A CSP model and LNS approach for the railway saturation problem

F. Degoutin [1], H. Cambazard[2]

1. INRETS-ESTAS 20 rue Elisee Reclus, BP 317, F-59666 Villeneuve d'Ascq Cedex - FRANCE and LAMIH-ROI - University of Valenciennes,
degoutin@inrets.fr
2. LINA - Ecole des Mines de Nantes, 4 rue Alfred Kastler. B.P. 20722, F-44307 Nantes Cedex 03 - FRANCE
cambazard@emn.fr

Abstract: Optimization of railway infrastructures includes a lot of challenging combinatorial problems. We study the evaluation of the railway infrastructure capacity of a junction. This work is part of the RECIFE (research on the capacity of the railway infrastructure) [8] project and was first studied by Delorme [3]. We propose in this paper a CSP Model for the saturation problem and a local seach approach based on the metaheuristic Large Neighborhood Search. Both approaches are evaluated on random and real instances.

Keywords: railway planning ; constraint programming ; local search

1 Introduction

Railway traffic management covers different kinds of problems, three levels can be defined : strategic (long term), tactical (medium term) and operational (short term). This research concerns the planification and programmation occuring on the strategic level, more precisely the railway infrastructure capacity evaluation. This problem, called the railway saturation problem, is modelised in a CSP and solved using a metaheuristic. It provides the railway authority with help when they have to choose between different infrastructure investment projects.
The aim of this paper is to describe the improved CSP model for this problem and to present the metaheuristic used to solve the complete problem.

2 The saturation problem

Management of railway lines is increasingly becoming an important issue for transport systems. Thus, the evaluation of the railway infrastructure capacity can help to choose the best modification. The definition of this problem [4] is :

Definition 1 (Saturation Problem) *The saturation problem consists of introducing the maximum number of trains among a predefined train set (which can be empty) that can be operated on the junction. The additional trains represent the absolute capacity margin of the infrastructure for the predefined train set.*

In order to follow the movement of a train within the infrastructure, this one is divided into zones of detection, where Z denotes the set of detection zones. R corresponds to the set of routes associated to the type of trains. The order in which trains are scheduled in a sequence of common zones must respect security rules which allow to obtain the minimum gap between two trains. The sequence of common zones between two routes is called a "conflict" :

Definition 2 *A running conflict is a quadruplet* $(r_1, r_2, z_1, z_2) \in R \times R \times Z \times Z$ *such that the set of zones between* z_1 *and* z_2 *of the route* r_1 *is equal to the set of zones between* z_1 *and* z_2 *of the route* r_2*.*

The schedule of trains, which takes routes r_1 and r_2, is carried out by a conflict arbitration :

Definition 3 *A conflict arbitration* (r_1, r_2, z_1, z_2) *is the choice of a couple of scheduled trains which take the routes* r_1*,* r_2 *on the sequence of common zones between* z_1 *et* z_2*.*

For this study, we consider the case of an empty time-table (trains can have any routes and starting time) for the Pierrefitte-Gonesse node with the 3 main categories of trains (TGV, Freight and Inter City) and 3 random instances with similar data to the real.

3 The formulation of the model and resolution

3.1 Model

The adopted formulation is the one of a constraint satisfaction problem (CSP). Two CSP formulation have been proposed by Rodriguez : [6] and [5]. The model described here is an extension of [6].
As presented in section 2, three types of variables are needed for this problem's modelling :

1. variable r_i, this variable will be called "route variable",
2. "variables incompatibility bounds" $\underline{t}_{ij}^k, \bar{t}_{ij}^k$ with $dom(\underline{t}_{ij}^k) \in \underline{I}, dom(\bar{t}_{ij}^k) \in \bar{I}$ stand for the values of the k^{th} incompatibility interval bounds between the trains i and j. To simplify the model, only the first interval of incompatibility is considered in the rest. Other intervals are integrated on the fly during resolution .
3. variable "start time" st_i represents the train entrance time in the node. The discretization step is of one second, so $dom(st_i) \subset \mathbb{N}^+$.
4. variable "gap" δ_{ij} represents the gap between the starting time of st_i and st_j.

In order to take conflicts into account, two constraints types are necessary : The first links the variables routes of the couple of trains and their incompatibility bounds variables. This constraint enables the enumeration of the acceptable tuples between variables $r_i, r_j, \underline{t}_{ij}, \bar{t}_{ij}$:

$$enum(r_i, r_j, \underline{t}_{ij}, \bar{t}_{ij}), \forall i, j \in T^2. \quad (1)$$

The second one models the incompatibility constraints between every couple of trains. It allows to settle of the conflicts between trains :

$$\delta_{ij} \notin]\underline{t}_{ij}, \bar{t_{ij}}[\quad (2)$$

with $st_i - st_j = \delta_{ij}, \forall i, j \in T^2$. This constraint works as follow :

- if $inf(\bar{t_{ij}}) > sup(\delta_{ij})$, $inf(\underline{t}_{ij}) = inf(\delta_{ij})$ and $sup(\delta_{ij}) = sup(\underline{t}_{ij})$.
- if $sup(\underline{t}_{ij}) < inf(\delta_{ij})$, $sup(\bar{t_{ij}}) = sup(\delta_{ij})$ and $inf(\delta ij) = inf(\bar{t_{ij}})$.

The use of this constraint allows the propagation to occur from t_{ij} variables, i.e incompatibility bounds towards the route variables on which the enumeration is performed, which would not be the case with a boolean constraint modelling such as our previous model.
The searching space described by those constraints presents symmetries. Hence, the constraint of trains succession is added : $st_i \leq st_{i+1}, \forall i \in T, i < N - 1$.
The proposed criterion is the minimisation of the infrastructure occupation time, comparable to the minimization of the *makespan* of the scheduling problem [7]. This is modelised by, in our case :min(st_N), N the last train.
Some improvements have been made to the model. They are presented in [2], to summarize those are : remove constraints between trains which can not be in conflict due to their far away, add cuts with the help of calculations of intermediate makespan, add an upper bound obtained on a sub-sequence of identical trains.

A new improvement have been tested, the idea is as follow : Here, further symetries are removed, for a solution with a paire of trains wich have differents routes and same start time, a symetrical solution is obtained with routes switched. As an example on 2 trains assigned, the solution : $r_0 = 0, r_1 = 5, st_0 = 0, st_1 = 0$ is equivalent to $r_0 = 5, r_1 = 0, st_0 = 0, st_1 = 0$. Symetries are broken through the use of nogoods gathered within a single global constraint that forbids all the permutation of values associated to the trains starting at the same time.

3.2 Resolution

An exact resolution has been implemented. It uses the algorithm implemented by ILOG SOLVER. This algorithm uses propagation and a search in depth-first. A specific branching heuristic is used trying first to assign better routes to train. The variables assigned first are the routes as explained in [1], then starting time variables. It allows the exact resolution on a few number of trains, until 14 in a suitable time.
To solve such a problem on a realistic number of train and resolution time (100 trains in an hour and less than 30 minutes of computation time) a metaheuristic must be used. Our knowledge of the problem is that the obtained solutions have often identical sub-sequences of routes. So the metaheuristic Large Neighborhood Search (LNS), proposed par Shaw [9] seems to be suitable. LNS works as follows : First choose an initial solution for the local search. Then define an operator for the local search (which variables have to be relaxed). And finally, explore the neighborhood, in our case by a Branch & Bound algorithm applied on the relaxed variables.
Such a metaheuristic use parameters, which are, in our case :

- initial solution : the upper bound described in section 2 or a bad solution composed of trains taking always the same routes.
- neighborhood : the number of relaxed variables are from 1 to 9, all variables are relaxed at least one time with a recoverage of 40%.
- type of neighborhood : variables are relaxed in chronological order or randomly but always on following trains. Indeed following trains have more conflicts together than trains far from each other.

4 Results

type of resolution	TGV_Freight		all routes		randoms instances	
	# nodes	time	# nodes	time	# nodes	time
incompability/previous	0.69	0.85	0.64	0.75	0.53	0.61
symetries/previous	0.98	1.4	0.93	1.06	0.999	1.2
complet/previous	0.69	0.86	0.63	0.82	0.53	1

Table 1: Ratio between improvements proposed and the previous model

The table 1 shows the impact of the two new constraints implemented, the incompatibility one and the one which allow to break symetries. *previous* refers to the resolution without the incompatibility constraint and the breaking of symetries. As we can see the incompatibility constraints is the most efficient with a decrease of 27% on time consumption and about 38% in the number of node explored. The constraint which allows to break symetries is efficient to reduce the number of node, but take too much time, so it is not used for the LNS resolution.

The table 2 presents LNS ratio between the model with the incompatibility constraint and without, then the ratio between LNS using random removing of sequence of variables or chronological order and the last one presents the ratio between LNS with a good initial solution and a bad one. As we can see the incompatibility constraint allow a reduction of the resolution time. The removal of random or chronologicaly ordered sequences of variables doesn't affect the solving process. However, the initial solution, the one obtained by our upper bound, has a big

type of resolution	TGV_Freight		all routes		randoms instances	
	makespan	time	makespan	time	makespan	time
incompatibility/simple	1	0.79	1	0.96	1	0.96
random/fixed	1	1	1	0.83	0.98	1.1
good/bad initial solution	0.91	0.11	0.93	0.27	1.03	1.3

Table 2: Ratio between different LNS combination

impact, specially on real instances. More than 100 trains can be scheduled in less than an hour and fastly (less than 20 minutes in 80% of the test). The resolution time and size requirements are filled with such an approach, so LNS seems a good practical way to tackle our problem.

5 Conclusion

The railway infrastructure saturation problem has been exposed and modelised as a constraint satisfaction problem. The results obtained on the pure CP approach by improving propagation, and trying to break more symetries have allowed to improve the previous model but are still not competitive on realistic size instances of around hundred trains. However the Large Neighborhood Search metaheuristic, applied on this model, is able to solve the problem very efficiently and to provide better solution than the ones obtained previously by Delorme [3].

References

[1] F. Degoutin and J. Rodriguez. Evaluation of depth-bounded discrepancy on a CSP model of the railway saturation problem. In *17th IMACS World Congress Scientific Computation, Applied Mathematics and Simulation*, Paris, France, july 2005.

[2] F. Degoutin, J. Rodriguez, and X. Gandibleux. Évaluation des performances d'un modèle CSP pour le problème de saturation d'infrastructures ferroviaires. In J.C. Billaut and C. Esswen, editors, *ROADEF'05,* 6^{ieme} *congrès de société franaise de recherche opérationnelle et d'aide à la décision*, volume 1, pages 277–294. Presse universitaire François Rabelais, Février 2005.

[3] X. Delorme. *Modélisation et résolution de problèmes liés à l'exploitation d'infrastructures ferroviaires.* Thèse, Université de Valenciennes et du Hainaut-Cambresis, Valenciennes, 2003.

[4] P. Hachemane. *Évaluation de la capacité de réseau ferroviaire.* Thèse, École Polytechnique Fédérale de Lausanne, Lausanne, Suisse, 1997.

[5] J. Rodriguez, X. Delorme and X. Gandibleux. Railway infrastructure saturation using constraint programming approach. In J. Allan, R.J. Hill, C. A. Brebbia, G. Sciutto and S. Sone, editors, *Computers in Railway VIII, proceedings of the Eighth International Conference on Computer Aided Design, Manufacture and Operation in the Railway and Other Advanced Mass Transit Systems (COMPRAIL 2002)*, volume 13 of *Advances in Transport*, pages 807–816. Wit Press, 2002.

[6] J. Rodriguez. Analyse des conflits de circulations ferroviaires. Rapport technique RR-02-734-FR, INRETS-ESTAS, 2002.

[7] P. Esquirol, P. Lopez, and M.J. Huguet. *Ordonnancement de la production*, chapitre Propagation de contraintes (5), pages 131–167. Hermes Science Publications, 2001.

[8] J. Rodriguez. Projet RECIFE : Analyse de la capacité ferroviaire. Technical report, INRETS, 2001.

[9] P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. *Lecture Notes in Computer Science*, 1520:417–431, 1998.