

# A Topological Model Based on Railway Capacity to Manage Periodic Train Scheduling\*

M.A. Salido<sup>1</sup>, F. Barber<sup>2</sup>, M. Abril<sup>2</sup>, P. Tormos<sup>3</sup>, A. Lova<sup>3</sup>, L. Ingolotti<sup>2</sup>

DCCIA<sup>1</sup>, Universidad de Alicante, Spain

DSIC<sup>2</sup>, DEIOAC<sup>3</sup>, Universidad Politécnica de Valencia, Spain

{msalido, fbarber, mabril, lingolotti}@dsic.upv.es

{ptormos, allova}@eio.upv.es

## Abstract

Railway capacity has been a significant issue in the railway industry. Over the last few years, numerous approaches and tools have been developed to compute railway capacity. In this paper, we compute railway capacity to design a topological model for solving periodic train scheduling, developed in collaboration with the National Network of Spanish Railways (RENFE). This topological technique transforms the railway optimization problem in subproblems such that a traffic pattern is generated for each subproblem. These traffic patterns will be periodically repeated to compose the entire running map. The results show that this technique improve the results obtained by well known tools as LINGO and ILOG Concert Technology (CPLEX).

## 1 Introduction

Rail transport has played a major role in the economic development of the last two centuries. It represented a major improvement in land transport technology and has obviously introduced important changes in the movement of freight and passengers. Over the last few years, railway traffic has increased considerably, which has created the need to optimize the use of railway infrastructures. This is, however, a very difficult task. Thanks to developments in computer science and advances in the fields of optimization and intelligent resource management, railway managers can optimize the use of available infrastructures and obtain useful conclusions about their topology.

The overall goal of a long-term collaboration between our group at the Polytechnic University of Valencia (UPV) and the National Network of Spanish Railways (RENFE) is to offer assistance to help in the planning of train scheduling, to obtain conclusions about the maximum capacity of the network, to identify bottlenecks, to determine the consequences of changes, to provide support in the resolution of incidents, to provide alternative planning and real

---

\*This work has been partially supported by the project DPI2001-2094-C03-03 from the Spanish Government and the project GV04B/516 from the Generalidad Valenciana.

traffic control, etc. Besides of mathematical processes, a high level of interaction with railway experts is required to be able to take advantage of their experience.

Railway capacity has been a significant issue in the railway industry. Many approaches and tools have been developed to compute railway capacity. However, capacity is a complex and loosely defined term that has numerous meanings. In general, within a rail concept, capacity can be described as "a measure of the ability to move a specific amount of traffic over a defined rail line with a given set of resources under a specific service plan" [7]. This could mean anything from the number of tons moved, speed of trains, on-time-performance, available track maintenance time, service reliability, or maximum number of trains per day that the subdivision can handle. Kreuger classified the capacity is different kinds: theoretical capacity, practical capacity, used capacity and available capacity.

Different models and mathematical formulations for train scheduling have been created by researchers [9], [3], [8], [5], [4], [2], etc. Several European companies are also working on similar systems. These systems include complex stations, rescheduling due to incidents, rail network capacities, etc. These are complex problems for which work in network topology and heuristic-dependent models can offer adequate solutions.

In this paper, we formalize railway capacity (theoretical capacity) to design a topological model for solving periodic train scheduling. This technique has been inserted in our system [1] and it is committed to solve this problem in order to obtain as good and feasible running map as possible. The system is able to plot the obtained running map. A running map contains information regarding railway topology (stations, tracks, distances between stations, traffic control features, etc.) and the schedules of the trains that use this topology (arrival and departure times of trains at each station, frequency, stops, junctions, crossing, etc.) (section 2). In our system, the railway running map problem is formulated as a Constraint Optimization Problem (COP). Variables are frequencies, arrival and departure times of trains at stations. Constraints are composed by user requirements and the intrinsic constraints (railway infrastructures, rules for traffic coordination, etc.). These constraints are composed by the parameters defined using user interfaces and database accesses. The objective function is to minimize the journey time of all trains. The problem formulation is (traditionally) translated into a formal mathematical model to be solved for optimality by means of mixed integer programming techniques. In our framework, the formal mathematical model is partitioned in two different subproblems: integer programming problem composed by the constraints with integer variables and linearized problem in which integer variables have been instantiated. Therefore, the problem constraints are classified such that most restricted constraints are studied first [10]. The most restricted constraints are considered to be composed of integer variables. In this way, our system studies first the integer programming problem and then it solves the linearized problem. The integer programming problem will be partitioned in a set of subproblems such that the solution of each subproblem will generate a traffic

pattern. The partition is carried out through the stations that take part in the running map. Each block of the partition is composed by contiguous stations, so that each traffic pattern represents the running map corresponding to each block of constraints. In Figure 1, a possible block of the partition may be composed by the first four stations: *Malaga Cent*, *Malaga Renfe*, *Los Prados* and *Aeropuerto*. Each traffic pattern will be periodically repeated to composed the entire running-map.

## 2 Problem Topology

A sample of a running map is shown in Figure 1, where several train crossings can be observed. On the left side of Figure 1, the names of the stations are presented and the vertical line represents the number of tracks between stations (one-way or two-way). The objective of the system is to obtain a correct and optimized running map taking into account: (i) the railway infrastructure topology, (ii) user requirements (parameters of trains to be scheduled), (iii) traffic rules, (iv) previously scheduled traffic on the same railway network, and (v) criteria for optimization.

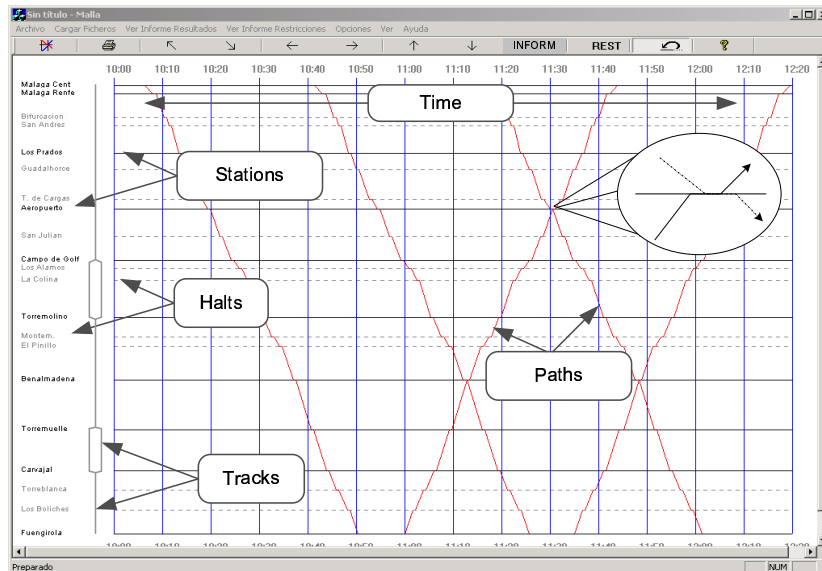


Figure 1: A sample of a running map

On a rail network, the user needs to schedule the paths of  $n$  trains going in one direction and  $m$  trains going in the opposite direction, trains of a given type and at a desired scheduling frequency.

The type of trains to be scheduled determines the time assigned for travel between two locations on the path. The path selected by the user for a train

trip determines which stations are used and the stop time required at each station for commercial purposes. In order to perform crossing in a section with a one-way track, one of the trains should wait in a station. This is called a *technical stop*. One of the trains is detoured from the main track so that the other train can cross or continue. (Figure 2).

For periodic train scheduling, we assume that all the trains in the same direction are of the same type and they stop in the same stations.

## 2.1 Railway Traffic Rules, topological and requirement constraints

A valid running map must satisfy and optimize the set of existing constraints in the periodic problem. Some of the main constraints to be considered are:

1. **Traffic rules** guarantee crossing operations. The main rules to take into account are:
  - *Crossing constraint*: Any two trains and going in opposite directions must not simultaneously use the same one-way track. The crossing of two trains can be performed only on two-way tracks and at stations, where one of the two trains has been detoured from the main track (Figure 2). Several crossings are shown in Figure 1.

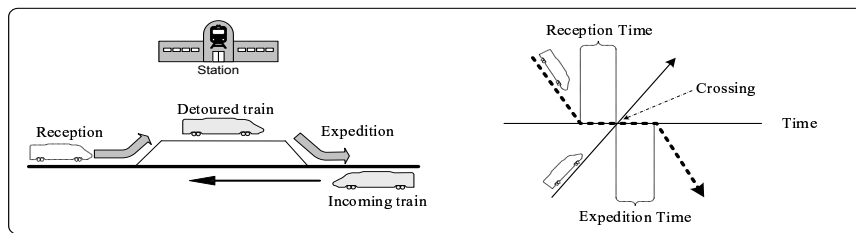


Figure 2: Constraints related to crossing in stations

- *Expedition time constraint*. There exists a given time to put a detoured train back on the main track and exit from a station.
  - *Reception time constraint*. There exists a given time to detour a train from the main track so that crossing or overtaking can be performed.
2. **User Requirements**: Constrains due to user requirements are:
    - *Type of train and Number of trains* going in each direction to be scheduled and *travel time* between locations.
    - *Path of trains*: Locations used and *stop time* for commercial purposed in each direction.

- *Scheduling frequency.* The frequency requirements of the departure of trains in both directions. This constraint is very restrictive, because, when crossing is performed, trains must wait for a certain time interval at stations. This interval must be propagated to all trains going in the same direction in order to maintain the established scheduling frequency.

In accordance with user requirements, the system should obtain the best solutions available so that all constraints are satisfied. Several criteria can exist to qualify the optimality of solutions: minimize duration and/or number of technical stops, minimize the total time of train trips (span) of the total schedule, giving priority to certain trains, etc.

## 2.2 General System Architecture

The general outline of our system is presented in Figure 3. It shows several steps, some of which require the direct interaction with the human user to insert requirement parameters, parameterize the constraint solver for optimization, or modify a given schedule. First of all, the user should require the parameters of the railway network and the train type from the central database (Figure 3).

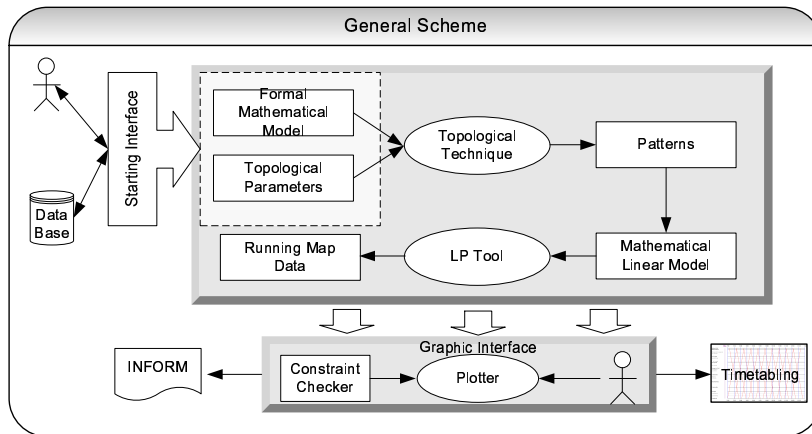


Figure 3: General scheme of our tool.

This database stores the set of locations, lines, tracks, trains, etc. Normally, this information does not change, but authorized users may desire to change this information. With the data acquired from the database, the system generates the formal mathematical model. This model is composed by a large number of mixed-integer constraints. To translate the mixed-integer problem into a linear problem, a topological technique is carried out to assign value to each integer variable. This technique carries out a partition of the stations such that each block of stations represents a subproblem and a traffic pattern (solution)

must be generated for each subproblem. This traffic pattern is generated based on the problem topology such as the number of stations, the train frequency, the type of stations, and mainly the distance among the stations. Once the traffic patterns are generated, the integer variables are instantiated and the linearized problem is straightforwardly solved returning the running map data. If the mathematical model is not feasible, the user must modify the parameters, mainly the most restrictive ones. If the running map is consistent, the graphic interface plots the scheduling. Afterwards, the user can graphically interact with the scheduling to modify the arrival or departure times. Each interaction is automatically checked by the constraint checker in order to guarantee the consistency of changes. The user can finally print out the scheduling, to obtain reports with the arrival and departure times of each train in each location, or graphically observe the complete scheduling topology.

### 3 Railway Capacity

Railway capacity is a quite old argument. How can a railway section be full, if at every cross section only once in 10 minutes or so a train passes? Most of the time there is nothing to see [11]. Certainly, there exist many definitions of railway capacity, but there does not yet exist an accepted definition of capacity on railway networks. So that, we will adopt the following definition of railway capacity.

*Definition.* Railway capacity is the maximum number of trains which can be scheduled in the railway in a fixed period of time.

Based on this definition, railway capacity will be subjected to a topological restrictions of the railway (distance between trains, number of lines, speed of trains, etc). For instance, if all sections have two-ways tracks, railway capacity will be higher than in sections with only one-way tracks. On the opposite, if there is two very distant cities joined by a one-way track, the railway capacity will be conditioned by this track, because once a train departs from one station to the other, no train can depart from this last station since the former train arrives. These tracks are bottlenecks and they condition the railway capacity. This example answers the question below (for many minutes there is no train to see).

In this section, we formalize the distance that trains must maintain to guarantee a feasible schedule. This distance will determine railway capacity. So, our aim is to geometrically study each train with tracks between stations.

We assign to each type of train and each track an angle that represents the speed of this train in this track. As we maintain in our database the distance of each track ( $d$ ) and the journey time for each train-track ( $t$ ), the angle  $\alpha$  can be straightforwardly deduced (Figure 4).

In this way, we define the angle  $\alpha$  as arc tangent of distance (in meters) divided by time (in seconds). This angle is ranged between 0 and 90 as intuitively can be seen in the running-map. Thus an angle  $\alpha \rightarrow 0$  means that the train has traveled a very short distance in a long time. Alternatively, an angle

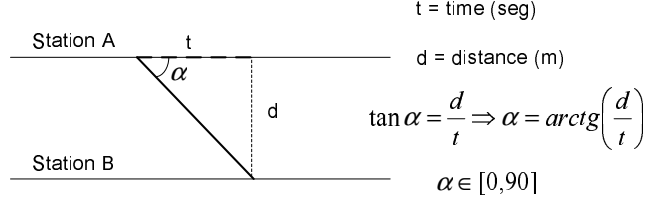
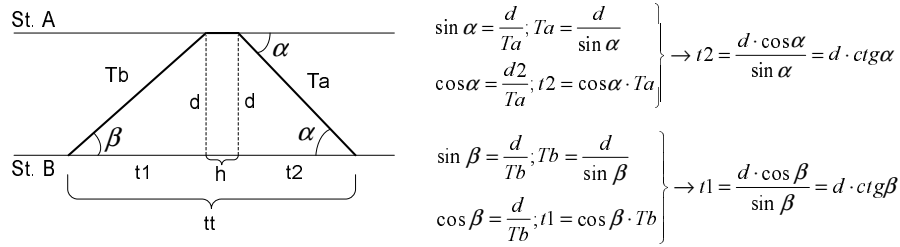


Figure 4: Angle per train and track.

$\alpha \rightarrow 90$  means that the train has traveled a very long distance in a short time.

Once, we have ranged the speed in the interval  $[0, 90]$ , we will study railway capacity by means of the minimum security distance for each track. We will focus on periodic trains and one-way tracks. Thus, all trains in each direction are of the same type.

For each track of distance "d" between two contiguous stations and each pair of trains in opposite direction, the total time (tt) of both journeys (up direction and down direction) is  $tt = h + t1 + t2 = h + d \cdot (\cot \alpha + \cot \beta)$ , where  $h = mrt + met$  is the sum of the *minimum reception time* (mrt) of the train that arrives plus the *minimum expedition time* (met) of the train that departs. These values (t1 and t2) are trigonometrically obtained as can be seen in Figure 5.



$$tt = h + d \cdot ctg \alpha + d \cdot ctg \beta = (ctg \alpha + ctg \beta) \cdot d$$

Figure 5: Minimum security distance of a track.

Applying this formula to each of the  $k$  tracks of the network, the railway capacity (denoted by  $C$ ) for a time period  $tp$  is:

$$C = 2 \cdot \left\lfloor \frac{tp}{\text{Max}\{h_i + d_i \cdot (\cot \alpha_i + \cot \beta_i)\}_{i=1 \dots k}} \right\rfloor \quad (1)$$

It must be taken into account that the bottleneck of the railway will occur in the station that satisfies  $\text{Max}\{h_i + d_i \cdot (\cot \alpha_i + \cot \beta_i)\}_{i=1 \dots k}$ . So that, the minimal frequency must be greater than the bottleneck.

**Theorem.** Given a railway  $R$  with one-way tracks with capacity  $C$  in a period  $tp$ , no additional train can be scheduled.

*Proof.* (Proof by Contradiction.) Assume to the contrary that there is a new train  $T$  that can be scheduled in  $R$ . Given the capacity  $C$  and the period  $tp$ , we can obtain the track  $j$  with maximum total time. Due to track  $j$  is a one-way track and train  $T$  is scheduled in this track, then to avoid crossing this train can only arrive or departure from critical station (all trains join) in the temporal interval  $h$ , that is, between the arrival of the predecessor train and the departure of the successor train. In this case  $tt_k = h_k + t1 + t2$ , where  $h_k$  is the sum of  $mrt$  plus  $met$  plus the necessary time for train  $T$  to cross the station ( $cs$ ). Obviously  $cs > 0$ , so that  $h_k = mrt + met + cs$ . However, we define  $h_k = mrt + met \rightarrow cs = 0$ . # contradiction. So, any new train can be scheduled in railway  $R$ .  $\square$

## 4 Topological Constraint Optimization Technique

The railway optimization problem is considered to be more complex than job-shop scheduling [6], [12]. Here, two trains, traveling in opposite directions use tracks between two locations for different durations, and these durations are causally dependent on how the scheduling itself is done (ie: order of tasks), due to the stopping, and starting time for trains in a non-required technical stop, expedition, reception times, etc. Some processes (detour from the main railway) may or may not be required for each train at each location. In our system, the problem is modeled as a COP, where finite domain variables represent frequency and arrival and departure times of trains of locations. Relations on these variables permit the management of all the constraints due to the user requirements, topological constraints, traffic rules, commercial stops, technical operations, maximum slacks, etc. Hundred of trains, of different types, in different directions, along paths of dozens of stations have to be coordinated. Thus, many variables, and many and very complex constraints arise. The problem turns into a mixed-integer programming problem, in which thousands of inequalities have to be satisfied and a high number of variables take only integer values. As is well known, this type of model is far more difficult to solve than linear programming models. So, our objective is to solve this problem previously assigning values to integer variables such that the mixed-integer programming problem is transformed into a linear programming problem. Then, the linearized problem is easily solved. In this way, the topological constraint optimization technique is committed to this goal.

The topological constraint optimization technique generates traffic patterns based on our definition of railway capacity.

For example, if two stations are so far away so that a train takes more than ten minutes in travelling this itinerary, there exist a physical constraint (bottleneck), because it is not possible to assign a five minutes frequency due to trains in opposed directions must be crossed between both stations. As we



pointed out in section 3, these type of tracks determine the railway capacity. Figure 6 shows this feature. Thus, the frequency must be greater than the minimal allowed frequency between both stations.

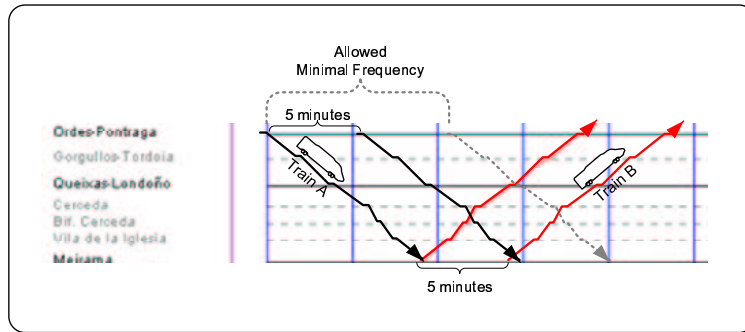


Figure 6: Example of bottleneck.

#### 4.1 Topological Technique

The main idea of this technique is to generate a traffic pattern based on our definition of railway capacity. Each traffic pattern is generated for each set of stations such that the union of these traffic patterns determine the journey of each train. Figure 7 shows a possible set of stations (block).

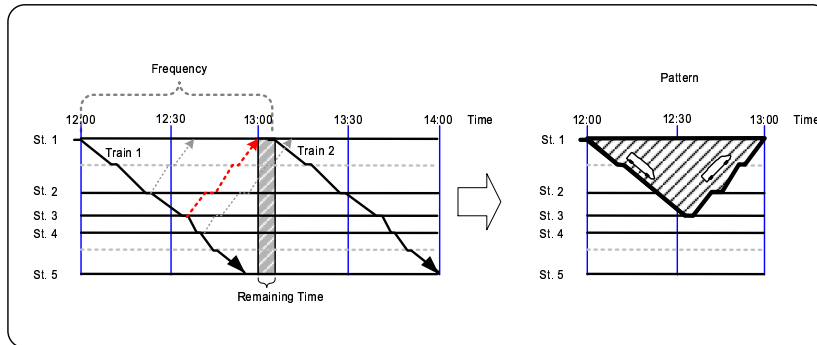


Figure 7: First traffic pattern generation.

The block of stations will be selected taking into account the speed of the trains, the distance among stations and mainly the frequency inserted into the problem. Each traffic pattern covers the block of stations necessary for a train to go from the first station of the block to the last station of the block and return from the last station to the first one (round trip). This round trip must arrive to the first station (St.1) as close but before to the following train departure (Train 2) as possible. Thus, our objective is to minimize the

remaining time between the frequency and the round trips. Each possible round trip will involve a different set of constraints. The round trip that minimize the remaining time will be selected as the *pattern*. This traffic pattern will be composed by a higher number of stations than the rest of possible round trips.

Once the first traffic pattern has been generated, the block of stations involved in this traffic pattern are temporally removed in order to study the following pattern with the remaining stations. Figure 8 shows the generation of the second pattern using the same strategy.

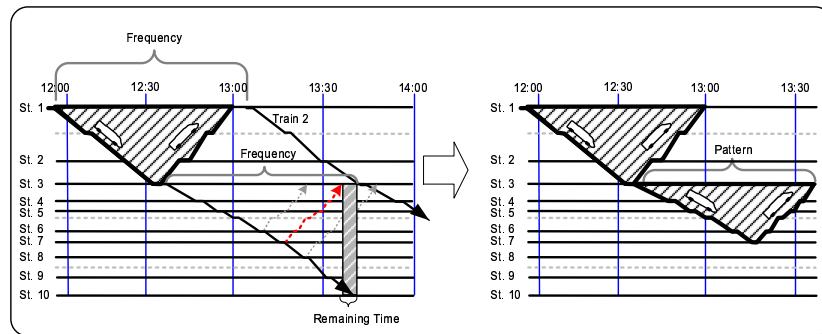


Figure 8: Second Pattern generation.

Therefore, when the second traffic pattern is generated, the topological technique studies the following traffic pattern until there is no station left. In Figure 9, we can observe an example of running map with three complete traffic patterns and some stations without traffic pattern. However, it is usual that there are some stations left. These stations are not involved in any traffic pattern. We must take into account that the best traffic pattern in a block of stations implies to start the following block of stations in the last station of the previous block. We must check all traffic patterns together in order to obtain the journey. Moreover, the first combination of traffic patterns may not be the best solutions due to existence of some combinations of traffic patterns. This combination depends on the number of stations that are not involved in a traffic pattern.

Figure 9 shows an example in which three stations are not involved in any traffic pattern. So, some combinations are possible and they are restricted to the set of stations involved in the first traffic pattern. Thus, these three stations can be sorted between the first and the last traffic pattern. In this way, the first traffic pattern may start at the second or third station and the last traffic pattern may finish in the penultimate or last but two station. However, due to efficient use of resources, or depending on the importance of the station, it is more appropriate the first traffic pattern (last traffic pattern) starts (finishes) at the first (last) station.

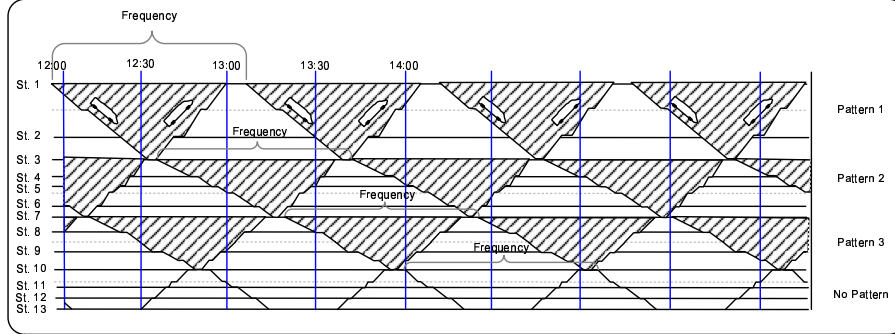


Figure 9: Periodic Pattern generation.

## 5 Evaluation

The application and performance of this system depends on several factors: Railway topology (locations, distances, tracks, etc.), number and type of trains (speeds, starting and stopping times, etc.), frequency ranges, initial departure interval times, etc.

In this section, we compare the performance of our topological technique with some well-known COP solvers: CPLEX and LINGO, because they are the most appropriate tools for solving these types of problems. However, the system carried out important preprocessing heuristics [1] before executing these well-known COP solvers in order to significantly reduce the size of this type of problems.

This empirical evaluation was carried out integrating both different types of problems: benchmark (real) problems and random problems. Thus, we defined random instances over a real railway infrastructure that joins two important Spanish cities (La Coruña and Vigo). The journey between these two cities is currently divided by 40 dependencies between stations (23) and halts (17).

In our empirical evaluation, each set of random instances was defined by the 3-tuple  $\langle n, s, f \rangle$ , where  $n$  was the number of trains in each direction,  $s$  the number of stations/halts and  $f$  the frequency. The problems were randomly generated by modifying these parameters. Thus, each of the tables shown sets two of the parameters and varies the other one in order to evaluate the algorithm performance when this parameter increases.

In Table 1, we present the running time and the journey time in problems where the number of trains was increased from 5 to 50 and the number of stations/halts and the frequency were set at 40 and 90, respectively:  $\langle n, 40, 90 \rangle$ . The results show that CPLEX obtained better running time and journey time than LINGO. However, it can be observed that the running time is lower using the topological technique than the other two COP tools. Furthermore, our technique always obtained the same journey time (lower than CPLEX and LINGO) due to the fact that it generates the corresponding traffic patterns and

it is independent of the number of trains. Figure 10 shows the system interface executing our technique with the instance  $\langle 10, 40, 90 \rangle$ . The first window shows the user parameters, the second window presents the best solution obtained in this moment, the third window presents data about the best solution found, and finally the last window show the obtained running map.

Table 1: Running time and journey time in problems with different trains.

$\langle n, 40, 90 \rangle$	CPLEX		LINGO		TOPOLOGICAL	
Trains	runtime	journey time	runtime	journey time	runtime	journey time
5	5''	2:29:33	8''	2:30:54	3''	2:22:08
10	8''	2:26:04	17''	2:31:37	4''	2:22:08
15	13''	2:26:18	24''	2:31:51	5''	2:22:08
20	16''	2:26:25	35''	2:31:58	5''	2:22:08
50	55''	2:31:09	1302''	2:32:11	10''	2:22:08

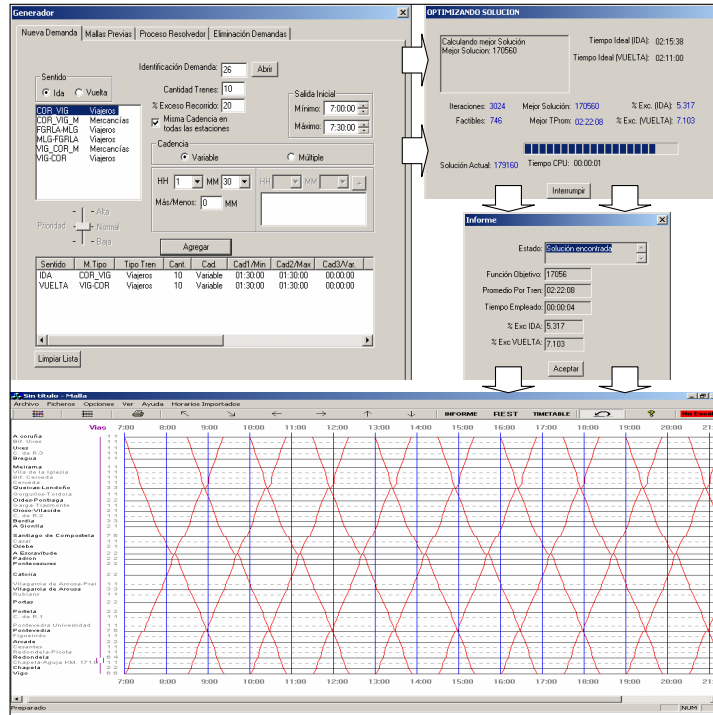


Figure 10: System Interface.

Table 2 shows the running time and the journey time in problems where the number of stations was increased from 10 to 60 and the number of trains and the frequency were set at 10 and 90, respectively:  $\langle 10, s, 90 \rangle$ . In this case, only stations were included to analyze the behavior of the techniques. It can be

observed that the running time was lower using our technique in all instances. The journey time was also improved using our topological technique. It is important to realize the difference between the instance  $\langle 10, 40, 90 \rangle$  of the Table 1 and the instance  $\langle 10, 40, 90 \rangle$  of the Table 2. They represents the same instance, but in Table 2 we only used stations (no halts), so that the number of possible crossing between trains is much more larger. This item reduced the journey time from 2:22:08 to 2:20:22, but the number of combination increased the running time from 4" to 7". Furthermore, CPLEX and LINGO maintained similar behaviors.

Table 2: Running time and journey time in problems with different number of stations.

$\langle 10, s, 90 \rangle$	CPLEX		LINGO		TOPOLOGICAL	
<i>Stations</i>	<i>runtime</i>	<i>journey time</i>	<i>runtime</i>	<i>journey time</i>	<i>runtime</i>	<i>journey time</i>
10	2"	0:58:36	4"	0:58:06	1"	0:57:36
20	3"	1:04:11	20"	1:04:11	2"	1:04:11
30	15"	1:45:08	42"	1:45:38	4"	1:45:08
40	56"	2:23:16	28"	2:24:36	7"	2:20:22
60	340"	3:44:28	326"	3:44:22	40"	3:32:15

In Table 3, we present the running time and the journey time in problems where the frequency was increased from 60 to 140 and the number of trains and the number of stations were set at 20 and 40, respectively:  $\langle 20, 40, f \rangle$ . It can be observed that the frequency the topological technique improved the journey time when the frequency increased. As in previous results, the running time of the topological technique was lower than CPLEX and LINGO.

Table 3: Running time and journey time in problems with different cadencies.

$\langle 20, 40, f \rangle$	CPLEX		LINGO		TOPOLOGICAL	
<i>Frequency</i>	<i>runtime</i>	<i>journey time</i>	<i>runtime</i>	<i>journey time</i>	<i>runtime</i>	<i>journey time</i>
60	>43200"	-	>43200"	-	36"	2:32:11
90	17"	2:26:25	32"	2:31:58	5"	2:22:08
100	18"	2:23:10	34"	2:22:55	3"	2:19:09
120	16"	2:16:17	27"	2:18:47	4"	2:16:00
140	17"	2:20:18	27"	2:16:19	4"	2:17:03

## 6 Conclusions

We have formalized railway capacity to design a topological model for solving periodic train scheduling in collaboration with the National Network of Spanish Railways (RENFE). This technique has been inserted into the system to solve more efficiently periodic timetables. This system, at a current stage of integration, supposes the application of methodologies of Artificial Intelligence in

a problem of great interest and will assist railways managers in optimizing the use of railway infrastructures and will help them in the resolution of complex scheduling problems.

## References

- [1] Barber, F., Salido, M.A., Ingolotti, L., Abril, M., Lova, A., Tormos, P. *An Interactive Train Scheduling Tool for Solving and Plotting Running Maps*, Current Topics in Artificial Intelligence LNAI 3040, 646-655 (2004).
- [2] Bussiecky, M.R., Winter, T., Zimmermann, U.T. *Discrete optimization in public rail transport*, Mathematical Programming 79(3), (1997), 415-444.
- [3] Caprara, A., Fischetti, M., Guida, P., Monaci, M., Sacco, G., Toth, P. *Solution of Real-World Train Timetabling Problems*, 34th Annual Hawaii International Conference on System Sciences **3** (2001).
- [4] Chiu, C.K., Chou, C.M., Lee, J.H.M., Leung, H.F., and Leung, Y.W., *A Constraint-Based Interactive Train Rescheduling Tool*, Constraints **7** (2002), 167-198.
- [5] Kaas, A.H., *Methods to Calculate Capacity of Railways*, Ph. Dissertation (1998).
- [6] Kreuger, P.; Carlsson, M.; Olsson, J.; Sjoland, T.; Astrom, E. *The TUFF train scheduler: Two duration trip scheduling on multi directional track networks*. In Proc. of the workshop on Tools and Environments for (Constraint) Logic Programming, (1997).
- [7] Krueger, H. *Parametric Modelling in Rail Capacity Planning*. Canadian National Railway, (1998).
- [8] Lindner, T., *Train schedule optimization in public rail transport*, Ph. Dissertation, Technische Universitat Braunschweig, Germany, (2000).
- [9] Oliveira, E., Smith, B.M., *A Job-Shop Scheduling Model for the Single-Track Railway Scheduling Problem*, Research Report 2000.21, University of Leeds, (2000).
- [10] Salido, M.A., Barber, F., *A Constraint Ordering Heuristic for Scheduling Problem*, 1st Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2003), (2) 476-491, (2003).
- [11] Weits, E., *Railway Capacity and Timetable Complexity*, 7th International Workshop on Project Management and Scheduling, Euro (2000).
- [12] Zuidwijk, R.A., Kroon, L.G., *Integer Constraints for Train Series Connections*, Erasmus Research Institute of Management (ERIM), Discussion Paper. (2002).