# The contribution of state resources in a constraint-based scheduling model for conflict solving at railway junctions

J. Rodriguez
*INRETS (Institut National de Recherche sur les Transports et leur Sécurité), Villeneuve d'Ascq, France*

## Abstract

Scheduling is the process of allocating resources to activities over time. In a scheduling problem, resources are scarce and constrained in various ways (e.g. in the order of activities and the capacity of resources), and one is looking for a schedule of the activities that both satisfies the constraints and is optimal according to some criterion.

Over the last decade, constraint-based scheduling (CBS) has become the dominant form of modelling and solving scheduling problems.

This paper deals with a CBS model of real time management of train traffic through junctions, it focuses on the use of state resources to improve the model of conflicts between trains running in opposite directions.

## 1 Introduction

The railway industry has to improve the quality of service provided in order to increase freight and passenger market-shares. One important parameter that affects the quality of service is the efficiency and the effective use of resources. To achieve this aim, one solution is the use of computer-aided systems in planning and traffic control. A significant part of these computer-aided systems involves the model and the solution method of the optimisation problem, associated with real situations and decisions to be taken.

When traffic on connecting lines of a junction is heavy, the junction is likely to be a critical resource. As a consequence, small disruptions are amplified. A few seconds originally can quickly lead to a delay of more than 5 minutes. To limit this

WIT Transactions on The Built Environment, Vol 88, © 2006 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR060581

phenomenon, new tracks, points and fly-overs can be built. Space and investments are needed for such extensions. However, such solutions are unlikely to be feasible in the short or medium term or impossible to implement in urban areas.

This provides the motivation for the study of new methods and models to optimise the use of scarce resources like rail junctions.

Concerning traffic control at a junction, the operator must perform the following tasks :

1. Analyse and select relevant information coming from the field,
2. Compare data with planned schedules,
3. Detect or anticipate conflicts,
4. Select and evaluate alternative solutions that reduce delays caused by conflicts,
5. Choose and implement a solution.

In this context, a computer-based assistance can be used for the first four tasks to improve the quality of the final solution. Task 4 can be formulated as an optimisation problem, the decision variables are the *selection of the alternative routes and sequences for trains* and the criterion is the *sum of delays*. This model and the solution method is part of the computer-aid system; nevertheless, the final decision must be left to the operator.

During the last decade, the problem of railway traffic management has been many times addressed by using a constraint-based scheduling (CBS) approach. An earlier attempt was in [1]: the application was the compilation of a railway timetable i.e. to construct a schedule of trains arrival-departure at each station of a line. The arrival-departure times must satisfy a set of temporal constraints.

Further to this contribution, constraint propagation and scheduling theory has been drawing the attention of researchers to solve railway traffic management problems. We can mention here some studies and applications :

- The generation of timetables for the suburban railway network of Melbourne [2],
- SAGITAIRE : a tool for the design of one day operation of the schedules, platform and route assignments of the trains for “Gare du Nord” station in Paris [3],
- PRaCoSy : a project for the automation of the preparation and updating of the running map for a section of the Beijing-Guangzhou line in China [4],
- CAPRES : a tool to analyse the capacity of railway networks with a saturation method [5],
- The scheduling of trains on single track networks [6, 7],
- COMBINE2 : a European project which aim was to define a conflict resolution model to use in control centres for fixed and moving block signalling systems [8],
- A generic tool for the National Network of Spanish Railways which covers many functions such as analysing the capacity, identifying bottlenecks, providing alternative planning and real-time traffic control [9].

Our first CBS formulation of the traffic management problem at junction was in [10], the last updated and detailed description of the model was in [11].

In this paper, we present an extension of the model of [11] by using *state resource constraints*. These kinds of constraint are redundant constraints. The state resource constraints allow a better management of conflicts between trains running in opposite directions in a section of track.

The paper is organised as follows: our CBS formulation of the railway traffic management is introduced in section 2. Within the framework of this model, we define in section 3 the opposite direction conflicts and the state resources associated. In section 4 the constraints between the activities and the state resources are defined according to different value assignments of the decision variables.

Finally, section 5 gives some preliminary results and issues.

## 2 Constraint-based scheduling model

The basic idea of the model is that *a train passing through a junction is a job.* According to scheduling theory, the concept of job is a set of activities linked by a set of precedence constraints. The movement of a train is a sequence of activities. Each activity is an elementary movement of the train through a track circuit. This is illustrated in figure 1.

As the train remains on track circuit until the next one becomes available for running, this limitation is named a "blocking constraint" in scheduling theory. Therefore our model is similar to that of *blocking job shop scheduling problem* [8, 12].

The constraints of our model will be roughly introduced, there is a more detailed formulation in [11]. The constraints of the problem can be formulated as follows:

- As each track circuit is a resource, the choice of a route for a train is turned into resource assignments for a sequence of activities. A constraint enumerates the combination of tuples of values allowed for the route and track circuit variables.
- The track circuits are modelled as unary resources, this leads to the constraint that two activities requiring the resource cannot overlap in time.
- Within the duration of an activity, we distinguish the detection phase. For each train, a constraint links the route values with the earliest start and finish time of detection phase of each activity.
- For each activity, a waiting time variable models the time spent when the next resource is not available. This time is added in the expression of the duration of the activity.
- The headway constraint between successive trains due to the block signalling system is formulated with a "synchronisation constraint". Let us consider a block signalling system with two aspects. In that case,a train enters a block if no train is detected inside. Therefore, to enter a block, all track circuits inside the block must be available at the same time. The start of each activity related to one block has to be *synchronised* with the start of the detection on the first track circuit of the block. For the general case of a block system with $n$ aspects, the synchronisation is established with

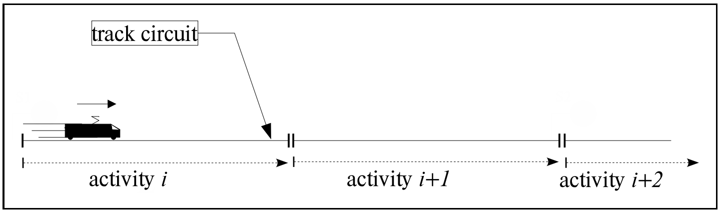

Figure 1: Train movement as a sequence of activities.

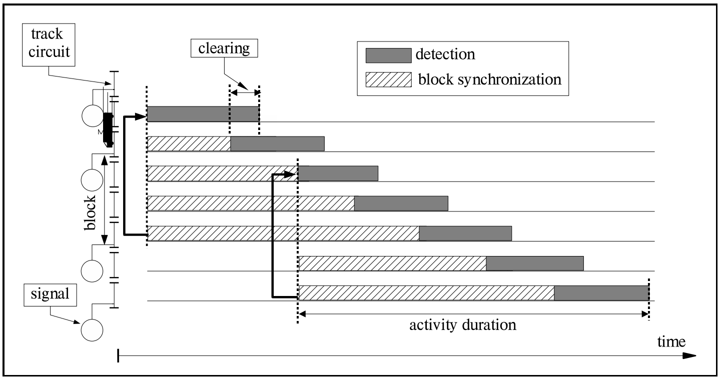

Figure 2: Gantt chart of activities for 3-aspect block signalling system.

the entrance in the first track circuit of the $n - 2^{th}$ previous block (e.g. see hatched rectangles for $n = 3$ in figure 2).

For train scheduling, the criterion frequently used is the sum of train delays caused by conflicts. This criterion is formulated with the sum of the waiting time variables.

## 3 Opposite direction conflicts

Let us define a “conflict sequence” as a common sequence of track circuits requested by two trains. There will be a running conflict when both trains request one track circuit of the conflict sequence during a common time interval. To arbitrate the conflict, the train circulations are ranked on the conflict sequence, i.e. the corresponding activities are ranked according to the train rank decision. For the case of a conflict sequence between routes running in the same direction, a rank decision on one activity implies all ranks of the other activities. This propagation of the rank decision is due to the blocking constraint (see section 2). However, in the case of an opposite direction conflict, there is no propagation of the rank decision; all the activities of the conflict sequence have to be ranked. The search algorithm for a feasible schedule is based on rank decision of activities. When there are opposite direction conflicts, the search algorithm will slow down as it needs more backtracks to find feasible ranks.

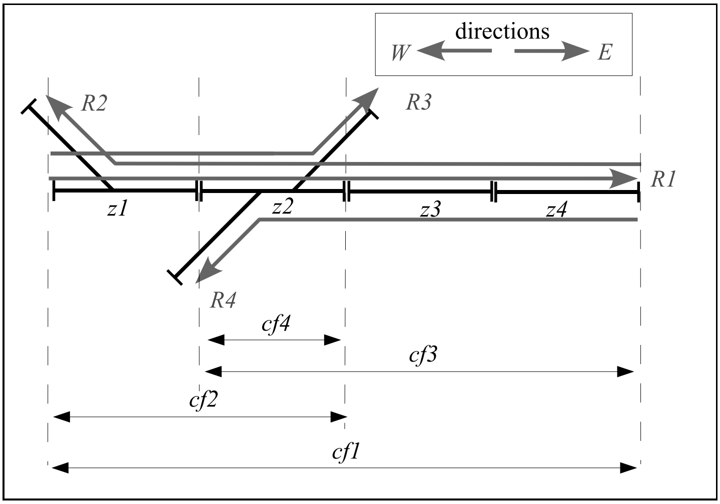

Figure 3: Example of 4 opposite conflict sequences.

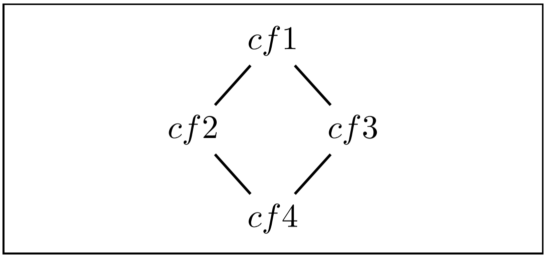

Figure 4: The Hasse diagram of conflict sequences.

To propagate the rank decision along the activities of an opposite direction conflict, we suggest using a state resource. A state resource is characterised as follows:

- It has an infinite capacity,
- The state can vary over time.
- Each activity may, during its execution, require it to be in a given state.
- Two activities may not overlap if they require incompatible states.

In the context of a railway traffic optimisation problem, a state resource models the allowed running direction during time along the conflict sequence. Therefore a state resource with 2 states (one for each direction) is associated with each opposite conflict sequence. The next section describes the constraints to post in order to obtain a correct propagation of rank decisions.

## 4 State resource constraints

The state resource constraints that can be posted depend on the resource assignment variables chosen for extending a partial solution during the search. Two cases have been considered: route assignment and track circuit assignment.

### 4.1 Route assignment

In this section we consider that the state resource constraints are posted after the routes are assigned to trains during a search algorithm. When a train is entering on the conflict sequence, the running direction will be constant from the start time of the first activities until the end time of the last activity. To satisfy this requirement, we define a constraint requiring that the state of the state resource has be constant during all the activities of the conflict sequence.

To formulate the constraint, let Ct$(t, cfi, state)$ be the constraint of the state resource of $cfi$ in state $state$ for a train $t$. This constraint is the conjunction of the state resource constraints of all activities of the conflict sequence i.e. :

$$\text{Ct}(t, cfi, state) = \bigwedge_{z_j \in cfi} activity(t, z_j).\text{requires}(cfi.resource, state)$$

To illustrate this constraint, let us consider the example of figure 3. The opposite direction of the routes $R1$ and $R2$ yields the conflict sequence $cf1 :< z1, z2, z3, z4 >$. If a train is running on route $R1$, the state resource associated to $cf1$ must be on the state $E$ during the running through all the track circuits of $cf1$. The state constraint of the route $R1$ is then :

$$\begin{aligned}\text{Ct}(t, cf1, E) \;=\;\; & activity(t, z_1).\text{requires}(cf1.resource, E) \wedge \\ & activity(t, z_2).\text{requires}(cf1.resource, E) \wedge \\ & activity(t, z_3).\text{requires}(cf1.resource, E) \wedge \\ & activity(t, z_4).\text{requires}(cf1.resource, E)\;.\end{aligned}$$

The state resource of the route $R2$ is similar, but with the state $W$.

The rank decisions to arbitrate a conflict between two trains $t_1$, $t_2$ requiring any track circuit of $cf1$ will be translated as one of the rank decisions :

- $activity(t_1, z_4) \prec activity(t_2, z_4)$,
- $activity(t_1, z_1) \prec activity(t_2, z_1)$.

This results is obtained by the constraints Ct$(t1, cf1, E)$ and Ct$(t2, cf1, W)$ which propagates the decision along the conflict sequence.

### 4.2 Track circuit assignment

Let now consider the case of extending a partial solution of a problem with track circuits assignments to trains. The question is then : what are the sufficient conditions to post a state resource constraint ? To answer the question, let us consider again the example of figure 3. There are 4 conflict sequences :

- $cf1 :< z1, z2, z3, z4 >$,

- $cf2 :< z1, z2 >$ ,
- $cf3 :< z2, z3, z4 >$ ,
- $cf4 :< z2 >$.

The conflict sequence set with the inclusion relation (symbol $\subseteq$) is a partially ordered set. This order can be graphically displayed as a Hasse diagram. Figure 4 is the Hasse diagram of the conflict sequences of the example of 3. In the Hasse diagram, each conflict sequence is represented as a vertex and a line is drawn upward from $cf_i$ to $cf_j$ if $cf_i \subseteq cf_j$, and there is no $cf_k$ such that $cf_i \subseteq cf_k \subseteq cf_j$, we say that $cf_j$ covers $cf_i$.

When a track circuit is assigned to a train, the following rules are used to post constraints:

**Rule 1:** If a the track circuit $z_i$ is assigned to a train then post the state resource constraint of *least conflict sequence* which includes $z_i$ (noted $lcs(z_i)$).

**Rule 2:** For all conflict sequence $cf_i$ covered by $lcs(z_i)$, post state resource constraint of $cf_i$ .

To illustrate Rule 1, let us consider the example of figure 3, and the case of the assignment of track circuit $z1$ to a train in the E direction. According to the rule, the state resource constraint requiring the conflict sequence $cf2$ in E direction is posted. Indeed, $cf2$ is the $lcs(z1)$, i.e. none of the conflict sequences which include $z1$ is a subset of $cf2$. Note that $cf1$ includes $z1$, but it is not the $lcs(z1)$ as $cf2$ is one of the subsets of $cf1$ (see the upward line of the Hasse diagram in figure 4).

The second rule concerns the conflict sequence covered by the least conflict sequence. We can take the as an example the figure 3 and the case of the assignment of track circuit $z1$. As the resource of $cf2$ is required by the train in direction E, to avoid inconsistencies, we must require the resource of the covered conflict sequence $cf4$ in the same direction and during the same interval.

So, after the assignment of $z1$ to the train, the state resources of $cf2$ and $cf4$ will be required in the state E, whatever the route assigned afterwards.

## 5 Results and conclusion

The extension of our CBS model presented here has been implemented and successfully validated on the infrastructure of the Pierrefitte-Gonesse junction. Table 1 shows some preliminary results on a set of growing size problem instances.

The column headings in the table have the following meanings:

| | |
|---|---|
| #t | : Number of trains of the problem instance, |
| CBS | : CBS model of section 2 , |
| CBS+SRC | : CBS model with the state resource constraints extension, |
| Crit | : Criteria of the best solution found, |
| CPU | : Computation time in CPU seconds to find the best solution, |
| BT | : Number of backtracks of the search algorithm. |

Table 1: Computational results.

| | | CBS | | | CBS+SRC | | |
|---|---|---|---|---|---|---|---|
| Inst. | #t | Crit | CPU[a] | BT | Crit | CPU [a] | BT |
| 1 | 6 | 64 | 1.63 | 206 | 64 | 1.91 | **138** |
| 2 | 8 | 64 | 1.75 | 631 | 64 | 2.42 | **322** |
| 3 | 8 | 66 | 2.4 | 721 | 66 | 3.1 | **304** |
| 4 | 10 | 161 | 33.42 | 8774 | 161 | **22.33** | **1172** |
| 5 | 12 | 161 | 68.66 | 9848 | 161 | **43.91** | **1286** |
| 6 | 14 | 313 | 339.6 | 27 323 | **258** | 832.12 | **13 646** |
| 7 | 16 | 350 | 704.93 | 24 789 | **258** | 1199.3 | **14 379** |
| 8 | 18 | 498 | 955.67 | 9055 | **461** | **646.02** | **7739** |
| 9 | 20 | 512 | 345.2 | 11 994 | **498** | 1585.14 | **7611** |
| 10 | 22 | 648 | 718.08 | 5084 | **609** | 1593.16 | **4135** |

[a] In seconds using a 2.66GHz Pentium IV processor, Ilog Solver 5.3 and Ilog Scheduler 5.2

The bold face values show the improvements of the state resource constraints extension of the CBS model.

For the problem instances used here, searching for an optimum value is not a realistic goal. In order to apply a resolution method and obtain *good solutions within a limited time*, an heuristic approach has been considered (see [11]). The two models have been tested with the same CPU limits.

The first observation is that the number of backtracks has been significantly reduced for all instances. As we also obtain a better or the same criteria solution, this shows that the redundant constraints have pruned many branches of the search tree. For example in the instance #5, the number of backtracks has been reduced from 9848 to 1286 and the criteria has been reduced from 68.66 to 43.91.

In what regards the criteria values, we remark that the improvements have been obtained with the large instances. This can been explained by the computational time needed to calculate the state resource constraints conditions (see section 4). This time increases very slowly with the instance sizes and therefore has more impact in small instances than in large instances.

As a conclusion, the state resource constraints show very promising preliminary results in what regards the resolution performances. This first attempt of using these kinds of constraint need further researches. Some improvement issues have been identified regarding :

- The reduction of the number of constraints needed,
- The combination of assignment and scheduling heuristics of the solution algorithm,
- The testing of the practical limits of some complete scheduling algorithms,

- An interesting future task is also to study in depth the performances of different models and resolution methods as initiated in [13], having in mind the goal of the potential integrations of different approaches.

## References

[1] Fukumori, K. & Sano, H., Fundamental algorithm for train scheduling based on artificial intelligence. *Systems and Computers in Japan*, **18(3)**, 1987.

[2] Gosselin, V., Train scheduling using constraint programming techniques. *Charme Technical Bulletin*, **(3)**, 1993.

[3] Poujade, M., Moyon, P. & J.F., P., Sagitaire: Allocation des quais de la gare Paris-Nord. *Conférence IA'95*, 1995. (in french).

[4] Chiu, C.K., Chou, C.M., Lee, J.H.M., Leung, H.F. & Leung, Y.W., A constraint-based interactive train rescheduling tool. *Second International Conference on Principles and Practice of Constraint Programming*, pp. 104–118, 1996.

[5] Hachemane, P., *Évaluation de la capacité de réseaux ferroviaires*. Thèse 1632, École Polytechnique Fédérale de Lausanne, 1997. (in french).

[6] Kreuger, P., Carlson, M., Olsson, J., Sj, T. & o Astr, Trips scheduling on single track networks - the tuff train scheduler. *CP'97 Workshop on Industrial Constraint - Directed Scheduling*, pp. 1–12, 1997.

[7] Oliveira, E.S., *Solving Single-Track Railway Scheduling Problem Using Constraint Programming*. Ph.D. thesis, School of Computing - University of Leeds, 2001.

[8] COMBINE2 project IST-2001-34705, Traffic regulation and co-operation methodologies. Deliverable D3, Univerita degli Studi Roma Tre, 2003.

[9] Barber, F., Salido, M.A., Abril, M., Ingolotti, L., Lova, A. & Tormos, P., An interactive train scheduling tool for solving and plotting running maps. *LNCS/LNAI*, **(3040)**, pp. 659–668, 2004.

[10] Rodriguez, J. & Kermad, L., Constraint programming for real-time train circulation management problem in railway nodes. *Computers in Railways VI*, WITPress: Lisbonne-Portugal, 1998.

[11] Rodriguez, J., A constraint programming model for real-time trains scheduling at junctions. *Transportation Research Part B : Methodological*, 2006. In press.

[12] Mascis, A. & Pacciarelli, D., Job-shop with blocking and no-wait constraints. *European Journal of Opera- tional Research*, **(143)**, pp. 498–517, 2002.

[13] Delorme, X., Rodriguez, J. & Gandibleux, X., Heuristics for railway infrastructure saturation. *Electronic Notes in Theoretical Computer Science 50 N1(2001) URL: http://www.elsevier.nl.locate/entcs/volume50.html*, pp. 41–55, 2001.