Models and Algorithms
for Traffic Management
of Rail Networks

Alessandro Mascis, Dario Pacciarelli, Marco Pranzo

RT-DIA-74-2002 November 2002

alessandro.mascis@it.transport.bombardier.com, pacciarelli@dia.uniroma3.it,
mpranzo@dia.uniroma3.it
ABSTRACT

In this paper we report on the results of a research project on train traffic control systems, supported by the European Commission. The results of the project include the development of new optimisation models and algorithms for traffic management, and a general architecture for train traffic control, capable of managing both fixed block and moving block signalling safety concepts. This paper focuses in particular on models and algorithms for real time conflict resolution. Computational results are reported, based on a portion of the Dutch railway network, on the high-speed line Paris-Brussels-Amsterdam.
1 Introduction

This paper deals with the results of a research project on train traffic control systems supported by the European Commission, entitled Project No. TR4004 IV FP - DG XIII Telematics, acronym COMBINE. The project involves suppliers and users of rail traffic systems, software houses and universities from different European Countries. Its goal is to analyze opportunities and problems for traffic management related to the introduction of the moving block signalling standard ERTMS. The results of the project include the development of a general architecture for a train traffic control system and new optimization models and algorithms for traffic management.

Due to its inherent complexity, the management and control of rail operations is usually organized in a hierarchically structured planning process to generate and maintain train schedules. The strategy consists of off-line developing a detailed timetable for each train, often called master schedule, and by operating in real time with strict adherence to these timetables [4]. When unforeseen events occurs, such as the temporary unavailability of some resources, which make unfeasible the planned timetables, it is necessary to partially modify in real time the master schedule in order to restore feasibility. Modifications may include changing precedence between trains and/or their planned speed. This on-line process is called train dispatching or conflict resolution (CR) in the first case, and speed regulation (SR) in the second case.

Even if the resolution of conflicts is presently performed by human dispatchers all over the world, several computerized Traffic Management Systems (TMS) have been designed and implemented to support them to re-schedule the train movements and to prevent them from taking wrong decisions, such as causing a deadlock situation. Among the published results, we cite the papers by Adenso-Díaz et al. [1], Cai et al. [3], Higgins et al. [6], Şahin [10] and the papers of Harker et al. [7, 4, 5]. In any case, models at the on-line control and planning level are not designed to replace the human decision maker, who is always in charge to take the decision of implementing a solution.

One aim of COMBINE project is to move a step further in the direction of automating the train traffic control process, by enabling the TMS to implement some traffic control actions, without the authorization of the human dispatcher. A significant difference between a decision support system and a partially automated system, like the COMBINE TMS, is that while the former one can provide a solution which is not feasible in reality, a partially automated TMS must either provide a solution which can be really implemented, or ask for the help of a human decision maker. To this aim, detailed optimization models are necessary, in order to guarantee that a solution, which is feasible for the optimization model, is always also physically feasible.

It is worth noting that the TMS is not in charge of the safety of the rail network. In fact, there exist underlying safety systems that, when necessary, can take the control of the trains by imposing emergency braking in order to avoid collisions between trains.

The paper is organized as follows. Section 2 introduces the train scheduling problem, or conflict resolution problem. Section 3 introduces and describes the architecture of the COMBINE TMS. In Section 4 we first introduce the notation and the alternative graph formulation, then we formulate the conflict resolution problem by means of an alternative graph. Finally, we describe the solution procedure adopted to solve the conflicts. Section 5 deals with the solution procedures for the Speed Regulation System. In Section 6 we illustrate the computational experiences, which is based on the so-called Breda triangle, in
the Dutch part of the high-speed line Paris-Brussels-Amsterdam. Finally some conclusions follow in Section 7.

2 Problem Description

In this section we introduce the conflict resolution problem. There are two different technologies to ensure safety in the railway networks: the fixed block technology and the moving block technology. Since there are many different national standards, in this paper we refer to the Dutch NS54 fixed block signalling and to the European standard ERTMS for the moving block technology.

In its basic form a fixed block railway network is composed by track segments and signals. Signals allow to control the traffic on the network, and to avoid any potential collision among trains. There are signals before every station, passing loop, junction, etc., as well as along the lines. A block section is a track segment between two signals. Signalling systems vary quite a lot from country to country. However, the basic mechanism is as follows. A signal may turns into three or more colors, say red, yellow, or green. A red signal means that the subsequent block section is either out of service or occupied by another train, a yellow signal means that the subsequent block section is empty, but the following block section is occupied by another train, and a green signal means that the next two block sections are empty. A train is allowed to enter a block section depending both on its speed and on the signal color. Slow trains can enter a block section only if the signal is either green or yellow, fast trains can enter a block section at high speed only if the signal is green. Hence, each block section can host at most one train at a time. A block section takes a minimum time to be traversed, which is known in advance for each train, depending on the train and infrastructure characteristics. Besides the traversing time, a delay may occur at the end of a block section if the signal is red or yellow. The combinatorial structure of the train scheduling problem is therefore similar to that of blocking job shop scheduling problem, a block section corresponding to a blocking machine, and a train corresponding to a job.

With the moving block technology, at any time the exact position and speed for each train are known. Signals are not necessary in this case, since the safety of the trains is ensured by regulating and controlling their respective speeds. Safety standards impose a maximum speed for each train, depending on the distance from the preceding train, necessary to grant the space for completely blocking the train in case of emergency. Hence, track segments in this case are multiple capacity resources.

In both cases i.e., fixed and moving blocks, stopping or slowing a train causes a remarkable loss of time and energy, due to the long braking distances, followed by acceleration of large masses. More important, if a railway line slopes up over a certain gradient, then there are some freight trains that should not decrease their speed under a certain limit, otherwise they would not be able to reach the top, due to horsepower reasons. Therefore, in a feasible schedule, there are some freight trains that must not decelerate too much, and however, in a good schedule, fast trains should always have a good speed profile i.e., a speed profile that permits low energy consumptions. This means that in a fixed block railway network some trains should always find green signals, whereas slow train should always find green or yellow signals. On the other hand, in a moving block railway network fast and freight trains should not suffer too many speed variations.
The real-time management of rail operations requires checking if the off-line timetables are coherent with the current trains positions and speeds. If unforeseen events cause a train not to follow exactly its planned timetable, then an action is required in order to restore the feasibility in the schedule. In this paper we deal with this short term planning process, which is often called conflict resolution. More precisely, a conflict is any unforeseen event which makes the planned timetables infeasible (see, for example, Kraay and Harker [7]). A conflict occurs, e.g., when two trains require the same resource i.e., the same segment of track, at the same time. The conflict resolution problem requires determining a new feasible plan of meets and overtakes as close as possible to the master schedule i.e., such that the delay at all the stations is minimized. In particular, in this paper we address the problem of minimizing the maximum delay.

3 Traffic Management System Architecture

In this section we describe the architecture of the Traffic Management System (TMS) developed in the COMBINE project, as far as the modules for automated train control are concerned. The architecture of the TMS is shown in Figure 1, where two different layers inside the TMS can be distinguished: the Conflict Resolution system and the Speed Regulator.

At the highest hierarchical level there is the human dispatcher in charge for controlling
the rail network. The dispatcher evaluates the rail network status, and controls the traffic flows in the network. The human dispatcher, in the COMBINE TMS, focuses on important planning decisions only, and let to the TMS all other minor decisions. In other words, while the human dispatcher is able to take major decisions, such that canceling a connection or changing the route of a train, the computerized dispatcher can only reschedule train movements, thus maintaining in real time a conflict-free schedule for each train, compatible with the real time situation. Three different operating possibilities can be identified:

- (Manual Mode) The dispatcher decides to manually solve the conflicts.
- (Mixed Mode) The dispatcher can interact with the TMS modifying the planned timetable or imposing precedence relations between trains.
- (Supervision Mode) The dispatcher supervises the work of the automatic TMS.

In the Manual Mode the dispatcher manually solves every conflict arising in the rail network.

In the Mixed Mode the dispatcher can impose to the TMS some constraints in order to guide the solution process. A typical constraint is a fixed precedence relation among two trains or a given route for a train. By constraining the TMS the dispatcher can influence the behavior of the system guiding the algorithm towards good quality solutions.

In the Supervision Mode, the TMS is in charge of solving the conflicts, and the main role of the dispatcher is to control the work of the TMS. In any case and at any time, in the Supervision Mode, the dispatcher can switch to the Manual Mode to assure a better circulation. Moreover in some critical situations the TMS could be not able to find a feasible solution, thus requiring the dispatcher’s help. In these situations the dispatcher has to take the control of the network by solving the arising conflicts manually. Usually in these situations major changes in the timetable are required in order to restore a feasible situation.

The Conflict Resolution layer takes as input the position, the speed and the planned timetable, usually obtained by some off-line algorithm, for each train circulating in the rail network. Moreover as mentioned before, in the mixed mode, a set of precedence relations could be directly added to the problem by the dispatcher. In other words, given the current network status, the aim of the CRS is to obtain in real time a conflict-free schedule, as close as possible to the planned timetable.

The output of the CRS is a set of precedence constraints among trains and a set of goals for each train. A goal specifies a relevant point along the line to be met by the train, such as a station, a junction, or the end of the current resource, an interval [earliest, latest] possible time to reach the position, and an interval [minimum, maximum] speed for the train at the goal position.

The SR module is in charge for regulating the speed profile of each train in the network with the aim of respecting all goals and saving energy. In other words, the SR module generates a speed profile for each train, such that the train is able to reach the position specified by every goals within the given margins of time and speed. Speed regulation is expected to become a significant aspect of traffic control under the moving block technology, whereas it is usually managed with simple static rules under the traditional fixed block technology. The Speed Regulator layer takes the feasible plan produced by the
CRS as input, and for each train decides the train speed needed to reach the goal while reducing the energy consumption.

Finally, the output of the speed regulator is sent to the field level. In our experiments, the field has been modeled using a detailed rail simulator compliant to the NS54 signalling system and the ERTMS standard.

In the COMBINE TMS the SR procedure is executed every time the rail network status is updated, whereas the CRS is invoked, and a new feasible plan is obtained, only if the SR is not able to reach all the goals. Note that, as long as the SR is able to reach all the goal the CRS algorithm is not executed. In this way the CRS is executed only a small number of times, and the solution of the TMS is “stable” i.e., it changes rarely in the time. If the CRS is not able to respect all the planned timetable constraints then the help of the dispatcher is requested.

4 Conflict Resolution

In this section we describe in details the CR system developed in the COMBINE project. First we introduce the mathematical notation used to model the train scheduling problem, then we show how the alternative graph formulation [8, 9] is able to represent in details the train scheduling problem. Finally, we describe the algorithm developed for the Conflict Resolution System, based on the alternative graph formulation. As already observed in Section 2, the combinatorial structure of the train scheduling problem is similar to that of blocking job shop scheduling problem, a block section corresponding to a blocking machine, and a train corresponding to a job. In what follows we describe the alternative graph formulation for the blocking job shop problem, we then extend the model to the conflict resolution context.

4.1 Models

Following the traditional terminology used in scheduling theory, we refer to a train as a job, whereas we refer to a track segment as a machine (i.e. a resource that is used by a job). In the usual definition of the job shop problem a job must be processed on a set of machines (i.e. a train must pass through a given set of track segments). The sequence of machines for each job is prescribed; the processing of a job on a machine is called an operation and it cannot be interrupted. We have therefore a set of operations \{o_0, o_1, \ldots, o_n\} which have to be performed on m machines \{m_1, m_2, \ldots, m_m\}. Each operation o_i requires a specified amount of processing p_i on a specified machine M(i), and cannot be interrupted from its starting time t_i to its completion time c_i = t_i + p_i. o_0 and o_n are dummy operations, with zero processing time, that we call “start” and “finish” respectively. Each machine can process only one operation at a time.

There is a set of precedence relations among operations. A precedence relation (i, j) is a constraint on the starting time of operation o_j, with respect to t_i. More precisely, the starting times of the successor o_j must be greater or equal to the starting time of the predecessor o_i plus a given time lag f_{ij}, which in this model can be either positive, null or negative. A positive time lag may represent, for example, the fact that operation o_j may starts processing only after the completion of o_i, plus a possible setup time. A time lag smaller or equal to zero represents a synchronization between the starting
times of the two operations. Finally, we assume that \( o_0 \) precedes \( o_1, \ldots, o_n \), and \( o_n \) follows \( o_0, \ldots, o_{n-1} \). Precedence relations are divided into two sets: *fixed* and *alternative*. Alternative precedence relations are partitioned into pairs.

A *schedule* is an assignment of starting times \( t_0, t_1, \ldots, t_n \) to operations \( o_0, o_1, \ldots, o_n \) respectively, such that all fixed precedence relations, and exactly one for each pair of the alternative precedence relations, are satisfied. Without loss of generality we assume \( t_0 = 0 \). The goal is to minimize the starting time of operation \( o_n \). This problem can be therefore formulated as a particular *disjunctive program* i.e., a linear program with logical conditions involving operations “and” (\( \land \), conjunction) and “or” (\( \lor \), disjunction), as in Balas [2].

**Problem 4.1**

\[
\begin{align*}
\min & \quad t_n - t_0 \\
\text{s.t.} & \quad t_j - t_i \geq f_{ij} \\
& \quad (t_j - t_i \geq a_{ij}) \lor (t_k - t_h \geq a_{hk}) \quad (i, j) \in F \\
& \quad ((i, j), (h, k)) \in A
\end{align*}
\]

Associating a node to each operation, Problem 4.1 can be usefully represented by the triple \( G = (N, F, A) \) that we call *alternative graph* [9]. The alternative graph is as follows. There is a set of nodes \( N \), a set of directed arcs \( F \) and a set of pairs of directed arcs \( A \). Arcs in the set \( F \) are *fixed* and \( f_{ij} \) is the length of arc \((i, j) \in F \). Arcs in the set \( A \) are *alternative*. If \((i, j), (h, k) \in A \), we say that \((i, j) \) and \((h, k) \) are *paired* and that \((i, j) \) is the *alternative* of \((h, k) \). Finally, \( a_{ij} \) is the length of the alternative arc \((i, j) \).

A *selection* \( S \) is a set of arcs obtained from \( A \) by choosing at most one arc from each pair. The selection is *complete* if exactly one arc from each pair is chosen. Given a pair of alternative arcs \((i, j), (h, k) \in A \), we say that \((i, j) \) is *selected* in \( S \) if \((i, j) \in S \), whereas we say that \((i, j) \) is *forbidden* in \( S \) if \((h, k) \in S \). Finally, the pair is *unselected* if neither \((i, j) \) nor \((h, k) \) is selected in \( S \). Given a selection \( S \) let \( G(S) \) indicate the graph \((N, F \cup S) \). A selection \( S \) is *consistent* if the graph \( G(S) \) has no positive length cycles. With this notation each schedule is associated with a complete consistent selection on the corresponding alternative graph. The *makespan* of a consistent selection \( S \) is the length of a longest path from node 0 to node \( n \) in \( G(S) \). Given a selection \( S \), we denote the value of a longest path from \( i \) to \( j \) in \( G(S) \) by \( l^S(i, j) \).

### 4.2 Train Scheduling Formulation

In this section a description of the alternative graph model for the conflict resolution problem is given. We first address the case of fixed block signalling system. Then, at the end of this section, we extend the results to deal with the moving block case and with mixed situations.

A railway network can be modeled as a set of track lines and signals, as described in Section 2, and a block section is a track segment between two signals. In the alternative graph model of the conflict resolution problem a node in the alternative graph corresponds to the time at which a given train enters a given block section. In this model fast trains require two or more empty block sections at a time, in order to travel at their maximum speed, and this can be easily modeled by suitably choosing the alternative pairs. Figure 2 shows an example for the case of two trains moving in the same direction: train \( A \) is a slow train and train \( B \) is a fast train, nodes \( i \) and \( j \) refer to the same block section \( k \).
Here, $p_{hk}$ is the travel time for train $h$ and block section $k$. If train $B$ precedes $A$ on block section $k$, train $A$ must wait until the section is empty i.e., until train $B$ enters section $k+1$. On the contrary, if train $A$ enters block section $k$ before $B$, then train $B$ must wait until the next two sections are empty i.e., until train $A$ reaches block section $k+2$.

![Figure 2: The graph representation for a slow and a fast train](image)

We observe that different trains have different further requirements. For energy saving and horsepower reasons, fast trains and freight trains should not decrease their speed under a certain limit. These constraints can be easily modeled by specifying a maximum time for moving from one point to another of the network. The requirement that a passenger train should not be too late at the stop stations can also be easily modeled as a due date constraint.

Figure 3 shows a small railway network with four block sections (denoted as 1, 7, 9, and 10), a simple station with two platforms (denoted as 3 and 4), and four special resources, called routes (denoted as 2, 5, 6, and 8), each of them including all the track segments in a junction. These resources have capacity one. At time $t$ there are three slow trains in the network. Train $A$ is a freight train, going from block section 1 to block section 10, and passing through platform 3 without stopping. Here, $\alpha$ is the time needed for train $A$ to pass through all block sections at the lowest speed allowed. Train $B$ is a passenger train going from block section 9 to block section 1, and passing through platform 4. Train $C$ is a passenger train going from block section 7 to block section 1, and stopping on platform 4. Its departure time from the station is $\bar{\beta}$. Finally, the planned times for trains $A$, $B$ and $C$ to leave the network are $\gamma$, $\delta$ and $\chi$, respectively.

![Figure 3: A small rail network](image)

In Figure 4 the alternative graph for this example is reported. For the sake of clarity we make use of a different notation, here. Each node of the alternative graph is denoted by the pair (train, block section). A pair of alternative arcs is represented by connecting
the two arcs with a small circle in Figure. Each alternative pair of arcs is associated to the usage of a common resource. In particular, trains A and B share resources 1, 2, 5, 6, and 8. Trains A and C share resources 1, 2, 5, and 6. Trains B and C share resources 1, 2, 4, 5, and 6. Note that the initial position of train A implies that B and C are not allowed to precede A on block sections 1 and 2, and therefore we have the selected alternative arcs \((A2, B1), (A2, C1), (A3, B2)\) and \((A3, C2)\). The respective forbidden alternative arcs are not depicted. On all the alternative arcs there is an arbitrarily small weight \(\epsilon > 0\).

The fixed arcs with negative weight represent the minimum speed constraint for train A and the delays of the three trains at some relevant points of the network. In particular, arc \((A10, A1)\), with weight \(-\alpha\), corresponds to requiring a maximum time \(\alpha\) for train A to travel from block section 1 to 10. Due to minimum and maximum travel time constraints, in a feasible schedule the train speed is always kept within the feasible interval.

The planned departure time \(\beta\) of train C from the station (resource 4) is modeled with arc \((C2, n)\) with weight \(-\beta\). Similarly, arcs \((A12, n), (B11, n)\) and \((C11, n)\) with weight \(-\gamma, -\delta\) and \(-\chi\), respectively, model the planned exit time of each train from the network. With this model, given a complete consistent selection \(S\), the length of the longest path from 0 to \(n\) in \(G(S)\) equals the maximum delay of the three trains in the associated schedule. In fact, \(l^S(0, C2)\) is the departure time of train C from the station, and therefore \(l^S(0, C2) - \beta\) is the delay of train C at the station. Similarly, \(l^S(0, C11), l^S(0, A12),\) and \(l^S(0, B11)\) are the exit times of the three trains from the network, and therefore \(l^S(0, C11) - \chi, l^S(0, A12) - \gamma,\) and \(l^S(0, B11) - \delta\) are their respective exit delays.

The case of a moving block signalling system is now addressed. This case is slightly more complicated to model than the fixed block one. A moving block section can be represented as a resource with multiple capacity in which two consecutive trains cannot enter simultaneously, but rather with a minimum time lag depending on train speed. Since the overtaking is not allowed within a resource, the model must represent this fact.
Figure 6 shows an example for a moving block section with two trains (A and B). There are two pairs of alternative arcs \((i, h), (k, j)\) and \((h, i), (j, k)\). The minimum separation at the beginning [at the end] of the block section equals the length of arcs \((i, h)\) and \((h, i)\) \([\(j, k\) and \((k, j)\)]. The non-overtaking constraint follows from the fact that, if an arc from any of the two pairs is selected, then an arc from the other pair is forbidden. For example, if \((i, h)\) is selected from the first pair, then \((h, i)\) must be forbidden in the second in order to avoid positive length cycles in the graph.

It is worth noting that this representation is not able to limit the number of trains simultaneously using the same moving block section, thus resulting in an infinite capacity resource. However, in practical applications, the capacity of a moving block section is rarely reached, and number of trains simultaneously using the same moving block section can be easily checked in a post-processing phase.

Figure 6 shows an example of a mixed situation. In this case the junction in bold, labeled with number 3, is equipped with fixed block technology, while the following block section, numbered with 4, is equipped with the moving block technology.

The alternative graph for the train A and the train B is shown in Figure 7, where the shaded nodes represent the actual position of the two trains. In this example there are three pairs of arcs, the pair \((j, k), (l, i)\) representing the conflict arising in the block section (resource 3), and the pairs \((j, l), (m, h)\) and \((l, j), (h, m)\) representing the conflict arising in the multiple capacity resource 4.

### 4.3 Conflict Resolution Procedure

The Conflict Resolution System (CRS) is responsible for train scheduling, and it is the critical system from the computational perspective. In fact, finding the optimal solution to a problem formulated by means of the alternative graph is an \(\mathcal{NP}\)-hard problem. More in general, the problem of deciding whether a deadlock-free schedule exists or not, being fixed the initial positions and routes of the trains is an \(\mathcal{NP}\)-complete problem [9]. Unfortunately, within a real time environment it is necessary to solve the problem under
severe time requirements. Hence, the COMBINE CRS uses a fast heuristic algorithm to find a feasible solution to the Problem 4.1. If the algorithm fails in finding a feasible solution, it means either that there is no feasible solution respecting all the constraints, or that the heuristic is unable to find one. In both cases the system requires the help of the human dispatcher to restore feasibility.

In order to respect the strict time bound the CRS only considers those trains that are or will be present in the network within a given time window, called planning horizon, thus obtaining a significant reduction in the size of the problem. With a short planning horizon only few trains, and few conflicts, are considered whereas a longer planning horizon leads to a larger number of circulating trains, and a larger number of possible conflicts. There is a trade-off between the size of the planning horizon time window and the quality of the solution found by the CRS. In fact the solutions found with few circulating trains could be myopic, since the CRS does not take into account conflicting trains not in the planning horizon. On the other hand a conflict arising far in the future, is not important as a closer conflict, since other unforeseen events could still affect the far conflict. In other words there is a priority in the conflicts, conflicts arising in near future are more important than other that could arise far in the future. Moreover the size of the resulting alternative graph is strictly dependent on the number of circulating trains i.e., the smaller the planning horizon the smaller the alternative graph is.

The Conflict Resolution algorithm can be considered basically as a sequence of three independent phases: pre-processing, plan creation and post-processing. Every time a sequence is completed the output of the algorithm are given as input to the Speed Regulator. In what follows we describe in details the three phases composing the algorithm.

4.3.1 Pre-processing

The pre-processing phase can be divided in two basic subtasks: the update scenario phase and the graph building phase.

The update scenario phase is responsible for filling the internal data structures of the Conflict Resolution System with the current route status and train position and speed and, when available, with a new plan received by the dispatcher. The current position and the speed of a train influence the minimum travel time needed for moving through the subsequent track segments.

The second task of the pre-processing operations is the graph building phase. In the graph building phase the alternative graph representing the rail network is built. Every train is represented in the alternative graph by a chain of nodes and fixed arcs, representing the sequence of actions to be performed by the train: e.g. perform route $x$, enter track $y$, enter track $z$ etc. A travel time is associated with each action; this time is evaluated in the update scenario task, assuming the train running at a constant speed and without taking into account any conflict. In order to reduce computational times we update the alternative graph instead of rebuilding it completely. New trains are added to the alternative graph model as they enter the planning horizon. The duration of each operation is updated according to the new position and speed of the train and the length of the arc is modified accordingly. If the train route is modified by the dispatcher, the train is removed and added again as a new train entering in the planning horizon.

As mentioned before the dispatcher has the chance of imposing some precedence constraints between trains i.e., imposing that a train should enter a conflicting resource before
Procedure Conflict Resolution

1. while a conflict is found
2. begin
3. Add to the graph the alternative pair representing the conflict.
4. Solve the conflict by selecting the pair.
5. if the graph is infeasible then
6. begin
7. Perform backtrack and choose the alternative arc.
8. if no backtrack is possible then exit (found an infeasible solution).
9. end
10. end
11. exit (feasible solution found).

Figure 8: The Conflict Resolution procedure.

an another train. The set of constraints received by the dispatcher is represented with a set of fixed arcs that is added to the alternative graph during the building graph task. A check is performed to verify if the graph is feasible i.e., with no positive length cycles. If the resulting graph is infeasible then a new plan is required to the dispatcher, and the TMS switches to the manual mode.

In order to reduce the computing time, the build graph subtask does not generate in the alternative graph all the pairs needed to represent the problem. The alternative pairs are added to the graph only when needed. More precisely, in the preprocessing step a plan of earliest/latest possible arrival and departure times for the trains at a set of key points is computed. Then, for each resource in the network, a conflict can arise only for those pairs of trains that are allowed to pass through the resource at the same time i.e., such that the respective intervals of earliest/latest possible arrival/departure times for the trains overlaps. Hence, we add a pair of alternative arcs only for these trains and resources. A time window, and consequently the number of alternative pairs, is increased whenever a train violates it. Computational experience shows that even a large network with high traffic conditions can be modeled with an reasonable number of pairs of alternative arcs, thus allowing us to solve it within a very short time.

4.3.2 Plan Creation

Our scheduling procedure, shown in Figure 8, is a constructive greedy algorithm that repeatedly enlarges a feasible partial solution. If an infeasible selection is reached, the algorithm performs a backtrack and explores another branch of the enumeration tree. Aim of the search is to find a feasible solution such that the maximum delay of a train at each stop is never larger than a given quantity.

A conflict arises when a train asks for a resource already in use by another train in case of fixed blocks or when a train overtakes another train in the moving block case. More precisely in the fixed block case it arises when a train $A$ enters a resource $R_x$ before train $B$ leaves the resource $R_x$. Whereas in the moving block case a conflict occurs if
train A enters resource $R_x$ before train $B$ and train $B$ exits from $R_x$ before train $A$.

The conflicts are detected by means of a topological visit of the alternative graph, and the algorithm solves the conflicts with higher priority first. The Conflict Resolution algorithm solves the conflicts giving the precedence to the conflicting train that minimizes the increase in the delay. More formally let $((i, j), (h, k))$ be the alternative pair detected by the topological visit. The pair is selected according to the following expression

$$
\min\{l^S(0, i) + a_{ij} + l^S(j, n), l^S(0, h) + a_{hk} + l^S(k, n)\}
$$

(1)

where $l^S(x, y)$ denotes the length of the longest path in $G(S)$ from node $x$ to node $y$. In other words, the criterion adopted to solve the conflicts can be considered as giving the precedence to the a posteriori more delayed train.

Note that in some situations there is no choice on how to select an alternative pair. For example, let us consider an alternative pair $((i, j), (h, k))$ such that there exists a path in $G(S)$ from node $j$ to node $i$, and let $l^S(j, i)$ be its length. Then, if

$$
l^S(j, i) + a_{ij} > 0
$$

(2)

selecting the arc $(i, j)$ would cause a positive length cycle in the graph. Hence, that arc has to be forbidden and its alternative selected. For some resources the planned timetable defines intervals earliest/latest on the earliest and latest entry time allowed on that resource. If selecting an alternative pair causes a train not respecting those constraints, then the condition 2 permits to identify positive length cycles in the graph and thus immediately to select the pair in the other direction.

### 4.3.3 Post-processing

When a satisfactory solution has been found by the Conflict Resolution algorithm, a post-processing is applied on it. The main task of the post-processing phase is to specify a set of goals for each train and each relevant point visited by the train. A goal contains the following information:

- a relevant point along the line to be met by the train, such as a station, a junction, or the end of the current resource,
- an interval [earliest, latest] possible time to reach the position,
- an interval [minimum, maximum] speed for the train at the goal position.

In other words, each train has to reach its next goal within given margins of time and speed. The definition of goals starts from the output of the plan creation phase, in which trains are scheduled to travel at maximum speed through all block sections. If a train reaches a station early with respect to the timetable, or if a train has to wait for another train at a junction, then in the post-processing phase the earliness or the waiting time is distributed backwards along the train path whenever this does not cause a delay to the previous trains. So doing, the train is allowed to travel at a lower speed, thus saving energy, while reaching on time all the relevant points.

After the post-processing phase, the resulting goals and precedence relationships between booking actions, that we call the plan graph, are sent to the Speed Regulator. More
precisely, the plan graph is composed of a chain for each train and precedence relationships between the route bookings. Each chain is a sequence of arcs representing resources, and nodes representing route bookings for each train. And the arc direction indicates the direction used by the train running in the resource.

5 Speed Regulation

In this section we will describe the Speed Regulator system. First, the relevant data to be exchanged by the CRS and the Speed Regulator are described, and then a detailed description of the Speed Regulator algorithm is given.

As shown in the previous section the Conflict Resolution System process sends a plan to the Speed Regulator. A CRS plan contains an ordered set $R$ of resources and the associated goals and routes for each train and a set of precedence relationships between the routes to be booked. When a route is set, other routes cannot be available, therefore it is necessary to define the Conflicting Routes Table (CRT).

A goal contains the following information for identifying the position, i.e. the end of the current resource, and regarding the earliest (latest) possible time to reach the position, and the minimum (maximum) possible speed at the goal position. In other words, each train has its own goal $A = (t_y, v_y)$ which can be reached with a margin $(\pm \delta_t, \pm \delta_v)$. Reaching the goal $B = (t_y + \delta_t, v_y - \delta_v)$ allows to reduce the energy consumption, but it may cause delays. Reaching the goal $C = (t_y - \delta_t, v_y + \delta_v)$ allows to reduce the delays but causes an increasing of the energy consumption. Since punctuality is the first objective, we prefer to let a margin for the trains.

The SR is responsible for controlling the train speed. Different Speed Regulator procedures are necessary when dealing with fixed block and moving block technologies. In fact, in the moving block technology the advisory speed is simply given by the minimum between possible local speed restrictions and the speed related to the distance from the next train, whereas in the fixed block case the speed depends on signals. In other words, advisory speed depends on the status of the next block sections (available/not available). In moving block technology this difference does not affect the architecture of the system, but only the computation of the SR. In this section we will first address the moving block case, which is simpler, and then the fixed block case.

At the start up of the process three look-up tables ($A$, $B$ and $C$) containing data on rolling stocks are also loaded. These tables are used by the SR, given the train type and the train speed, to calculate the needed space and time for each train circulating in the rail network, to accelerate $A$, to to brake $B$ and to cost $C$, i.e. decelerate without braking. In both cases the SR performs a sequence of four independent phases: update scenario, safety check, speed evaluation, and route booking. Every time a sequence is completed a new sequence can start. Now we discuss in details all the phases of the Speed Regulator.

5.1 Update Scenario

This action is responsible for filling the internal data structures of the Speed Regulator with the current route status and train position and, when available, with a new plan received by the Conflict Resolution.
5.2 Safety Check

A simple and very fast safety check is performed to verify if the train is able to stop at the end of its current Movement Authority. In this paper we call Movement Authority (MA) of a train the maximum speed allowed to the train. Two different limitations on the maximum speed allowed to a train can be distinguished: a “static” limitation due to a route that has not been set yet and a “dynamic” limitation due to a preceding train having a smaller speed. The MA definition is different depending on the fixed or moving block case. In fixed block case the MA is always dependent on static limitations, in particular it could depends on the train position and the first red signal. Whereas in the moving block case the MA could be dependent on both the static and dynamic limitations. In fact the maximum speed allowed to a train could depend on the train position and the position of the first train ahead on the same resource, or it could depend on the fact that the next route has not been set yet.

In what follow \( MA_L \) denotes the length of the Movement Authority, \( MA_D(v) \) deontes the distance with the preceding train (note that this distance also depend on the speed of the train ahead).

The safety check is very important in order to avoid that the underlying safety system takes the control of the train with undesired safety braking. This phase is performed differently in moving and fixed block technology.

In moving block the check is based on a table, say Brake Table \( B(v) \), in which is reported the information concerning the needed space (in meters) and the needed time (in seconds) to stop the train, starting from any given speed \( v \). In what follows we call \( B(v)_S \) the needed space to stop and \( B(v)_T \) the needed time to stop. Hence, the check is performed by comparing the needed space to stop the train at the current speed \( v \), i.e. \( B(v)_S \), with the current distance from the end of the Movement Authority. In practice the Safety Check consists in checking the following inequality:

\[
B(v)_S < MA_L + \sigma_1
\]  

where \( B(v)_S \) is the needed space to stop at the current speed \( v \), \( MA_L \) is the Movement Authority Length and \( \sigma_1 \) is an internal parameter (Safety Parameter) that depends on several technological parameters such as the uncertainty in the train position, the delay of the communication system in communicating the train position to the Speed Regulator, and the delay in implementing the actions required by the Speed Regulator.

The table \( B(v) \) contains static data and it is loaded when the process starts. Therefore this check is very fast. If the inequality is TRUE then the train can continue following its current speed (that can be updated by the next step). If the inequality is FALSE then the train has two possibilities:

1. if the \( MA_L \) is due to a preceding train (having smaller speed) then the current train will adapt its speed to the speed of the preceding train, i.e. the speed of the current train will become a bit smaller than the speed of the preceding train.

2. if the \( MA_L \) is due to a route that has not been set yet, then the current train will start stopping.

With fixed block technology, the Movement Authority is simpler, since there are static speed restrictions. In this case signals at the beginning of each block section give the
Movement Authority for the train entering the section (e.g., decrease speed down to 60 km/h). In this case we simply adapt the speed of the train to be compliant with the Movement Authority. If the train speed is already greater than the current speed, we simply consider it as a Movement Authority speed in the current block section.

5.3 Speed Evaluation

The speed evaluation phase is composed of two basic subtasks: the first one (goal check) has to be executed in all the cycles of the Speed Regulator algorithm, the second one (speed analysis) can be executed with a lower frequency, in order to match the strict time requirements that are given for the Speed Regulator.

5.3.1 Goal Check

The goal check simply verifies if the train can reach the goal (without taking into account the position of the other trains). If the train cannot reach the goal the Speed Regulator sends a warning to the CRS. This phase is very similar under moving and fixed block technologies. The goal check can be efficiently performed by means of the tables $B(v)$ and $A(v)$. The Acceleration Table $A(v)$ contains all the information concerning the needed space and time to let the train reach its maximum speed $v_{\text{max}}$, starting from any given speed $v$. In what follows we will call $A(v)_S$ the needed space to reach $v_{\text{max}}$, starting from $v$, and $A(v)_T$ the needed time to reach $v_{\text{max}}$, starting from $v$. Let $t$ be the current time, $x$ the current position of a train, $v_x$ its current speed, $y$ the goal position of the train (the smallest value if an interval is given), $t_y$ the goal time, and $v_y$ the goal speed. The goal check consists in verifying if there exists an intermediate speed $v_c$ such that the train is able to reach the goal position and speed within the time $(t_y - t)$ switching from the speed $v_x$ to $v_c$ and, finally, to $v_y$. Note that the goal speed $v_y$ can be partially specified (for example $v_y < v$) or not specified at all. In these cases we will consider the most favorable case (for example $v_y = v_c$ or $v_y = v_x$ in the case $v_c > v_x$). Hence $v_c$ can be either greater, smaller or equal to $v_x$ and $v_y$. We define the following quantities:

\[
T(v_1, v_2) = \max\{A(v_1)_T - A(v_2)_T, B(v_1)_T - B(v_2)_T\} \tag{4}
\]

\[
S(v_1, v_2) = \max\{A(v_1)_S - A(v_2)_S, B(v_1)_S - B(v_2)_S\} \tag{5}
\]

where $T(v_1, v_2)$ is the time needed to switch from $v_1$ to $v_2$, and $S(v_1, v_2)$ is the space needed to switch from $v_1$ to $v_2$. Note that if $v_1 > v_2$ then $A(v_1)_T - A(v_2)_T < 0$, $B(v_1)_T - B(v_2)_T > 0$, and vice versa. Using the equations 4 and 5 we can define the residual time $\Delta t$ and the residual distance $\Delta x$ to be covered at constant speed $v_c$ as follows:

\[
\Delta t = t_y - t - T(v_x, v_c) - T(v_c, v_y) \tag{6}
\]

\[
\Delta x = y - x + \sigma_2 - S(v_x, v_c) - S(v_c, v_y) \tag{7}
\]

The parameter $\sigma_2$ is an internal parameter of the TMS (Safety Parameter) that may also depend on the technological parameters already mentioned in the previous section $\sigma_1$ (e.g., the uncertainty in the train position, the delay of the communication system in
communicating the train position to the Speed Regulator, and the delay in implementing the actions required by the Speed Regulator).

The goal check for a train is therefore successful if the following system has a feasible solution:

\[
\begin{align*}
v_c \Delta t & \geq \Delta x \\
v_c & \leq \min\{MA, v_{max}\} \\
\Delta t & \geq 0 \\
\Delta x & \geq 0
\end{align*}
\]  

(8)

where \(MA\) is the maximum speed allowed to the trains circulating between \(x\) and \(y\), and \(v_{max}\) is the maximum speed allowed to the train. The solution can be easily found by means of the algorithm of Figure 9 (where \(\epsilon\) is a small positive constant):

5.3.2 Speed Analysis

As stated before the speed evaluation phase is composed of two basic subtasks: goal check and speed analysis. The rest of this section is devoted to the description of the speed analysis subtask.

Let \(TR_i\) be the ordered list of trains currently running on resource \(R_i\). The Speed
Procedure Speed Profile

1. Start considering the first train in $TR_i$ (the one that is forward) and calculate its needed speed profile to reach the goal in the normal way (i.e. as scheduled by the CRS).

2. Consider the next train and calculate its speed profile on the basis of its goal and the dynamic expected Movement Authority depending on the speed profile of the previous train.

3. Repeat step 2. until the speed profile is computed for the last train in $TR_i$.

4. If a train is not able to reach the goal, then trains that are forward try to reach the goal in the fastest way (i.e. as scheduled by the CRS, minus the given tolerance) in order to let the delayed train go faster.

Figure 10: The Speed Profile procedure.

Profile procedure calculates the speed profile for all the trains in a branch (i.e. with moving block technology), starting from the last one, i.e. from the first one to reach the next goal.

This means that Speed Profile procedure is performed the first time with the trains trying to reach the goal in the normal way. If it turns out that train $A$ cannot reach the goal, then Speed Profile procedure is performed again with the train $B$ (that precede $A$) trying to reach the goal in the fastest way.

Note that in the moving block technology the Movement Authority Speed $MA$ is not constant between $x$ and $y$, i.e. is a function $MA(x, t)$. However, the Movement Authority constraint can be easily satisfied, once the speed profile of the preceding train is known. In fact, the speed profile of each train is well known. Each train must first switch its speed from the current value to the value $v_c$ and, after a while, to the goal value $v_y$. We also know the precise position in which the preceding train will start varying its speed: the first time it is from position $x$ to position $x + S(v_x, v_c)$, the second time it is from position $y - S(v_c, v_y)$ to the goal position $y$. Let indicate the preceding train with the superscript $A$ and the following train with the superscript $B$. $x^A$ and $t^A_y$ are therefore, for example, the current position and the goal time for the first train, respectively. Note that train $B$ cannot reach the goal $y$ before time $t^A_y + k$, where $k$ is a time interval depending on the speed of train $B$ in proximity of the goal position $y$. For example, we can assume:

$$k = MA_D(v^B_c)/v^B_c + \min\{0, d(v^B_c - v^A_c)\}$$  \hspace{1cm} (9)

where $MA_D(v^B_c)$ is the Movement Authority Distance between the trains $A$ and $B$ (at the speed $v^B_c$), and $d$ is a safety parameter acting only if train $B$ is faster than train $A$. In fact, this is the only case in which we must pay attention to the Movement Authority: the parameter $d$ ensures that train $B$ will not reach train $A$ before the goal. If $t^B_y < t^A_y + k$ than train $B$ will not be able to reach its goal. In this case the Speed Regulator must re-calculate the speed profile, for all the preceding trains, trying to force them to reach their respective goals in the fastest possible way.
In other words, the trains should first try to reach goal \( A = (t_y, v_y) \). If all the trains can reach their goals with no delays, we have finished. Otherwise, as soon as we find a train that cannot reach its goal, then we must re-compute the speed profile for the preceding train, now based on the goal \( C = (t_y - \delta_t, v_y + \delta_v) \). This may force to re-compute also the speed profile of the preceding train and so on. If the computation propagates back to the first train of the list \( TR_i \), the Speed Regulator sends a message to the CRS.

Summarizing the above discussion: the time goal of train \( A \) will be either \( t_y^A \) or \( t_y^A - \delta_t \). The first situation arises when \( t_y^B \geq t_y^A + k \), the second situation arises when this inequality is not true. In the second situation we must compute the speed profile for train \( B \) with the target \( t_y^B = \max\{t_y^B, t_y^A - \delta_t^A + k\} \). If, in particular, \( t_y^B + \delta_t^B \geq t_y^A + \delta_t^A + k \), then the train \( B \) will not be able to reach its target within the given margins, and a message will be sent to the CRS. In any case, the updated target should allow to automatically satisfy the Movement Authority constraint for train \( B \). Hence, in what follows, train \( B \) will not consider explicitly the Movement Authority constraints for train \( B \), but just the maximum speed allowed on the given resource (referred to as \( MA_S \), Movement Authority Speed).

In the fixed block case, it is necessary to add a further check to verify that, once the speed profile of train \( A \) is computed, train \( B \) will never violate the Movement Authority. This is never the case when the trains are far enough one from the other, whereas it must be specifically evaluated when a distance comparable with the length of the current block section separates the two trains.

The speed analysis can be efficiently performed by means of the tables \( B(v) \) and \( A(v) \) described in the goal check section, plus an additional table \( C(v) \). The Costing Table \( C(v) \) contains all the information concerning the needed space \( C(v)_S \) and time \( C(v)_T \) to let the train cost, i.e. stop starting from any given speed \( v \) and switching the engine off (slowing down without braking). Also, let define the following quantities:

\[
\hat{T}(v_1, v_2) = \max\{A(v_1)_T - A(v_2)_T, C(v_1)_T - C(v_2)_T\} \tag{10}
\]

\[
\hat{S}(v_1, v_2) = \max\{A(v_1)_S - A(v_2)_S, C(v_1)_S - C(v_2)_S\} \tag{11}
\]

where \( \hat{T}(v_1, v_2) \) and \( \hat{S}(v_1, v_2) \) are the time and the space needed to switch from \( v_1 \) to \( v_2 \) without braking. Hence, the residual time and space to be cover constant speed, can be computed as follow in the cases of braking and casting respectively:

\[
\Delta^B t = t_y - t - \hat{T}(v_x, v_c) \tag{12}
\]

\[
\Delta^B x = y - x - \hat{S}(v_x, v_c) \tag{13}
\]

\[
\Delta^C t = t_y - t - \hat{T}(v_x, v_c) \tag{14}
\]

\[
\Delta^C x = y - x - \hat{S}(v_x, v_c) \tag{15}
\]

Note that, \( \Delta^C x \geq \Delta^B x \) and \( \Delta^C t \geq \Delta^B t \) since \( C(v)_S > B(v)_S \) and \( C(v)_T > B(v)_T \) for all \( v \).

The procedure Speed Analysis (Procedure 11) is called several times for each resource. In particular, we start from the first train in the resource, and call procedure Speed
Procedure Speed Analysis (current status $t_x, v_x$, goal $t_y, v_y$)

1. $v_c' := \min\{MA, v_{\max}\}$, $v_c'' := \min\{v_x, v_y\}$, check:=negative.

2. Compute $\Delta^C t$ and $\Delta^C x$ for $v_c := v_c'$.
   if $(\Delta^C t \geq 0) AND (\Delta^C x \geq 0)$ then go to 3,
   else go to 4.

3. Check inequality $v_c \Delta^C t - \Delta^C x \geq 0$.
   if it is verified then check:=positive, find $v_c$, feasible for 8, by means of a binary search in the interval $(0, v_c')$,
   else check:=negative ($y - x$ is too long), exit.

4. Compute $\Delta^C t$ and $\Delta^C x$ for $v_c := v_c''$.
   if $(\Delta^C t \geq 0) AND (\Delta^C x \geq 0)$ then go to 5,
   else check:=negative ($t_y - t$ or $y - x$ is too short for adjusting the speed), exit.

5. $v_c'' := 0$; find a feasible value of $v_c$ in the interval $(v_c'', v_c')$ such that $0 \leq v_c \Delta^C t - \Delta^C x \leq \epsilon$, $(\Delta^C t \geq 0) AND (\Delta^C x \geq 0)$ by means of a binary search.
   if $v_c \Delta^C t - \Delta^C x > \epsilon$ then $v_c' := (v_c' + v_c'')/2$ else $v_c'' := (v_c' + v_c'')/2$.
   if $v_c$ has been found then the check is positive.
   if $v_c' - v_c'' < \epsilon$ exit (the check is negative).

6. if the check is negative then try a new search by using the values $\Delta^B t$ and $\Delta^B x$, and repeat steps 1 to 5.

Figure 11: The Speed Analysis procedure.
Analysis (current status $t_x, v_x$, goal $t_y, v_y$). We continue by computing a speed profile for train $i$ in the list by checking first the inequality $t_y^i \geq t_y^{i-1} + k$. If this inequality holds, we execute Speed Analysis $(t_x, v_x, t_y, v_y)$, otherwise we have to re-compute the speed profile of train $i-1$ first. We do this by calling procedure Speed Analysis $(t_x^{i-1}, v_x^{i-1}, t_y^{i-1} - \delta_t^{i-1}, v_y^{i-1} - \delta_t^{i-1})$. This re-computation may propagate back to train $i-1$ and so on, until either we find feasible profiles for the first $i$ trains or we find a train that already reaches the goal $t_y^{i-1} - \delta_t^{i-1}, v_y^{i-1} - \delta_t^{i-1}$. In this case we compute a profile for train $i$ with the goal $t_y^i = \max\{t_y^i, t_y^{i-1} - \delta_t^{i-1} + k\}$ and send a message to the CRS.

5.4 Route Booking

Given the Plan Graph the Speed Regulator will book all the routes that are not blocked. A route $R_a$ is said blocked when there is at least a precedence arc $(b, a)$ in the Plan Graph such that route $R_b$ is not yet released. Finally, this module sends to the field the resulting speed values and booking actions coming from the previous phases.

6 Computational Experiences

The test site chosen for verification and evaluation purposes is the Breda triangle in the Dutch part of the high-speed line Paris-Brussels-Amsterdam (hereinafter called Breda junction). The test site is depicted in Figure 12. A mini-station with a loop enables passing and recovery of required train orders on the area boundary. A junction of two train tracks enables crossing movements of trains, and it is assumed that there are no power supply limitations. The maximum speed on the bold lines is 300 km/h, and in the tunnel is 280 km/h. On the medium tracks the maximum speed is 170 km/h, whereas on the thin lines the maximum speed is 140 km/h in the first 400 meters from the main line, 110 km/h otherwise. TGV’s run on the main line from Amsterdam to Brussels. Shuttle trains run from Rotterdam to Breda and from Brussels to Breda, where merging and exiting is done via fly-overs. The high speed line will be used by TGV’s. Some of its sections will be jointly used by national high speed Shuttle trains.
This site has been chosen as a case study for the COMBINE project since at the time of writing this was one of the first undergoing real world implementations of the ERTMS level 3 system. The approach has been tested by using a detailed rail simulator fully compliant with ERTMS Level 3 specifications. The rail simulator takes into account the characteristics of the rolling stocks, rail tracks, radio transmissions, driver reaction times, etc. In particular, we call control loop delay the minimum time between two consecutive updates of the rail network status. The control loop delay is dependent on a number of technological variables, such as radio transmission delays and others. In any case the TMS should be able to obtain a new solution within the control loop delay time, otherwise the safety layer could take control of the trains and imposing undesired emergency braking. In the computational experiments the control loop delay is fixed, for all tests, at 20 seconds.

In all the test the TMS optimization algorithm is compared with a simple dispatching rule (First In - First Out). The comparison between TMS and FIFO is carried out showing a set of information, presented in graphical form. The description of such information, as well as definitions necessary to avoid any misunderstanding for the reader, is presented in the following. Let us define the “entry delay” as the difference between the actual entry time and the planned entry time, i.e. the difference between the instant when the observed train enters the control area and the instant when the observed train is scheduled to enter the control area according to the timetable. Moreover we call “exit delay” the difference between the actual exit time and the planned exit time, i.e. the difference between the instant when the observed train leaves the control area and the instant when the observed train is scheduled to leave the control area according to the timetable. In “exit vs. entry delay distribution” the whole set of observed trains is shown in histograms where trains are classified according to their delay when leaving the control area. The “normalized energy consumption” shows the energy consumption as percentage of the energy consumption for the reference case, i.e. the FIFO case. The “total tardiness normalized to entry delay” shows the sum of the exit delays, as percentage of the sum of the entry delays.

Two sets of tests are carried on. We report the first test set in Section 6.1, where different traffic and disturbance scenarios are considered. The aim of these tests is to achieve the maximum exploitation of the Breda junction test site, for this reason the timetables have been defined addressing the full exploitation of the test site; to this aim, scheduled speeds, for each train, are the train or line maximum speed. This means that, in these tests, no significant margins are planned in the timetables to recover entry delays. In order to address the behavior of the system also in conditions where a significant delay recovery margin is available, an additional set of tests has been performed, as shown later in Section 6.2.

6.1 Tests with Maximum Speed Timetable

For evaluation purposes, several traffic conditions have been considered: Normal Traffic (NT) representing the traffic planned over the high speed line for year 2015, Heavy Traffic (HT) representing the planned traffic for year 2015 but with more Shuttles, Extreme Traffic (ET) as the Heavy Traffic but with additional national traffic on the secondary lines between Rotterdam and Breda. It is worth noting that the secondary lines are not used in the other scenarios. In all the traffic scenarios all trains are regularly spaced in time. Table 1 shows the frequency of the trains (on each line and on each direction) expressed in minutes, for all the three traffic scenarios.
For each traffic condition, the following disturbance scenarios have been considered:

- **(Disturbance scenario D1) Small stochastic disturbances** which should not affect the planning. All entering trains are stochastically delayed with respect to the plan according to an exponential distribution with a mean of 0.75 minute and truncated to a maximum size of 3 minutes, over the whole simulated period.

- **(Disturbance scenario D2) Large stochastic disturbances** which will cause minor to major conflicts. All entering trains are stochastically delayed with respect to the plan according to an exponential distribution with a mean of 2.5 minutes and truncated to a maximum size of 10 minutes, over the whole simulated period.

- **(Disturbance scenario D3) Small stochastic disturbances, and large deterministic disturbance** which will cause major conflicts. All entering trains are stochastically delayed with respect to the plan according to an exponential distribution with a mean of 0.75 minute and truncated to a maximum size of 3 minutes, over the whole simulated period. Additionally, the second TGV from Rotterdam to the border is 10 minutes delayed. This train is chosen because it will cause a cascade of disturbances.

Since each test involves stochastic disturbances, and in order to collect sufficient data for a statistically sound analysis, each test consisted of 4 replications of 5 consecutive hours, where the first hour (warm-up time) has been discarded in the analysis of the results. Given the complexity of these tests and the huge amount of data they provide, the results are presented in this section summarizing data about train delays and energy consumption for each traffic condition.

This first tests set addressed the behavior of the TMS when the full exploitation of the Breda junction is aimed. These tests were characterized by the fact that all trains were planned to run throughout the controlled area at maximum allowed speeds, i.e. addressing

Table 1: The three traffic scenarios

<table>
<thead>
<tr>
<th></th>
<th>NT</th>
<th>HT</th>
<th>ET</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGV Rot-Bel</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>TGV Bel-Rot</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Shuttle Bre-Rot</td>
<td>30</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Shuttle Rot-Bre</td>
<td>30</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Shuttle Bre-Bel</td>
<td>60</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Shuttle Bel-Bre</td>
<td>60</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Shuttle Bre-Rot (secondary)</td>
<td>-</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>Shuttle Rot-Bre (secondary)</td>
<td>-</td>
<td>-</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 2: Energy consumption normalized to FIFO case

<table>
<thead>
<tr>
<th></th>
<th>NT</th>
<th>HT</th>
<th>ET</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>TMS</td>
<td>99.5%</td>
<td>98.3%</td>
<td>98.3%</td>
</tr>
</tbody>
</table>
Figure 13: NT Exit delay distribution (Entry delay as disturbance indicator)

Figure 14: HT Exit delay distribution (Entry delay as disturbance indicator)
minimum travel times from their origins to their destinations. Moreover, there was not any priority distinction among trains.

In this case, the two control strategies (TMS or FIFO) show similar delay figures, see Figures 13, 14, 15, whereas all the TMS solutions are associated with minor benefits as far as energy consumption is concerned, as shown in Table 2. Such tests also revealed a close connection between the increase of traffic complexity and the decrease of the normalized energy consumption.

6.2 Tests with Flexible Timetable

This tests set addresses the behavior of the TMS in situations somehow more interesting in order to assess the effectiveness of optimization algorithms. Also in these tests there is not any train priority management, but they are characterized by the fact that timetables has been defined taking into account suitable delay recovery margins. In other words, planned travel times, for each train, are higher than their minimum values. Hence we describe here two different test situations, in the following AT1 and AT2, and we analyze the influence and the benefits of the TMS versus the FIFO control strategy.

Here we describe the first additional test case (AT1). The Shuttle 138604 from Belgium to Breda enters the control area with large delays (between 780 and 840 seconds), so that hindering conflicts arise with the TGV 104 from Belgium to Rotterdam.

With the FIFO rule, the Shuttle 138604 passes through the mini-station on the secondary line and joins the high speed line preceding TGV 104. The TGV is hindered by the Shuttle until the latter leaves the high speed line. This turns out into significant...
Figure 16: AT1. Entry and exit delay delays for TGV 104, whereas Shuttle 138604 recovers most of its initial delay. The delay collected by TGV 104 causes a convergence/hindering conflict with Shuttle 138601 from Breda to Rotterdam. In this case Shuttle 138605 joins the high speed line preceding TGV 104, which leaves the control area with a large delay.

Whereas TMS uses the secondary line inside the mini-station in order to allow TGV 104 to overtake Shuttle 138604, that is slowed down below the maximum speed allowed inside the station, so that it is no more hindered by the latter and leaves the control area on schedule. No other conflict arises.

As shown in Figure 16, with the FIFO rule, the Shuttle 138604 is able to drastically reduce its delay from 835 seconds to 263 seconds, but the TGV 104 exits with 584 seconds of delay, and all the other trains exit 90 seconds before their scheduled time, since they all drive at maximum speed. When the TMS is running, the exit delay of Shuttle 138604 is doubled in comparison with the FIFO case, but it is halved in comparison with the entry delay. Besides all the other trains respect the timetable.

Now we address the second additional test case (AT2). Trains coming from Rotterdam enter the control area with large delays (between 800 and 900 seconds for TGVs, between 300 and 360 seconds for Shuttles), so that convergence/hindering conflicts are likely to arise between the TGV 101 from Rotterdam to Belgium and the Shuttle 138602 from Breda to Belgium, when joining the high speed line.

With the FIFO case Shuttle 138602 runs with the speed scheduled by the original plan and approaches the convergence point before the delayed TGV 101, joining the high speed line preceding it. The TGV is hindered by the Shuttle up to the control area border and its exit delay is larger then the entry one. Shuttle 138602 leaves the control area on schedule.

When TMS is active the algorithm slows down Shuttle 138602 before the convergence point so that it joins the high speed line just behind the delayed TGV 101. This has some consequences on Shuttle punctuality, but allows the TGV 101 to recover a significant part of its initial delay, running at maximum speed throughout the control area.

With the FIFO rule, as shown in Figure 17, the Shuttle 138603 and Shuttle 183606 are able to recover partially their entry delay from 355 seconds to 81 seconds, and from...
Figure 17: AT2. Entry and exit delay

Table 3: AT1, AT2. Total Tardiness normalized to entry delay, and energy consumption normalized to the FIFO case.

<table>
<thead>
<tr>
<th></th>
<th>AT1</th>
<th>AT2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Tardiness</td>
<td>Energy Consumption</td>
</tr>
<tr>
<td>FIFO</td>
<td>100.9%</td>
<td>100%</td>
</tr>
<tr>
<td>TMS</td>
<td>53.1%</td>
<td>89.1%</td>
</tr>
</tbody>
</table>

The TGV 103 exits with 690 seconds of delay, thus reducing the entry delay, whereas the delay of TGV 101 increases from 817 to 1051 seconds. All the other trains exit before their scheduled time, since they all drive at maximum speed. On the other hand when the TMS is running, the exit delay of Shuttles 138603 and 183603 are completely recovered, but Shuttle 138602 exits the Breda junction with 162 seconds of exit delay. Both the TGV 101 and 103 are capable of reducing their delays from 817 to 584 and from 897 to 663 seconds.

This additional test set revealed definitely better performances of the TMS solutions with respect to those provided by the FIFO rule, as far as both punctuality and energy saving (Table 3) are concerned. In particular, such tests demonstrated the benefits deriving from the implementation of optimization algorithms which take decisions based on the knowledge of the global traffic status, with respect to a system where simple, almost “blind” control rules are used.

6.3 Discussion

The analysis of the test results reported allows to carry out some significant assessments concerning the effectiveness of the advanced optimization algorithms implemented by the TMS. Such assessments turn out from the comparison of the TMS solutions with respect to those provided by the reference FIFO solutions. Two different sets of tests have been
performed, considering very different infrastructure utilization philosophies on the same, realistic, test site.

In terms of punctuality increase and energy savings, the comparison with the reference case leads to different assessments when the results of the two test sets are considered. In fact, for the particular traffic conditions addressed in the first test set, the implementation of the advanced optimization algorithms seems to be redundant with respect to FIFO control rule. On the other hand, for the traffic conditions addressed in the second test set, the benefits of the optimization algorithms become apparent and definitely significant. In this case the FIFO rule is outperformed by the TMS algorithms.

When the timetables are defined with trains operating at maximum speed there is no room for having considerable energy savings and better quality service. In fact in this case once a train is delayed it is almost impossible to recover the delay, since the train is supposed to travel always at maximum speed. More flexible timetables, in which trains are planned to travel at less than maximum speed, offer two mains advantages:

1. when trains are late, it is possible to speed-up the trains in order to recover delays, thus increasing the probability of arriving at destination on time.

2. when trains are on time considerable energy savings can be achieved if trains do not travel at maximum speed.

As pointed out by Kraay and Harker [7], “planning at maximum velocity does not provide this flexibility”.

In order to improve the quality of the service and obtaining energy savings the Master Schedule of the railroad should be defined in a flexible way, i.e. taking into account suitable delay recovering margins and using advanced optimization algorithm. The tests confirm this hypothesis, in fact in the test with maximum speed timetables the TMS performs no better than the FIFO rule, since there is no margin to recover delays, but on the other test cases flexible timetable and the use of an advanced optimization algorithm permits to obtain significant increases in terms both of quality of service and energy savings.

7 Conclusions

In this paper we discussed models and algorithms capable of describing a rail network equipped both with fixed block and moving block signalling safety systems.

Performance tests aimed at showing whether advanced optimization algorithms are useful to manage railway traffic. Results showed that the optimization algorithms turned out into valuable advantages in terms of better punctuality and energy saving, when compared with simple dispatching rules, whenever appropriate slacks are present in the train timetables.

8 Acknowledgments

This work was partially supported by the European Commission, Grant number TR 4004, project COMBINE (enhanced COntrol center for a Moving Block signalling system).
References


