Enhancement of train routing and scheduling at a terminal station with constraints programming and temporal decomposition

Joaquin Rodriguez

INSTITUT NATIONAL DE RECHERCHE SUR LES TRANSPORTS ET LEUR SÉCURITÉ

Joaquin Rodriguez, (INRETS-ESTAS)

ENHANCEMENT OF TRAIN ROUTING AND SCHEDULING AT A TERMINAL STATION WITH CONSTRAINTS PROGRAMMING AND TEMPORAL DECOMPOSITION

Décembre, 2007
Programme TAT - Projet T32 "Optimisation - Recife"
Arrêté du Conseil Régional N° 06 23 0009
INRETS/INRETS/RR - 07 - 731 - FR

Nom du projet : RECIFE (REcherche sur la Capacité des Infrastructures FErroviaires)

Partenaires :
- INRETS, Unité de Recherche Évaluation des Systèmes de Transport Automatisés et de leur Sécurité,
- UVHC, LAMIH - UMR CNRS 8530 - Équipe Recherche Opérationnelle et Informatique
- SNCF, Délégation Régionale Infrastructure de Lille, Études prospectives Infrastructures, Pôle exploitation.

Intervenant :
- J. Rodriguez (Directeur de Recherches, INRETS-ESTAS)

Type	Programme TAT - Projet T32 "Optimisation - Recife"
Convention	N° 02 31 0045
Décision	Arrêté du Conseil Régional N° 06 23 0009
Ref.	INRETS/RR - 07 - 731 - FR

Table des matières

Enhancement of train routing and scheduling at a terminal station with constraints programming and temporal decomposition

Joaquín Rodriguez
INRETS (Institut National de Recherche sur les Transports et leur Sécurité),
20, rue Élisée Reclus, F-59666 Villeneuve d'Ascq Cedex, France

Résumé

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 criteria.

This paper deals with an extension of a constraint-based scheduling model of real time management of train traffic through junctions and large stations. This extension allows the improvement of the model of conflicts between trains running in opposite directions with the use of state resources. To tackle large instances, a temporal decomposition has been used. We tested the model and algorithms by applying to problem instances of a real terminal station. Computational results show that with the temporal decomposition method we are able to obtain good solution in short computation time.

keywords : real time traffic management, train routing and scheduling, temporal decomposition, opposite direction conflicts

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 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.

Concerning traffic control at a station, 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.

Task (iv) can be formulated as an optimisation problem. The problem consists in defining a schedule in real time for all the trains circulating in the station. For each train it is necessary to define a routing in the station area and the sequence of use of track segments and platforms. In case of disturbances, the objective is to minimise the delays.

In a terminal station, the inbound and outbound trains can run in a common segment of the track in opposite directions. This leads to opposite direction conflicts which are difficult to manage as they consume unequal capacity in comparison with the other types of conflicts. As mentioned by Carey and Carville [5] busy stations may be the most complex part of the network to schedule.

In this context, a computer-based assistance can be used to improve the quality of the final solution. The 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 addressed many times. For an overview of the proposed approaches, reader should refer to the articles of Bussieck at al. [4], Cordeau at al. [8] and Törnquist [28].

Most of published results deals with routing and scheduling trains at strategic and tactic decision level. and deal with simplified models, in which stations have often unlimited capacity. But there is a limited amount of published works on routing and scheduling trains at busy stations.

Zwaneveld et al. [31] consider the routing of trains through railway stations, given the detailed layout of the stations and a tentative timetable. They formulate the problem as a weighted node packing problem and develop an algorithm for solving this routing problem to optimality. Following this work, Lusby [19] presents a novel formulation as a set packing problem but without computational results. Velásquez and al. [29] use also a set packing problem formulation of the tactical and operational problem of routing and scheduling trains. Good results are obtained for a test example with 8 trains running through a junction

Carey and Carville [5] develop scheduling heuristics analogous to those successfully adopted by train planners. They considers one train at a time and defines for it an arrival time, a departure time and a platform. If there is a conflict in the schedule, arrival or departure times for some train are increased until there is no longer a conflict.

Other published works are Dariano at al. [9] and Flamini et al. [10]. They both used the alternative graph formulation of Mascis and Pacciarelli [21], which allows modelling job shop scheduling problems with blocking constraints. In [9], the model is used to forecast and minimise delay propagation, the time horizon of the problem is decomposed into tractable intervals to be solved in cascade. Computational tests are carried out on the dispatching Utrecht Den Bosch area. In [10], the train scheduling problem of a metro rail terminus is modelled and solved as a bicriteria job shop scheduling problem.

Like Spzigel [27], we also proposed in [26] a job shop modelling of the traffic management problem at junction. The model is formulated with a constraint programming approach. The last updated and detailed formulation of the model is in [25]. In this paper, we present an extension of the model of [25] by using *state resource constraints*. These kinds of constraint are redundant constraints. The state resource constraints allow a better management of opposite direction conflicts occurring in a terminal station.

The paper is organised as follows : the CBS formulation of the railway traffic management is introduced in section 2. Within the framework of this model, we define in section 3 the management of the opposite direction conflicts and the state resources constraints associated. Section 4 presents a global and a temporal decomposition solution method. Section 5 gives the results obtained with problem instances from traffic of the Lille-Flandres railway station.

2 Constraint programming model

Like Spzigel [27], 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* [7, 21].

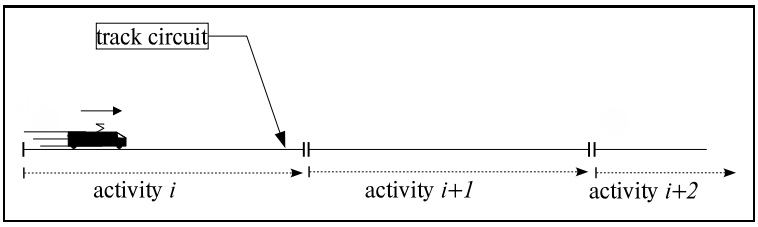

FIG. 1 – Train movement as a sequence of activities

The constraints of our model will be outlined briefly (there is a more detailed formulation in [25]). 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 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).

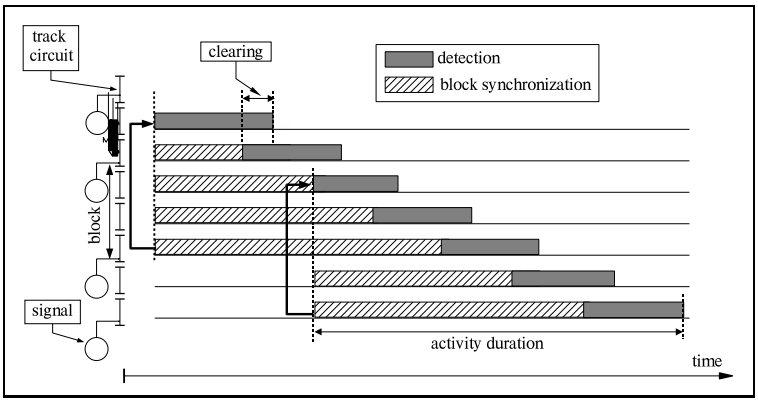

FIG. 2 – Gantt chart of activities for 3-aspect block signalling system

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 Management of opposite direction conflicts

In this section we introduce the problem of managing the opposite direction conflicts. Then, we present the use of a state resource to model the use of one circulation direction in a sequence of track circuits.

When two trains need to run through a common track, let us define as ”conflict sequence” the common sequence of track circuits requested by the trains.

Figure 3 illustrates an opposite direction conflict. The conflict is due to the direction E of the route $R1$ of $train1$ and the direction W of the route $R2$ of $train2$. The opposite directions of the routes yield the conflict sequence $cf1 :< z1, z2, z3, z4 >$.

The search algorithm for a feasible schedule with the CBS model is based on rank decision of activities. For the case of a conflict sequence between routes running in the

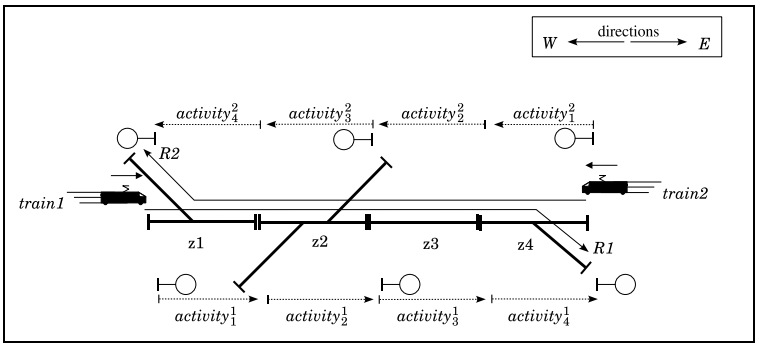

FIG. 3 – An opposite direction conflict

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 of the model (see section 2).

However, in the case of an opposite direction conflict, there is *no propagation of the rank decisions* ; all the activities of the conflict sequence have to be ranked. When there are opposite direction conflicts, the search algorithm will slow down as it needs more backtracks to find feasible ranks.

To propagate the rank decisions along the activities of the opposite direction conflicts, we suggest extending the CBS model by using state resources.

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 to each conflict sequence. In the example of figure 3, a state resource with states E and W is associated to the conflict sequence $cf1$.

The management of opposite conflicts with state resources is formulated with three constraints. The first constraint is a *meta-constraint*. It is a logical relation between a route assignment and the state resource constraint detailed afterwards.

To formulate this constraint, let $r(t)$ be the variable of the route of a train t, let R^i be a route running through a conflict sequence cfi. SRCt$(t, cfi, state)$ denotes the constraint for a train t requesting the state resource of cfi in state $state$. The meta-constraint to post a state resource constraint is :

$$(r(t) = R^i) \Rightarrow \text{SRCt}(t, cfi, state). \tag{1}$$

The second constraint is a *covering constraint* between the set of the activities of the conflict sequence and a dummy activity associated to the conflict sequence. Let

$activity(t, z_i)$ be the activity of the elementary movement of train t through a track circuit z_i and $activity(t, cfi)$ be the dummy activity associated to the conflict sequence cfi. The covering constraint is :

$$activity(t, cfi) = \text{covers}(\bigcup_{z_i \in cfi} activity(t, z_i)). \quad (2)$$

The third constraint links the covering activity with the state resource :

$$\text{SRCt}(t, cfi, state) = activity(t, cfi).\text{requires}(cfi.resource, state). \quad (3)$$

The constraints (1) (2) and (3) mean that when the train t is entering on the conflict sequence cfi, the running direction of cfi associated to $state$ will be constant from the start time of the first activity until the end time of the last activity.

To illustrate these constraints, let us consider the example of figure 3. If a train is running on route $R1$ (resp. $R2$), the state resource associated to $cf1$ must be on the state E (resp. W) during the running through all the track circuits of $cf1$. The constraints for the state resource constraints of trains $train1$ and $train2$ are :

$(r(train1) = R1) \Rightarrow \text{SRCt}(train1, cf1, E)$,
$(r(train2) = R2) \Rightarrow \text{SRCt}(train2, cf1, W)$,
$activity(train1, cf1) = \text{covers}(activity_1^1, activity_2^1, activity_3^1, activity_4^1)$,
$activity(train2, cf1) = \text{covers}(activity_1^2, activity_2^2, activity_3^2, activity_4^2,)$,
$\text{SRCt}(train1, cf1, E) = activity(train1, cf1).\text{requires}(cf1.resource, E)$,
$\text{SRCt}(train2, cf1, W) = activity(train2, cf1).\text{requires}(cf1.resource, W)$.

The rank decisions to arbitrate the conflicts between trains $train1$, $train2$ requiring the track circuits of $cf1$ will be translated as one of the rank decisions $activity_4^2 \prec activity_1^1$ or $activity_4^1 \prec activity_1^2$. This result is obtained by the constraints $\text{SRCt}(train1, cf1, E)$ and $\text{SRCt}(train2, cf1, W)$ which propagates any decision along the conflict sequence.

4 Solution methods

Subsequently, section 4.1 gives a brief presentation of the solution methods for solving a constraint programming model, section 4.2 focuses on a complete method to solve our CBS model and section 4.3 present the temporal decomposition heuristic.

4.1 Solution method for a CP model

In a CP model, a *domain* is the set of values a variable can take. *Constraints* represent the combinations of values authorised for a problem. To solve a problem, the method is roughly a search tree where two steps are performed at each node :

- A *labelling* procedure applies to variables,
- A *consistency* procedure applies to constraints.

During the labelling procedure, a value of the domain variable is selected. The *current domain* of that variable is reduced to the selected value. This domain reduction triggers other domain reductions through the consistency procedure. When the domain of a variable is empty, there is no solution in the sub-tree. In the latter case, a new value from a previous labelling variable is tried. A feasible solution is found when all variables have been labelled.

A consistency (or constraint propagation) procedure removes values which are inconsistent with the constraints. Several types of consistency have been defined, for example arc-consistency [20], path-consistency [22], k-consistency [11] and arc-B-consistency [18].

In the case of a CSP with an objective function, the resolution method uses a *Branch and Bound* algorithm. For a problem which involves minimising a criterion c, each time a solution is found with a value c^* for the criterion, the constraint $c < c^*$ is added to the model for all the other nodes in the search tree.

The performance of a CSP resolution method depends on three basic parameters : the variable and value selection order of the instantiation procedure, the consistency procedure(s) and the backtracking technique.

4.2 Solution method for the CBS model

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.

The first step of the solution method is to solve the allocation problem, after that, we search for a solution to the scheduling problem.

For the experiments presented in this paper, the allocation problem is solved with a *complete labelling procedure* of the route variables of the trains. A static order based on the domain size has been used.

The scheduling problem has been resolved by an *exact method* taken from the Ilog Scheduler library [15, 2]. The algorithm uses the following steps :

1 Select a resource among the resources required by unordered activities.

2 Select the activity to execute first among the unordered activities that require the chosen resource. Post the corresponding precedence constraints. Keep the other activities as alternatives to be tried upon backtracking.

3 Iterate step 2 until all the activities that require the chosen resource have been put in order.

4 Iterate steps 1 to 3 until all the activities that require a common resource have been put in order.

In step 1, the resource selected is the one whose ”slack” is minimal. The slack is the difference between the availability (or ”supply”) of a resource and the demand for the resource over a specific period of time ([15]). The slack is equivalent to a measure of the criticality of a resource. In step 2, the activity to schedule first is the activity with the minimal earliest start time and, in case of ties, the activity with the minimal latest start time.

In the first experiments to use this algorithm, we obtained weak time performances. The algorithm could not find a feasible solution of many instances within the time limit. Like in [1], to improve the algorithm, we start with a *greedy algorithm* to find in a short time a feasible solution. The greedy algorithm used apply a well known dispatching rule : First Come First Served (FCFS). The upper bound of the criterion is set with the solution of the greedy algorithm and we proceed with the scheduling algorithm.

With regard to consistency procedures, in addition to the arc-consistency, three mechanisms are used to propagate the resource utilisation constraints and adjust the time limits of activities [17] :

- The first mechanism relies on an explicit timetable of the variation of resource utilisation and resource availability over time.
- The second mechanism is the "disjunctive constraint propagation" ; it consists in maintaining arc-B-consistency [18] on temporal constraints.
- The third mechanism known as "edge finding" [6], considers an arbitrary set of activities $\Omega \cup A_i$ requiring the same resource and determines whether A_i must take place before or after all activities in Ω.

These three propagation mechanisms make it possible to prune many non-feasible decisions by adjusting the time limits of activities and thus improve the efficiency of the search.

Backtracking signifies the choice of a new node to explore. This choice is made after finding a feasible solution or proving that there is no solution in the sub-tree of the current node. For all results reported here, we have only used the *chronological backtracking* : the new node to explore is the last variable labelled ; if all domain values of this variable have been tested, backtracking moves to the previous variable.

4.3 Temporal decomposition

Applying the previous solution method to scenarios of heavy traffic of a large station can result into computationally intractable instances. To overcome the difficulty we have used a decomposition approach.

Many authors [23, 24, 3] have studied the benefits of temporal decomposition heuristics. The methods allow to quickly obtain solutions for large-scale scheduling problems. In these methods, only a small subsection of the horizon is dealt with at a time, thus reducing the combinatorial complexity of the problem.

The objective of the temporal decomposition process is to partition the given problem's time window based on an analysis of resource contention.

We assume that the number of trains is a good function to evaluate the resource contention during an interval. Therefore, the number of trains is used to select the best partition. For each subinterval, a subproblem is created to cover the initial time window.

Let $\mathcal{P}$, be the initial problem associated to an ordered set of trains $T =< t_1, \ldots, t_N >$. The set T is sorted according to the earliest start time of the first activity of each train. Let I_d be the number of subintervals for the initial time window. $\mathcal{P}_i$ denotes the subproblem of $\mathcal{P}$ associated to the subinterval i. The sets of trains T_i are disjoint subsets of T such as :

$$T_i =< t_{1+(i-1)\times\lfloor\frac{N}{I_d}\rfloor}, \ldots, t_{i\times\lfloor\frac{N}{I_d}\rfloor} >, \quad i = 1, \ldots, I_d - 1$$

$$T_{I_d} =< t_{1+(I_d-1)\times\lfloor\frac{N}{T_d}\rfloor}, \ldots, t_N >$$

With the temporal decomposition heuristic, the solution to the initial problem is composed by the subproblem solutions but global feasibility has to be guaranteed. In our model, the capacity and the state constraints of the resources are global constraints, e.g. they apply to the resources used by all trains. These global constraints ensure the coordination in order to avoid infeasibilities of the constructed solution. Other coordination constraints are also posted in order to consider relations between trains (use of same rolling stock, connections, ...) for different subproblems.

The algorithm is applied to each subproblem, a stopping condition limits the CPU time spent for solving each subproblem.

5 The computational results

In this section we presents the experiments to evaluate the resource constraints extension of the CBS model. Then, the temporal decomposition heuristic is compared to the global resolution procedure of the problem.

5.1 The infrastructures considered

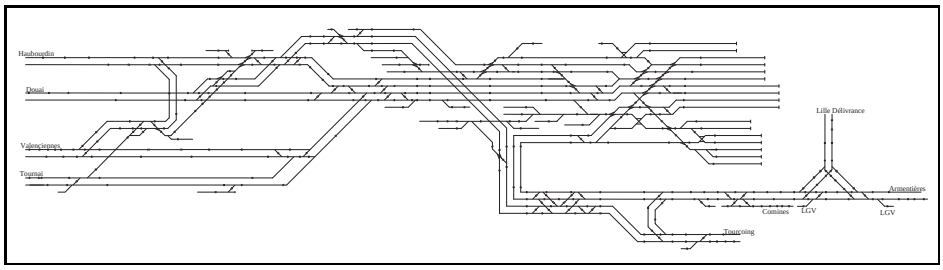

FIG. 4 – The layout of the Lille-Flandres station

To evaluate the impact of modelling the opposite direction conflicts, 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 4).

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 kilometres within 6 minutes of running time.

5.2 Numerical instances

For this experimental study, a sets of 10 problem instances were considered (Table 1).

		CBS		CBS + SRCt	
Inst.	$\|T\|$	# variables	# constraints	# variables	# constraints
1	6	1950	2247	2939	3822
2	8	2548	2748	3834	4774
3	10	3430	3527	6840	8567
4	12	4040	4032	8051	9986
5	14	4427	4685	7850	9753
6	16	5284	5376	9952	12257
7	18	5895	5875	11141	13487
8	20	6499	6392	12525	15218
9	22	7233	7095	13933	17004
10	24	7594	7430	14671	17907

TAB. 1 – Instances characteristics

The instances are generated by selecting train movements from a real timetable. To get difficult instances, the start time and the initial routes of trains are set in order to increase the number of potential conflicts.

In table 1, the columns headed “CBS” refer to the figures of the constraints based scheduling model of section 2. The columns headed “CBS+SRCt” refers to the figures of the CBS model extended with the state resource constraints described in section 3.

These figures show that there are almost as many variables as constraints. The addition of the state resource constraints increases significantly the number of variables and constraints and therefore the time needed to set up the model. The increase in variables (resp. constraints) for the 10 instances is between 33% and 48% (resp. 41% and 58%).

The number of state resources is between 64 and 411 and the number of activities is between 10 and 30.

5.3 Evaluation of the state resource extension

For all the experiments, a limit of 180 seconds of CPU time has been set.

Table 2 compares the CSB model (section 2) and the extension with the state resource constraints (section 3). The results were obtained with the resolution method of section 4.2. The column headings in the table have the following meanings :

Inst. : the instance number,
$|T|$: the number of trains,
GS : the greedy solution value found,
BS : the best solution value found by the method within the time limit,
GAP % : the percentage improvement over the greedy solution,
CPU : the CPU time needed to find the best solution (expressed in seconds),

Columns 4-6 and 7-10 show the results with the CBS model and its extension, respectively.

With the CBS model, only one optimal solution has been found within the time limit. For the instances #6,#9,#10 no better solution than the greedy one has been found. The rate of improvement between the greedy solution and the best solution found is 32 %.

			CBS			CBS + SRCt		
Inst.	$\|T\|$	GS[a]	BS[a]	% GAP[b]	CPU[c]	BS[a]	% GAP[b]	CPU[c]
1	6	242	0*	100	1,11	0*	100	1,22
2	8	975	365	62.56	1,58	79 *	91.9	1,79
3	10	1434	928	35.29	1,61	87 *	93.93	39,22
4	12	1806	1064	41.09	124,48	98 *	94.57	146,96
5	14	1468	1119	23.77	39,87	302	79.43	64,19
6	16	1304	1304	0	1,64	635	51.3	88,03
7	18	1880	1304	30.64	2,65	955	49.2	30,02
8	20	2252	1676	25.58	2,67	966	57.1	85,27
9	22	2294	2294	0	2,1	1252	45.42	147,33
10	24	2294	2294	0	2,3	1294	43.59	38,24

* Optimal solution
[a] In seconds
[b] Relative gap of delays in %,i.e., 100×(GS - BS)/GS
[c] In seconds using a 1.66 GHz Intel® Duo CoreTM T2300 processor, Ilog Solver 5.3 and Ilog Scheduler 5.2

TAB. 2 – CBS formulation versus the state resource constraints extension

With the extended CBS+SRCt model 4 optimal solutions have been found. For all the instances, we obtain a good improvement of the solution found after the greedy solution. The propagation of the state resource constraints allows a significant reduction of the search space. This enables the improvement of the quality of the solution within the time limit. We can notice that the instances #1 to #5 have better rate of improvement than the instances #6 to #10. The rate of improvement with the extended CBS+SRCt model for all instances is 70 %, which is twice as efficient as the CBS model. From this set of experiments, we can conclude that the state resource constraints enhances the previous CBS model.

These optimisation models have been integrated within various tools for processing and displaying the results. For example, the figure shows the display tool of a Gantt chart with the best solution for instance #10. This type of tool permits detailed analysis of the solutions and allows us to identify the critical resources for a journey.

5.4 Temporal decomposition method

Table 3 compare the results obtained with a global resolution method and the temporal decomposition method with the parameter $I_d = 2$.

Columns 4-6 and 7-10 show the results with the global and temporal decomposition, respectively.

From the results of instances #1, #2, #3, #4, we can notice actually that the temporal decomposition approach gives sub-optimal solutions.

On the other hand, temporal decomposition gives better results in terms of delay minimisation for the larger instances (#6 to #10). The rate of improvement over the greedy solution for these instances is 82 %. With the global approach the rate of improvement for these instances is only 49 %.

From this set of experiments, we can conclude that the global approach gives better

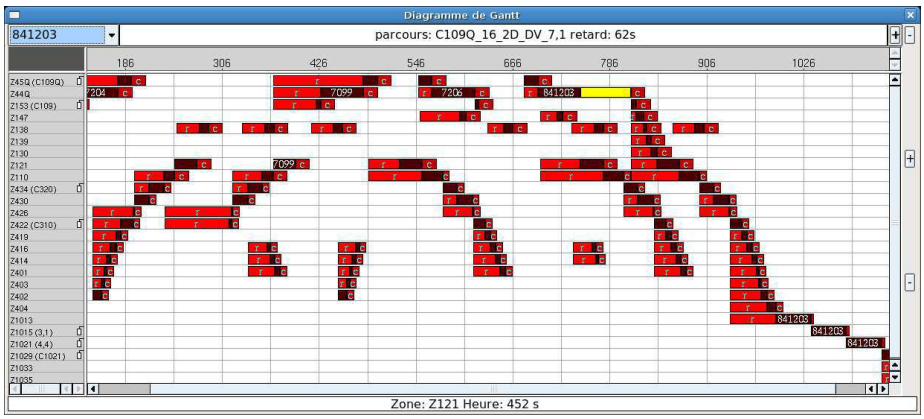

FIG. 5 – Tool for displaying the Gantt chart for a solution

results for instances up to 14 trains (instances #1 to #5). Beyond this size, a temporal decomposition allow to keep almost the same quality of solutions in less than 180 seconds thus being suitable for real time purposes.

6 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.

The state resource constraints show very promising results in relation to the resolution performances. We have shown that a temporal decomposition heuristic has also very good performances from the points of view of the quality of the solutions and the computational time for the larger 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 [12])) or DDS (Depth-bounded Discrepancy Search [30]) which have produced good results for some CSPs.

As has already been stated by Törnquist [28], many models which use other approaches have also provided very interesting results for similar problems (e.g. [9]). 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 CP model. The published results on the used of hybrid approaches for scheduling problems in other spheres (see [16, 13, 14]) show

			Global			Temporal decomposition		
Inst.	# trains	GS[a]	BS[a]	% GAP[b]	CPU[c]	BS[a]	% GAP[b]	CPU[c]
1	6	242	0*	100	1,22	7	97.11	1,12
2	8	975	79 *	91.9	1,79	86	91.18	1,28
3	10	1434	87 *	93.93	39,22	87	93.93	2,69
4	12	1806	98 *	94.57	146,96	105	94.19	7,4
5	14	1468	302	79.43	64,19	781	46.8	90,27
6	16	1304	635	51.3	88,03	370	71.63	94,95
7	18	1880	955	49.2	30,02	370	80.32	147,48
8	20	2252	966	57.1	85,27	381	83.08	132,17
9	22	2294	1252	45.42	147,33	344	85	94,64
10	24	2294	1294	43.59	38,24	235	89.76	99,63

* Optimal solution
[a] In seconds
[b] Relative gap of delays in %,i.e., 100×(GS - BS)/GS
[c] In seconds using a 1.66 GHz Intel® Duo CoreTM T2300 processor, Ilog Solver 5.3 and Ilog Scheduler 5.2

TAB. 3 – Global resolution versus temporal decomposition

that this is a very promising direction for research.

Acknowledgements

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

Références

[1] P. Baptiste, C. Le Pape, and W. Nuijten. Constraint-based optimization and approximation for job-shop scheduling. In *Proceedings of the AAAI-SIGMAN Workshop on Intelligent Manufacturing Systems, IJCAI*, Montréal, Québec, 1995.

[2] Philippe Baptiste. *A Theoretical and Experimental Study of Resource Constraint Propagation*. PhD thesis, University of Compiègne, 1998.

[3] Matthew H. Bassett, Joseph F. Pekny, and Gintaras V. Reklaitis. Decomposition techniques for the solution of large-scale scheduling problems. *AIChE Journal*, 42(12) :3373–3387, 1996.

[4] M.R. Bussieck, T. Winter, and U.T. Zimmermann. Discrete optimization in public rail transport. *Mathematical Programming*, 79 :415–444, 1997.

[5] Malachy Carey and Sinead Carville. Scheduling and platforming trains at busy complex stations. *Transportation Research Part A*, 37 :195224, 2003.

[6] J. Carlier and E. Pinson. ”Adjustment of heads and tails for job-shop problem”. *European Journal of Operational Research*, (78) :146–161, 1994.

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

[8] J-F. Cordeau, P. Toth, and D. Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32(4) :380–404, 1998.

[9] Andrea D'Ariano and Marco Pranzo. An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. In *2nd International Seminar on Railway Operations Modelling and Analysis*, Hannover, Germany, March 2007.

[10] Marta Flamini and Dario Pacciarelli. Real time management of a metro rail terminus. *European Journal of Operational Research*, 2007. In press.

[11] E.C. Freuder. "A sufficient condition for backtrack-free search". *Journal of the ACM*, 29(1) :24–32, 1982.

[12] W.D. Harvey and M.L. Ginsberg. Limited discrepancy search. In C. S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95) ; Vol. 1*, pages 607–615, Montréal, Québec, Canada, August 1995. Morgan Kaufmann, 1995.

[13] John N. Hooker. An integrated method for planning and scheduling to minimize tardiness. *Constraints*, 11(2-3) :139–157, 2006.

[14] John N. Hooker. Planning and scheduling by logic-based benders decomposition. *Operations Research*, 55(3) :588–602, 2007.

[15] Ilog. *Ilog Scheduler 5.2 Reference Manual*, 2002.

[16] Vipul Jain and Ignacio E. Grossmann. Algorithms for Hybrid MILP/CP Models for a Class of Optimization Problems. *Informs Journal on Computing*, 13(4) :258–276, 2001.

[17] C. Le Pape. Three mechanisms for managing resource constraints in a library for constraint based scheduling. In *INRIA/IEEE Conference on Emerging Technologies and Factory Automation*, Paris-France, 1995.

[18] O. Lhomme. "Consistency techniques for numeric CSPs". In *13th International Joint Conference on Artificial Intelligence*, pages 232–238, Chambery-France, 1993.

[19] Richard Lusby. Routing trains through railway junctions : Dual formulation of a set packing model. In *ORSNZ 2004*, 2004.

[20] A. Mackworth. "Consistency in network of relations". *Artificial Intelligence*, 8 :99–118, 1977.

[21] A. Mascis and D. Pacciarelli. Job-shop with blocking and no-wait constraints. *European Journal of Operational Research*, (143) :498–517, 2002.

[22] H. Montanari. "Network of constraints : Fundamental properties and applications to pictures processing". *Information Sciences*, 7 :95–132, 1974.

[23] M.C. Portmann. *Méthodes de décomposition spatiale et temporelle en ordonnancement de la production*. Thèse d'état, Université de Nancy 1, 1987. in French.

[24] Narayan Raman, Ram V. Rachamadugu, and F. Brian Talbot. Real-time scheduling of an automated manufacturing center. *European Journal of Operational Research*, 40(2) :222–242, 1989.

[25] J. Rodriguez. A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B : Methodological*, 41(1) :231–245, 2007.

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

[27] Bernado Spzigel. Optimal train scheduling on a single track railway. In M. (ed.) Ross, editor, *OR'72*, pages 343–352. North Holland Publishing Co., 1973.

[28] J. Törnquist. Computer-based decision support for railway traffic scheduling and dispatching : A review of models and algorithms. In *ATMOS2005 (Algorithmic MeThods and Models for Optimization of RailwayS)*, Palma de Mallorca, Spain, October 2005. Dagstuhl Research Online Publication Server (DROPS).

[29] Rafael Velasquez, Matthias Ehrgott, David Ryan, and Anita Schöbel. A set-packing approach to routing trains through railway stations. In *ORSNZ 2005*, 2005.

[30] T. Walsh. Depth-bounded discrepancy search. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2, pages 1388–1393, August 1997.

[31] P. J. Zwaneveld, L. G. Kroon, and S. P.M. Van Hoesel. Routing trains through railway a station based on a node packing model. *European Journal of Operational Research*, 128 :14–33, 2001.