



**HAL**  
open science

# Scheduling theory and constraint programming applied to rail traffic management

Joaquin Rodriguez

► **To cite this version:**

Joaquin Rodriguez. Scheduling theory and constraint programming applied to rail traffic management. Scheduling theory and constraint programming applied to rail traffic management, in Farewell address, Prof. Ingo Hansen, TECHNICAL UNIVERSITY DELFT, p133-145, 2012. hal-00959593

**HAL Id: hal-00959593**

**<https://hal.science/hal-00959593v1>**

Submitted on 20 Mar 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Scheduling theory and constraint programming applied to rail traffic management

Joaquín Rodríguez

ESTAS - IFSTTAR / PRES Université Lille Nord de France

Scheduling theory (ST) is a formal framework which have been used many times over the last decade as starting point to model railway traffic management. Graph theory (GT), mathematical programming (MP) or constraint programming (CP) have also be used as additional steps to scheduling theory. This is illustrated in figure 1. The last step is a solution algorithm that is many times based a complete or partial enumeration procedure (EP) of feasible solutions where large subsets are discarded. This article will focus on the path  $ST \rightarrow CP \rightarrow EP$  to solve railway management problems. I will first briefly introduce each field, and after applications to the railway traffic management.

## 1 Scheduling theory

Scheduling theory has been studied since 1950s, the first problems were taken on the field of management of operations in workshops. Scheduling has been used in many other areas but a specific scheduling problem were defined for each situation. This result a huge number of scheduling algorithms for a large number of specific problems. A widely-used category of problem is the job-shop scheduling (JSS) problem. A JSS problem consists of a set of  $n$  jobs that can run on  $m$  machines. A job consists of operations which may also be referred to as tasks or activities. Each activity requires a machine (also named a resource) during its execution. There are two types of constraints in a JSS problem:

- precedence constraints between activities;
- resource constraints that assert that machines are unary resources, i.e. no two activities requiring the same resource may execute at the same time.

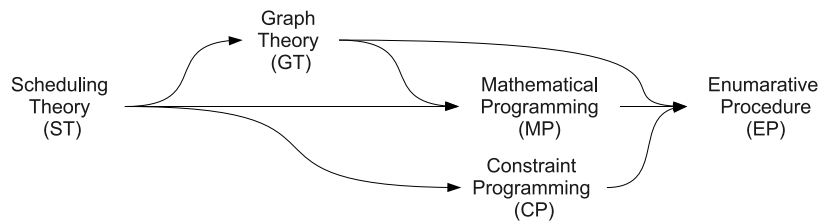


Figure 1: Steps to model railway traffic management with scheduling theory

The preemption is not permitted, i.e. the processing of an operation of a job cannot be interrupted. Each job may follow one or more machine sequences. If each job follows one machine sequence, operations are pre-assigned to particular resources, as a result scheduling means the decision when to execute operations. If each job may follow different machine sequences, there may be several alternative resources to which the operation can be allocated. In that case resource allocation is part of the scheduling task. Some widely used objectives are : the minimization of the maximal completion time (so called makespan), the minimization of the maximal lateness, the minimization of the sum of all the flow times that a jobs spend in the workshop (so called flow times), the minimization of the sum of all the tardiness of the jobs. As many JJS problems have been shown to be NP-hard, researchers often develop heuristic methods to find good solutions rather than exact methods to find optimal solutions.

## 2 Scheduling theory in railway traffic management

One the first work which use the scheduling theory for rail traffic management modeling go back to the 70's with the work by Spzigel [Spzigel, 1973]. The problem considered was the timetable construction of a single-track line. In a single-track line, each pair of stations are linked by a track segment where at most one train can be at any time. Spzigel pointed out the similarity of the time-table construction and the JSS problem : each train is an ordered sequence of activities. In his model, track segments are unary resources used to perform an activity. In the case of stations, Spzigel's model assumes that stations can accept any number of trains, i.e. that their capacity is infinite. The objective function is the weighted sum of the journey times which is equivalent to the weighted sum of the times each train enters the last segment. Resolution uses a Branch and Bound method, with the values being chosen in priority from the disjunctive constraints between the trains that are in conflict on a track segment. The instances used relate to a line with 5 track segments, with half the trains traveling in each direction. The journey time on the segments is the same and the number of trains varies between 4 and 10.

Other authors [Higgins et al., 1996, Kreuger et al., 1997, Isaii and Singh, 2000] follow the idea of modeling segments of single-track lines as unary resources or by posting disjunctive constraints between activities which require the segment.

When trains are running on the same direction a more detailed model is needed to take into account the safety constraints to protect against rear collision. To avoid rear collisions, the block signals enforce train headway such that at most one train can be in a block at any time. Therefore the idea was to model each block segment with a unary resource. It can be notice that this model include the model of Spzigel and it is the most widely used approach [Brännlund et al., 1998, Oliveira, 2001, Kreuger et al., 2001, Mascis and Pacciarelli, 2002, Barber et al., 2004, Geske, 2005, Mazzarello and Ottaviani, 2005, Schlenker, 2006, D'Ariano et al., 2007, Tornquist and Persson, 2007, Corman et al., 2010]

In [Mascis and Pacciarelli, 2002] Mascis et al. studied specific JSS model with blocking and no-wait constraints. This category of JJS model was used in [Mascis et al., 2002] and

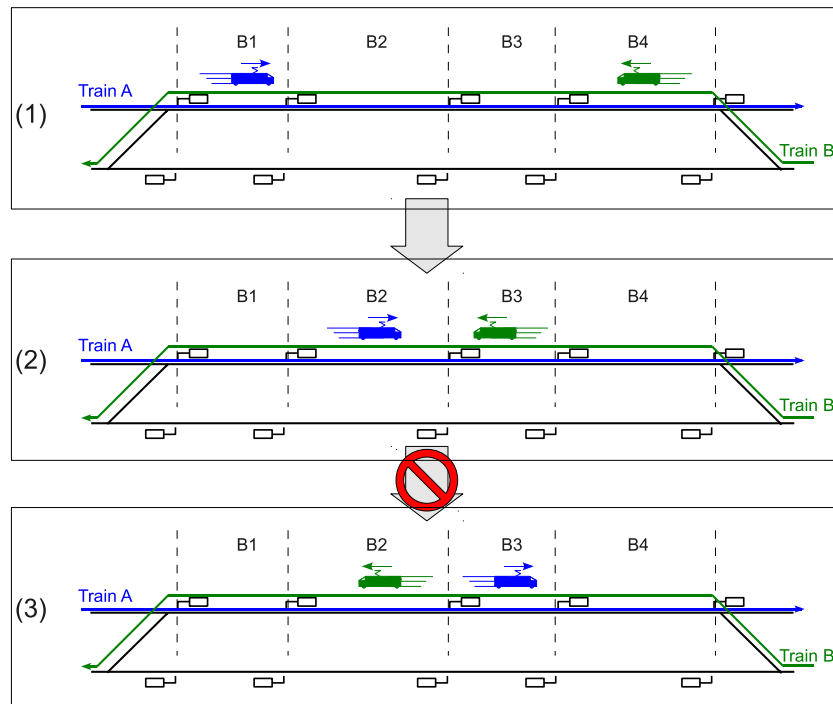


Figure 2: Deadlock example forbidden by no swap blocking constraint

other works are based on this approach [Mazzarello and Ottaviani, 2005, D'Ariano et al., 2007, Corman et al., 2010]. The no-wait constraints allow to model that each activity will start immediately after its direct predecessor in a job. The blocking constraints allow to model that each activity of a job after complete processing on a resource will remain in it until the resource for the next activity become available for processing. Due attention must be paid to use no swap blocking constraints or any other equivalent technique to not missing some deadlocks. The figures 2 and 2 illustrate a deadlock example which could be missed without the no swap blocking constraints. This kind of possible infeasible train movements of the JSS block model was recently pointed out in [Harrod, 2011] and solved with a directed hypergraph model.

Our first work using a JSS model for rail traffic management was in [Rodriguez and Kermad, 1998]. The problem was to schedule and route trains to minimize secondary delays during operation on complex junctions or stations. We defined a more accurate model than the JSS block model because a block is too imprecise to represent the incompatibilities correctly i.e. potential train conflicts or deadlocks. Track circuits are more appropriate for specifying the resource used to determine the incompatibilities between train times. Therefore, the idea was to model each track circuit for train detection with a unary resource. This is illustrated in the figure 4. This model take into account the temporal position of the head and tail of train, i.e. a train is not considered as a point but it has a length. The start

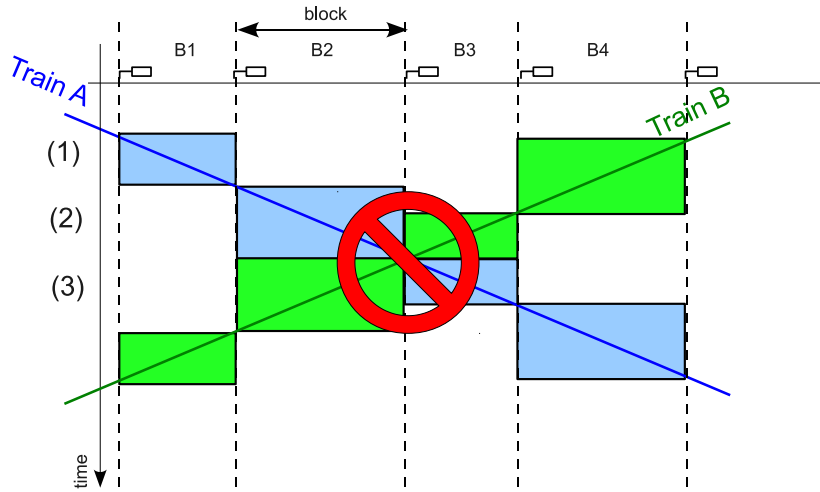


Figure 3: Time-Distance diagram of the deadlock example of fig. 2

and end variables of the activities are an expression of the time when the head enters or the tail clears the track circuits. This modeling feature allows to easily formulate temporal constraints for the signal watching time, several aspects signaling, the clearing time, the switching time and the sectional route release implemented by the interlocking system. One can notice that all these constraints are complying with the so-called «blocking time theory» defined in the UIC leaflet 406 [UIC, 2004]. The clearing time constraint between consecutive activities can prevent infeasible swap movement when there are deadlocks such as in figure .

### 3 Constraint programming

Constraint programming is a technique used to represent constraint satisfaction problems. In very general terms, a constraint satisfaction problem (CSP) is defined by the following ordered triple  $\mathcal{P} = (X, D, C)$  such that:

- $X = \{x_1, \dots, x_m\}$  is a set of variables,
- $D = \{d_1, \dots, d_m\}$  is a set of finite domains where a domain  $d_i$  is a set of possible values for the variable  $x_i$ ,
- $C = \{c_1, \dots, c_e\}$  is a set of constraints where each constraint  $c_i$  limits the combinations of possible values for a subset of variables  $X$ .
- 

For a given CSP instance, it is possible to either look for one solution, all the solutions, or an optimum solution with respect to an objective function. In the last case, a

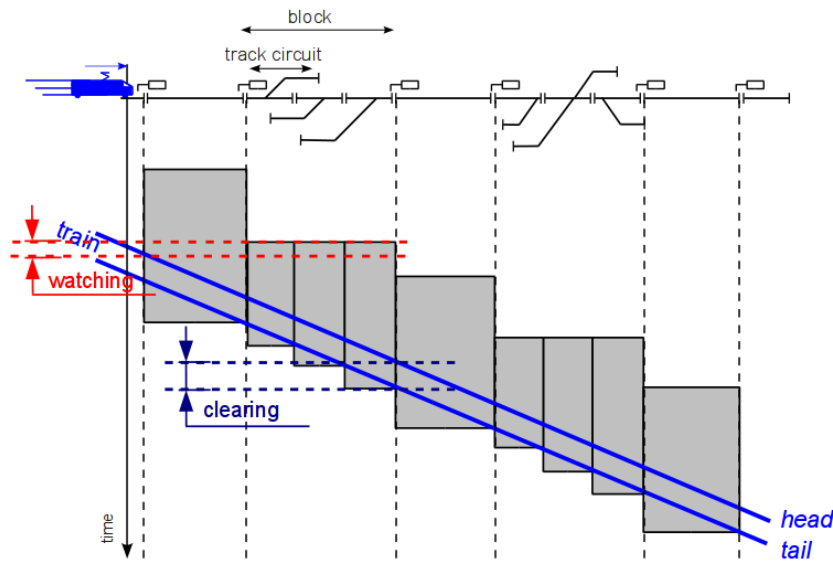


Figure 4: Time-Distance diagram of a JSS model with track circuit resources

CSP associated with an objective function is a combinatorial optimization problem. Constraint programming (CP) provides the languages and tools which allow us to define a CSP and program resolution algorithms. These tools use a constraint-based process to reduce the search space which reduces the amount of computer processing required to resolve the problem. This process, which is referred to as constraint propagation, not only validates the values that have been selected for some variables, but, above all, provides a way of removing values from the domains. The removed values can no longer feature in a solution in view of the decisions that have been made. It is this removal of values from the domains which makes it possible to converge towards a solution to the problem or, on the contrary, demonstrate that there is no solution.

## 4 Constraint programming in railway traffic management

To our knowledge, Fukumori [Fukumori, 1980, Fukumori and Sano, 1987] was among the first to propose the use of constraint programming in order to model a rail traffic management problem. This research dealt with constructing a timetable on the basis of specifications of a set of constraints that applied to the trains. The decision variables related to trains overtaking or crossing in stations. Consequently, the main variables in the model were departure and arrival times at stations. The criterion to be optimized was each train's journey time in order to limit excessive train standing time in stations. The model's constraints expressed:

(Ct. 1) the minimum headways which trains must observe in the sections of track between

stations,

(Ct. 2) the minimum durations for stopping, crossing and overtaking in stations;

(Ct. 3) the maximum number of trains which each station can take.

A Depth First Search (DFS) resolution algorithm was used. This makes trade-offs with regard to track use conflicts on the basis of the priorities assigned to the trains. The instantiation order of the variables is the same as the journey order of the trains. When this research was carried out, CP tools were not in common use and a constraint propagation algorithm was therefore developed specifically. This algorithm seeks consistency with regard to the bounds of the variable domain, which is similar to arc-B-consistency [Lhomme, 1993]. Fukumori's model was validated on problem instances from one day's operation of the Osaka and Nara lines [Fukumori and Sano, 1987]. The examples given in the article relate to the instances for 14 stations and 40 trains.

A number of subsequent studies have dealt with the timetable construction problem using Fukumori's model as a basis. The CAPRES model developed by Hachemane [Hachemane, 1997], uses the same constraints with the addition of some constraints which are specific to the applications in question. The most striking extension is that stations may be modeled with greater precision. For example, this model can include incompatibilities between routes into and out of stations. However, this option is only used to prove the feasibility of a solution. The constraint consistency algorithm is of the type used by Fukumori and in this case too it is implemented without a CP library. The CAPRES model has been used in many capacity studies (see a survey in [Rivier et al., 2001]), and was subsequently included in the Viriato software package [SMA und Partner AG, 2005].

The PRaCoSy project model described by Chiu et al. [Chiu et al., 1996] also makes use of the constraints defined by Fukumori. It involves a different application, that of assisting an operator and designing a timetable in order to cope with disruptions. Two new criteria have been considered in this case:

- Minimizing the modifications to the theoretical timetable,
- Minimizing the largest delay in comparison with the theoretical timetable.

Two resolution heuristics have been tested. The first uses an instantiation order based on the smallest value in the variable domain. The second uses an instantiation order which as far as possible conserves the values in the initial timetable. Thus, the variables for the conflicting trains and for the trains which share part of their route with the conflicting trains are the last to be instantiated. The second heuristic gave better results for the problem instances from the Nanjingxi-Shanghai line. Implementing this instance with the CP Ilog Solver library gave a problem with 598 variables and 3005 constraints. However, the paper quotes no indicators to give an idea of the difficulty of the problems dealt with (e.g. the number of conflicting trains).

Ingolitto et al. [Ingolitto et al., 2004] also make use of Fukumori's model and CSPs. As with CAPRES, the problem involved timetable saturation. A resolution heuristic is used

which divides the line in two in order to make a scheduling decision between the saturating trains. It should be noted that the algorithm only verifies the constraints for the use of single tracks. If the verification fails, the times of the trains are changed.

Isaai and Singh [Isaai and Singh, 2000] propose a model of the same type as Fukumori in order to solve a timetable construction problem. Their resolution method combines the CP heuristic in order to obtain a solution that is feasible and a taboo heuristic to improve this solution in a neighborhood. The 5 data sets used for the tests were obtained from two lines with single and double tracks. These lines had a total of about 50 stations. The number of conflicts was estimated at between 25 and 62. The paper's results show that the taboo method improves the initial solution given by the CP heuristic.

The decision variables used in the above research principally relate to train scheduling with shared resources (station platforms, track sections). Constraint programming has also been applied to problems with additional variables for train routing, i.e.; the resources allocation. Adding of these variables greatly increases the size of the search space which makes it very difficult to find a solution which is both acceptable and optimum.

To our knowledge, Gosselin in [Gosselin, 1993] describes the first application which uses constraint programming to deal simultaneously with the problems of train scheduling and train routing. In view of the number of variables and constraints, the problem was limited to the search for an acceptable solution. To be able to do this, it is essential to define the constraints that propagate between the times of station arrival and departure events and alternative routes. The resolution method is guided by a route evaluation function which enables it to find solutions more rapidly. In order to restrict the size of the search space, the number of possible routes between stations was limited to 4. The model was applied to 10 instances on the Melbourne suburban network, with the size of the instances varying between 200 and 2000 trains. Resolution times were less than one hour which satisfied customer requirements.

Poujade et al. [Poujade et al., 1995] also used a CP model to deal with a routing problem. The application involved allocating platform lines at the Gare du Nord in Paris. In this application, the train times at the reference points were fixed. The difficulty of the problem lay in the number of routes and the number and type of constraints to be satisfied. The initial problem in the application was over-constrained, i.e. there was no possible solution. The constraints were then divided into three categories: those that absolutely had to be satisfied, those which could be modified and those which had to be satisfied as far as possible. In the resolution method, the possible number of routes for a train was so great (several thousand) that these routes were only calculated after the platform lines had been assigned (terminus and/or intermediate stops). Then, route selection focused on those which were as direct as possible. In the event of failure, new platform line assignments were tried out. If the search ended in failure, the sets of conflicting trains were identified in order for the user to be able to choose which constraint to relax and then initiate a new search. This application is operational and used on a daily basis. The routes cover a zone with a radius of 16 km around the Gare du Nord which includes 10 stations, 100 platforms for 1700 trains and the model has a resolution time of 15 minutes.



The «track circuit» JSS model [Rodríguez and Kermad, 1998] mentioned in section 2 was formulated as a Constraint Satisfaction Optimization Problem and implemented with a CP library (IBM Ilog Solver/Scheduler). It was successively improved to deal with more complex instance problems [Rodríguez, 2007a]. In [Rodríguez, 2007b] we introduce the use of state resources to early prune scheduling decision leading to opposite direction conflicts. Thereafter [Rodríguez and Sobieraj, 2009] we define a new heuristic which performs incrementally decisions at very local level i.e. at track circuit level. At each step, the algorithm performs decisions of allocation of a track circuit or of scheduling a pair of train runs on this track circuit. For this later solution method, the experimental study was a set of 18 instance problems with increasing difficulty. The difficulty of the instance problems is related to the number of trains (6 to 40 trains) and conflicts between trains (1 to 26). The instances were generated by selecting train services from the services of a peak hour of the station of Lille-Flandres, the main station of Lille in the North of France. The instances considered correspond to instances with 2000 to 22000 variables and 3000 to 26000 constraints. This later solution method have shown that good quality solutions can be obtained with processing times (less than 30 seconds) which are compatible with the operational constraints.

## 5 Prospect

This paper has briefly presented ST and CP models for the railway traffic management problems. We focused on one path  $ST \rightarrow CP \rightarrow EP$  of the graph of figure 1. Despite contributions to improve the methods of solving problems, this model is still only predictive but not explanatory as it is argued in [Hooker, 2007]. Indeed all models are generally confined to a set of computational techniques but they do not help to understand the phenomenon they are studying : the railway traffic management. To our knowledge, there is still no modeling contribution that highlight the structure of a problem of railway traffic management and can be used to understand why a solution is good. Moreover, there is no reason to consider a single path in the graph of figure 1. On the contrary, all paths, i.e. all techniques, should be available to exploit the structure of the problem and solve it. Therefore, future prospects are to integrate different optimization techniques (ST, GT, IP) to obtain efficient predictive models but also explanatory models of the railway traffic management.

## References

- [Barber et al., 2004] Barber, F., Salido, M. A., Abril, M., Ingolotti, L., Lova, A., and Tormos, P. (2004). An interactive train scheduling tool for solving and plotting running maps. *LNCS/LNAI*, (3040):659–668.
- [Brännlund et al., 1998] Brännlund, U., Lindberg, P. O., Nou, A., and Nilsson, J.-E. (1998). Railway timetabling using lagrangian relaxation. *Transportation Science*, 32(4):358–369.

- [Chiu et al., 1996] Chiu, C. K., Chou, C. M., Lee, J. H. M., Leung, H.-F., and Leung, Y. W. (1996). A constraint-based interactive train rescheduling tool. In *Second International Conference on Principles and Practice of Constraint Programming*, pages 104–118.
- [Corman et al., 2010] Corman, F., D’Ariano, A., Pacciarelli, D., and Pranzo, M. (2010). A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B*, 44:175–192.
- [D’Ariano et al., 2007] D’Ariano, A., Pacciarelli, D., and Pranzo, M. (2007). A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183:643–657.
- [Fukumori, 1980] Fukumori, K. (1980). Fundamental scheme for train scheduling: Application of range-constriction search. Technical Report A.I. Memo No. 596, Massachusetts Institute of Technology Artificial Intelligence Laboratory, USA.
- [Fukumori and Sano, 1987] Fukumori, K. and Sano, H. (1987). Fundamental algorithm for train scheduling based on artificial intelligence. *Systems and Computers in Japan*, 18(3):52–64.
- [Geske, 2005] Geske, U. (2005). Railway scheduling with declarative constraint programming. In *INAP’05*, pages 117–134.
- [Gosselin, 1993] Gosselin, V. (1993). Train scheduling using constraint programming techniques. *Charme Technical Bulletin*, (3):3–13.
- [Hachemane, 1997] Hachemane, P. (1997). *Évaluation de la capacité de réseaux ferroviaires*. Thèse 1632, École Polytechnique Fédérale de Lausanne. (in french).
- [Harrod, 2011] Harrod, S. (2011). Modeling network transition constraints with hypergraphs. *Transportation Science*, 45(1):81–97.
- [Higgins et al., 1996] Higgins, A., Kozan, E., and Ferreira, L. (1996). Optimal scheduling of trains on a single line track. *Transportation Research Part B*, 30(2):147–161.
- [Hooker, 2007] Hooker, J. N. (2007). Planning and scheduling by logic-based benders decomposition. *Operations Research*, 55(3):588–602.
- [Ingolitto et al., 2004] Ingolitto, L., Barber, F., Tormos, P., Lova, A., Salido, M., and Abril, M. (2004). An efficient method to schedule new trains on a heavily loaded railway network. In Reyes, C. and Gonzales, J., editors, *Advances in Artificial Intelligence, IBERAMIA 2004, 9th Ibero-American Conference on AI*, Lecture Notes in Computer Science, pages 164–173, Puebla, Mexico. Springer Verlag.
- [Isaai and Singh, 2000] Isaai, M. and Singh, M. (2000). An intelligent constraint-based search method for a single-line passenger-train scheduling problems. In *The International Conference on the Practical Application of Constraint Technologies and Logic Programming*, pages 79–91.

- [Kreuger et al., 1997] Kreuger, P., Carlson, M., Olsson, J., Sjöland, T., and Aström, E. (1997). Trips scheduling on single track networks - the tuff train scheduler. In *CP'97 Workshop on Industrial Constraint - Directed Scheduling*, pages 1–12.
- [Kreuger et al., 2001] Kreuger, P., Carlsson, M., Sjöland, T., and Aström, E. (2001). Sequence dependent task extensions for trip scheduling. Technical Report SICS-T-2001/14-SE, Swedish Institute of Computer Science (SICS),.
- [Lhomme, 1993] Lhomme, O. (1993). "Consistency techniques for numeric CSPs". In *13th International Joint Conference on Artificial Intelligence*, pages 232–238, Chambéry, France.
- [Mascis et al., 2002] Mascis, A., Pacciarelli, D., , and Pranzo, M. (2002). Models and algorithms for traffic management of rail networks. Technical Report DIA-74-2002, Dipartimento di Informatica e Automazione, Università Roma Tre.
- [Mascis and Pacciarelli, 2002] Mascis, A. and Pacciarelli, D. (2002). Job-shop with blocking and no-wait constraints. *European Journal of Operational Research*, (143):498–517.
- [Mazzarello and Ottaviani, 2005] Mazzarello, M. and Ottaviani, E. (2005). A traffic management system for real-time traffic optimisation in railways. In *1st International Seminar on Railway Operations Modelling and Analysis (RailDelft2005)*, Delft - The Netherlands.
- [Oliveira, 2001] Oliveira, E. S. (2001). *Solving Single-Track Railway Scheduling Problem Using Constraint Programming*. PhD thesis, School of Computing - University of Leeds.
- [Poujade et al., 1995] Poujade, M., Moyon, P., and J.F., P. (1995). Sagitaire: Allocation des quais de la gare Paris-Nord. In *Conférence IA'95*. (in french).
- [Rivier et al., 2001] Rivier, R., Lucchini, L., and Curchod, A. (2001). Analysing the capacity of railway networks: Summing up the experience. In *9th World Conference on Transportation Research*.
- [Rodriguez, 2007a] Rodriguez, J. (2007a). A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B : Methodological*, 41(1):231–245.
- [Rodriguez, 2007b] Rodriguez, J. (2007b). A study of the use of state resources in a constraint-based model for routing and scheduling trains. In *2nd International Seminar on Railway Operations Modelling and Analysis*, Hannover, Germany.
- [Rodriguez and Kermad, 1998] Rodriguez, J. and Kermad, L. (1998). Constraint programming for real-time train circulation management problem in railway nodes. In *Computers in Railways VI*, pages 597–606, Lisbonne-Portugal.

- [Rodriguez and Sobieraj, 2009] Rodriguez, J. and Sobieraj, S. (2009). A study of an incremental texture-based heuristic for the train routing and scheduling problem. In *3rd International Seminar on Railway Operations Modelling and Analysis*, Zurich, Suisse.
- [Schlenker, 2006] Schlenker, H. (2006). Distributed constraint-based railway simulation. In Van Hentenryck, P., editor, *Principles and Practice of Constraint Programming - CP 2002*, volume 2470 of *Lecture Notes in Computer Science*, pages 275–282. Springer Berlin / Heidelberg.
- [SMA und Partner AG, 2005] SMA und Partner AG (2005). Future module : CAPRES - Elaboration And Saturation Of Railway Network. In *Viriato|OpenTrack User-Conference*, Zurich, Suisse.
- [Spzigel, 1973] Spzigel, B. (1973). Optimal train scheduling on a single track railway. In Ross, M. e., editor, *OR'72*, pages 343–352. North Holland Publishing Co.
- [Tornquist and Persson, 2007] Tornquist, J. and Persson, J. A. (2007). N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B*, 41:342–362.
- [UIC, 2004] UIC (2004). Capacity (uic code 406). Technical report, International Union of Railways (UIC), Paris, France.