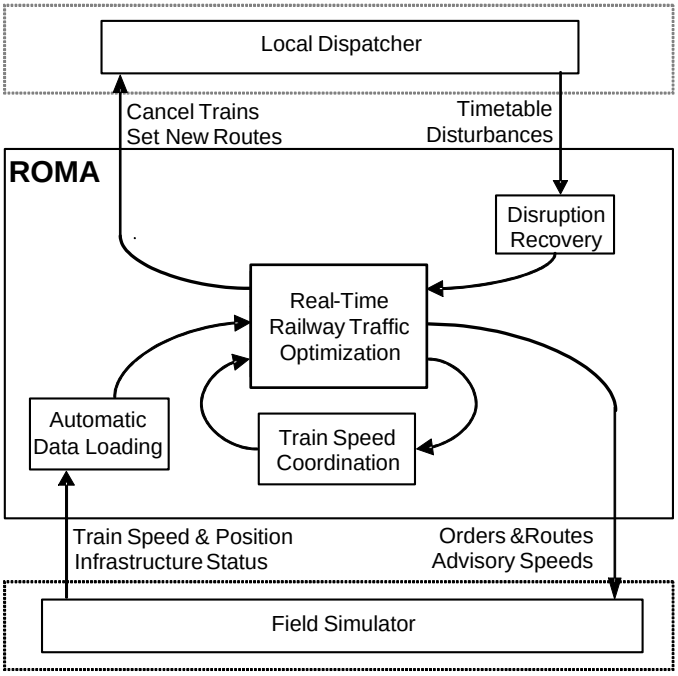

Figure 2: ROMA dispatching support tool architecture.

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.

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 center. A more reliable method may be to use an automatic data mining tool, such as TNV-Conflict [6], that is based on actual track occupancy data at specified timetable points.

The `disruption recovery` module is in charge of providing a feasible route for each train when some track section is unavailable to traffic, e.g. due to infrastructure disruptions. A list of train routes is given to this module, that select 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 [7]).

Several heuristic methods can be used to solve the CDRFR problem. A straightforward method is to impose the order prescribed by the timetable, another one is to follow the simple rule 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 [2], is usually adopted as far 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 [8]. 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 Preemptive Schedule [12] 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 [5]. 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 of trains and the corresponding blocking times and verifying the compatibility by means of alternative graphs, such that the resulting train schedules comply with the constraints of the signaling and safety system [9].

3.4 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 recognize 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 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.

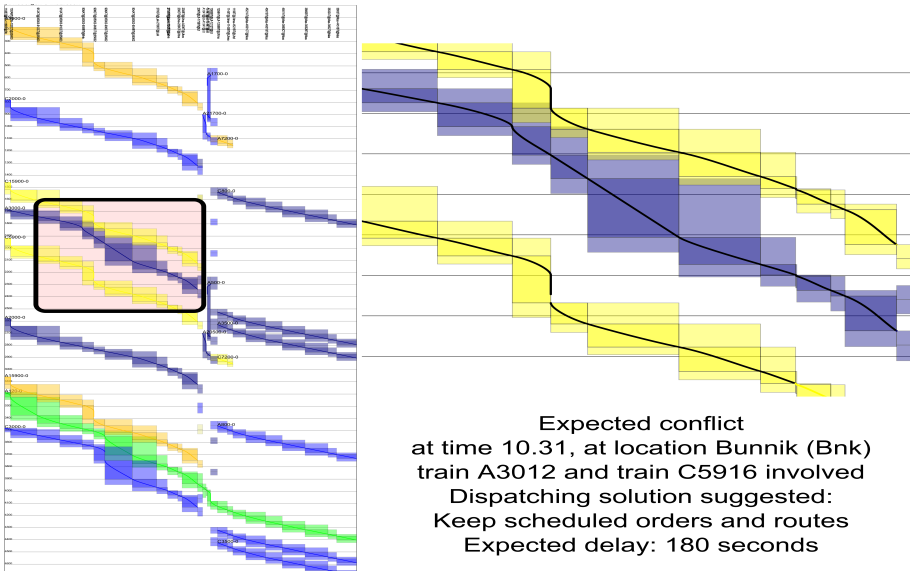

Figure 3: An interface based on the blocking time plot of the proposed control actions.

Figure 3 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 while the visualization of conflicts and route booking actions is quite complex in station areas. So, specific points of interest, where a train would experience hinder, and the suggested rescheduling solutions are highlighted with a sufficient amount of detailed information (e.g. location of conflicts, suggested orders and routes, experienced delay) to let the dispatcher understand the proposed dispatching actions and their impact on operations.

3.5 Discussion on real-time dynamic setting

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 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 (i.e., 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 4):

t_0 : time at which the current position and speed of each train are measured;
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.

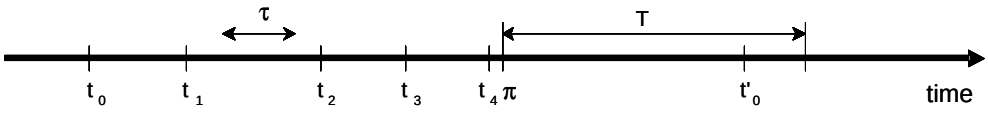

Figure 4: Time intervals to apply the dispatching support tool during operations.

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 center. This can be transmitted via GSM-R on-board units of trains or by interpolating real-time train detection and signaling 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 needed 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 up them 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 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 is the interval t_4 - t_0, 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 given 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 downstream conflicts too far away in time should be avoided since the prediction uncertainty would enlarge. However, the computation of optimal dispatching solutions requires to take into account global information regarding train traffic. To this end, the dispatching tool should be able to manage traffic in a large area with dense traffic.

Another important issue to study is the frequency of rescheduling. In fact, t'_0 is a variable time since the dispatching support tool could be run periodically or event-based (discussions on rescheduling under uncertainty can be found e.g. in [21, 1, 11]). 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 t_0, t_1, t_2, t_3, t_4 and their link with the time horizon T and the starting time of traffic prediction π is required. To this end, experimental verification would 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 5 main directions crossing each other (see Figure 5). The main lines relaying the North and South regions of the Netherlands are connected to the lines to the West and the 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 long.

This is the most complex station area in the Netherlands and the interlocking area of the main station has, alone, 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

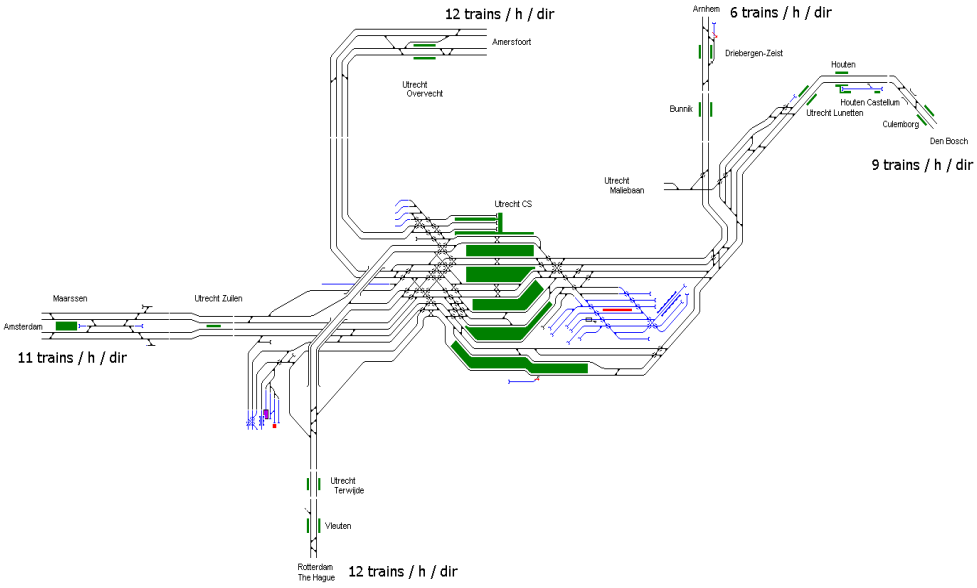

Figure 5: Utrecht Central Station in the center of the diagram; the scheduled hourly traffic per direction is given for each line of the dispatching area.

transformed into around 200 aggregated blocks, with a total amount of around 250 virtual machines necessary to detect the incompatibilities between aggregated blocks.

Utrecht Central Station provides 20 platform tracks, most of which might be reserved to two trains at a time, for instance when coupling or splitting. Most of the platform tracks are used by through traffic, i.e., 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 cyclic 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 travelers at Utrecht Central Station is around 150.000 per day.

In order to limit the number of routings, we only consider two routes for each train. So, the total number of routes is 160. Most of the alternative routes consider adjacent platforms since railway managers suggest to limit passengers' discomfort and avoid extra waiting time to move passengers from a platform to another one.

We generate timetable perturbations that cause initial delays in the network. To this end, we analyze a total amount of more than 33000 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 (as shown in [23]) to set the three parameters of the Weibull distributions, which are adopted to characterize the delays

of different trains and the variation in the dwell time process (see Table 1).

Train	Disturbed Entrance Times			Dwell Time Extensions		
Type	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

Table 1: Parameters of the Weibull distributions.

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. Precisely, the average disturbance is around 30 seconds per train while the maximum disturbance is 685 seconds. This instances represent an average level of perturbed operations over a whole month. We combined each instance with the three infrastructure scenarios: *(i)* all infrastructure available; *(ii)* platform 2 in Utrecht unavailable for traffic; *(iii)* switch 1423A in Utrecht unavailable for traffic (so that platform 15 results blocked). In the given scenarios, we have that, 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 prediction (T) and with one minute of maximum time to compute a dispatching solution (τ).

4.2 Computational results

The main aim of the study is to assess how much train delays could be minimized by choosing suitable dispatching actions. We compare the dispatching solutions obtained by simple rules and advanced algorithms. Precisely, we tested 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 (i.e., computes better solutions to the CDR problem).

The average behavior 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), 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 in the timetable has a poor quality. This is due to long waiting times for solving the train 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

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

Table 2: Average behavior of the analyzed dispatching procedures.

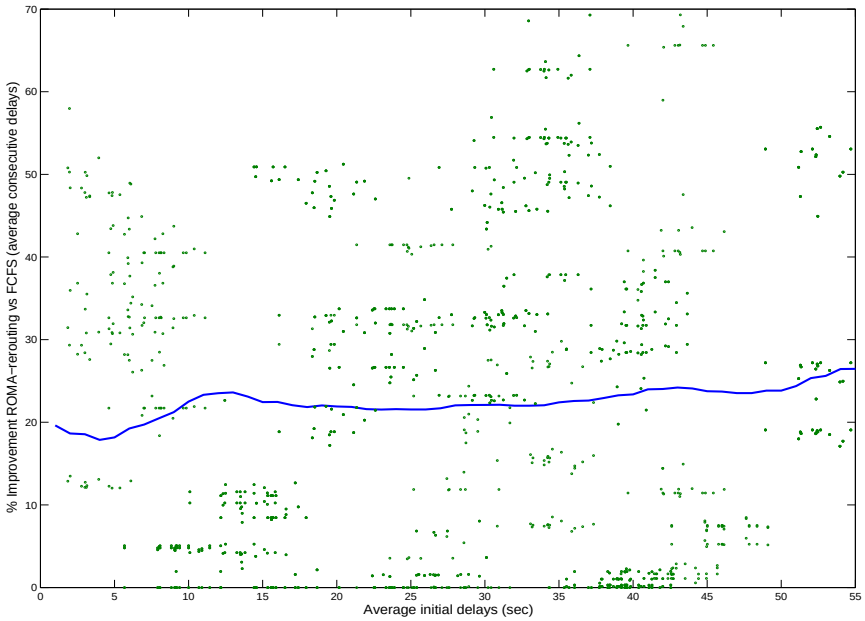

Figure 6: % Improvement of ROMA-rerouting over FCFS, in terms of average consecutive delays, for varying the average initial delays.

optimal solution to the CDRFR problem in 89% of the cases within the given time limit of computation. When comparing 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 regard 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 perturbated situation). On the other hand, significant results are obtained by ROMA-rerouting. In particular, Figure 6 shows the 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 relatively small initial disturbances.

5 Conclusions

This paper describes a laboratory train dispatching support tool that has been designed for the management of complicated and densely occupied station areas, i.e., many trains per hour dwelling at a large set of platform tracks. The novel features consist of implemented 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 discuss the real-time use of the dispatching support tool, and generalize the applicability of earlier developed models and algorithms to manage complex interlocking areas.

In order to assess the performance of the proposed algorithms, we propose 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 Station. A large set of disturbances is generated and multiple control actions are required to restore feasibility and minimize the delay propagation. For the given set of instances, we investigate the computational effort and efficiency of advanced reordering and rerouting algorithms compared to simple dispatching procedures taken from the traffic management practice. The tested instances aim at simulating the average behavior in case of perturbed traffic for one month of operations. The reordering solutions computed by ROMA are 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 is not yet 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 has been considered.

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

Acknowledgments

We thank the Dutch infrastructure manager ProRail (specially D. Middelkoop and L. Lodder) for providing the instances, D. Pacciarelli for fruitful discussions. This work is partially

supported by the Dutch foundation “Next Generation Infrastructures” and by the Italian Ministry of Research, Grant number RBIP06BZW8, project FIRB “Advanced tracking system in intermodal freight transportation”.

References

[1] H. Aytug, M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy. Executing production schedules in the face of uncertainties: a review and some future directions. *European Journal of Operational Research*, 161(1):86–110, 2005.

[2] N. Berends and N. Ouburg. Beschrijving ARI-functionaliteit. Technical Report 20, ProRail Internal Specification (in Dutch), Utrecht, the Netherlands, 2005.

[3] G. Caimi, D. Burkolter, T. Herrmann, F. Chudak, and M. Laumanns. Design of a new railway scheduling model for dense services. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

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

[5] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. A tabu search algorithm for rerouting trains during rail operations. Technical Report RT-DIA-127-2008, Dipartimento di Informatica e Automazione, Università degli Studi di Roma Tre, 2008.

[6] W. Daamen, R. M. P. Goverde, and I. A. Hansen. Non-Discriminatory Automatic and Distinct Registration of Primary and Secondary Train Delays. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[7] A. D'Ariano. *Improving Real-Time Train Dispatching: Models, Algorithms and Applications*. PhD Thesis, TRAIL Thesis Series T2008/6, The Netherlands, 2008.

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

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

[10] I. A. Hansen and J. Pachl. *Railway Timetable and Traffic: Analysis, Modelling and Simulation*. Eurailpress, 2008.

[11] K. Hozak and J.A. Hill. Issues and opportunities regarding replanning and rescheduling frequencies. *International Journal of Production Economics*, 2008. to appear.

[12] J. R. Jackson. Scheduling a production line to minimize maximum tardiness. Technical Report 43, University of California, Los Angeles, 1955. Management Science Research Project.

[13] J. Jacobs. 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, editors, *Computers in Railways IX*, pages 603–612. WIT Press, Southampton, UK, 2004.

[14] M. Lüthi, G. Medeossi, and A. Nash. 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, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis* . Hannover, Germany, 2007.

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

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

[17] J. Rodriguez. 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, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[18] A. Schrijver and A. Steenbeek. Dienstregelingontwikkeling voor Railned: Rapport CADANS 1.0. Technical report, Centrum voor Wiskunde en Informatica, Amsterdam, the Netherlands, 1994. In Dutch.

[19] P. Serafini and W. Ukovich. A Mathematical Model for Periodic Event Scheduling Problems. *SIAM Journal on Discrete Mathematics*, 2 (4):550–581, 1989.

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

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

[22] S. Wegele, F. Corman, and A. D'Ariano. Comparing the effectiveness of two real-time 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, editors, *Computers in Railways XI*, pages 535–544. WIT Press, Southampton, UK, 2008.

[23] J. Yuan. *Stochastic Modelling of Train Delays and Delay Propagation in Stations* . PhD Thesis, TRAIL Thesis Series T2006/6, The Netherlands, 2006.

[24] P. J. Zwaneveld, L. G. Kroon, E. H. Romeijn, M. Salomon, S. P. M. Dauzère-Pérès, S. Van Hoesel, and H. W. Ambergen. Routing Trains Through Railway Stations: Model Formulation and Algorithms. *Transportation Science*, 30:181–194, 1996.