

TMS (Traffic Management System)

Maura Mazzarello, Douwe de Vries – June 2010

Contents

Overview	1
TMS core	3
TMS GUI	7
Examples of TMS results	10
Comparison with other methods	19

Overview

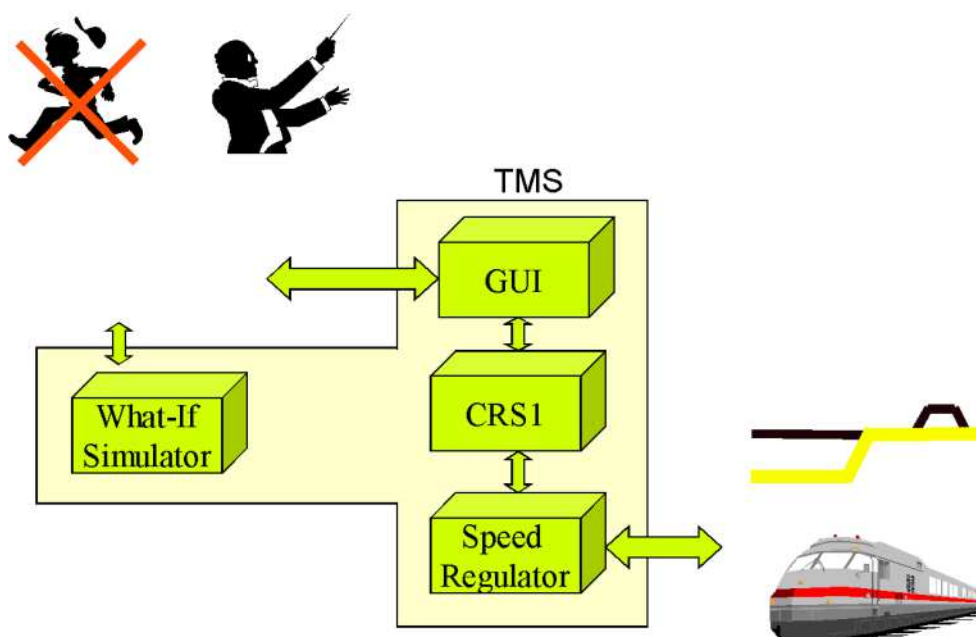
What is TMS

TMS is an autonomous system for the complete real-time management of railway traffic over a given network.

The goal of the TMS is to execute a predefined plan as well as possible, given a certain optimization criterion. The TMS detects and solves future conflicts and generates a new conflict free plan by shifting timetable target times and exploiting alternative routes. The TMS also supports the implementation of this plan by giving advisory speeds (or time-location-speed slots) to train drivers and advices about routes, route setting times and order changes to dispatchers.

The current TMS is meant for moderate disturbances, not for large disruptions (it cannot cancel trains or assign a train to another train-path).

This short paper concentrates on the kernel modules of the TMS architecture, which are responsible for automatic local traffic optimisation and control. The TMS kernel has a hierarchical structure consisting of two layers, respectively named Conflict Detection & Resolution-1 (CRS1) and Speed Regulator (SR). Additionally the TMS has a graphical user interface, named CRS2, where the original plan, the new plan and actual plan execution is shown.

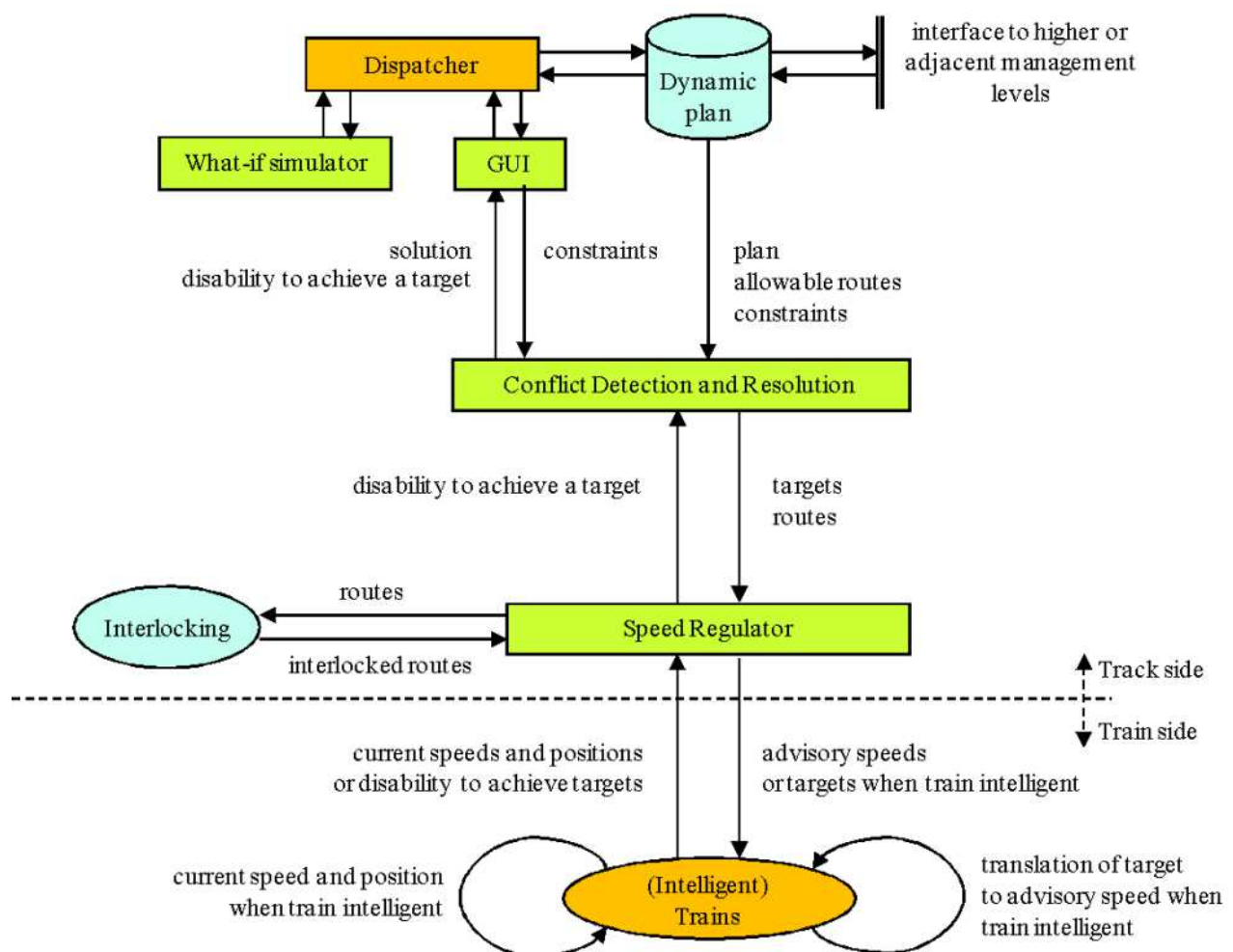


Application

The TMS is highly suitable to support dispatchers, solving routine conflicts with a high level of automation (which can be adapted to the desires of a dispatcher), thus freeing the dispatcher of this work so that he/she can focus on tactical decisions and also preventing that small deviations become large ones by intervening at an early stage.

Even problems with a limited number of trains on a limited network (e.g. 5 trains approaching 's-Hertogenbosch from different directions) have many solution possibilities and involve many relevant details, while there can be large differences in quality between various solutions. This quickly grows beyond human capabilities and response times. At the same time, computers are very good at this.

On the other hand, humans are very good at a tactical level, requiring a broader overview and creativity, dealing with problems that as yet cannot easily be formulated in mathematical terms or are too complex to solve.



The Rail system is not unique, not in process, not in complexity, not in operator interfacing needs. Decision support and control systems have been implemented successfully for similar systems. Clearly, a step by step approach in implementation is advisable. Due to its modularity TMS is suitable for this.

A first step, requiring only a moderate investment and without requiring an implementation in an actual traffic control centre, would be to incorporate the TMS in the FRISO-PRL gaming environment. In this environment dispatchers can experiment with the system and evaluate it. Also, interfaces and procedures can be tuned to dispatcher needs in this environment.

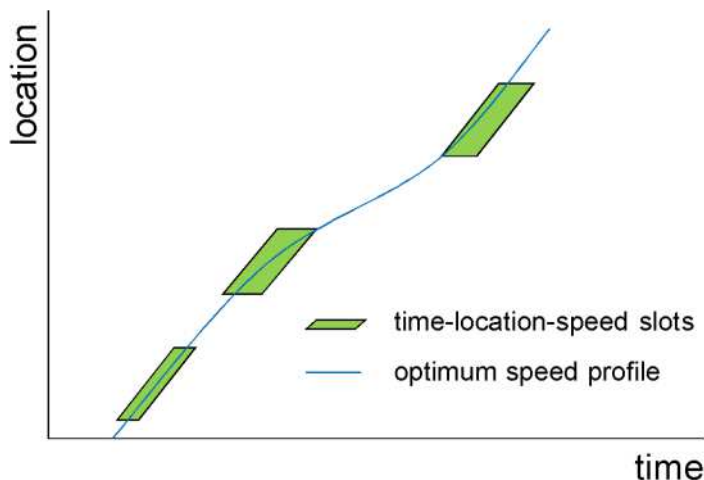
In simulation studies and a pilot implementation TMS already has shown its value, increasing punctuality, reducing energy consumption and reducing the number of non-commercial stops for freight trains.

TMS core

Modules

The three TMS modules are presented briefly below. Subsequently the TMS functionality is described in more detail.

- **CRS1** is responsible for automatic real-time train scheduling and routing. Given the current timetable, a set of constraints, the current traffic disturbances, and the position and speed of each train in the area, CRS1 detects and solves future conflicts and creates a new conflict-free plan by means of an alternative graph in which each node represents an event and each arc represents a time precedence relationship between two events. The scheduling algorithm minimises a suitable function of exit delays, acting both on train precedence relations at conflict points and on train routings. The new plan generated by CRS1 consists of time-location-speed goal windows and routes. It exploits allowable alternative routes, order changes and speed variations. CRS1 performs a global optimization. The optimization is over the whole voyage from entry to exit of the network of all trains currently running in the network or expected to enter in the next 15 minutes. So all trains, the whole network and a sufficient part of the future are taken into account. This time horizon is adaptable. Also the optimization criterion itself is adaptable.



- **SR** is responsible for the execution of the plans produced by CRS1. SR books routes and computes an optimal speed profile for each train, in order to make the CRS1 plan being executed in a safe and fuel-efficient manner. An advisory speed is sent to the trains whenever a change of speed is required, as result of estimations on the current plan goals and current state of running trains and infrastructure. If due to new disturbances SR detects that it no longer can comply with the goal windows, it informs CRS1 and asks for a new plan.
- **CRS2** provides a graphical user interface with dispatchers. It shows in a time-distance graph the original plan, the new plan and actual plan execution. It also provides time-speed graphs, distance-speed graphs and a graph with the infra-layout and current train positions.

Thanks to the modularity of TMS architecture, CRS1 can operate as stand-alone module, as well as linked to SR and other communication tools. As stand-alone tool, CRS1 generates optimal

schedules, given different traffic scenarios. Connected to SR and external modules, CRS1 exploits its capability to react to traffic disruptions, building and re-building in real time optimised plans, managing the order of trains, or using alternative routes.

Main functions

The main functions performed by TMS are:

- Conflict detection and resolution
- Complete schedule generation
- Advisory speeds computation
- Route booking

Theoretical foundations

The TMS internal model is based on the alternative graph scheme, originally developed for job shop problems and successively extended and adapted for the rail traffic scheduling problems. In the graph model the nodes represent the passage of trains at a give set of network points, and arcs represent time relations between nodes. A detailed description of the graph model can be found in the paper “A Traffic Management System for Real-Time Traffic optimization in Railways”, *Transportation Research Part B* 41 (2), pp. 246-274 (2007).

How it can be used

TMS is conceived to operate in a fully automatic way. The main current use of TMS is in combination with a train simulator, providing all train data and executing the suggested actions (advisory speed for trains, order changes, modified routings, route settings, ...). However, there are also several reduced operational modes, depending on the required level of automation. The simplest one is a DSS-mode, in which only the train scheduling function is active.

The intended use of TMS is in a traffic control centre to support dispatchers and train drivers. A successful proof-of-concept pilot has been executed at the Lage Zwaluwe junction (see the examples at the end of this paper).

TMS input

TMS manages both static and dynamic input data:

- Static data
 - Network topology (blocks, switches, signals, slopes, speed restrictions,...) including the length of infra
 - Maximum speeds on the infra (speed boards, signal aspects)
 - Minimum speeds on infra (e.g. on crossings with roads)
 - Signal positions and signal aspects patterns
 - Rolling stock characteristics (length, maximum speed, acceleration & braking curves, ...)
 - Train priorities (e.g. delays for an IC are weighted heavier than for a freight train)
 - Train routes (planned and alternatives)
 - Phased setting rules
- Dynamic data
 - Initial infra and traffic status
 - Initial and current plan
 - Connections between trains
 - Train initialisations (including trains expected to enter in the near future)
 - Train positions and speeds
 - Route that are set
 - Setting rules (Dutch: Instelvoorschriften)
 - Current delays

Static data are stored in a database, accessed by TMS either directly or by suitable translation software. Dynamic data are communicated to TMS by a message passing interface. Output data are managed by the same messaging scheme.

Currently, the availability of train position data at a sufficiently high rate (e.g. each 10 seconds) is considered as mandatory for TMS full operation. In absence of train position information, TMS could operate also with only track occupation information, but this is still a research issue.

TMS output

The final goal of TMS is to provide a smooth and efficient traffic regulation, through a continuous update of train speeds (by the communication of advisory speeds when required) and a progressive modification of the current train schedule in order to dynamically solve future conflicts and optimize the chosen cost function (e.g. the total expected delay). Optimization is the core activity of TMS and it is based on a set of heuristics allowing quick identification of the best set of train precedence rules. When it is needed, alternative routes are explored in order to achieve a better solution. Given routings and ordering of the trains, the running time/speed profiles are computed, and so the train schedule contains an estimate of passing time/speed of each train at every future control point on the network (goals). Control points are distributed in the model in such a way to avoid any possible ambiguity in train motion (block limits, switches, speed restriction boards and so on).

Operating cycle

At each cycle, TMS evaluates if a conflict will occur in the future, and it modifies the current schedule in order to solve it in the best way. Conflicts may arise in different ways (delayed trains, mismatches between predicted and actual positions/speeds, other disturbances). Conflict resolution is performed hierarchically by the following single actions, or by a combination of them:

- Adapting train speed
- Changing train precedence
- Modifying train routing

The best modified schedule is evaluated keeping in account all the possible railway constraints (e.g. blocking, connections, min/max speed, turning and many others). This is possible, as the railway constraints are modelled into the internal graph representation. The schedule is guaranteed to be feasible with respect to train dynamic characteristics, because acceleration/deceleration curves are explicitly used in travel time computations.

Finally, the schedule is also compatible with the already booked routes. Booking policy is a key aspect of TMS, because it is strongly related to traffic smoothness. Booking is performed generally as late as possible, in order to improve resource allocation. However, it can also be anticipated in order to allow the so-called “green wave” traffic flow.

Cost functions

The final goal is pursued at each level by optimizing cost functions whose selection can be tailored by the user needs. Currently we mainly focus on punctuality, averaged delay and energy consumption as major terms to be evaluated, but every other computable quantity can be used. Train speed optimal profiles are evaluated in order to maximize smoothness of the speed function itself, minimizing the number of speed changes.

Communication issues

TMS is conceived as a centralised system, managing messages coming from the field and producing new messages in order to execute the required actions. In this way a key issue is the communication delay between a datum (e.g. a train position) and the implementation of a suggested action (e.g. new target speed). In order to cope with such a delay (T seconds), TMS operates always in forecasting mode, that means it always evaluates train position/speed in the near future after T seconds according to the current schedule. That is, when sending a new target speed TMS takes care that it is correct for the moment T seconds later when it is actually

implemented. Doing so, delayed communication effects are circumvented and efficiency of the suggested actions is maintained.

TMS capabilities

A short list of the most important TMS capabilities, available with the current TMS version, is:

- Drive each train according to the railway constraints (e.g. speed restrictions, minimal headway).
- Deal with different train types and priorities (from freight to IC).
- Generate updated time/speed estimations of each train at a large set of control points in the future, allowing precise monitoring of the expected evolution of the current scenario.
- Ensure smooth speed profiles avoiding unnecessary speed changes.
- Forecast the future effect of each decision in order to select the best choice not for its immediate benefit, but for its global impact on delays.
- Exploit alternative routes in order to find the most efficient solution to conflicts.
- Ensure that every routing modification could be realized with proper and efficient information to dispatchers and customers.
- Model every type of network element, either for double or single direction.
- Model all types of railway constraints among different trains (connections, fixed order,...).
- Tune the objective function giving different emphasis on punctuality, delay, or energy consumption.
- Manage complex railway mechanisms like phased route setting and turning/coupling through of trains.
- Consider trains as physical objects, with length, mass and dynamic characteristics, described by specific laws.
- Deal with unavoidable communication delays, compensating them with short term forecasting.

Previous TMS versions showed the capabilities to address also the following issues. Through an adaptation process, they could be added to the current version too:

- Guarantee proper speed control on positive and negative slopes.
- Use coasting.
- Manage efficiently fixed block and moving block signalling, even in a mixed mode.
- Deal with infrastructure and train degradations.
- Cope with the presence of uncontrolled trains, not sending/receiving messages to/from TMS.

TMS possibilities

TMS is able to generate a new consistent global plan in the face of disturbances. TMS can show this plan to the dispatchers, together with the original plan, so that dispatchers can see the differences and can quickly see the consequences of the TMS proposals.

TMS can very well act as a decision support system, where even the level of automation can be decided upon (some routine task could be automated so that TMS measures are followed unless blocked by the dispatcher, while for other tasks TMS makes proposals and shows their consequences but these are not followed unless approved by the dispatcher).

Although not implemented, TMS is by construction and applied optimization method suitable to support the following functions:

- Present advices to dispatchers, who can accept or reject parts of the proposed solution and who can suggest or force their own measures. It can be shown if an advice involves a non-local order change (i.e. order change on open track) or the choice of an alternative route. Subsequently TMS takes these forced components of the desired solution into account as constraints in its optimization.
- Present advices to dispatchers and show the value of the optimization criterion for different solutions (e.g. doing nothing, following TMS, dispatcher proposal).

- Show a prediction of what will happen for a scenario (what-if simulator, possibly in combination with FRISO).
- Adapt the optimization criterion and time horizon over which this criterion is being evaluated.
- Send time-location-speed goal windows to train drivers instead of advisory speeds. Software in the train could support the driver to calculate a speed profile (part of SR or software developed by Prorail e.g. Economy meter). The train driver could notify the TMS if he no longer expects to be able to reach his current goal (expects to violate allocated slot). This is a more distributed system. Advantages are that more freedom and responsibility is given to the train drivers, and that train drivers and the support software implemented in the specific train can follow their own strategy and optimization criterion in reaching the given goal (punctuality, energy, throughput, interval). As a side effect the computational load decreases for the central system.

Note that TMS currently works with:

- Receiving train position and speed at a sufficiently high rate (down-link with trains).
- Sending advisory speeds to trains when needed (up-link with trains).

Considering the down-link, TMS could operate also with only section occupation information. This probably means less accurate data and hence reduce TMS performance. This is still a research issue.

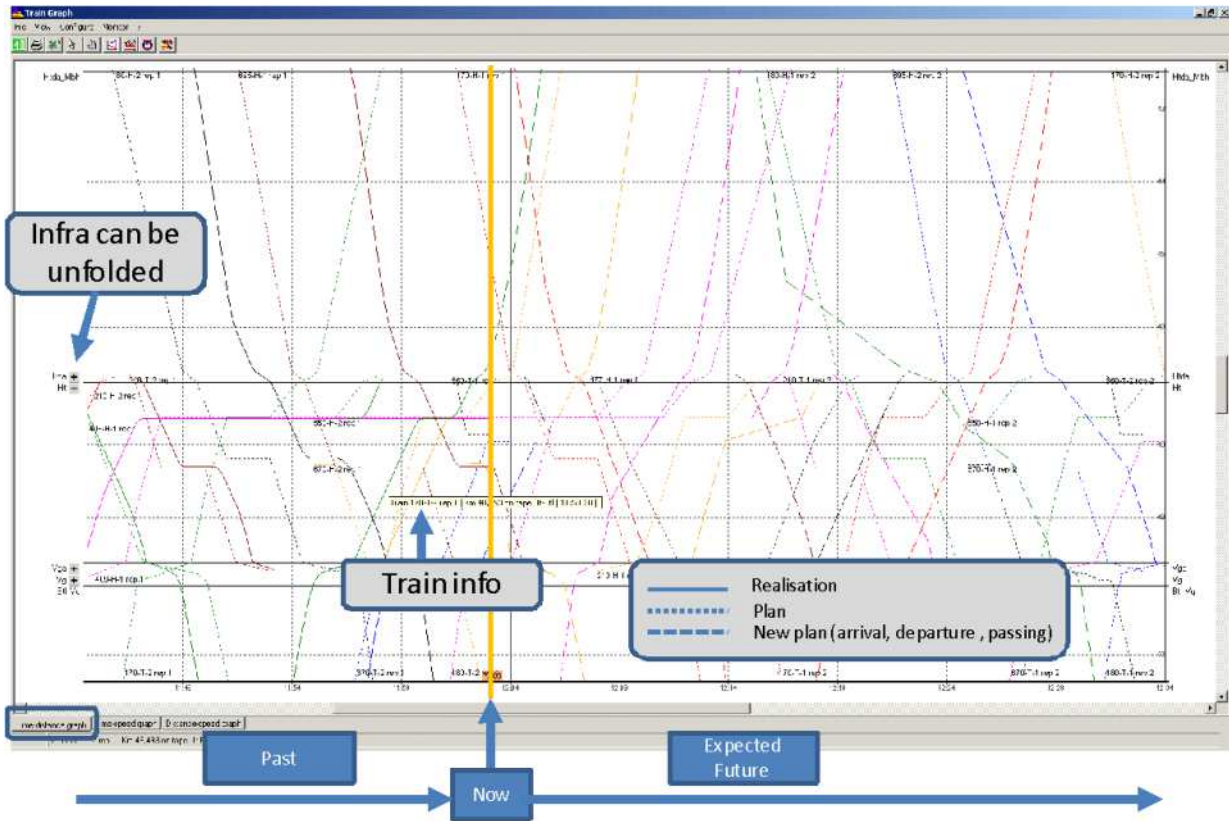
Considering the up-link, TMS can function without and only use route choice and timing to control the flow of traffic. Again, this will affect performance, trains may meet more yellow and red signals which affects punctuality, throughput and energy efficiency.

TMS GUI

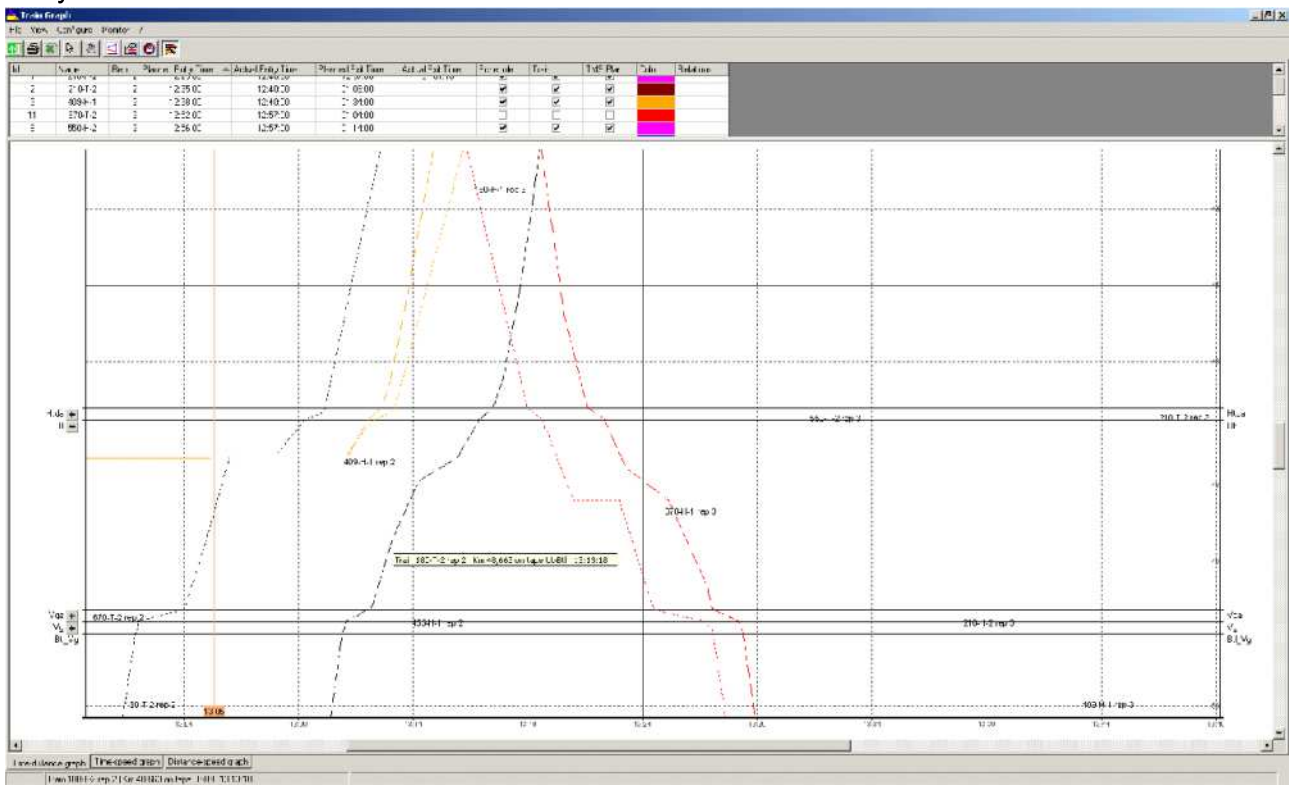
The TMS contains among more the following screens:

- Dynamic Time-distance graph, with original plan, new plan and plan execution.
- Dynamic Time-speed graph.
- Dynamic Distance-speed graph.
- Dynamic Graph with the infra-layout and current train positions and allowed possible routes.

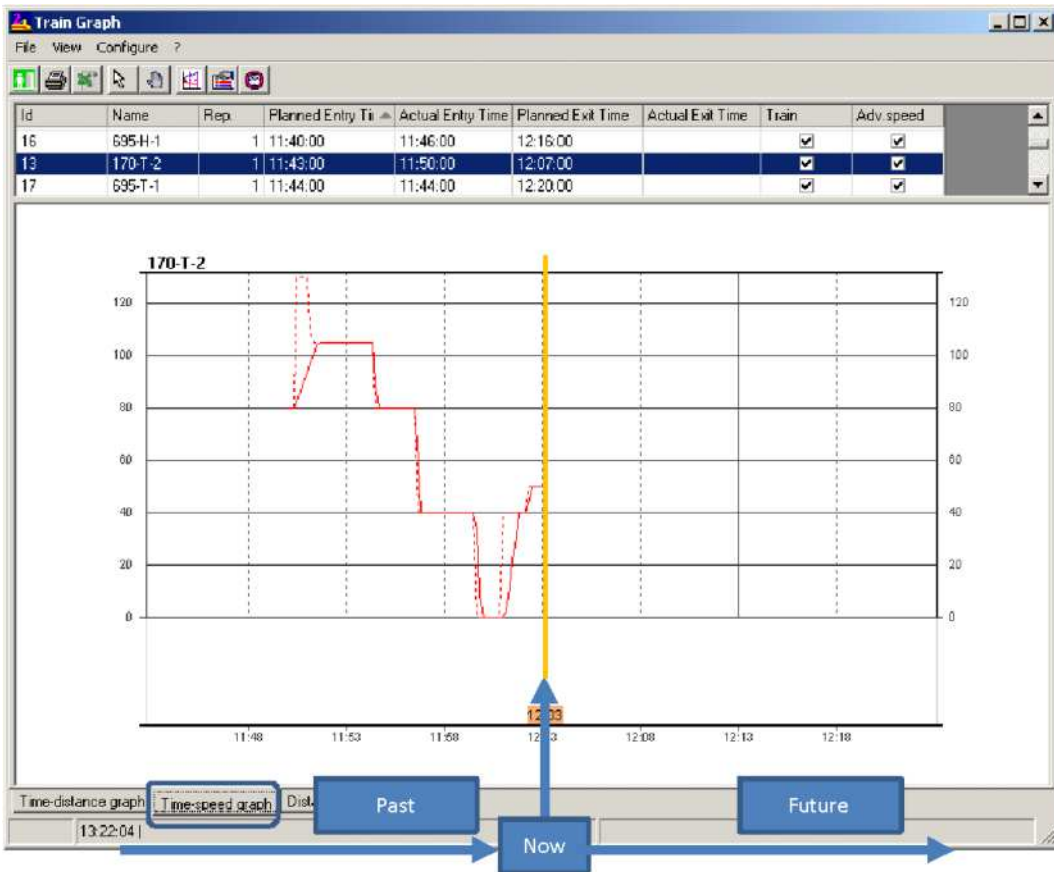
Time-distance



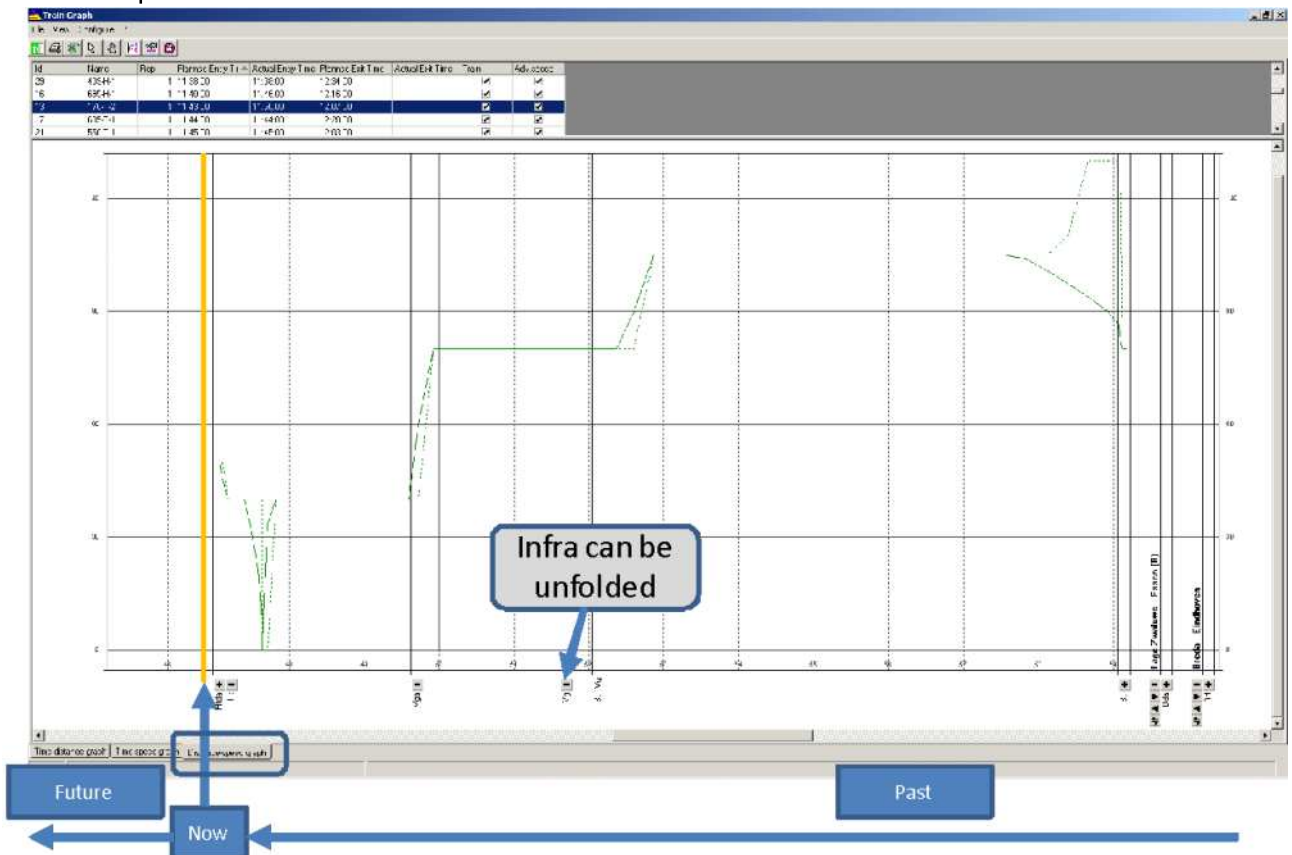
Time-distance, showing that TMS proposes an order change in Ht between train 180-T-2 which is delayed and train 409-H-1.



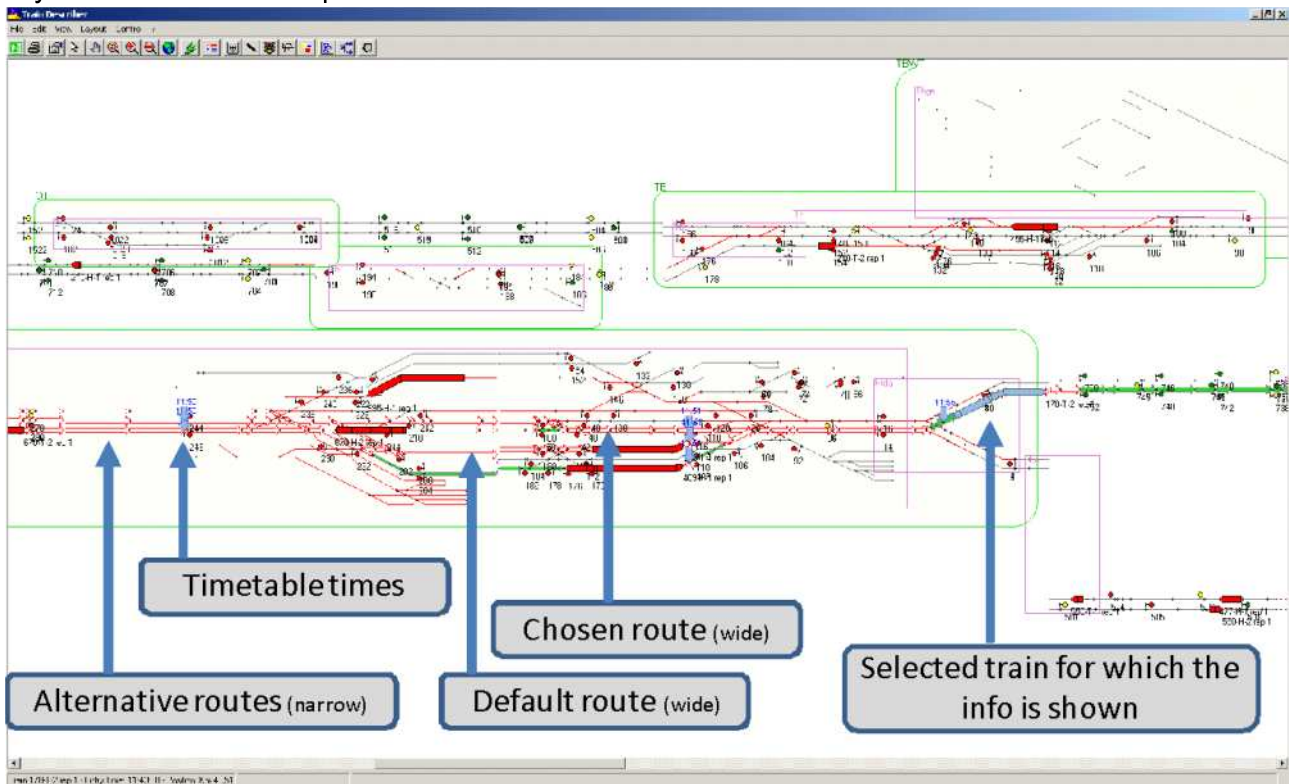
Time-speed



Distance-speed



Layout with current train positions



Examples of TMS results

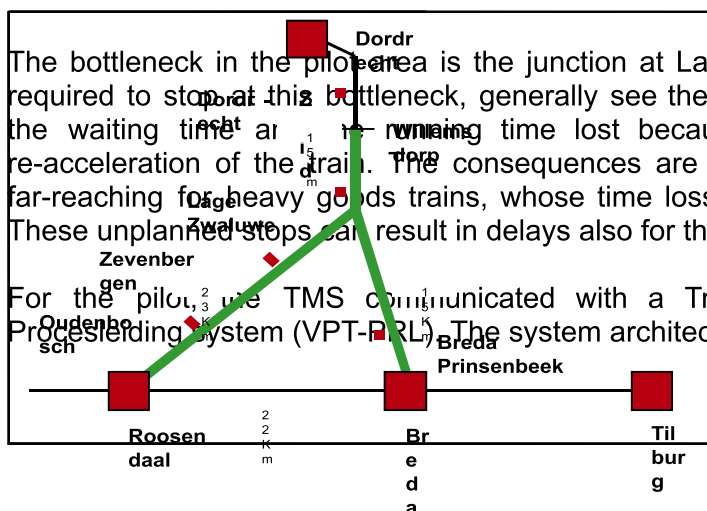
Below three examples of TMS applications will be given:

- Green Wave pilot implementation for Lage Zwaluwe junction.
- Schiphol simulation study.
- 's-Hertogenbosch simulation study.

The green wave 2004

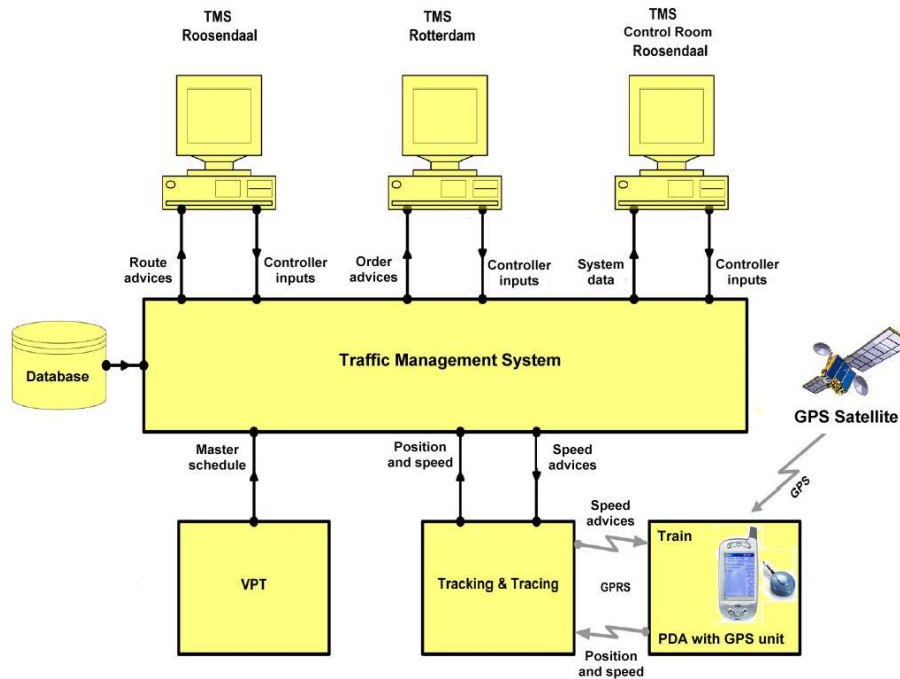
The pilot “The Green Wave”, actually called “de Groene Golf” (dGG) was carried out in June 2004, and was a first attempt to test TMS in practice.

The pilot area starts at Breda at signal 88 and at Roosendaal at signal 82 and ends at Willemsdorp at signal 615. The Breda and Roosendaal tracks are outside the scope of the pilot area.



The bottleneck in the pilot area is the junction at Lage Zwaluwe. Trains that, due to delays, are required to stop at this bottleneck, generally see their delay becoming ever longer, as a result of the waiting time and the remaining time lost because of the slowing down and subsequent re-acceleration of the train. The consequences are disadvantageous for all trains, but they are far-reaching for heavy goods trains, whose time losses can amount to as much as 10 minutes. These unplanned stops can result in delays also for the following trains.

For the pilot, the TMS communicated with a Tracking & Tracing system (T&T) and the Processing system (VPT-PRL). The system architecture is sketched below.



The TMS received, on a regular basis, position and speed information from the running trains, equipped by mobile units.

Combining the actions of CRS1 and SR, the TMS issues 3 types of recommendations, namely speed recommendations, order recommendations and route recommendations. Network controllers and train traffic controllers are the addressee of order and route recommendations. The speed recommendations are intended for the driver. By following the speed recommendations, the driver can ensure that his train is not forced to stop at a red signal. If network controllers and train traffic controllers comply with the order and route recommendations, the result is an optimum flow across the entire control area. In this way, the delay effects for other trains are minimised.

The main “quantities” of the pilot are:

- Test days: 11
- Pilot days: 8
- Train runs: 993
- Passenger trains: 766
- Freight trains: 211
- Special pilot trains: 16
- Train-attendants: 89
- Mobile units: 50

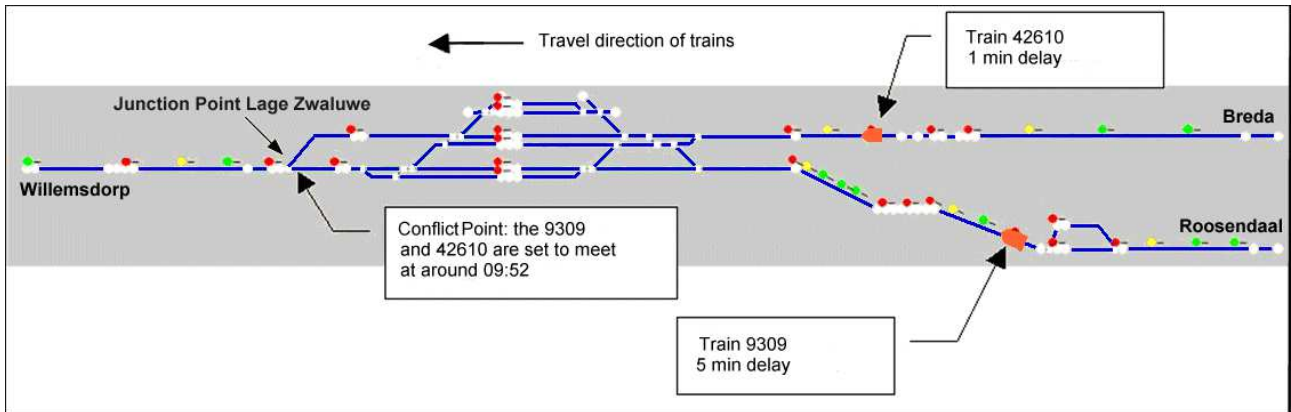
TMS must work in real-time, so its recommendations have to be produced as fast as possible. This can be a key aspect for order and routing recommendations, involving a new plan generation by CRS1. However, with TMS core modules (CRS1+SR) installed on a dedicated PC, TMS was able to produce a new a plan in a few seconds, allowing timely reaction to perturbations.

Results of the pilot

The effects of the TMS were demonstrated using a number of cases occurring during the pilot and analyzing the precise course of events. In some cases, it was verified that major problems caused by trains running a few minutes late, would in fact have been prevented by taking up the TMS plan and speed advices. An example from the analysed cases is reported in the following.

Starting situation

This example dates from 29/06/2004 at around 09:45 hours, and describes a case where the TMS came up with a solution for eradicating the delay, for a number of trains.



Train	Type	From	Planned Time	Entry	Entry Delay
9309	Thalys	Roosendaal	09:34:34		+5 min.
42610	Freight	Breda	09:38:08		+1 min.
5128	Stop train	Roosendaal	09:39:30		+4 min.
2528	Intercity	Breda	09:47:29		On time
					(+10 min.)

In this practical example, in particular the 9309 and the 5128 from Roosendaal suffered some delay. In this example, an indication is provided of what recommendation was issued by the TMS and what happened when a train ignored the recommendation. By way of comparison, it also is indicated what would have happened if the train in question had taken up the recommendation.

The TMS recommendation

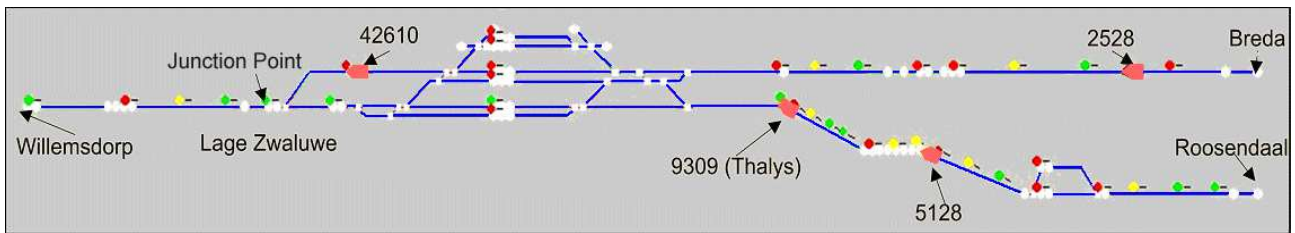
If the trains had travelled according to the plan, nothing would have gone wrong. The Thalys (9309) would have arrived at the junction point at Lage Zwaluwe well before the 42610. The two trains could then subsequently have continued on their way, unhindered.

Due to the delay of the 9309 (5 minutes), the situation turned out to be very different. TMS detected that a conflict was due to occur at around 9:52 at the junction point at Lage Zwaluwe, and calculated the best means of avoiding the conflict. In this solution, TMS also included the subsequent trains (5128 and 2528) and TMS attempted to maintain the original order, in other words first the 9309 and then the 42610.

In this schedule, the 42610 should have been slightly delayed by reducing the recommended speed to prevent the train having to stop at a red signal. At the same time, the 9309 should have been slightly accelerated to maximum section speed, in order to in part make up the area entry delay and at the same time to delay the 42610 as little as possible.

Result of not taking up the recommendation

In practice, no agreement had yet been made with the drivers to take up the TMS recommendations. Actually, the 9309 did take up the recommendation, but the 42610 ignored it. The 42610 travelled at a speed of 65 km/hour rather than the recommended 40 km/hour. As a consequence, the 42610 was forced to stop at the red signal at the junction point at Lage Zwaluwe. The figure below shows that the Thalys was granted its route across Lage Zwaluwe (red line) whilst the 42610 had to wait for a red signal. The other two trains (2528 and 5128) are also visible in this picture.



The table below shows the consequences of ignoring the recommendation. Failing to take up the speed recommendation by the 42610 resulted in a total delay of 14 minutes.

Train	Type	Entry Delay	Exit Delay	Extra Delay
9309	Thalys	+5 min.	+4 min.	-1 min.
42610	Freight	+1 min.	+7 min.	+6 min.
5128	Stop train	+4 min.	+6 min.	+2 min.
2528	Intercity	On time	+7 min.	+7 min.
		(+10 min.)	(+24 min.)	(+14 min.)

Result if TMS recommendation had been taken up

From the analysis of TMS calculation, it emerged that the above-described additional delay would not have occurred if the 42610 had taken up the recommendation. If we assume that the Thalys in that case had also stuck to the recommendations, the 42610 would have reached the junction point several minutes later, at the recommended speed of approximately 40 km/h. As a result of which the train would not have been forced to halt for a red signal, but would have been able to pass through a green signal.

Train	Type	Entry Delay	Exit Delay	Extra Delay
9309	Thalys	+5 min.	+4 min.	-1 min.
42610	Freight	+1 min.	+2 min.	+1 min.
5128	Stop train	+4 min.	+2 min.	-2 min.
2528	Intercity	On time	+1 min.	+1 min.
		(+10 min.)	(+9 min.)	(-1 min.)

Differences

On balance, taking up the speed recommendation represents a gain of 15 delay minutes. In terms of punctuality, whereby a train is only in fact delayed if the delay is greater than 3 minutes, this means an output punctuality of 75% as opposed to an initial punctuality of 50%. Because in the practical situation the recommendation was not taken up, all four trains were delayed, resulting in an output punctuality of 0%.

Conclusion

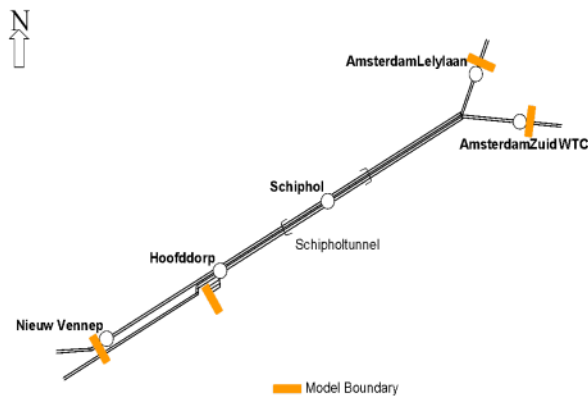
This example shows the strength of the TMS for improving punctuality (and the secondary objective: saving fuel by avoiding a stop) by preventing conflicts and guiding trains on the basis of recommended speeds. The example also shows that an apparently simple problem, namely a Thalys running a few minutes late, can grow into a considerable disruption with major delays for other trains.

A global estimation, based on the analysis of a number of practical situations occurring during pilot days and on feedbacks from the involved personnel, was carried out. These figures and the other experiences acquired during the trial indicate that the TMS does have sufficient capacity to become a positive addition to the existing systems. The pilot demonstrated that on a weekday, gains of approximately 210 train delay minutes could be achieved for the entire pilot area, when equipped to cover all trains. On an annual basis, the gain for the pilot could be of 1100 delay hours. The pilot has also demonstrated that the forecast energy saving of 10% is easily achieved.

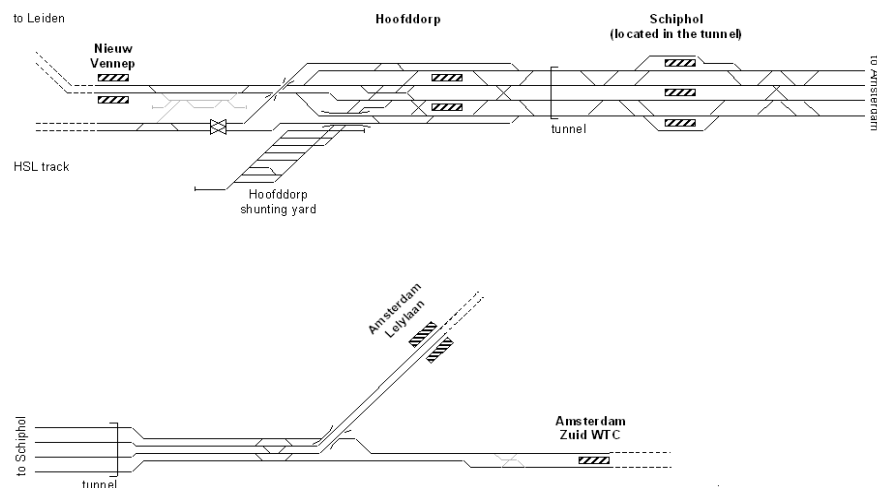
The Schiphol bottleneck 2007

A study was carried out to verify if the use of a TMS could be effective to manage the forecasted increased traffic at Schiphol bottleneck in 2007. The study addressed the problem of efficiently using the capacity of the existing railway network and improving reliability and punctuality of train operations. The study verified the effectiveness of the TMS global approach by means of simulations on different traffic scenarios.

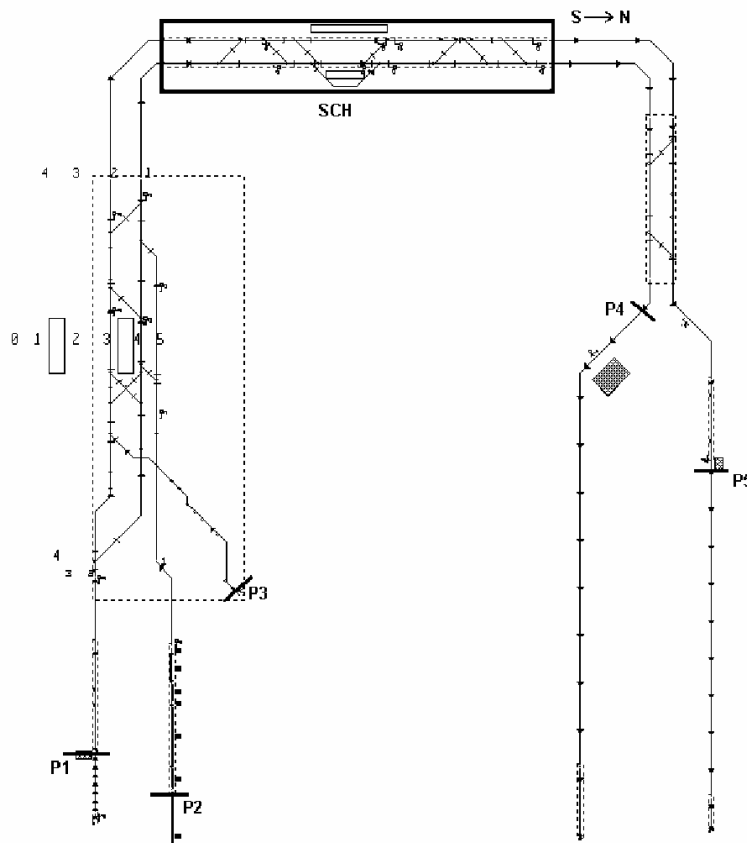
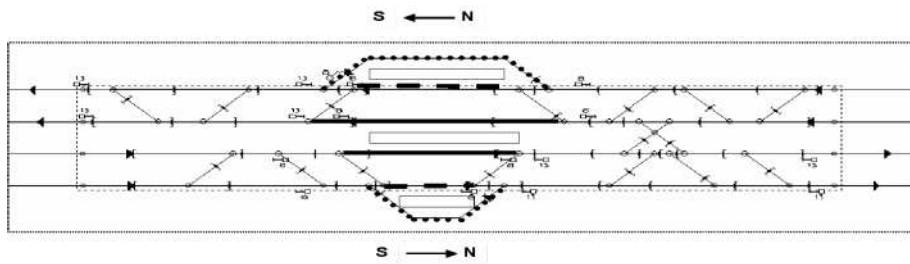
The research area comprises all tracks between Nieuw Venneep on the east side, Amsterdam Lelylaan on the north side and Amsterdam Zuid WTC on the south side. At the Hoofddorp location a number of storage sidings operate as origin or destination for trains.



The next figure gives an overview of the infrastructure, including the 4-track bottleneck at Schiphol, where in each direction 27 trains will run. This involves trains in/from two directions northbound (Amsterdam Zuid WTC and Amsterdam Lelylaan) via three platform tracks and three directions southbound (HSL, Leiden, shunting yard) via three platform tracks.



The next figure shows the standard routes per line at Schiphol. For each direction, 9 possible routes have been considered in schedule optimisation.



In the figure, “SCH” identifies the Schiphol station, concerning possible routes and platform tracks (Northbound direction). A set of points P1...P5 identifies the Schiphol area limits.

The train characteristics considered in the model are summarised in the following table.

Rolling stock	Length (m)	Weight (ton)	Max speed (km/h)
TGV-P 10	200	417	220
ICE22 8	200	465	220
ICM3 6	161	287	160
IRM3 9	242	431	160
IRM4 8	214	394	160
E17D2 Dd 8	245	550	140
E17D2 Dd 6	193	459	140

The study considered different values for the number of trains per hour (19, 23, 27, 29, 32), starting from reference timetables. Two different kinds of perturbations were added to each timetable:

- entry delays for approaching trains (randomly sampled from a “Pearson T5” distribution)
- stop extensions at Schiphol and Hoofddorp (randomly sampled from a normal distribution)

According to the above context (timetable + perturbations), a large set of different realisations was generated, each one considering 8 hours of traffic. Every 10 min, CRS1 generated a new optimal conflict-free schedule for all the trains inside the controlled area, by acting on both precedence relations and track changes (allowed only in the Schiphol station).

Results of the Schiphol study

The overall result from the study is that it is possible to manage the requested 27 trains per hour timetable through the Schiphol bottleneck, so the existing infrastructure is adequate for the objective. The result came from a relevant number of simulation hours, leading to about 2600 averaged trains for each one of the 5 analysed timetables.

The study showed the benefits that can be expected. The most important are:

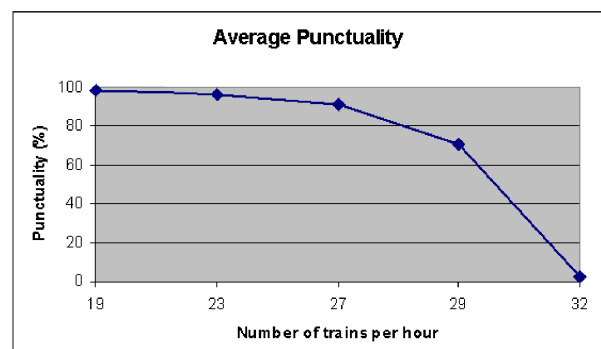
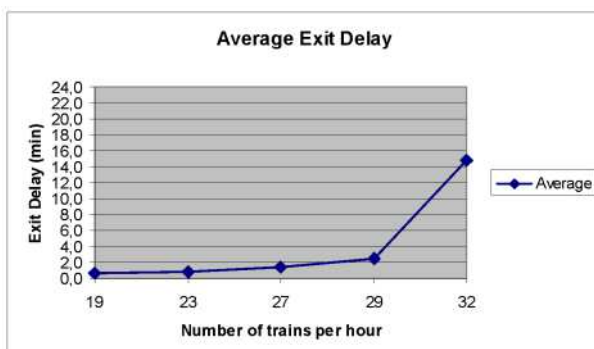
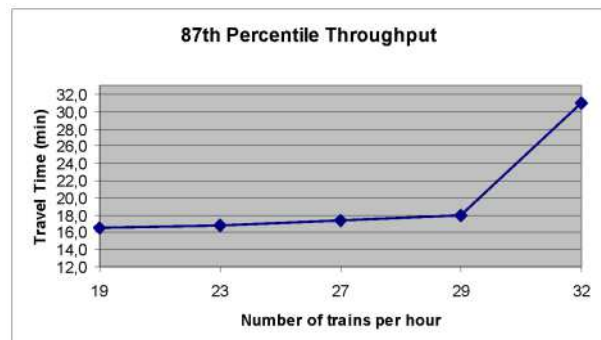
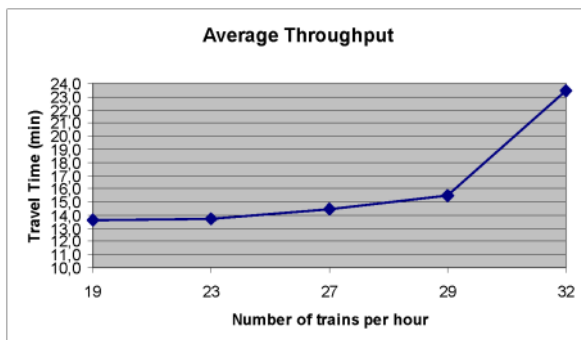
- travel times are lower and reliability is higher.

27 trains per hour	Average Throughput Time [min.]	
Direction	Reference	CRS1
Northbound	17,8	14,4
Southbound	17,5	15,0

- severe disturbances can be handled easier, or bad consequences can be weakened.
- the energy consumption is lower.
- there is a better overall control of the traffic.

Below, the most important results of this study are reported by synthesis graphs.

- Average Throughput (number of trans per hour)
- Percentile Travel Time (87th percentile of the travel time distribution)
- Exit delay
- Punctuality (many trains have an exit delay less than 3 minutes)

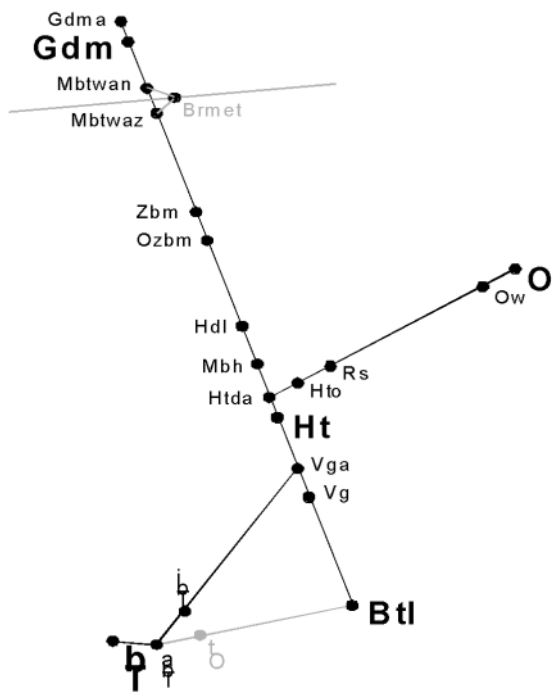


The graphs show that 27 trains per hour can be managed with reasonable performance degradations in all the computed parameters. However, a further increase in train density is not sustainable. This is due to the Schiphol station bottleneck where the 3 existing platforms and the approaching tracks are not sufficient to handle the traffic (with perturbations) at more than 30 trains per hour.

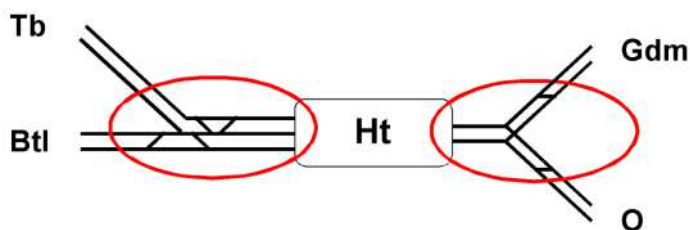
's-Hertogenbosch 2009

A simulation study has been executed for an area of reasonable size and complexity, where an actual control challenge currently exists. TMS results have been compared statistically to a number of other strategies (FCFS, VaVo: FCFS with maintaining timetable order for trains having the same end-track in a route).

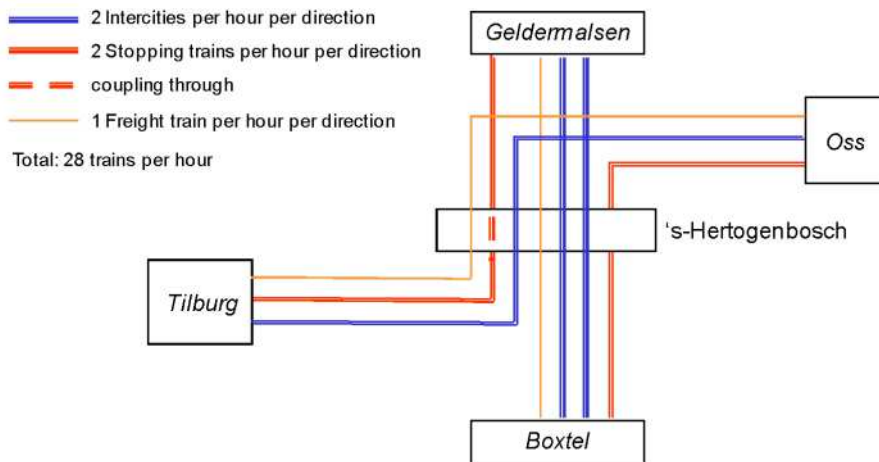
The study area is delimited by and including Tilburg and Boxtel on one end and by and including Geldermalsen and Oss on the other end.



The main conflict area's are at Vught Aansluiting (Vga) and at Diezebrug Aansluiting (Htda) and also in 's-Hertogenbosch (Ht) itself. Additionally, headway conflicts frequently occur on the open tracks (e.g. between intercities and stopping trains or freight trains).



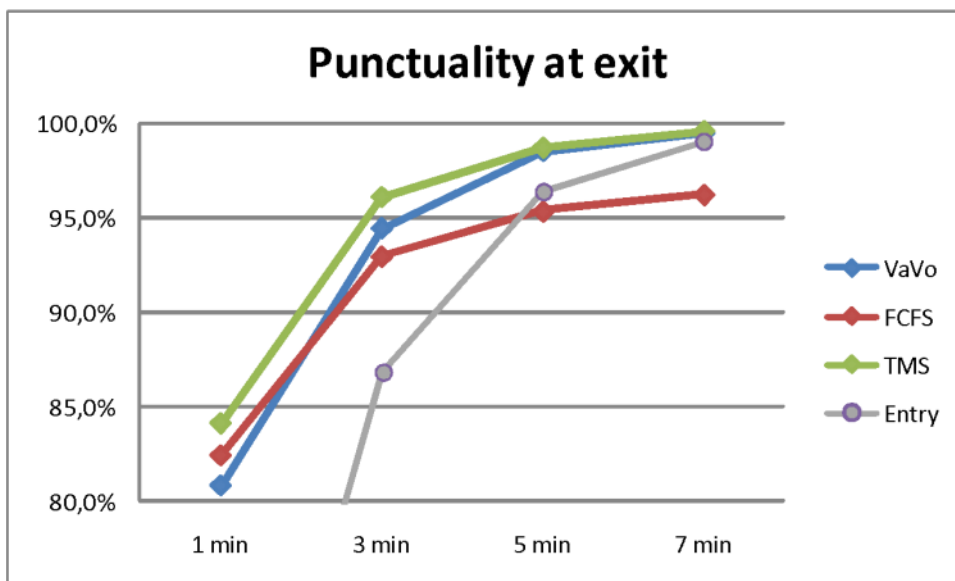
The timetable contains 28 trains per hour.



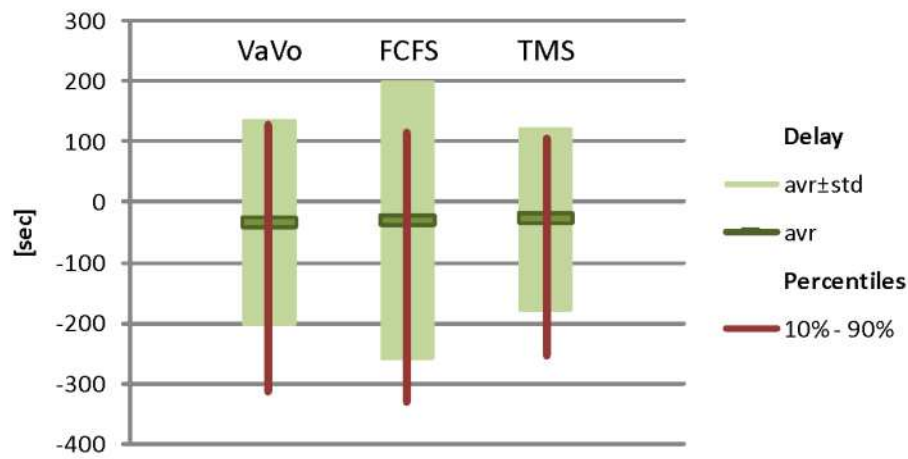
Entry delays in the simulation are generated from distributions based on measured data (peak hours 2007).

Simulation results have been gathered over 48 hours (12 runs of 6 hours where the first 2 hours are discarded as warm-up time).

Punctuality at exit and bandwidths at exit are depicted below. It can be seen that TMS performs better than FCFS and VaVo. In turn VaVo outperforms FCFS, which is not surprising since disturbances are not large. Comparing TMS with VaVo on punctuality, TMS has significantly more trains with delays less than 1 minute or 3 minutes. Also, looking at the percentiles and standard deviations, TMS achieves more compact intervals containing most trains (smaller deviations from planned timetable paths).



Bandwidth at exit



Comparison with other methods

A short comparison on a conceptual level will be given with:

- SMD.
- ROMA.

SMD

SMD can be seen as a method to generate a set of priority rules for trains approaching a junction, depending on different levels of delay and current speed of the trains approaching the junction and achievable speed on the merging track (which can be inferred from the position and speed of trains on that track plus the block layout and signal aspect relations). So, an exhaustive set of TAD rules is generated.

SMD is a stochastic approach and is timetable-free, focusing on the conflict and aiming at minimizing the average stay time of trains instead of their delays. It has been shown that SMD also performs quite good when looking at delays and on a complex corridor (Utrecht-Gouda).

Conversely, TMS currently aims at minimizing delays (deviations from the timetable). However, another suitable cost function also could be employed, for instance throughput or stay time.

Both SMD and TMS produce advices for the train order.

Both approximate the true optimal solution.

At the moment a comparison on simulation level for an actual case is not available.

Below, SMD is explored in more detail, in order to present a fair comparison scheme with TMS and to delineate a possible cooperation between them.

Note that SMD addresses the Conflict Detection and Resolution (CDR) problem in railways. It provides order choices for trains approaching a junction. It does not provide time-speed goals, advisory speeds, route setting times or route choices (which the TMS also does provide). So, a fair comparison should concentrate on the CDR part of both approaches.

TMS as a CDR method

The CDR module of the TMS is inside the CRS1 process and it is responsible for the generation of an optimised schedule, minimizing the total delay of the trains.

In TMS the CDR problem is tackled with a graph optimization approach, where the graph represents how trains use infrastructure, according to the time table. Using the graph, TMS (specifically, the CDR module of CRS1) foresees conflicts and solves them by looking for minimal paths in the graph. In order to avoid an exhaustive path enumeration, some heuristics have been developed in the past. The one currently used includes the best ideas that emerged during years of tests. Let us call it the "TMS-CDR heuristic". It is based on a forecasting of different delay patterns produced by different ordering choices, and the delay forecasting is implicitly produced by the graph itself.

Still, let us recall that the final aim of the real-time scheduler (i.e. the CRS1 process) is not only to define the best train ordering at confluences, but in general to provide an updated set of time-speed goals for all trains. CDR is a prerequisite for conflict-free scheduling, but it should not be confused with the full CRS1 process.

SMD as a CDR method

SMD is a statistical method to provide an optimal ordering of trains at a confluence, based on a rough description of the layout (in terms of blocks), block occupation and train attributes (delay, speed and type). Optimization of a suitable cost function is performed off-line and the output is stored as a set of rules, similar to the TAD rules used by dispatchers in the Netherlands to solve train conflicts. In this sense, SMD seems a good way to get more effective rules using statistical information. Doing so, rules can be more robust to schedule deviation, and they can be defined even without a predefined timetable. The underlying idea is that railway traffic is often unpredictable (even in the near future) and so only statistical information is reliable.

At runtime, the system must recognize which is the correct rule to be applied, looking at train positions, speeds and delays. Then it can apply the train ordering suggested by the best matched rule.

TMS-CDR versus SMD

By the above observation a direct comparison between TMS and SMD should be performed by focusing on the two CDR approaches, maintaining fixed any other aspect. This is what has already been done to compare SMD with usual railway heuristics like FCFS, showing the superior performances of SMD.

However this kind of comparison between TMS-CDR and SMD is not possible at present, because SMD should be included in FRISO stand-alone, while the TMS-CDR heuristic is of course part of the TMS itself. So the “natural” solution is to look at SMD as a FRISO improvement and to compare FRISO stand-alone (with SMD) and FRISO-TMS as usual. Doing so, the focus of the comparison is partially lost, because the different environments mask the effects of the different ordering choices. Apart from ordering choices, TMS for instance also can influence train speeds (advantage) and suffers from communication delays (disadvantage).

Possible cooperation TMS-SMD

Looking at SMD as a statistical way to generate effective TAD rules, a possible cooperation between TMS and SMD can be conceived as follows. TMS may access a set of rules and use them as a suggestion at runtime. A new source of knowledge about best train ordering may be useful to deal with cases in which the heuristic is not able to identify a choice which is definitely better than the other. In fact, the different ordering decisions commonly taken by TMS may be classified as follows.

- Nearly obvious decisions: one choice is much better than the others, so that every heuristic should take the same decision (e.g. the involved trains are well separated in time, so the conflict does not occur in practice).
- Method-dependent decisions: the choice depends on the currently applied heuristic, but analyzing data it is possible to decide which is the best one (e.g. looking at the total delay it can be proved that one decision produces a delay value meaningfully lower than the others).
- Controversial decisions: the different choices produce similar results, and it is not clear which is the best one. Only an external source, like dispatcher knowledge, may suggest the best choice (e.g. the total delays are similar but the delay distributions over the trains are different).

In the latter case, the SMD ordering rules, being based not only on the current data, but keeping in account the statistical and repetitive nature of the problem, could enrich the TMS-CDR decision making. For example, the TMS-CDR could include the train ordering suggested by SMD as one of the choices and use it if it performs better than other choices. So, instead of one or the other, both approaches can cooperate.

Note that this is true more generally. If the optimisation criterion can be clearly formulated and a fast algorithm/simulation is available that can forecast how different choices (train speeds, train orders, route setting times, route choices, or even plan changes) affect this criterion, then different methods can be combined (TMS, SMD, dispatcher ideas, and so on) and the best solution can be applied, where of course the dispatcher has the final choice.

Final remarks

To conclude, a few remarks concerning the feasibility of a statistical approach to the railway scheduling problem.

Given the nature of the problem and the unavoidable perturbations in traffic, predicting the future effects of a decision can be quite difficult (in TMS-CDR this task is accomplished by the graph-based approach). The periodicity of the timetable allows the accumulation of statistical evidence that can be used to define robust rules to be applied when needed.

However, the learning of the rules, which is performed offline, requires a stable situation in terms of traffic flow. That is, any change in the timetable structure (e.g. adding a new train or changing the frequency of a service) could affect the optimality of the learned rules. The implicit hypothesis in a statistical approach is that there are no modifications of operating condition between learning

phase and running phase. This partially can be dealt with by making SMD rules for a number of expected operating conditions.

On the other hand, the TMS rescheduling approach, being based only on current data, may deal with every type of modification occurring at run time (new trains, changed schedules, blocked tracks, etc.), simply generating the new schedule on-the-fly.

ROMA

On a conceptual level ROMA and TMS are quite similar. They both aim at a global optimization of train traffic over a network. Both use a two-level approach. In the first level both use a CDR module based on the alternative graph formulation to compute conflict-free schedules with admissible speed profiles. This involves re-timing, re-ordering and re-routing. The second level in both approaches is to detail the speed profiles (and go back to the first step if no acceptable profile can be found).

The two approaches differ in the algorithms used in computation and in the way constraints are taken into account (how they are formulated and maybe on which level they are applied).

ROMA may offer benefits in producing more optimized schedules in terms of deeper search of the solution space. However, when dealing with congested time tables and with unpredictable field behaviour (e.g. delayed route setting, variable communication delays with the trains, entry delays, stop and departure disturbances, etc.) a given schedule, even if optimal, rarely survives for a long time. Updates are always necessary, so TMS preferred to get a good solution in a short time and go further instead of consume time to get a better one, which could be already out-of-date when produced. Obviously, if a better solution could be produced in the same time with other algorithms these are very welcome.

At the moment a comparison on simulation level for an actual case is not available.

Also, a direct comparison between TMS and ROMA will be possible only when ROMA also performs all the tasks currently performed by TMS, but not directly related with scheduling. For instance, the route booking task. This is probably the most critical issue with respect to performance. Every error in the booking/setting process (e.g. a delayed route setting) can destroy the best schedule. A wrong desired setting time in a route booking may introduce a perturbation in train behaviour, invalidating the most optimal schedule.