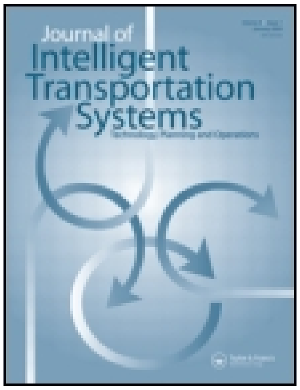

Journal of Intelligent Transportation Systems: Technology, Planning, and Operations
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gits20

A Discrete Event-Based Simulation Model for Real-Time Traffic Management in Railways

Jose Luis Espinosa-Aranda [a] & Ricardo García-Ródenas [a]

[a] Escuela Superior de Informíca, Universidad de Castilla La Mancha, Ciudad Real, Spain

Available online: 09 Mar 2012

Journal of Intelligent Transportation Systems, 16(2):94–107, 2012

ISSN: 1547-2450 print / 1547-2442 online
DOI: 10.1080/15472450.2012.671713

A Discrete Event-Based Simulation Model for Real-Time Traffic Management in Railways

JOSE LUIS ESPINOSA-ARANDA and RICARDO GARCÍA-RÓDENAS
Escuela Superior de Informíca, Universidad de Castilla La Mancha, Ciudad Real, Spain

Rail systems are highly complex and their control requires mathematical-computational tools. The main drawback of the models used to represent railway traffic, and to resolve any conflicts that occur, is the large computational time needed to obtain satisfactory results. Therefore, the purpose of this article is to study and design a discrete event-based model, characterized by the positioning of trains in block sections, that can represent the rail system, including the dynamic aspect, and a fixed block signaling system able to proactively detect and resolve potential conflicts that may occur within this system. This model can be developed and integrated as a specific module of a railway passenger-dedicated intelligent transportation systems (ITS) oriented to real-time traffic management in order to improve the service quality of a railway system's operations. The aim is to reduce the computational cost as much as possible and implement the proposed model in a railway network. A numerical investigation based on the Renfe Cercanís Madrid rail network (Spain) shows the high computational performance of the proposed approach in a real-life application context.

Keywords Discrete Event Modeling; Railway Traffic Management; Real-Time Traffic; System Capacity Check; Train Control

INTRODUCTION

Railway systems provide large-scale public transport services in an eco-friendly and sustainable manner. In order to increase performance and maintain a high-quality service, new strategies for increasing capacity are needed, including the building of new infrastructures and the improvement of existing ones. The improvement of these railway infrastructures requires heavy economic investment, which will on occasions require limited changes to the existing infrastructure, which may not be possible for other reasons. Therefore, production-based strategies for increasing capacity by allowing more trains to be operated on the same infrastructure, or by train scheduling, are being developed.

This research has been partly financed by the Ministerio de Fomento (Development Ministry) via CEDEX and the PT-2007–003-08CCPP research project. Our thanks to Renfe Operadora for provision of the data to test the model and the tool, and for the assistance given in the development of the SIMEIFER tool.

Address correspondence to Ricardo García-Ródenas, Escuela Superior de Informática, Universidad de Castilla La Mancha, C/ Paseo de la Universidad, 4 13004 Ciudad Real, Spain. E-mail: Ricardo.Garcia@uclm.es

Within the normal working of railway systems, such systems may suffer technical failures and disturbances, causing what are known as *primary delays*. The interaction between trains as a result of these primary delays may cause knock-on effects among other rail services, causing *secondary delays*. This problem is being addressed via an *off-line* approach, developing analytical methods for obtaining robust timetables that are able to deal with minor delays occurring in real-time, and *on-line*, by new technology that introduces intelligent transportation systems (ITS) (Zandi & Tavana, 2011); these allow railway operators to improve traffic control actions that reduce buffers between trains and that improve line capacity.

These ITS are designed as decision support systems for dispatchers, and are divided into distinct modules. This type of design allows better maintenance and modification of the ITS since the adjustment of any of the components or the existence of any internal error will not affect the rest of the system; furthermore, any of the modules may be used individually. A prototype design/architecture based on ROMA (D′Ariano & Pranzo, 2009) is shown in Figure 1, in which three main modules may be observed:

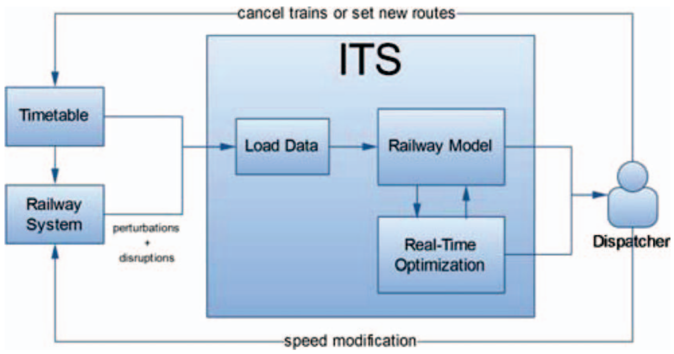

Figure 1 ITS architecture (color figure available online).

1. Load Data module. This module has the function of obtaining real-time system state data. It loads information on the *master schedule* and periodically verifies the state of the lines to check that the service schedules are being satisfied.
2. Railway Model module. The information obtained by the previous module is sent to the *Railway Model* module, which assesses the current state of the system and the initial schedule with the aim of anticipating possible disruptions, secondary delays, and problems that could occur on the railway system. This module is responsible for automatic traffic supervision.
3. Real-Time Optimization module (Corman, D′Ariano, Pacciarelli, & Pranzo, 2010; D′Ariano, Corman, Pacciarelli, & Pranzo, 2008). If the *Railway Model* detects the existence of a disturbance it will activate the *Real-Time Optimization module,* which resolves possible conflicts and offers solutions that minimize as far as possible both the knock-on effects of delays and any other situation which may disrupt the system. This module is responsible for real-time train scheduling and routing. This module may also compute the optimal speed profile for each train (Corman, D′Ariano, Pacciarelli, & Pranzo, 2009; D′Ariano, Pranzo, & Hansen, 2007).

The ITS system assists the dispatcher in the taking of decisions, making real-time adjustments either to the original network schedule or the planned services, by canceling trains, setting new routes, modifying speeds, or using other rail traffic control strategies.

Simulation models provide a detailed and dynamic description of the railway system and are the ideal analytical tool for representing the *Railway Model module*.

This article concentrates on providing a computational assessment of the performance of a new simulation model for the real-time management of railway traffic representing the *Railway Model* module by an ITS as a part of the complete Decision Support System. The article uses a simulation scheme for discrete events defined by the location and speed of the trains on the block sections (Flamini & Pacciarelli, 2008) and the signal controls.

The simulation model described in this article is able to represent microscopically the working of existing rail systems with reduced computational cost in order to facilitate its future inclusion in an ITS, as shown in Figure 1. In this way it can also be used to predict the future state of the system, enabling possible disruptions to be anticipated and different options for system planning to be assessed.

The numerical experiments carried out on a real problem show the suitability of the model for on-line use, as well as other, off-line, uses, such as assessing alternative operation plans or emergency plans.

The article is organized as follows. The second section discusses previous research, the third section proposes an improved simulation model based on discrete events and presents the algorithm for solving the simulation model, in the fourth section the software application implemented (SIMEIFER) is presented and two computational experiments are reported, and finally the article concludes with a discussion of our findings and future work.

PAST RESEARCH

Rail systems are highly complex and their planning and management require mathematical-computational tools. Traditionally a set of stages has been used, with different time horizons, which deal with certain types of established problems (D′Ariano, 2008).

A great deal of research is currently being carried out to find solutions to all these problems, with the aim of improving the quality of service provided. These approaches are generally divided into two areas, the off-line management of rail traffic, which focuses on the study of planning rail schedules from scratch as effectively as possible, and the real-time management of rail traffic, which deals with the solving of problems where there is already a schedule.

Where there is a predefined schedule, although it may be highly robust (Goverde, 2005) and reliable (Goverde & Hansen, 2000; Landex, Kaas, & Hansen, 2006), in unexpected cases or specific situations it is not able to respond appropriately to the changing needs of the system, nor to fulfil the original planning criteria (Vromans, Dekker, & Kroon, 2006), in which case it is necessary to find, within a time window, a solution that can return the system to the scheduled state or to a state in which new disturbances are reduced as much as possible.

Furthermore, in order to be in a position to produce the most effective schedules it is necessary to make an estimate of the waiting times in the system. To retain validity when they are transferred to the system, various approaches have been used, such as the addition of buffer times to compensate for delays to trains (Nie & Hansen, 2005) or canceling a particular train departure because of the problems it causes to the rest of the schedule (Wendler, 2007).

Recently the train scheduling problem has attracted great attention since several tracks are using new technologies that allow the dispatcher to know the position and speed of the trains in real time. This information can be used to develop analytic tools capable of anticipating conflicts of rail traffic that can cause delays in the railway system. One approach to solving this problem in real time has focused on finding optimal solutions to certain nonlinear programming models (Cordeau, Toth, & Vigo, 1998; Szpigel, 1973). This type of methodology has two major disadvantages:

1. Despite the high computational capacity of modern computers, solving a nonlinear programming problem with constraints for each service would involve a very high computational cost. For some problems this computational burden makes these methods unviable.
2. If a solution of this type is used, each time that a train cannot fulfill its predetermined timetable, the entire plan must be recalculated from the current state of the network.

One solution recently proposed in the literature is based on simulation models (Baohua, Wenzheng, Shaokuan, & Jianfeng, 2007; Ho, Mao, Yuan, Liu, & Fung, 2002; Martin, 1999), more exactly based on discrete events, incorporating into the models the concept of travel advanced strategy (TAS) (Dorfman & Medanic, 2004). These models reduce the complexity of the calculations and eliminate the need to recalculate all the schedules each time a delay occurs. A similar approach is used in (Li, Gao, Li, & Yang, 2008). These authors use a more effective strategy than the TAS, since they consider the speed of the train on a block section as a decision variable to represent different changes. Another approach is followed in Burdett and Kozan (2009) using compound buffers to maintain the correct occupancy levels of lines while allowing trains to pass through crossover points without additional routing decisions.

Simulation models have been applied in order to describe train movements. Depending on the level of detail required, there are two scheme types available: time-based models, and event-based models. The first of these has a high computational cost and is applied to specific problems of energy consumption and signal layout design. With the second scheme it is possible to describe the movements of trains, such as arrivals and departures at and from stations and behavior at intersections, as events. At the same time, this kind of model can be classified according to the basic principles of signaling. While on long-distance and high-speed routes moving-block signaling, which is widely treated in the literature, is used, on urban routes fixed-block signaling is more commonly used, and it is this system that is the main focus of this article.

Although the simulation uses algorithms of mathematical models, these simulations can also be combined with the real state of a system by data input/output when it is operating, which is useful for the real-time study of its behavior and to make predictions about it, intending to launch an optimizer similar to that presented in ROMA software (Corman, D′Ariano, Pranzo, & Hansen, 2011; D′Ariano, 2009) in the case of a conflict occurring.

Objectives and Contributions

The research in this area has led to the development of software tools that implement rail simulators with particular reference to long- and middle-distance railways, such as RailSys (Bendfeldt, Mohr, & Muöller, 2000), OpenTrack (Nash & Huerlimann, 2004), FRISO (Middelkoop & Loeve, 2006), and TOPSIM (Hellstroöm et al., 2003). The differences between these simulators and the approach followed in this article are that the proposed model is focused on urban railways, offers users more precise modeling and visualization of infrastructure, and is intended to be a conflict detection tool to launch the *Real-Time Optimization module*.

The differences between the model described in this article and the solutions proposed by Dorfman and Medanic (2004) and Li et al. (2008) are that they use a single track between stations. This article, like Toörnquist and Persson (2007), uses an approach aimed at solving the problem with several tracks on the same line, which gives a better representation of urban railways such as regional or underground trains (Flamini & Pacciarelli, 2008). It also includes a signaling system able to anticipate potential conflicts that may occur within this system, representing the real traffic lights used in railway infrastructures.

It is also possible to classify the model presented in this article as a synchronous and microscopic simulation model, because it is characterized by a detailed description of the track infrastructure, with the train paths and train disposition, accurately modeling the traffic flow in the railway network by using detailed information on track configuration, signaling system, timetable, rolling stock characteristics, and automated dispatching rules. This type of model can handle very large networks but requires extensive work to model infrastructure topology, signaling, and timetables.

Thus, the main contributions of this article are threefold:

- A clear description of the traffic control problem is given and a discrete event simulation model is proposed from the very beginning based on this description. The simulation model is focused on urban rail systems such as regional or underground railways, but can also be used in interurban networks.
- A computational experiment has been carried out on the real local Renfe network in Madrid, which proves the suitability for on-line use and thus for incorporation into the *Railway Model module* within an ITS.
- Because of the low computational cost the simulation model described can be integrated as a part of an ITS as a schedule analyzer or for studying the feasibility of the control strategies, such as ordering of trains, alternative routes, or proper speed recommendations (Cordeau et al., 1998). This use has been illustrated in the article with a case study of new alternative operation plans.

FORMULATION OF THE RAILWAY MODEL

This article formulates a simulation model that represents the *Railway Model* module by an ITS and can predict the future state of the system for a real situation and a given schedule. In order to formalize the railway model, some definitions

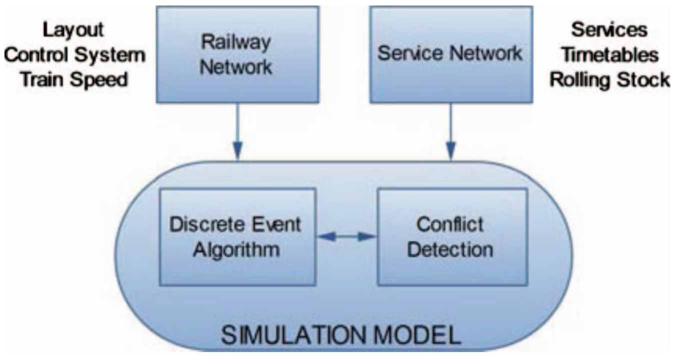

Figure 2 Railway model (color figure available online).

taken from D′Ariano, Pacciarelli, and Pranzo (2007) are will be used.

A railway system consists of a *physical network* that supports a *network of services*. This *railway network* comprises *track segments* and *signals*, with the type changing from one country to another. The signaling system to be understood in this article is the NS54 system, the standard used in many countries, including Spain. These signals are distributed along the length of the lines, intersections, and stations, and define the *block sections*, which are track segments between two block signals.

The passing of a train through a particular block section is called an *operation,* and the time required to carry it out is the *running time*. This time may experience a delay, depending on the entry signals for the train to the next track section. When a train leaves a track section a blocking period occurs until the entry of a new train is allowed; this is called the *setup time*. When two or more trains require the same block section at the same time a *conflict* occurs.

The model has been divided into three main parts (Figure 2):

1. Railway network model. The railway network is composed of track segments and signals which conform the block sections.
2. Service network model. This model represents the programmed services for a physical network.
3. Simulation algorithm. A simulation algorithm represents the working of the entire system. That is, it represents the performing and sequencing of operations. In this way, the instants in which operations will occur can be calculated and thus they can be compared with the original master schedule to identify any disruptions or delays.

Railway Network

Physical Layout

To represent the rail network in this model the typical make-up of urban railway networks, such as those of local or underground trains, has been borne in mind, in which each of the tracks to be used is divided into sections of different lengths called block sections. In each of these segments, for safety reasons, there must not be more than one train at a time. To avoid having more than one train enter these block sections a variety of blocking mechanisms may be used, of which the most common is the use of signals at the point where one segment ends and the next begins; this is the means used in this article.

Within the block sections themselves three main types can be distinguished: normal block sections, each with its individual restrictions, station block sections, where trains stop so that passengers may get on and off, and intersection block sections that represents the intersection zones. A further feature of these block sections is that they have a maximum speed V^i at which they may be crossed, which will depend on the physiognomy of the infrastructures or the state they find themselves in at a specific moment.

In summary, the main characteristics of the block sections will be the type (normal, station, or intersection), the length of it, maximum speed, and control traffic light state.

Thus, the physical network will be represented by a graph $G = (V,A)$ where the V are the block sections, and the links will represent whether there is a connection between them. Furthermore, in the nodes there will be three types, V_N, which are the normal block sections, V_E, which are the station block sections, and V_I, which are intersections, and so:

$$V = V_N \cup V_E \cup V_I \quad V_i \cap V_j = \emptyset \quad i \neq j \quad i, j \in \{N, E, I\} \tag{1}$$

For example, one part of the physical network of Renfe Cercanías Madrid (the local network) leads to a graph like that in Figure 3. The normal block sections can be seen in plain color, and the station block sections with hatching. This example does not include intersection block sections, which are fictitious block sections representing the crossover points that exist in the network in more complex zones (see Figure 12, later in the article).

Of the forms that exist to present a graph, the format used in the L2QUE algorithm has been chosen (Gallo & Pallotino, 1988), which is used for directed graphs, since this way of storing data only uses two vectors of non-null elements and can quickly recover any type of information contained therein.

The Signaling System in the Physical Network

In our model the signaling system works by means of fixed-block signaling. Each block section $j \in V$ will have a variable of state S^j that expresses the current state of the signal, showing whether or not it may be entered.

The rules of signal control depend on the direction of travel of the trains with respect to the orientation of the tracks. In the normal orientation if a train is on a block section the entry signal to the track is on RED (R), preventing any other train from

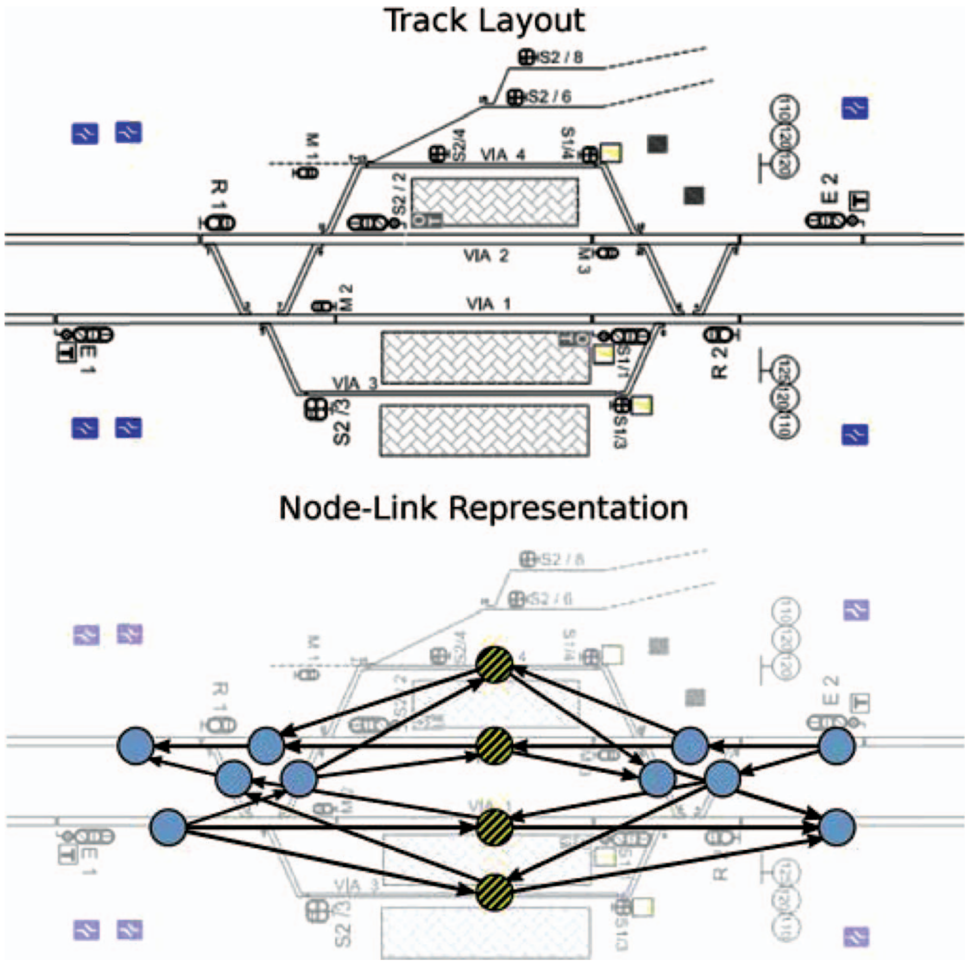

Figure 3 Example of node-link model of a railway track (color figure available online).

entering this block section; the entry signal to the adjacent track will be on YELLOW (Y), allowing entry to the block section but with certain precautions as to speed; and the signal to other tracks is on GREEN (G), allowing entry without any restriction. An example of a possible situation of the system may be seen in Figure 4. This represents a change of block section for the train that was in the block section V_i to a new block section V_j and the changes in the signals. The lowercase letters i, j, and k represent the identifier of the block section or the signal associated with this block section.

Therefore, when a train changes from a block section V_i to another block section V_j it will be necessary to update the signal S^j to R and, if it is occupied by another train, the signal S^i to Y and the signal S^K at the previous block section V_K to G.

Furthermore, in intersection block sections its signal remains at R until the train leaves the next block section it reaches.

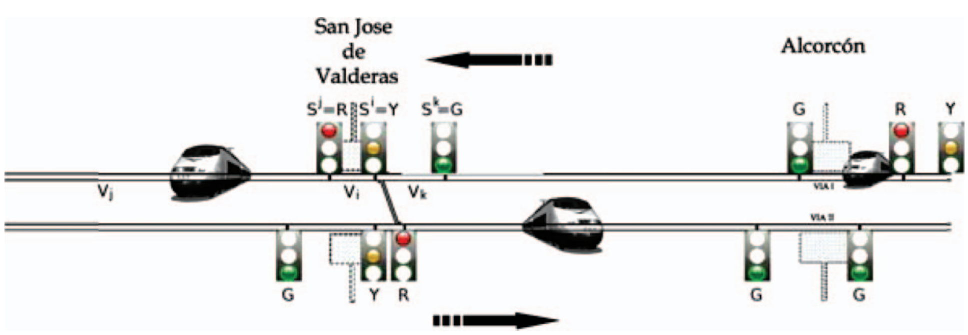

Figure 4 Example of the state of the signals at a particular moment on the C5 Line of Renfe Cercanías Madrid in the stations of *San José de Valderas* and *Alcorcón* (color figure available online).

Service Network

Services and Timetable

In order to model the trains that operate on the available infrastructure and their associated characteristics, the concept of services is introduced; services are individual options intended to satisfy the transport needs of the users of the rail system.

Each service comprises:

– A train with an associated maximum speed.
– A schedule to be kept to, defined by the time of departure and of passage through the stations.
– A route on the graph of block sections.

Let us suppose there are n services programmed on the rail network, collectively described as the set $\mathcal{S}$ and a set of m trains called $\mathcal{T}$ that will perform these services. Let t be an element of $\mathcal{T}$ and s an element of $\mathcal{S}$.

The timetable of each of the services ($\mathcal{H}_s$) will be given by a vector whose coordinates are associated to the moments of departure, when the train leaves block section $i \in V_E$ corresponding to a given station.

$$\mathcal{H}_s = (..., h_i, ...), i \in V_E \quad (2)$$

Each service s has an associated route r_s in the block section graph G. This sequence of block sections defines at each instant the next block section to advance to as a function of the current state of the train, and so defines:

$$\mathcal{R} = \{r_S | s \in \mathcal{S}\} \quad (3)$$

Thus, a service s is determined by the triplet:

$$s = \{t_s, \mathcal{H}_s, r_s\} \quad (4)$$

where t_s is the train that performs the service s.

Rolling Stock Restrictions

Different services share the same rolling stock, which means that a delay in certain services can affect the departures of subsequent services. These delays are managed by a variety of dispatching policies. This model considers that a service may not start until the train t that is due to perform the service is in the station, plus a turning buffer time between train routes (D′Ariano & Pranzo, 2009), which is a time margin between the end of a train route and the start of a new service using the same rolling stock.

Let us suppose that a train t performs two consecutive services s_1 and s_2 and that the planned timetables are:

$$H_{s_1} = \left(h_0^1, h_1^1, \ldots, h_n^1\right) \quad h_0^1 < \ldots < h_n^1 \quad (5)$$

$$H_{s_2} = \left(h_0^2, h_1^2, \ldots, h_m^2\right) \quad h_0^2 < \ldots < h_m^2 \quad (6)$$

Now:

$$\hat{h}_0^2 = max\{\hat{h}_n^1 + TBT_{s_1,s_2}, h_0^2\} \quad (7)$$

for $\hat{h}_n^1 \equiv$ predicted finishing time of service s_1 by train t
$\hat{h}_0^2 \equiv$ predicted starting time of service s_2 by train t
$TBT_{s_1,s_2} \equiv$ turning buffer time between services s_1 and s_2

Simulation Algorithm

During train operation two situations are distinguished. The first is movement within a block section (operation), and the second is the transition from one block section to another. In the first type of movement the train does not interact with any other train, while in the second a conflict could be produced that would require a sequencing of operations involving several trains. In the following subsections each of these movements is described.

Intrablock Section Movements (Operations)

Suppose that an operation is performed in which a train t moves within a block section $i \in V$ in the direction of block section $j \in V$ as shown in Figure 5. The model assumes that the intrablock section movements of the trains are determined by the speeds and the time instant at which the train t occupied the start and finish of block section i. Define the following variables:

– I_t^i: The *initial speed* at entry for train t in block section i.
– E_t^i: The *exit speed* of train t on leaving block section i due to the restrictions imposed during the passage and the state of the controls.
– γ_t^i: The instant at which train t enters block section i.
– τ_t^i: The instant at which train t reaches the end of block section i.

In this section are described the mechanisms for calculating the speed and the moment at which the train reaches the end of block section i as a function of the initial speed I_t^i and the instant γ_t^i at which it was at the start of the block section, the state of the system, and the characteristics of the block section.

From the previous characteristics three restrictions related to the propagation of speeds in the network as a function of the state of the system at a given moment are derived:

- The first of these is the conservation of speed when the train t passes from block section i to block section j (Figure 5):

$$E_t^i = I_t^j \quad (8)$$

- The second restriction is the speed as a function of signal regulation S^j of block section j which it is about to enter:

$$E_t^i := \begin{cases} V_t^{max} & if\ S^j = G \\ V^Y & if\ S^j = Y \\ 0 & if\ S^j = R \end{cases} \quad (9)$$

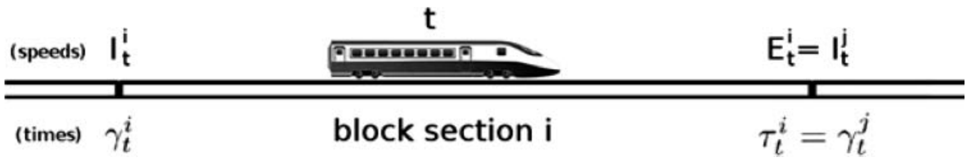

Figure 5 Conservation of speed and instant when there is a change of block section.

where S^j is the signal of block section j, $S^j \in \{G, Y, R\}$ are the different states, V_t^{max} is the maximum speed associated with the train t, and V^Y is a safe speed predetermined by the system managers. It is assumed that train t travels in block section i at a mean speed M_t^i:

$$M_t^i = \frac{I_t^i + E_t^i}{2} \tag{10}$$

- A special type of block section is considered that can limit the speed to a value V^i that cannot be surpassed at any time for safety reasons, and so if $max\{I_t^i, E_t^i\} \geq V^i$ then $E_t^i = min\left\{M_t^i, V^i\right\}$ and $M_t^i = V^i$. The running time, that is, the time taken to traverse a complete block section C_t^i, is

$$C_t^i = \frac{L_i}{M_t^i} \tag{11}$$

where L_i is the length of the block section. Note that this section is concerned here with urban networks, basically used for transportation between the different zones of a city and its outskirts, where the length of block sections is small, so formulas (10) and (11) provide a good approximation.

Finally, two characteristics related to the stops made in station block sections with a platform where passengers may board and alight have also been considered:

- To the running time C_t^i must be added a time T_p, defined as the time necessary for passengers to get on and off the train, in accordance with the safety regulations, plus the time required to carry out braking and accelerating maneuvers, which refers to the reduction of speed when entering a station until the train stops and the acceleration after the train has picked up all the passengers from the platform. $T_p = 0$ if the block section is not of the station type.
- Bearing in mind that there is a schedule H_s it is assumed that the trains can at no point be ahead of their scheduled time; that is, they cannot be in advance of their designated time for block section i given by h_i. This case could be obviated by following a different approach, because letting some trains arrive earlier at some points, if carefully managed, can be used to optimize railway traffic (D′Ariano, Pacciarelli, & Pranzo, 2008).

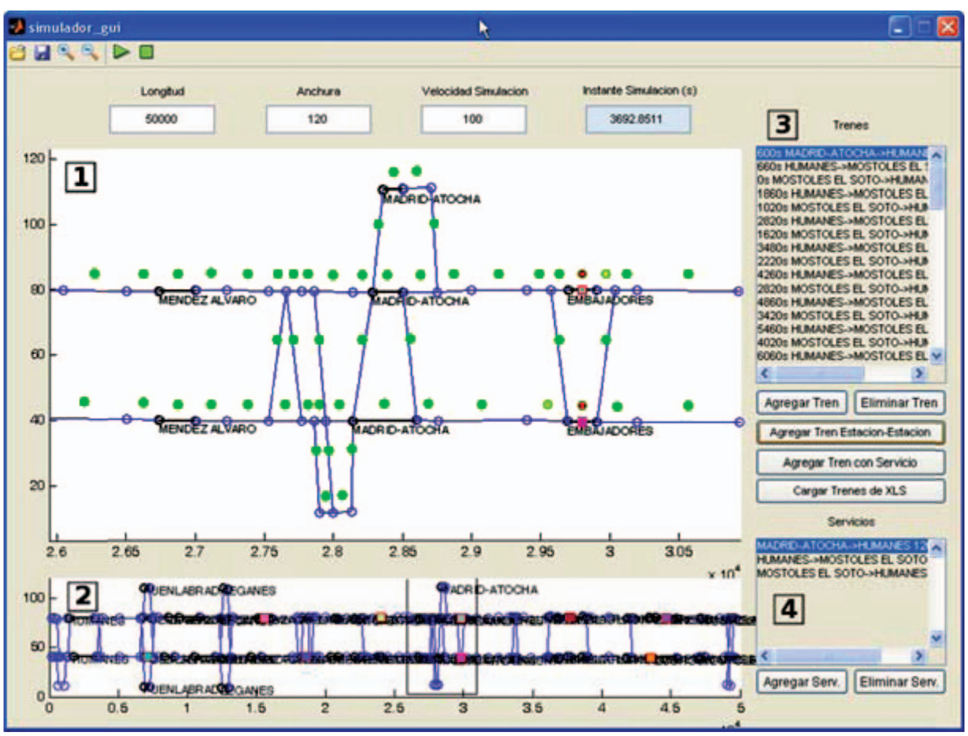

Figure 6 The SIMEIFER tool (color figure available online).

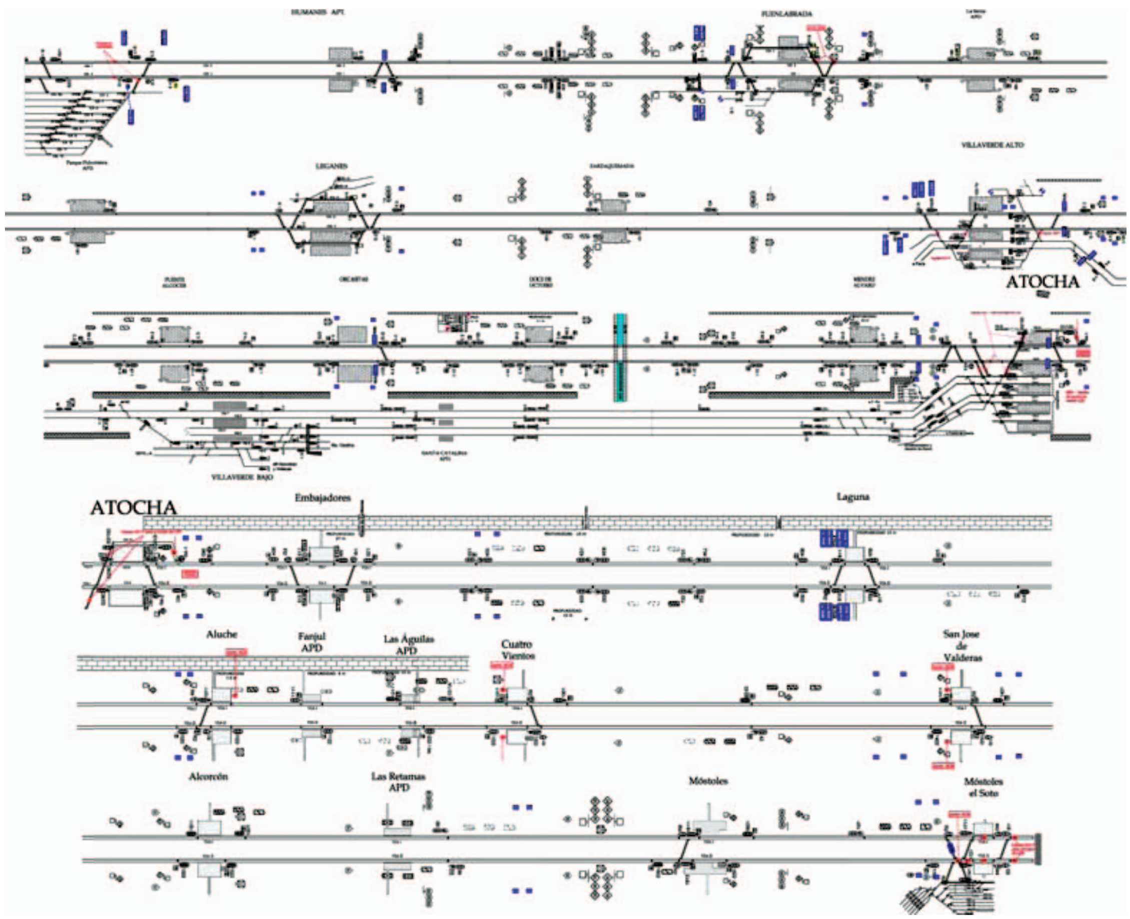

Figure 7 Line C5 of Renfe Cercanías Madrid (color figure available online).

For these reasons the time of arrival τ_t^i at the end of block section i is given by

$$\tau_t^i = max\left\{\gamma_t^i + C_t^i + T_p, h_i\right\} \tag{12}$$

where γ_t^i is the instant of entry to block section i; $\gamma_t^i = \tau_t^k$ where k is the block section previous to i.

It is necessary to bear in mind that speed is conserved not only at the change of block section, but also the instant, that is $\tau_t^i = \gamma_t^j$.

The mechanisms just described correspond to the propagation of speeds and instants in the unions of block sections. In order for the problem to be completely determined, only the initial conditions are missing. Assume that $I_t^i = 0$ for the initial stations i of a service and $\tau_t^i = \hat{h}_i$, that is, the real end time of the service performed by train t.

Interblock Section Movements (Events)

An event in the system is defined as a transition in the state of the system in which a service s leaves a block section $i \in V$ in which it was traveling and enters another block section $j \in V$. These events coincide with the different operations carried out in the system.

The key to performing the simulation is to analyze the events in the appropriate order, since they are interrelated and a movement of one train can affect the schedule of another. In other words, the space (block sections) and time in which the events occur must be appropriately analyzed. The dispatching rule used is first come, first served (FCFS), which is commonly adopted in railway management. This consists of giving precedence to the train arriving first at a block section, and sequencing the trains in order of arrival.

Other possibilities are to use the first leave, first served (FLFS) rule or the AMCC algorithm described in D′Ariano, Pacciarelli, and Pranzo (2007). The FLFS dispatching rule consists in giving precedence to the train that is able to leave the block section first, computing this time before entering it. The AMCC algorithm is a greedy heuristic algorithm based on local information and uses the concept of alternative graphs. Its objective is to compute near-optimal train schedules of practical size within a short computation time, selecting in each iteration one of two alternative arcs, which then fixes a precedence constraint between two trains at a potential conflict point.

Suppose that it is wished to simulate the system in the time interval $[T_I, T_E]$. Let E be the set of events that occur in the system in the interval $[T_I, T_E]$. For each event $e \in E$, the associated block section $i \in V$, the time τ_t^i at which event e

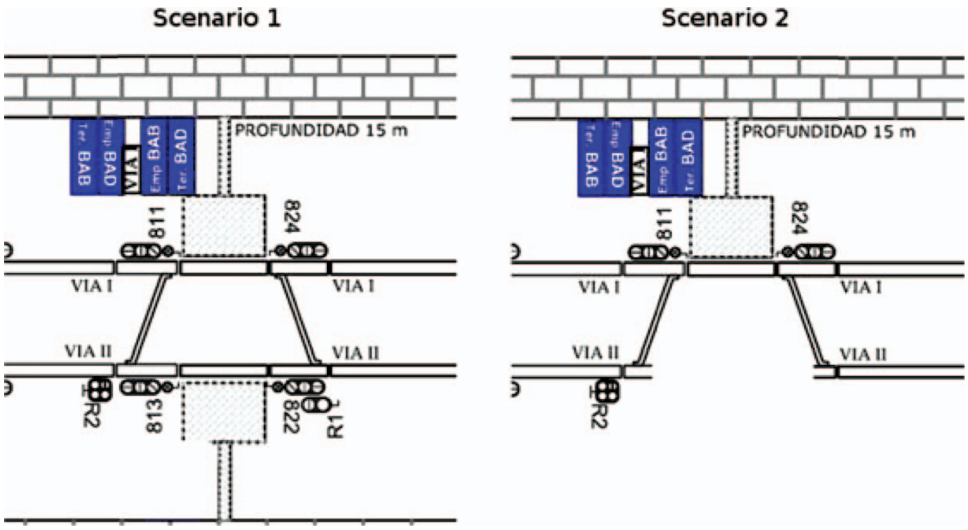

Figure 8 Modification to Laguna station (color figure available online).

should occur, and the service s_e associated with event e must be stored.

It may be seen that each event is associated with a train $t \in T$ and therefore at each moment the system state is completely determined by the distribution in block sections of this set of trains. This will allow the list of events E to be processed in an orderly manner in accordance with the FCFS dispatching rule. Suppose that at the current time T, the train t is in block section $i \in V$. Then set $Q_t = \tau_t^i$.

The set of instants at which the events that determine the state of the system occur is, at time T:

$$Q = \{Q_t | Q_t > T, t \in \mathcal{T}\} \tag{13}$$

The next instant at which the system may change state will be

$$Q_{t'} = min_{t \in T}\{Q_t\} \tag{14}$$

Note that the state of the system is unaltered in $[T, Q_{t'}]$, so the system clock (the time within the simulation) can be reset by $T = Q_{t'}$. Then, once the train t', its block section, and its service have been identified, the train is, if possible, moved to the next block section by the transition proper to it according to the intrablock section mechanisms previously described, and the new $Q_{t'}$ is calculated.

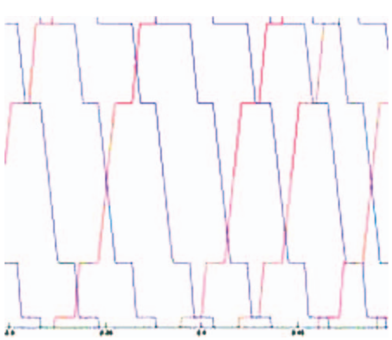

Figure 9 Time–station diagram for first scenario (color figure available online).

A conflict may occur, and therefore the block section j that the train t' seeks to enter is already occupied by another train t'' that has not kept to the schedule due to some type of unforeseen circumstance, so the former must wait until the block section is free. The minimum instant at which train t' may leave the block section is $Q_{\{t'\}} = Q_{\{t''\}} + \varepsilon$ where $\varepsilon > 0$ is the minimum established safety time known as the *setup time*.

The algorithm used for working with the list of events E should update the current state and obtain the new situations that will occur according to all the considerations in the model. This algorithm is described next.

Require: Graph of block sections $G = (V,A)$, set of programmed services $\mathcal{S}$, timetable sheets $\mathcal{H}_s$, $s \in \mathcal{S}$, list of trains $\mathcal{T}$, and end time of simulation T_E

(Initialization) Make list of current events E from the set of initial departures of T. Let T_I be the initial time of the simulation. Make $T = T_I$

while $T \leq T_E$

(Choose train associated with event to be handled) $t' =$ arg minimize$_{t \varepsilon T}$ $\{Q_t\}$

$T = Q_{t'}$

(Handle event) From t' calculate the next block section $j \in V$ for entry of service s from its route r_s.

if $\nexists\, j$ (it has reached the final station of the service)

Eliminate event of E.

Generate the next service s and event for train t' if it exists taking into account restrictions on rolling stock.

else

Calculate time of the next operation of train t' $Q_{t'}$ according to the criteria described for intra-block section movements.

if $S^j = R$

$Q_{t'} = max(Q_{t'}, Q_{t''}) + \varepsilon$ where $Q_{t''}$ is the instant of departure of the train in block section j at that moment and ε is the setup time.

end if

(Update signals) Update affected signals and new block section j that train t' is now in.

end if

end while

(End) View results of simulation.

COMPUTATIONAL EXPERIMENTS

This section works through a computational experiment with the simulation model described in a real case.

Two numerical tests have been designed. In the first the aim is to test whether the CPU time is lowered enough for the model to be applicable on-line. In the second test an off-line use is presented in which it is assessed, that is, integrating into an ITS for study a new use or operational plan, considering the number of conflicts that occur.

To carry out the computational experiments the SIMEIFER tool was developed following the United Software Development Process to implement the model mentioned in the previous section. The tool has been codified using MATLAB. This tool provides an easy to use visual editor of railway infrastructure, and allows input and editing of the trains and services that will participate in the simulation, or they may be acquired from databases in which the planning schedule is stored, and it can show and store the results of any simulation that has been performed. Some videos showing the working of this tool may be seen at http://www.inf-cr.uclm.es/www/Ricardo.Garcia/RICARDO/videos/SIMEIFER

Figure 6 shows a capture of the simulation tool. Area 1 represents the part the user is focusing on the railway system at the moment from area 2, which shows the whole infrastructure. Areas 3 and 4 are the parts of the system that handle the information about trains and services, respectively, including options to add or load trains or services to the system.

The computer used to perform the computational experiments has the following characteristics: Windows Vista 64 bits, with processor Intel QuadCore 2.83 GHz, and RAM 4 GiB.

Case Study of Line C5

To check the applicability of the simulation model, two computational experiments have been performed on Line C5 of Renfe Cercanías Madrid (Spain), which may be seen in Figure 7. This line is formed by a set of 256 block sections on which there are 23 stations over 45 km of track, carrying more than 400,000 passengers every day. It also has sensors that can show the complete state of the system at any time. This example uses information provided by Renfe Cercanías, which gives the schedules of the 327 trains that operate daily on the line.

The first scenario to which the model is applied corresponds to normal running of the system with initial timetables predefined, without including any disruption or disturbance.

The second scenario deals with one of the problems that is most frequently encountered in railway systems, the closing of a set of block sections because of some kind of emergency or line work. An example of this occurred on June 26, 2009, on Line C5 in Madrid, when a crash between two trains cut the links between *Madrid-Atocha* and *Móstoles el Soto*.

Using SIMEIFER it is possible to redirect the traffic automatically, removing the obstructed block sections from the

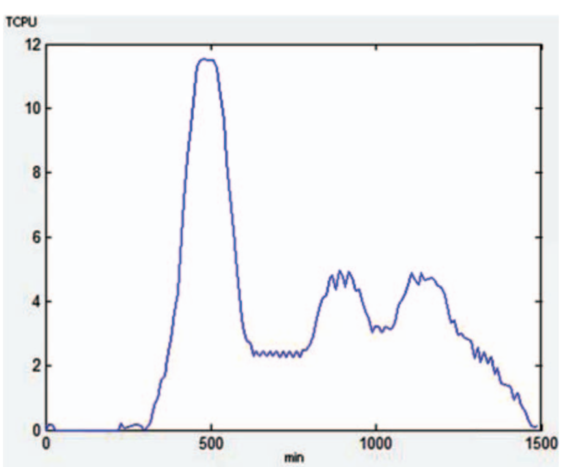

Figure 10 Graph of the running of the base example (color figure available online).

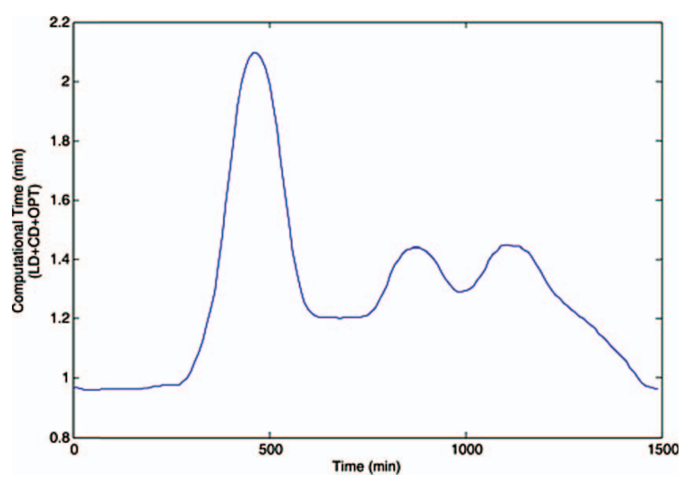

Figure 11 Computational cost of ITS with time horizon 1 hour every 10 minutes (color figure available online).

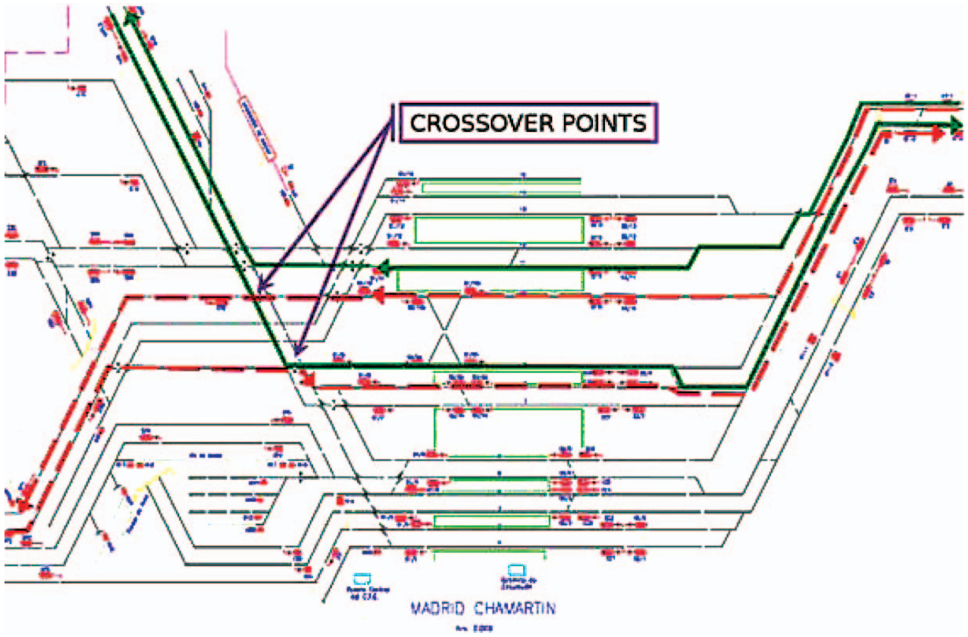

Figure 12 Example of crossover zones (color figure available online).

infrastructure, and advice could be obtained as to whether it is feasible to redirect the traffic through the available lines without causing further accidents or bringing the system to a halt.

In order to carry out this simulation, part of *Laguna* station on Line C5 has been removed, as seen in Figure 8.

In this case study the following results have been obtained:

- Simulated time (ST). Shows the total simulated time, that is, the time taken by the system to complete the given schedule.
- Simulation time (sT). Shows the total CPU time taken to perform the complete simulation of the given schedule.
- Waiting time (WT). Sum of all waiting times caused by trains which have met with a yellow or red signal.
- Delay time (DT). Sum of all the delays experienced by the trains in leaving a station later than the time established in the initial schedule.

The results for both scenarios are presented in Table 1. Figure 9 shows part of the time-station diagram for the first scenario which shows a double-track line since trains can meet and pass while running on open tracks. The factors that influence the computational costs of train dispatching can be seen in (D′Ariano & Pranzo, 2009).

It can be seen that for the first scenario the waiting times and system delays are small enough to be applied in real time. There is an average waiting time of 5.97 seconds for each train of the 327 planned, and an average delay time of 6.67 seconds per train, which, considering all the stations on the routes, is insignificant, especially if it is also taken into account that it must be divided by the number of block sections and stations that form part of the route.

The numerical experiments carried out with the network parameterization provided by Renfe Cercanías, taking into account average journey speeds, maximum speeds, length of block sections, time stopped at stations, etc., show clearly how the model described can reliably simulate a real railway system, getting the trains to pass through the stations at the scheduled moment, obtaining a waiting time or delay time similar to that predicted by Renfe Cercanías Madrid with an average delay of 6.67 seconds for an average journey of 55 minutes, reflecting the planned schedules very precisely.

In the results for the second scenario a general increase in the waiting times and delay times can be seen with respect to the first, but even so it is clear, once the simulation has finished correctly, that it would be possible to use the sole remaining part of the station to perform transfers in both directions. It can also be observed that the simulation time is small, and in less than eight minutes results could be obtained that would help the dispatchers to take the most appropriate decisions in this situation.

Finally, Figure 10 shows the evolution of the computational cost with respect to the instant simulated for the first scenario; the cost is highest at the 500th minute, which roughly coincides with the morning rush hour, about 8:30 a.m. This graph shows on the OY axis the number of seconds required to simulate 10 minutes (600 seconds) of the system at each instant T of the

Table 1 Results of the simulation for the standard timetable on Line C5.

Scenario	ST	sT	WT	DT
Normal	24 hr 16 min 40 sec	7 min 43 sec	32 min 33 sec	36 min 21 sec
Emergency	24 hr 40 min 26 sec	7 min 56 sec	2 hr 43 min 46 sec	3 hr 8 min 57 sec

Table 2 Results of simulation for different plans.

Plan	ST	sT	WT	CN	CT
1	19 hr 36 min 32 sec	3 min 33 sec	2 hr 49 min 7 sec	110	11 min
2	19 hr 23 min	2 min 27 sec	1 hr 30 min 44 sec	57	21 min
3	19 hr 23 min	2 min 3 sec	56 min 56 sec	30	39 min

simulation. Note that even in the rush hour this figure is less than 12 seconds, giving a ratio of $600/12 = 50$, which shows that the system can calculate 50 times more quickly than real time even in the worst situation.

This section next looks in more detail at the matter of whether the proposed simulation model can be included in an ITS to anticipate situations and as a means of predicting the future state of the system.

The on-line applicability of an ITS system like that shown in Figure 1 requires the following tasks to be performed:

- Load data (*LD*). Load system information in real-time.
- Conflict detection (*CD*). Run the *Railway Model module* to detect possible disturbances and if necessary run the *Real-Time Optimization module* and inform the human dispatcher.
- Operation optimization (*OPT*). Run the *Real-Time Optimization module* to resolve conflicts. The cost of running it will depend on the algorithm used.

The computational cost of carrying out the tasks *LD*, *CD,* and *OPT* depends on a variety of factors, of which the most important are the time horizon of the operations to be studied and the trains in service during that time window. The time horizon of practical interest to railway managers is usually less than an hour for real-time purposes.

A number of numerical trials have been performed on Line C5 with a time horizon of 1 hour, and the mean value obtained for the time necessary to load the data for the time window was $LD =$ 57.23 seconds (D′Ariano, Pacciarelli, & Pranzo, 2007) reports mean computational results obtained with the AMCC algorithm (element *Real-Time Optimization module*) for networks of size similar to that of Line C5 with a higher traffic density and with a time window of 1 hour as $OPT = 0.5$ seconds.

Figure 11 shows the estimated computational cost of running the ITS at each moment of time. If the system is refreshed every 10 minutes it can be seen that in the worst case the dispatcher would have 7 minutes and 54 seconds to take a decision and to communicate the necessary measures to resolve the disturbance.

Study of Alternative Operation Plans

Finally, another use for the model is to study possible future actions, such as the extension of the network by building new track sections, stations, or entry points, modernizing or improvement of infrastructures, or increasing services.

The choice of a plan of use from among a set of alternatives requires assessment from many perspectives (operating cost, demand, etc.) and also of the impact on current use. SIMEIFER allows this assessment.

One of the most important actions currently being undertaken by Renfe Cercanías Madrid consists of the building of new lines to Barajas airport from the station of *Chamartín* and three stations, *Manoteras*, *Valdebebas*, and *Barajas T-4*, on the route. This requires the study of alternative operation plans for the new, modified, system, seeking the lowest number of conflicts and the shortest waiting times.

Specifically, three alternative operation plans will be studied in which the intention is to put 475 trains into operation, and the aim is to find an optimal solution without greatly modifying the current plan of the system, and with a minimum number of conflicts at intersections (Figure 12).

In this case study the following results have been obtained, as well as the parameters ST, sT, and WT mentioned in the previous section:

– Conflict number (*CN*). The total number of occasions on which the trains in the system have had to wait on meeting a red signal.
– Time between conflicts (*CT*). The average time in the system between one conflict and the next. This parameter measures the complexity that the dispatchers have to deal with.

The results for the simulation of the three alternatives using our model are shown in Table 2 and Figure 13.

These results clearly show the difference that exists between the three alternatives considered for the system plan, in terms of both waiting times produced by conflicts and the number of conflicts produced, particularly between 8:00 and 9:00 a.m. These results are in agreement with the experience of the dispatchers and allow a precise assessment of the viability of the different alternatives by using the model.

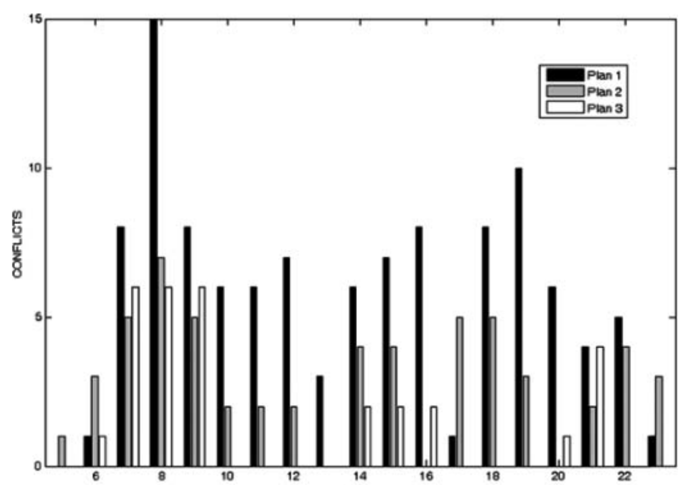

Figure 13 Conflicts per hour for alternative operation plans.

CONCLUSIONS

In this article has been described a simulation model for a railway system based on discrete events determined by the position of the trains on the different block sections that have mechanisms for signal regulation. This simulation tool can be integrated into *on-line* or *off-line* systems. In the first case it would form part of an ITS and would allow the prediction of the future state of the system for a real situation and a given schedule, and would thus anticipate possible conflicts and secondary delays. Off-line use is aimed at assessing and testing the feasibility of operation plans determined by a set of services and timetables.

The software tool SIMEIFER has been developed from this model, and it satisfies the requirements and offers the features that would be expected of a system of this nature. The tool has been validated by the performance of several experiments on the real situation of the Line C5 of Renfe Cercanías Madrid and on the development plans currently under consideration to connect the station of Chamartín with Madrid-Barajas airport. It has been shown that the computing time required to assess the evolution for the next 10 minutes takes 12 seconds in the most difficult case and an average of 5 seconds, which proves its viability for on-line use. In the numerical trials off-line use of the model for the assessment expansion plans for the rail system has also been demonstrated.

One of the essential characteristics of the ITSs developed for the control and management of rail traffic is their modular architecture, such that they allow the incorporation and combination of different modules capable of solving any of the difficulties involved in railway systems. For this reason, one of the future lines of work that makes use of this is the implementation of the *Railway Module* within a real system, with the aim of detecting disruptions that might occur during the running of the initial plan, giving a useful system for providing information to the dispatchers. Work is also being done on broadening the features of the model to simulate railway systems with moving-block signaling or with other strategies for conflict resolution, such as FLFS or the AMCC algorithm (D′Ariano, Pacciarelli, & Pranzo, 2007), which allows the problem of real-time conflict resolution to be addressed.

REFERENCES

Baohua, M., Wenzheng, J., Shaokuan, C., & Jianfeng, L. (2007). A computer-aided multi-train simulator for rail traffic. *International Conference on Vehicular Electronics and Safety,* December, **2007**, 1–5.

Bendfeldt, J., Mohr, U., & Muöller, L. (2000). RailSys, a system to plan future railway needs. *Advances in Transport*, **7**, 249–255.

Burdett, R. L., & Kozan, E. (2009). Scheduling trains on parallel lines with crossover points. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, **13**(4), 171–178.

Cordeau, J.-F., Toth, P., & Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation Science*, **32**, 380–404.

Corman, F., D′Ariano, A., Pacciarelli, D., & Pranzo, M. (2009). Evaluation of green wave policy in real-time railway traffic management. *Transportation Research Part C: Emerging Technologies*, **17**(6), 607–616.

Corman, F., D′Ariano, A., Pacciarelli, D., & Pranzo, M. (2010). A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological*, **44**(1), 175–192.

Corman, F., D′Ariano, A., Pranzo, M., & Hansen, I. A. (2011). Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. *Transportation Planning and Technology*, **34**(4), 341–362.

D′Ariano, A. (2008). *Improving real-time train dispatching: Models, algorithms and applications* (Doctoral thesis). The Netherlands TRAIL Research School, Delft, Holland. TRAIL Thesis Series no. T2008/6.

D′Ariano, A. (2009). Innovative decision support system for railway traffic control. *IEEE Intelligent Transportation Systems Magazine*, **1**(4), 8–16.

D′Ariano, A., Corman, F., Pacciarelli, D., & Pranzo, M. (2008). Reordering and local rerouting strategies to manage train traffic in real time. *Transportation Science*, **42**(4), 405–419.

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

D′Ariano, A., Pacciarelli, D., & Pranzo, M. (2008). Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research Part C: Emerging Technologies*, **16**(2), 232–245.

D′Ariano, A., & Pranzo, M. (2009). An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Network and Spatial Economics*, **9**(1), 63–84.

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

Dorfman, M., & Medanic, J. (2004). Scheduling trains on a railway network using a discrete event model of railway traffic. *Transportation Research Part B*, **38**, 81–98.

Flamini, M., & Pacciarelli, D. (2008). Real time management of a metro rail terminus. *European Journal of Operational Research*, **189**, 746–761.

Gallo, G., & Pallotino, S. (1988). Shortest paths algorithms. *Annals of Operations Research*, **13**, 3–79.

Goverde, R. (2005). *Punctuality of railway operations and timetable stability analysis* (Doctoral thesis). The Netherlands TRAIL Research School, Delft, Holland. TRAIL Thesis Series no. T2005/10.

Goverde, R., & Hansen, I. A. (2000). Tnv-prepare: Analysis of dutch railway operations based on train detection data. *Computers in Railways VII*, 779–788.

Hellstroöm, P., Kauppi, A., Wikstroöm, J., Andersson, A.W., S. B., Kvist, T., & Gideon, A. (2003). Experimental evaluation of decision support tools for train traffic control. *World Congress on Railway Research 2003*, September–October 2003, 670–677.

Ho, T., Mao, B., Yuan, Z., Liu, H., & Fung, Y. (2002). Computer simulation and modelling in railway applications. *Computer Physics Communications*, **143**, 1–10.

Landex, A., Kaas, A., & Hansen, S. (2006). *Railway operation* (Tech. rep. no. 4). Technical report, Technical University of Denmark, Centre for Traffic and Transport, Kongens Lyngby, Denmark.

Li, F., Gao, Z., Li, K., & Yang, L. (2008). Efficient scheduling of railway traffic based on global information of train. *Transportation Research Part B*, **42**, 1008–1030.

Martin, P. (1999). Train performance and simulation. *Winter Simulation Conference 1999,* December 1999, 1287–1294.

Middelkoop, A. D., & Loeve, L. (2006). Simulation of traffic management with FRISO. *Computer in Railways, X*, 501–509.

Nash, A., & Huerlimann, D. (2004). Railroad simulation using opentrack. *Computers in Railways, IX*, 45–54.

Nie, L., & Hansen, I. (2005). System analysis of train operations and track occupancy at railway stations. *European Journal of Transport and Infrastructure Research*, **5**(1), 31–54.

Szpigel, B. (1973). Optimal train scheduling on a single track railway. *Operations Research*, **72**, 343–352.

Toörnquist, J., & Persson, J. A. (2007). N-tracked railway traffic rescheduling during disturbances. *Transportation Research Part B*, **41**, 342–362.

Vromans, M., Dekker, R., & Kroon, L. (2006). Reliability and heterogeneity of railway services. *European Journal of Operational Research*, **172**(2), 647–665.

Wendler, E. (2007). The scheduled waiting time on railway lines. *Transportation Research Part B*, **41**, 148–158.

Zandi, F., & Tavana, M. (2011). An optimal investment scheduling framework for intelligent transportation systems architecture. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, **15**(3), 115–132.