

An incremental decision algorithm for railway traffic optimisation in a complex station

J. Rodriguez

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

Abstract

In heavy traffic nodes of rail networks, as in main stations, conflicts and subsequent train delays greatly propagate during operations. A disturbance which is at origin only a few seconds long can quickly lead to other delays of over five minutes. To limit this phenomenon, new methods and models are necessary to optimise the use of scarce resources like platforms and track sections.

The operator controlling traffic has to select and evaluate alternative solutions that reduce delays caused by conflicts. This task can be formulated as an optimisation problem, where the decision variables are the selection of the alternative routes and sequences for trains and the criterion is the sum of delays.

This optimisation problem is a joint scheduling and allocation problems and is considered to be a NP-hard problem which makes it hard to solve using exact methods for a reasonable problem size.

In this paper, we compare two heuristic methods for solving the problem. The first one uses a two-phase approach to perform independently resource allocation and scheduling. The second one performs incrementally the two kinds of decisions at track section level, i.e. at each step, the algorithm performs decisions of allocation of a track section or scheduling a pair of train runs on this section. The algorithm uses a measure of the criticality of the track sections to heuristically choose between making resource allocation decisions or scheduling decisions.

Keywords: rescheduling, traffic control, heuristic methods, constraint programming.



1 Introduction

As with all public transport modes, planning is an important aspect of railway transport. To improve modal transfer from the road, rail transport must adapt its planning to various changing and/or conflicting requirements: satisfying demand, providing a public service, achieving profitability, safety, security, quality of service, reducing pollution, etc. The need to meet these requirements makes planning one of this sector's key activities.

As is the case for other sectors, railway planning is often broken down into three levels of decision-making : strategic planning, tactical planning and operational planning. The configuration of resources is principally located at the strategic level. At this level, decisions relate, for example, to the construction of new infrastructure, the hiring of staff or the purchase of rolling stock.

Then, at the tactical level, is the design of the transport plan which consists of distributing resources within time and space while satisfying as well as possible the requirements mentioned above. The design of timetables for lines is the central component of the transport plan.

Last, the operational level relates to the implementation of the transport plan. The plans which are envisaged during the tactical phase are never completely applied in the operational phase. Incidents that affect the availability of resources or the modification of demand mean that supply needs to be re-planned.

At railway station or junction level, incidents and modifications are both frequent and numerous. The most common incidents are arrival or departure delays, additional trains, the failure of certain signalling components, track maintenance works, etc. For passengers, they lead to delays with respect to the timetable. In situations where traffic is dense and at the limit of capacity, incidents may cause very large delays.

One way of limiting the amplification of delays in such situations is to increase capacity by providing additional switches, tracks, flyovers or platforms. However, the lack of available space or the costs involved mean that these solutions are not feasible in urban areas. Consequently, when traffic is dense at a junction or a station, it is imperative to optimise the way the infrastructure is shared.

At operational level, the routing and order of trains must be reconsidered in order to exploit all the possible alternatives as regards routes and train scheduling. The operator who manages traffic during peak hours must choose rapidly under pressure from a large number of alternatives. In order to facilitate this task, a tool which proposes the most advantageous solutions may be extremely helpful. For this purpose, we have developed a model of the train routing and scheduling optimisation problem for an item of railway infrastructure such as a junction or a station. This model is characterised by the use of two approaches: constraint programming and scheduling theory. Our first constraint-based scheduling (CBS) formulations of this problem was in [1]. The last updated and detailed description of the model was in [2]. An extension of the model to allow a better management of conflicts between trains running in opposite directions has been presented in [3].



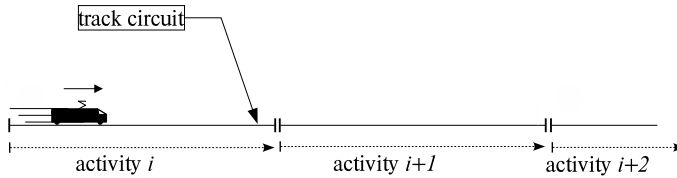


Figure 1: Train movement as a sequence of activities.

The published experiments with CBS model used a two-phase approaches for solving the problem, i.e. the first step is to solve the allocation problem, after that, we will search for a solution to the scheduling problem. In this paper, we present an experimental study of a new heuristic resolution method based on an *incremental approach*. The algorithm combines the local decisions of allocation track sections and scheduling train runs on these sections. The algorithm uses *texture measurements* to heuristically choose between making resource allocation decisions or scheduling decisions.

The paper is organised as follows: our CBS formulation of the railway traffic management is recalled in section 2. Section 3 presents the two phases and the incremental heuristic resolution methods we have compared. Section 4 gives the results obtained with problem instances from traffic of the Lille-Flandres railway station. Finally, section 5 gives some preliminary conclusions and issues.

2 A constraint-based approach to the rail traffic operational management problem

Like Spzigel [4], 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* [5, 6].

The constraints of our model will be outlined briefly (there is a more detailed formulation in [2]). 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.



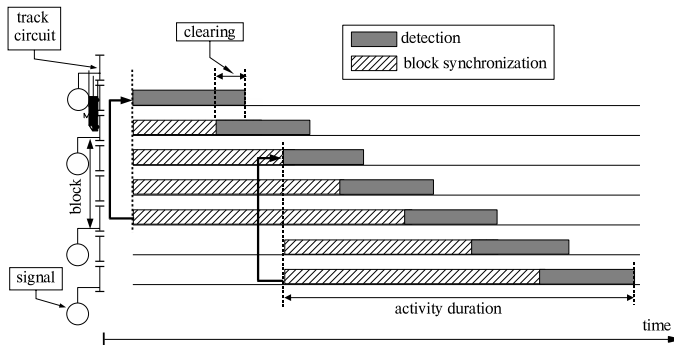


Figure 2: Gantt chart of activities for 3-aspect block signalling system.

- 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 resources of the 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 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).
- The extension of the model presented in [3] introduces redundant constraints to improve the resolution methods. These constraints relate to the opposite direction conflicts in track circuit sequences. For each opposite direction conflict, a two-state resource is specified. This state resource will make it possible to model the direction of train movements on this track circuit sequence. An activity is also defined which covers the sequence of activities of a train in the track circuit sequence. If a train has to run through an opposite direction conflict, the covering activity requires the resource to be in a state that corresponds to its direction of run. The covering constraint and the state constraint of the resource make it possible to propagate scheduling decisions on activities of other trains. The experiments reported in [3] show significant improvements of the resolution methods with the state resource constraints.

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 Heuristic methods

According to the classification of the conventional scheduling problems, the problem of the CBS model of section 2 is a joint scheduling and allocation problems i.e. there are degrees of freedom for deciding both which activities to perform and when, and which resources to allocate to these activities.

To deal with the combined resource allocation and scheduling problem, we consider two approaches : the first one is a *two-phase approach* which performs independently resource allocation and scheduling and the second one is an *incremental approach* where partial resource allocation and partial scheduling is repeated until a complete solution is generated.

This section gives a presentation of these heuristic methods that are compared in the experiments presented in next section.

Subsequently, subsection 3.1 gives the presentation of the two-phase heuristic method, and subsection 3.2 focuses on the incremental heuristic method.

3.1 Two phases algorithm

The idea behind this approach is to conduct search in two phases (see [2, 3] for more details) :

- In the first phase, we make all the train route decisions with a *complete labelling procedure* of the route variables of the trains for solving the allocation problem, i.e. all combination of routes are tried.
- In the second phase, we schedule trains in time with an *approximation algorithm* to solve the scheduling problem , it consists of a labelling all the start time variables of the activities; the labelling order is dynamic and is based on the minimum value of the domain of variables. This algorithm corresponds to the earliest release date scheduling rule.

If the scheduling part of the procedure determines infeasibility we may be able to backtrack to the first phase of the search to obtain a new route assignment. The main drawback of this approach is the limited ability of the allocation phase to foresee situations where the chosen routes can not improve the criteria, which may only be detected once many scheduling decisions have been made.

On the other hand, this approach is relatively simple to implement and allows to straightforwardly use different techniques for each phase of the search.

3.2 Incremental algorithm

The incremental algorithm is inspired from ([7, 8]) and is structured as follows :

1. Identifies a resource R with the critical time point (i.e. the maximum contention time point).
2. Identifies two activities, A and B, which require the resource R during the critical time point and that are unsequenced. Analyzes the consequences of each sequence possibility to determine the best one based on calculations of the temporal slack associated [9].



3. Identifies an activity C that has the highest contribution to the critical time point of the resource R and that still have alternative resources.
4. Choose between one of the two following choice points:
 - (a) Constrain the best sequence activity ($A < B$ or $B < A$) with the alternative being the opposite sequence.
 - (b) Constrain the activity C to not execute on the resource R with the alternative being that it must execute on this resource.

A key element of the algorithm is the calculation of the criticality time point as it imply the choice of the resource R and the activities A,B and C. The criticality measurement is done with texture measurement techniques [10]. These techniques are formed from metrics on the constraint graph representation that underlies a search state. In our case we use the contention for the resource as a texture measurement technique.

For an activity A requiring a resource R, an individual demand curve is calculated based in the amount of R, required by A. To estimate contention, the individual demands of each activity are aggregated for each resource. Aggregation is done by summing the individual activity curves for that resource. This aggregate demand curve is used as a measure of the contention for the resource over time (for more details see [7], [8]).

4 The computational results

In this section we presents the experiments to compare performances of the two heuristic resolution methods presented in previous section.

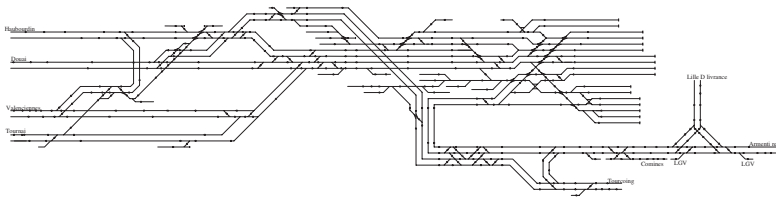


Figure 3: The layout of the Lille-Flandres station.

4.1 The infrastructures considered

To make the comparison, a real case study of a terminal station has been considered. It is the station of Lille-Flandres, the main station of Lille in the North of France (see Figure 3).

In the Lille-Flandres station, there are Regional trains, Inter City trains and TGV trains. The station is connected to 7 lines, trains can arrive at 17 platforms. Almost all tracks have the possibility of running in both directions. The running distance of the routes is more or less 4 kilometers within 6 minutes of running time.

Table 1: Two phases method versus incremental method.

Inst.	T	nb var.	nb ct.	BS ^a	TPH		INC		
					CPU ^b	BT	BS ^a	CPU ^b	BT
L1	4	2119	2918	0	2,05	4	0	1,1	1
L2	5	2598	3490	0	2,48	4	0	1,43	32
L3	6	2939	3822	0	2,61	15	0	1,57	37
L4	7	3247	4095	79	3,13	28	79	1,76	38
L5	8	3834	4774	79	3,5	43	79	2,43	64
L6	9	5131	6531	87	24,44	1683	87	6,4	159
L7	10	6840	8567	87	40,93	3182	87	4,75	114
L8	11	7573	9445	98	143,97	5320	98	6,44	132
L9	12	8051	9986	98	159,02	6158	98	16,08	466

^aIn seconds.

^bIn seconds using a 1.20 GHz Intel® Duo Core™ U7600 processor, Ilog Solver 5.3 and Ilog Scheduler 5.2.

4.2 Numerical instances

For this experimental study, a set of 9 instance problems were considered (Table 1, column 1-4).

The column 1-4 headings in the table have the following meanings:

Inst.: the instance number,

|T|: the number of trains,

nb var.: the number of variables,

nb ct.: the number of constraints.

The instance L1 has 4 trains, two trains in one direction and the others in the opposite direction. The instances L2 to L9 are generated from the instance L1 by successively adding one train. The trains added maintain the same proportion of trains in both directions and consequently increase the number of potential opposite direction conflicts. From these figures, we can see that almost all instances have as many variables as constraints. Regarding the number of state resources, the L[1-9] instances have between 46 and 245 state resources. Regarding the number of activities, the trains of L[1-9] instances have between 10 and 30 activities.

4.3 Evaluation of the heuristic resolution methods

For all the experiments, the limit of the computational effort was set to 180 seconds of CPU time (including setup and preprocessing of data).

Columns 5-10 of Table 1 compares the two heuristic methods, the column headings have the following meanings:



- TPH: results with the two-phase heuristic method (subsection 3.1),
 INC: results with the incremental heuristic method (subsection 3.2),
 BS: the best solution value found by the method within the time limit,
 BT : the number of backtracks,
 CPU: the CPU time needed to find the best solution (expressed in seconds),

For all the instances, the two methods find the same quality solutions. If we consider the time spent to find the best solution, the performance of the INC method is always better than the TPH method. The number of backtracks is another metric which confirms that the INC method is significantly better than the TPH method.

5 Conclusion

The model of the train routing and scheduling problem that has been presented is able to consider a large number of technical and commercial characteristics drawn from real situations. Trials with problems of increasing sizes have shown that good quality solutions can be obtained with processing times which are compatible with the operational constraints.

An incremental heuristic method has been compared with a two-phase heuristic method. The incremental heuristic shows very promising results in relation to the resolution performances from the points of view of the computational time. These are preliminary results which have to be confirmed with more experiments and with other instances.

Potential improvements to the resolution method relate to:

- evaluating other techniques used to break down the problem to make it possible to resolve large-sized problems,
- investigating the benefit of using search heuristics such as LDS (Limited Discrepancy Search [11]) or DDS (Depth-bounded Discrepancy Search [12]) which have produced good results for some constraints satisfaction problems.

As has already been stated by Törnquist [13], many models which use other approaches have also provided very interesting results for similar problems (e.g. [14]). The lack of publicly available data on problem instances means that it is impossible to compare the benefits and disadvantages of the different approaches. Comparisons of this type could result in hybrid models which include some Operations Research algorithms or graph theory algorithms in a constraint programming model. The published results on the use of hybrid approaches for scheduling problems in other spheres (see [15, 16]) show that this is a very promising direction for research.

Acknowledgements

The author wishes to thank the “Direction Régionale des infrastructures-Lille” of SNCF for supplying data used in the experiments.



This work has been partially supported by Conseil régional Nord-Pas de Calais and European Union research funds (FEDER).

References

- [1] Rodriguez, J. & Kermad, L., Constraint programming for real-time train circulation management problem in railway nodes. *Computers in Railways VI*, WITPress: Lisbonne-Portugal, 1998.
- [2] Rodriguez, J., A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B : Methodological*, **41**(1), pp. 231–245, 2007.
- [3] Rodriguez, J., A study of the use of state resources in a constraint-based model for routing and scheduling trains. *2nd International Seminar on Railway Operations Modelling and Analysis*, Hannover, Germany, 2007.
- [4] Spzigel, B., Optimal train scheduling on a single track railway. *OR'72*, ed. M.e. Ross, North Holland Publishing Co., pp. 343–352, 1973.
- [5] Mascis, A. & Pacciarelli, D., Job-shop with blocking and no-wait constraints. *European Journal of Operational Research*, (**143**), pp. 498–517, 2002.
- [6] D'Ariano, A., *Improving Real-Time Train Dispatching: Models, Algorithms and Applications*. Ph.D. thesis, Delft University of Technology, 2008. TRAIL Thesis Series no. T2008/6.
- [7] Beck, J.C., *Texture Measurements as a Basis for Heuristic Commitment Techniques in Constraint-Directed Scheduling*. Ph.D. thesis, University of Toronto, 1999.
- [8] Ilog, *Ilog Scheduler 5.2 Reference Manual*, 2002.
- [9] Smith, S.F. & Cheng, C.C., Slack-based heuristics for constraint satisfaction scheduling. *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pp. 139–144, 1993.
- [10] Fox, M.S., Sadeh, N., & Baykan, C., Constrained heuristic search. *Eleventh International Joint Conference on Artificial Intelligence*, p. 309316, 1989.
- [11] Harvey, W. & Ginsberg, M., Limited discrepancy search. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*; Vol. 1, ed. C.S. Mellish, Morgan Kaufmann, 1995: Montréal, Québec, Canada, pp. 607–615, 1995.
- [12] Walsh, T., Depth-bounded discrepancy search. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, pp. 1388–1393, 1997.
- [13] Törnquist, J., Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms. *ATMOS2005 (Algorithmic Methods and Models for Optimization of Railways)*, Dagstuhl Research Online Publication Server (DROPS): Palma de Mallorca, Spain, 2005.
- [14] D'Ariano, A. & Pranzo, M., An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *2nd International Seminar on Railway Operations Modelling and Analysis*, Hannover, Germany, 2007.



- [15] Milano, M. & Wallace, M., Integrating operations research in constraint programming. *4OR*, **4(3)**, pp. 175–219, 2006.
- [16] Hooker, J.N., Planning and scheduling by logic-based benders decomposition. *Operations Research*, **55(3)**, pp. 588–602, 2007.

