2011 International Conference on Networking, Sensing and Control
Delft, the Netherlands, 11-13 April 2011

Dispatching trains during seriously disrupted traffic situations

Francesco Corman, Andrea D'Ariano, Ingo A. Hansen, Dario Pacciarelli, Marco Pranzo

Abstract— **Train services are planned in detail, defining several months in advance the train order and timing at crossings, junctions and platform tracks. A robust timetable is able to deal with minor perturbations even though in case of large delays or blocking of some tracks multiple timetable modifications are required in order to recover the feasibility of operations. Due to the interaction between trains, such exceptional situations result in consecutive delays to other trains in the network, making the railway system very vulnerable to disruptions. This paper investigates disruption handling strategies for large and busy railway networks. We consider seriously disturbed traffic conditions on double track railway lines where some block sections of one track are unavailable for traffic, e.g., due to a temporary track blockage, and disrupted train services are rerouted in other areas while still with the same origin and destination. Centralized approaches are used to solve the whole scheduling problem. Distributed approaches are also presented to manage effectively larger networks, in which a coordinator sets ad-hoc constraints between areas and delegates the scheduling decisions to local schedulers. Computational experiments on a large Dutch railway network, actually controlled by ten dispatchers, assess the performance of the centralized and distributed procedures, showing that both the procedures face increasing difficulty in finding feasible schedules in a short computation time for increasing time horizons of traffic prediction.**

I. INTRODUCTION

Railway traffic management of large and busy networks is typically based on detailed (off-line) train timetables. Careful scheduling is necessary since safety rules between trains impose that at most one train at a time can occupy a block section, i.e., a portion of railway delimited by signals. In accordance to such rules, the sequence of trains traversing each block section of the railway network has to be accurately defined so that feasible timetables can be developed, without any deadlock. A set of trains cause a deadlock when each train in the set claims a block section ahead which is not available, due either to a disruption (i.e., a permanent and serious reduction of railway capacity) or to the occupation/reservation for another train in the set.

During operations, a disruption handling phase is needed in order to limit propagation of train delays and to avoid any deadlock in the network. Operational traffic management is mainly directed towards recovering from disruptive events as quickly as possible. Failures of rolling stock, extended dwell times due to larger flow of passengers, speed restrictions due to infrastructure breakdowns or adverse weather, and incidents with humans or animals result in initial delays against which little or nothing can be done. Due to the interaction of trains along open tracks and in interlocking areas at stations, such initial delays propagate widely as consecutive delays to other trains in heavily loaded networks. This domino effect reduces seriously service quality.

The train operating companies adjust, in short time, the personnel and rolling stock plans to comply with the actual traffic situation, avoiding unbalance that may result in unavailability of train units or crews. The dispatchers then manage the train traffic under shortage of spare capacity and several delayed trains. In the railway practice this real-time task is often hierarchically organized into two decision levels, with local dispatchers controlling single areas with a local view of the traffic flow, and network dispatchers coordinating them, responsible for the traffic management over a railway network of several areas and with a global overview of the traffic flow. Experienced dispatchers have developed strategies allowing them to foresee simple route conflicts due to perturbed operations and to take compensatory control actions based on local information. Traffic management in the Netherlands is supported by automatic train regulation for routes in interlocking areas. However, during operations, dispatchers only reschedule the route setting plan when trains have a considerable delay and become active only when train traffic is already highly disturbed.

In case of disruptions, the dispatchers apply predetermined if-then measures in order to decide a suitable train path until the next station and keep the (network) traffic control center informed. The intention is to limit modifications of the rolling stock and crew plans as much as possible. However, for some disruptions some of the original services may be canceled or some trains can be rerouted. To this end in the Netherlands, dispatchers are provided with emergency timetables, defined as a solution to typical incidents and infrastructure breakdowns. As the variety of possible disruptions is very large, some of them are grouped together in classes, so that the amount of emergency timetable is limited (now there are around 1200 emergency timetables).

Advanced approaches have been recently proposed, dynamically updating crew and rolling stock plans during exceptional situations (see, e.g., [7]). The main focus of those tools is on the macroscopic level, able to deal in an approximate way at national level, rather than delivering detailed and precise solutions for the dispatching problem. Those approaches neglect capacity restraints at stations and along open tracks, even though disruptions result in heavy limitations of available capacity. To support dispatchers, a number of decision support systems based on mathematical tools have been proposed, so far, to compute good quality solutions (see, e.g., the literature review in [4] and other recent contributions in [1], [8], [10], [12], [13], [14], [15], [16]). Most of the existing approaches lack of a thorough computational assessment and limit the analysis to simple

978-1-4244-9573-3/11/$26.00 ©2011 IEEE

networks or simple perturbation patterns. The analyzed delay patterns are often quite specific, e.g., only one train is delayed or the problem is limited to a single junction or to a simple line. Moreover, the models used in the literature for the assessment are often simplified and do not capture entirely the consequences of delays and other disturbances, limited capacity, potential conflicts and deadlocks.

The main contribution of this paper is the assessment of advanced rescheduling approaches for managing railway traffic in a large network with dense traffic during exceptional situations such as network disruptions. To this aim, we further develop the research work of Corman et al. [2] that address the coordination problem for a complex and busy railway station divided into two dispatching areas. Our approach features a network coordinator that sets constraints at the border between areas and delegates scheduling decisions to local schedulers, driving the search of the local dispatchers toward globally feasible schedules. The model used to reschedule trains in each local dispatching area is a microscopic formulation based on the *alternative graph* of [9] and on the blocking time theory [6].

In the computational experiments, we analyze a Dutch railway network, spanning over ten dispatching areas, under different network decompositions. The disturbances include multiple delayed trains and a serious and permanent disruption in the network, which requires the rerouting of several trains and the management of complex traffic situations with the risk of deadlocks. For each perturbed traffic situation, we explore the limits of centralized and decomposed scheduling approaches to recover feasibility of operations on increasingly large time horizons of traffic prediction. The tested approaches range from simple and fast dispatching rules to more sophisticated decentralized scheduling procedures and a state-of-the-art exact algorithm [5].

II. SOLUTION APPROACHES AND PROCEDURES

Fast and efficient management of railway traffic during operations is necessary to cope with severe disturbances that may cause infeasibility. Basically, it consists of rescheduling the trains running in the railway network, in terms of their routes, orders and times. Railway movements and expected conflicts are predicted over a given time horizon, and the delay propagation is considered over a large area divided into multiple dispatching areas.

The timetable describes the movement of all trains in a given time horizon, specifying, for each train, 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 after its scheduled arrival time. We distinguish between timetable perturbations, that result in a set of train delays at the entrance of the network or at scheduled stops, and infrastructure disruptions that instead relate to blocked tracks, requiring stronger modifications to be recovered, such as new speeds profiles and/or train routes.

The microscopic formulation of the railway traffic management process requires modeling of railway networks at the level of block sections and signals. A block section is a track segment between two main signals and may host at most one train at a time. The passage of a train through a particular block section is called an operation. A route of a train is a sequence of operations to be performed in a dispatching area during a service. For every train, a release time in an area is given, that is the earliest starting time for the operation on the first block section in the area. Each operation requires a given running time which depends on the actual speed profile followed by the train while traversing the block section. The minimum time separations among the running trains translate into a minimum setup time between the exit of a train from a block section and the entrance of the subsequent train into the same block section.

A potential conflict arises when two or more trains claim the same block section simultaneously. In this situation a decision on the train ordering has to be taken and one of the trains involved has to change running, departure, passing times according to the constraints of the signaling system. This results in delays: the total delay is the difference between the calculated train arrival time and the scheduled time at a relevant point in the network, and is divided into two parts. An initial delay is caused by disturbances (e.g., blocked tracks, rolling stock or infrastructure failures, entrance delays) that cannot be recovered by rescheduling train movements.A consecutive delay is caused by the interaction between trains running in the network, i.e., trains that are held in front of a red signal or brake for a yellow signal, in order to solve a potential conflict.

The railway traffic management for disturbed operations is modeled as a job shop scheduling problem with additional constraints and formulated as an alternative graph [9] using the blocking time theory [6] to compute time separations between operations. The alternative graph has been used to model and solve train scheduling problems in several papers (see, e.g., [2], [3], [4], [5]). The main value of this formulation is the detailed and flexible representation of network topology and signaling system.

The alternative graph is a triple $\mathcal{G} = (N, F, A)$, where N is the set of nodes (including dummy nodes 0 and *), F is the set of fixed arcs and A is the set of pairs of alternative arcs. A selection S of arcs from A is obtained by choosing at most one arc from each pair in the corresponding set. The selection is complete if exactly one arc is chosen from each alternative pair. A problem solution is represented by an alternative graph solution $(N, F \cup S)$ in which S is a complete selection. The selection is feasible if the graph contains no positive length cycles. With this formulation, the problem of deciding whether a feasible schedule exists or not is NP-complete [9]. In order to evaluate a schedule, we use the maximum consecutive delay as performance indicator of a solution, that measures the propagation of the initial delays to the other trains.

In case of fixed block signaling, each block signal corresponds to a node in the alternative graph and the arcs between nodes are used to model the blocking times. The alternative graph represents the routes of all trains in a given control area along with their precedence constraints (minimum headways) and release times. A train route is modeled in the alternative graph with a job that is a chain of operations (nodes) and associated precedence constraints (fixed arcs). This formulation requires that a feasible route for each train is given and a fixed traversing time for each block section is known in advance, except for a possible additional waiting time between operations to allow for solving train conflicts. A train schedule therefore corresponds to the set of the starting time of each operation. Since a block section cannot host two trains at the same time, at each block section, a passing order between trains requiring it must be defined. This is modeled in the alternative graph by introducing a suitable pair of alternative arcs for each pair of trains traversing the block section. A deadlock-free and conflict-free schedule is obtained by selecting one of the two alternative arcs from each pair, in such a way that there is no positive length cycle in the graph.

Figure 1 (up) presents a traffic situation with two trains (named T_A and T_B) that run in a simple network with four block sections (named 1–4). The train scheduling problem is to solve the potential conflict on block section 2.

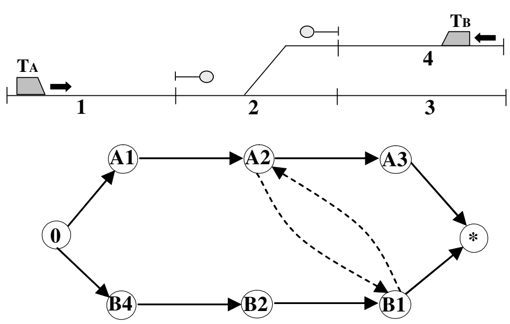

Fig. 1. A traffic situation (up) and an infeasible graph selection (down)

Figure 1 (down) shows a complete selection in the alternative graph formulation of the example of Figure 1 (up). The alternative (dotted) arc ($A2$, $B1$) imposes the ordering decision that T_A precedes T_B on block section 1 (this order is implied by the initial position of train A), while ($B1$,$A2$) corresponds to the precedence of T_B over T_A on block section 2. The proposed selection is clearly infeasible and this is detected in the graph by the positive length cycle between the alternative arcs ($A2$, $B1$) and ($B1$,$A2$). A conflict-free schedule is obtained by replacing ($B1$,$A2$) with its alternative arc ($A3$,$B2$) from the same pair (($B1$,$A2$),($A3$,$B2$)).

To solve the dispatching problem over a single area, the following centralized procedures are used. The First Come First Served (FCFS) dispatching rule, often used in the railway practice, solves train conflicts by assigning each block section to the first train that requires it. In other words, FCFS gives precedence to the train arriving first at each block section. Due to the myopic point of view, the procedure is straightforward, but may results easily in infeasible traffic situations. In this paper we consider an exact method, the branch and bound algorithm described in D'Ariano et al. [5], referred to in the following as BB. Specifically, we consider a larger set of starting heuristics, including the arc greedy heuristics presented in [11].

Another approach for the management of complex problems is to divide them into smaller, loosely interrelated subproblems, whose solutions are then composed to find a global solution. In this paper, a distributed approach is presented to limit, within an acceptable level, the computational complexity of the train scheduling problem. The network is considered as divided into local areas, so that the scheduling decisions in each local area are taken in a fully parallel fashion by the local schedulers (that consider the level of local dispatchers). Since the composed solution must be globally feasible, the main issue is to find coordination actions (at the level of network dispatchers) in such a way that the union of all local solutions is globally feasible.

Given a division of the network, we adopt a compact representation of variables and constraints of each local area in order to limit the size of the set of data to be managed by the coordinator, avoiding the coordination task to become the bottleneck of the procedure. Each local dispatcher sends to the coordinator the entrance/exit time of each train traversing its borders and, for each pair of trains entering/leaving the local area, the minimum temporal distance between the two events. This information is used to build the *border graph* $G_B = (V_B, A_B)$, which consists of the set V_B of all border nodes and the set A_B of all border arcs. Border nodes are associated to the operations representing trains crossing the borders between areas. Let b_i, b_j be two border nodes, a border arc is defined between them if a directed path from b_i to b_j exists in the graph associated to a solution of a local area. The arc is weighted by the length of a longest path from b_i to b_j in the local graph. As proved in [2], the local solutions produced by the local schedulers are globally feasible if and only if there are no positive length cycles in the resulting border graph.

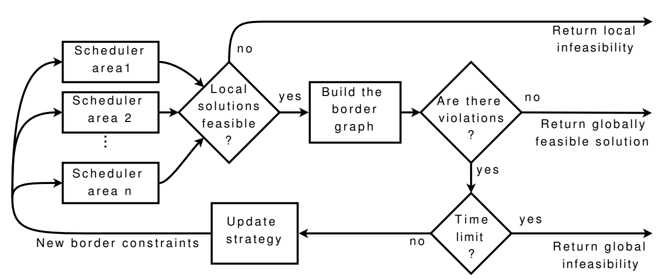

Fig. 2. General scheme of the coordination procedure.

We use a coordination procedure based on the border graph, following a generalized version of the heuristic rule described in [2]. Figure 2 presents the general coordination procedure for n local areas. First, each local scheduler computes the train scheduling solution of the corresponding local area. The local solutions, if locally feasible, are used to build the border graph. The border graph is used to check the global feasibility of the local solutions. A number of border constraints between each pair of adjacent dispatching areas have to be satisfied: (i) the exit time of a train from one area must be equal to the entrance time in the adjacent area, (ii) trains traversing a border on the same block section must keep the same order at the exit from an area and at the entrance in the adjacent area, (iii) locally feasible schedules must not cause deadlock situations from a global perspective.

Such constraints cannot be checked directly by the a single local dispatcher, since each subproblem has a myopic view of the entire process, whereas a global view over all areas is required. To this end, the coordinator may either impose precedence constraints among some trains at border nodes or may impose the same value for the exit time from an area and the entrance time in the subsequent area for some train. The update strategy used imposes new constraints at the borders, concerning the three cases analyzed before. For case (i) the coordinator sets the release time of the train entering the next area equal to the exit time of the same train from the previous area. For any infeasibility of type (ii), the train order of the previous area is imposed to be the same in the following area, in case the trains run in the same direction; if trains run in opposite directions, precedence is given to the first train reaching the border in any local area. When the infeasibility affects multiple areas as in case (iii), there must be at least one train, among those involved in the global deadlock, for which the exit time from an area is larger than the entrance time in the next area computed by the two associated local schedulers. The coordinator selects among these trains the earliest exit time τ scheduled by the local dispatchers, and then sets the release time of the associated train in the next area equal to τ. Each local scheduler then computes a new schedule with the new border constraints. The iterative procedure terminates when a globally feasible solution is found or when a local infeasibility is found or when the time limit of computation is reached.

III. COMPUTATIONAL EXPERIMENTS

This section presents our computational results. The algorithms are implemented in C++ and run on a PC equipped with a dual-core processor Intel Pentium D (3 GHz), 1 GB RAM and Linux operating system. We study a busy railway network in the South-East of the Netherlands that spans over ten dispatching areas of the Dutch railway network. The network layout comprises a combination of single and double-tracks of different length, with a diameter of 300 km. In total, there are more than 1200 block sections and platform stops at stations. The network studied includes the major stations of Utrecht Central, Arnhem and Den Bosch, plus other 40 minor stations. Two network decompositions are considered for the distributed architecture, as shown in Figure 3. For the 3-area division the border stations are Bunnik, Utrecht Lunetten and Nijmegen Dukenburg; for the 6-area division the border stations include also Den Bosch Oost, Arnhem Zuid, Utrecht Terwijde and Utrecht Zuilen.

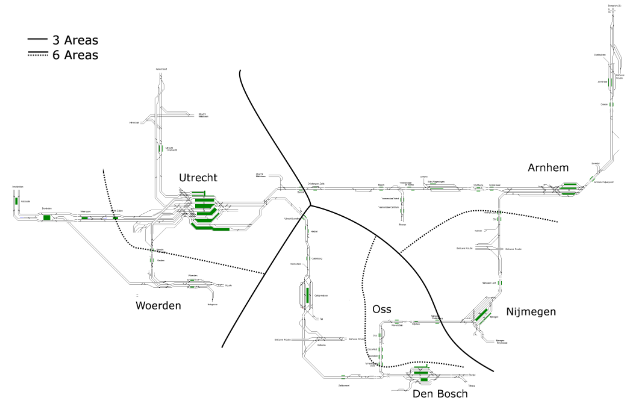

Fig. 3. The studied railway network and the two divisions in local areas.

We refer to an operational timetable that schedules local and intercity services and has periodicity of one hour. Increasing time horizons of traffic prediction (i.e., 30, 60 and 90 minutes) are investigated in order to study the delay propagation and the limits of the proposed algorithms.

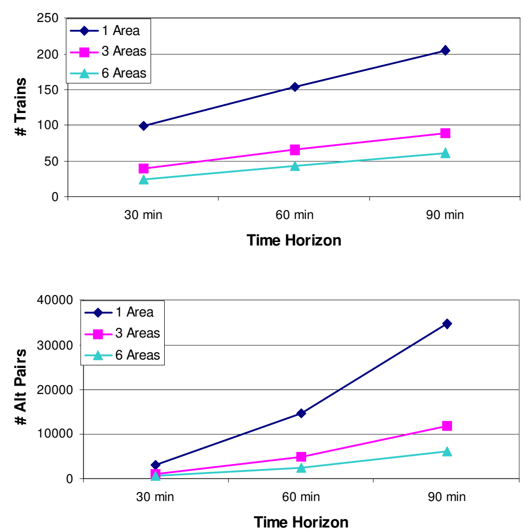

Fig. 4. Number of trains and alternative pairs for the different instances.

Figure 4 reports information on the problems being solved, in terms of the number of trains considered for each scheduling problem (top), and the number of alternative pairs A, i.e., elementary decision variables per area (bottom). Every plot reports those values for the three network decompositions into 1, 3 and 6 areas, and moreover for the three time horizons considered (x-axis). Enlarging the time horizon results in a linear growth of instance size, and a quadratic growth of the amount of variables.

We consider as infrastructure disruption a blockage of a single track, on the double track line that connects Utrecht to Den Bosch, between the stations of Zaltbommel and Den

Bosch (see Figure 5). In the original timetable, 12 trains per hour (6 per direction) are scheduled on the disrupted line, four of which are local services between Utrecht and Den Bosch while the other eight are intercity services connecting the northern region of the country with the southern region. We consider the following modifications to the train routes of the hourly timetable to recover the disrupted situation: Four intercity trains (2 trains per hour per direction) and four local trains (2 trains per hour per direction) are still scheduled on the line Utrecht - Den Bosch. In the vicinity of the disruption, the trains are locally rerouted for a stretch of 6 km, along the only available track that now serves a bidirectional flow. This results in a shortage of capacity that leads to delays propagating throughout the network. The other four intercity trains (2 trains per hour per direction) are globally rerouted, i.e., they change their routes but keep the same origin and destination stations, via the line going to Nijmegen and Arnhem, or vice versa. Note that the already busy interlocking area of Arnhem has to handle 4 additional trains per hour, leading to several conflicts for the increased amount of traffic. The running time required for the alternative route between Utrecht and Den Bosch is around 40 minutes longer than the original trip time, which is 30 minutes long.

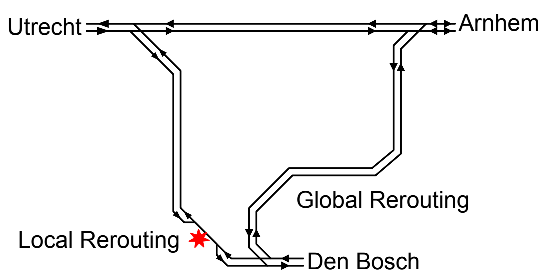

Fig. 5. Disrupted timetable with local and global rerouting.

We model timetable perturbations as random variations of the entrance times of all trains running in the network, represented by the Weibull distributions described in [3] that have been fitted to real-life data. This defines a set of expected deviations of entrance times for all trains running in the network. Five such delay cases are considered, resulting in an average initial delay of around 21.7 seconds.

Figures 6 and 7 report information on the computational results for each solution approach. The algorithms considered are the FCFS (considering only one area), the centralized scheduler of [5], labeled BB1, the distributed approach considering 3 areas (BB3), and the distributed approach considering 6 areas (BB6).

Figure 6 (top) shows the percentage of feasible schedules found by each algorithm. Figure 6 (bottom) reports the total computation time (in seconds) to compute a feasible schedule (if any) for the three algorithms FCFS1, BB3 and BB6. Since for the centralized branch and bound algorithm BB1 the total computation time is always 300 seconds, we report under BB1 the time required to find the best solution (if any).

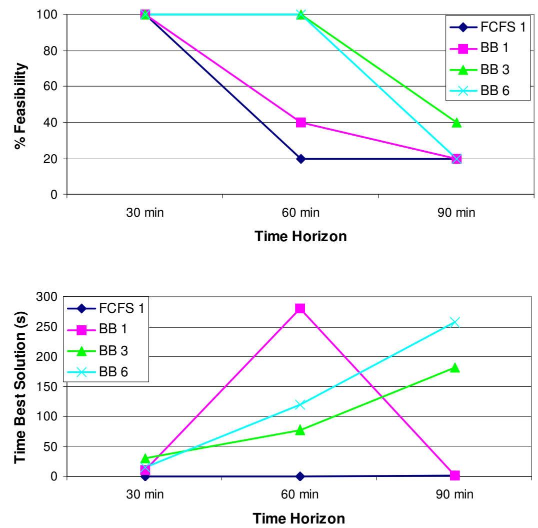

Fig. 6. Feasibility and computation times.

For the 30-minute time horizon, all tested algorithms are able to compute a feasible schedule within a few seconds. For the 60-minute time horizon, the distributed approaches (BB3 and BB6) always compute globally feasible solutions, while the centralized approaches (FCFS1 and BB1) often fail to find a feasible schedule. For the 90-minute time horizon, all algorithms have problems in the computation of feasible schedules. Specifically, BB1 is unable to improve the results of FCFS1 within the 300 seconds of computation, and this fact motivates the negligible time required by BB1 to find the best solution for the 90-minute instances. Also BB6 finds the same number of feasible solutions, while BB3 is able to solve a number instances double than the other algorithms.

The computational experience shows that disruption handling problems are very complex when there is a serious risk of deadlock. In our instances this is due to the large number of trains sharing the single track section in both traffic directions, and to those running over the busy interlocking area of Arnhem. The 90 minutes instances represent the current limits of the proposed method. In fact, these instances are both huge (up to tens of thousands of variables) and difficult to be solved and coordinated.

Figure 7 gives the performance of each algorithm in terms of maximum (top) and average (bottom) consecutive delay, both expressed in seconds. Due to the fact that not all algorithms are able to deliver a feasible schedule for each instance, specially in case of large time horizons, we deal with any infeasibility in the following way: if one algorithm does not find a globally feasible solution for one instance, then we consider a penalty equal to the worst value found by the other algorithms. Regarding to the solution quality, no algorithm outperforms the others, even if the average consecutive delay is better minimized by distributed approaches. In particular, BB6 very often achieves the best scores for the various time horizons.

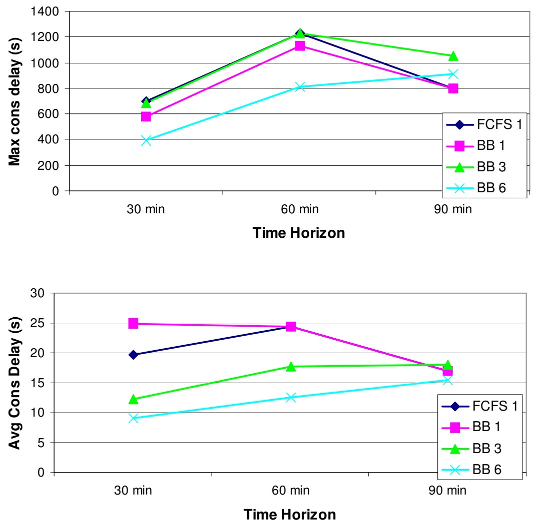

Fig. 7. Maximum and average consecutive delays.

On the whole, distributed approaches outperforms centralized approaches in terms of both the number and the quality of the feasible solutions found. In fact, centralized algorithms experience increasing difficulty in managing the complexity of the train scheduling problem and the high probability of conflicts and deadlocks when the size of the instances increase. Distributed approaches take advantage from the smaller size of the instances solved by each dispatcher. A key role in the performance of distributed approaches is played by the coordination procedure, which is able to drive dispatchers towards globally conflict-free and deadlock-free schedules despite the dense traffic and the scarce capacity. However, when the problem size and the number of areas increases, also the coordination problem becomes more complicated and the probability of finding globally feasible schedules decreases.

CONCLUSIONS AND FURTHER RESEARCH

This paper applies innovative approaches and procedures to support the railway dispatching process during emergencies caused by disrupted traffic situations. We present an extensive study of dispatching and coordination algorithms. The disruption handling procedures compute feasible train schedules for exceptional situations, including global and local rerouting of train services. The solutions are evaluated quantitatively, so that the dispatchers might choose the most effective disruption resolution scenario and the corresponding microscopically feasible plan of operations.

For time horizons up to one hour, the proposed algorithms are able to compute good quality solutions in a very short time of computation, with the distributed approach resulting in a better feasibility performance than the centralized ones. Increasing the time horizon results in instances that are too difficult to be solved and would require further investigation on better coordination procedures.

Future research should address the implementation of dispatcher user interfaces, enabling fast and simple communication of key indicators without information overload. Such an approach could also be used in the planning stage in order to evaluate the feasibility and performance of alternative timetables.

REFERENCES

[1] Caimi, G. (2009) Algorithmic decision support for train scheduling in a large and highly utilised railway network. PhD Thesis, Diss. ETH No. 18581, Switzerland.
[2] Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M. (2010) Centralized versus distributed systems to reschedule trains in two dispatching areas. *Public Transport: Planning and Operations*, to appear.
[3] Corman, F., D'Ariano, A., Pranzo, M., Hansen, I.A. (2010) Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. *Transportation Planning and Technology*, to appear.
[4] D'Ariano, A. (2008) Improving Real-Time Train Dispatching: Models, Algorithms and Applications. PhD Thesis, TRAIL Thesis Series T2008/6, The Netherlands.
[5] 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** 643-657.
[6] Hansen, I.A. and Pachl, J. (2008) Railway Timetable and Traffic: Analysis, Modelling and Simulation. Eurailpress, Hamburg, Germany.
[7] Jespersen-Groth, J., Potthoff, D., Clausen, J., Huisman, D., Kroon, L.G., Maróti, G. and Nielsen, M.N. (2009) Disruption Management in Passenger Railway Transportation. In R.K. Ahuja, R.H. Moehring, C. Zaroliagis (Eds.), Robust and Online Large-Scale Optimization (5868) (pp. 399-421). Berlin: Springer.
[8] Lüthi, M., Medeossi, G. and Nash, A. (2009) Structure and simulation evaluation of an integrated real-time rescheduling and train control system for railway networks. *Networks and Spatial Economics* **9(1)** 103-121.
[9] Mascis, A. and Pacciarelli, D. (2002) Job shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research* **143(3)** 498–517.
[10] 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.
[11] Pranzo, M., Meloni, C. and Pacciarelli, D. (2003) A new class of greedy heuristics for job shop scheduling problems. *Lecture Notes in Computer Science* **2647** 223–236.
[12] Ricci, S. and Tieri, A. (2008) A petri nets based decision support tool for railway traffic conflicts forecasting and resolution. In J. Allan, E. Arias, C. A. Brebbia, C. Goodman, A. F. Rumsey, G. Sciutto, and N. Tomii (editors), Computers in Railways XI, pages 483492. WIT Press, Southampton, UK.
[13] Rodriguez, J. (2007) A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B* **41(2)** 231–245.
[14] Salido, M.A., Abril, M., Barber, F., Ingolotti, L., Tormos, P. and Lova, A. (2007) Domain-dependent distributed models for railway scheduling. *Knowledge-Based Systems* **20(2)** 186–194.
[15] Törnquist, J. and Persson, J.A. (2007) N-tracked railway traffic rescheduling during disturbances. *Transportation Research Part B* **41(3)** 342-362.
[16] Wegele, S., Slovák, R. and Schnieder, E. (2007) Real-time decision support for optimal dispatching of train operation. In: I.A. Hansen, A. Radtke, J. Pachl and E. Wendler (editors), CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis, Hannover, Germany.