Trains do not vanish: the ROADEF/EURO challenge 2014

Christian Artigues, Eric Bourreau, Vincent Jost, Safia Kedad-Sidhoum, François Ramond

To cite this version:

Christian Artigues, Eric Bourreau, Vincent Jost, Safia Kedad-Sidhoum, François Ramond. Trains do not vanish: the ROADEF/EURO challenge 2014. Annals of Operations Research, 2018, 271 (2), pp.1091-1105. 10.1007/s10479-018-3066-x . hal-01883062

HAL Id: hal-01883062
https://laas.hal.science/hal-01883062

Submitted on 15 Sep 2021

Trains do not vanish: the ROADEF/EURO challenge 2014

Christian Artigues[1], Eric Bourreau[2], Vincent Jost[3],
Safia Kedad-Sidhoum[4], François Ramond[5]

[1] LAAS-CNRS, Universit de Toulouse, CNRS, Toulouse, France
[2] Université Montpellier, LIRMM, 161 rue Ada, 34392 Montpellier cedex 5
[3] Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000 Grenoble, France
[4] Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
[5] SNCF Innovation & Research, F-75611 Paris Cedex 12, France

1 Introduction

The ROADEF/EURO challenge is a contest jointly organized by the French Operational Research and Decision Aid society (ROADEF) and the European Operational Research society (EURO). The contest appears on a regular basis since 1999 and always concerns an applied optimization problem proposed by an industrial partner. The 2014 edition of the ROADEF/EURO challenge was led by the Innovation & Research department of SNCF, a global leader in passenger and freight transport services, and infrastructure manager of the French railway network.

The objective of the challenge is to find the best way to store and move trains on large railway sites, between their arrivals and departures. Indeed, between arrivals and departures in terminal train stations, trains never vanish. This aspect is sometimes neglected, as it is assumed that the capacity of train stations is sufficient. In the past, stations and their associated rail networks had enough capacity to handle all trains without much trouble, but this tends to become more and more difficult. Traffic has increased a lot in recent years and some stations have real congestion issues. The current trend will make this even more difficult to deal with in the next few years. Train management in large railway sites is of high interest for SNCF, which is why it was submitted to the operations research community as the industrial problem for the 2014 edition of the ROADEF/EURO challenge.

Special issues of international journals have been devoted to the challenge since 2005. This paper aims at introducing the special issue devoted to the ROADEF/EURO challenge 2014 subject, as well as the methods of the finalist teams and their results.

Section 2 presents the problem, as defined by SNCF. Some related works are described in Section 3. The characteristics of the problem instances provided by SNCF are detailed in Section 4. The results of the competitors for the qualification and for the final phases are analyzed in Section 5, where the papers of the special issue are also introduced.

2 Problem description

This section provides a general overview of the industrial problem. A formal and complete description of the problem can be found in [13].

The objective is to find the best way to handle trains between their arrivals and departures in terminal stations. In practice, this problem is shared between several departments at SNCF, so it is rather a collection of sub-problems which are solved in a sequential way. Consequently, the formulation is provided in an integrated way which follows a prospective research approach.

In the proposed model, we focus on the multiple dimensions of the problem, in the sense that different aspects are taken into account. We remain within geographically limited perimeters, typically a few kilometers in urban areas: we consider a train station and the surrounding railway infrastructure resources. Solutions to such problems involve temporary parking and shunting on infrastructure which are typically platforms in stations, maintenance facilities, rail yards located close to train stations and track groups linking them, used by trains to move from a resource to another one (these infrastructure resources constitute the "system"). An example of such a system is presented in Figure 1.

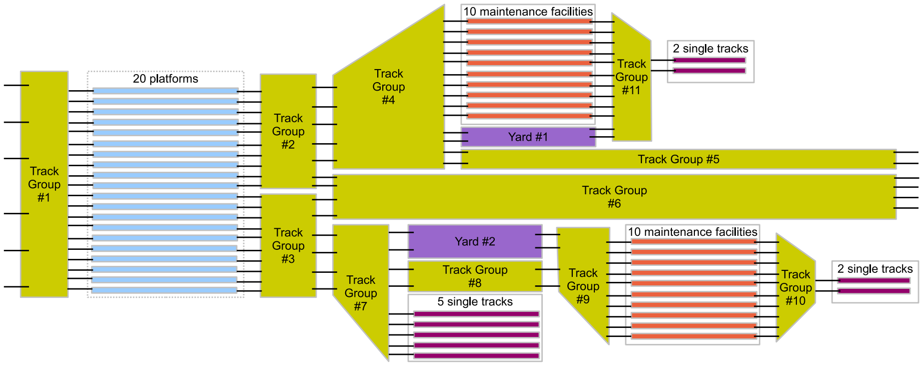

Figure 1: Example of infrastructure resources.

We consider physical trains during their presence in the system. Trains can be already in the system at the beginning of the planning horizon which typically varies, depending on the instances, between 1 and 14 days. Arrivals bring new trains from out of the system during the planning horizon, and trains leave the system through departures. Arrivals and departures are given as input data, along with their date/times.

All trains have an associated *condition*, reflecting their maintenance level. This condition is decomposed into two values: the remaining *time* before a maintenance operation is required for the train to continue running safely, and the remaining *distance*. Trains cannot be associated with a departure and leave the system if their condition does not allow to cover the whole trip. In this case, maintenance operations can be performed to improve their condition. This can be done only on dedicated facilities, which are part of the infrastructure. This implies moving the train from the station platform to the facility, and back to a station platform. Depending on the network configuration and the time window between arrival and departure, these moves on the infrastructure require sequential use of several resources, either to run or to park trains.

To allow more passengers on board, trains can be physically assembled to form longer train sets. This is possible when their categories are compatible. The operations to assemble trains are called junctions, and disjunctions refer to the inverse operation, i.e. separation of assembled trains.

A solution to the problem is then composed of a schedule for each train. A schedule is a sequence of events between the arrival and the departure of the train, representing either changes of resources, entering and leaving portions of tracks, used by the train, or operations on the train such as maintenance, or junction/disjunction. To avoid infeasible situations, it is allowed although highly penalized not to cover all the arrivals and departures. In this case, the associated trains do not enter/leave the system.

2.1 Constraints

To be feasible, a solution must respect four categories of constraints relative to either the validity of each individual schedule, or the respect of departure requirements by assigned trains, or some rules on the usage of each resource or the coherence of events of physically assembled trains. The main constraints of these four categories are described in the next sections.

2.1.1 Schedule properties

The schedule of every train must respect the following constraints:

- Each entrance of a train on a resource (portion of track) must be followed, later in time, by an exit of this resource.
- Each exit of a resource must be immediately followed by an entrance on a neighboring resource.
- Maintenance operations can only be performed on dedicated resources, known as maintenance facilities.
- Junctions between trains must last a certain duration, and can only be performed on trains located on adjacent positions on the same track.

- The duration of moves on dedicated tracks (track groups) must respect a given duration, and trains must exit on the side which is opposite to their entrance.

- Reverse of trains, i.e. moving a train in the opposite direction, is possible only on certain types of tracks, and has to respect a minimum duration.

2.1.2 Train-departure assignments

The train departure assignments must respect the following constraints:

- Only a subset of trains may be assigned to a given departure. Obviously, the train must already be in the system at the departure date/time, in a good maintenance condition and with sufficient time to board passengers.

- At most one train may be assigned to each departure. As mentioned earlier, a solution might assign no train to a departure assuming a penalty in the objective function.

- The condition of the train must be sufficient to cover the trip associated with the departure. If necessary, this can be achieved by performing maintenance operations on trains, on dedicated resources known as maintenance facilities.

- The train category such as high-speed train or local train has to be compatible with the required departure category.

2.1.3 Resource usage

This set of constraints focus on individual resources and make sure their expected usage by trains is feasible.

- Trains may only use resources which are compatible with their category.

- The capacity of resources, provided by the maximum number of simultaneous trains using them, must be respected.

- The duration of platforms usage by trains in station is limited to a maximal duration.

- There is a minimum duration of usage of resources.

- The sum of lengths of trains using simultaneously a particular track may not exceed its own length.

- The total number of maintenance operations per day is bounded.

- The train order on individual tracks must be coherent with their entrance order.

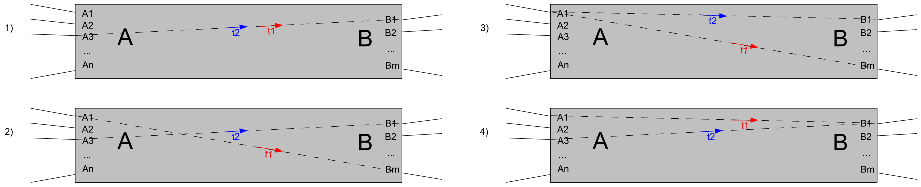

Figure 2: Track group conflicts with trains running in the same direction.

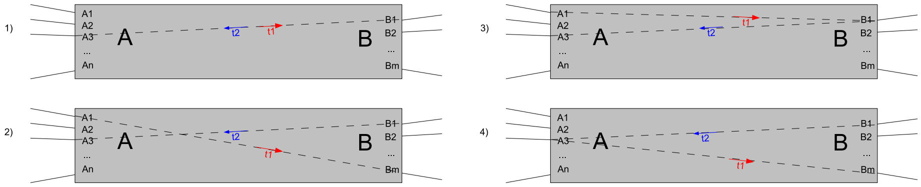

Figure 3: Track group conflicts with trains running in opposite directions.

- For safety reasons, minimum durations between trains running on the same track are imposed. These durations are different whether the trains move in the same or opposite directions. Figures 2 and 3 represent the conflicts occurring between the trains t_1 and t_2 using a common track in their path, respectively when running in the same direction both from side A to side B, and running in opposite directions, t_1 running from A to B and t_2 running from B to A.
- The capacity of a yard defined by the maximum number of trains which can be parked at a given time is limited.

2.2 Objective function

The problem aims at minimizing a weighted sum of individual costs which are associated with:

- Uncovered arrival/departure: the highest penalty concerns arrivals and departures which are not covered, as they represent trains which are not considered in the solution. The goal is to manage most trains, while respecting all constraints.
- Over-maintenance: whenever a maintenance operation is performed on a train which could still run for some distance or time, the remaining quantity is penalized.
- Train junction and disjunction operations: solutions should try to minimize the number of junctions and disjunctions, as this requires manual workforce (additionally, this induces some weaknesses in the global

timetable as it can propagate delays). Solutions where assembled trains remain together are preferred.

- Platform usage: platforms within stations are critical resources, using them for long durations is penalized.

- Platform assignment preferences: for commercial reasons, some platforms may be preferred for some trains. Penalties are given when the assignments of trains to platforms differ from these preferences.

- Non-satisfied train reuse: trains have a theoretical schedule expressing the departure which is supposed to be covered next by an arriving train. Differences in the solution from this schedule are penalized, but in some cases changing the theoretical schedule can be used to find globally better solutions.

3 Related works

A large body of literature on train routing problems is available. However, any exact or even similar matches from previous research with the considered problem could not be identified to the best of our knowledge. Only variations of some subproblems can be found such as the work of Lentink et al. [11] and Freling et al. [4].

The departure matching problem can be considered as the core problem of the challenge application which consists in finding a good and feasible matching of trains to departures while respecting train compatibility and maintenance constraints.

Freling et al. [4] identify the subproblem of matching departures and arrivals as part of a shunting problem which is related to the process of parking train units together with several related processes. The authors formulate a matching subproblem that considers the unattractiveness of producing matches that require breaking up trains into units matching different departures. The authors do not describe anything similar to linked arrivals or maintenance decisions and daily restrictions, which are the features of the addressed problem, making matches interdependent.

Kroon et al. [10] identify a matching subproblem as part of a larger station shunting problem. However there is no analogue to the linked arrivals of the departure matching problem. The authors do not solve the matching problem in isolation but as part of a larger formulation that includes shunting features not addressed in the challenge problem.

In the shunting literature there are many problems that share similarities with the challenge problem and potentially have a subproblem that is very similar to the departure matching problem. However shunting is not necessarily an important component of the challenge application because the station infrastructure has large and simplified yard resources. These are abstractions with only a maximum capacity, but train units can be parked in, or removed from, the yards without considering their ordering.

For a complete review on train routing and scheduling problems for various applications, we can quote the works of Cordeau et al. [3] and Lusby et al. [12]. Huisman et al. [8] provide a survey on some planning problems in the specific context of rolling stock management.

4 Challenge schedule and characteristics of the instances

The number of teams that registered to the ROADEF/EURO 2014 challenge reached 36, involving 107 people coming from 20 different countries, as illustrated in Figure 4.

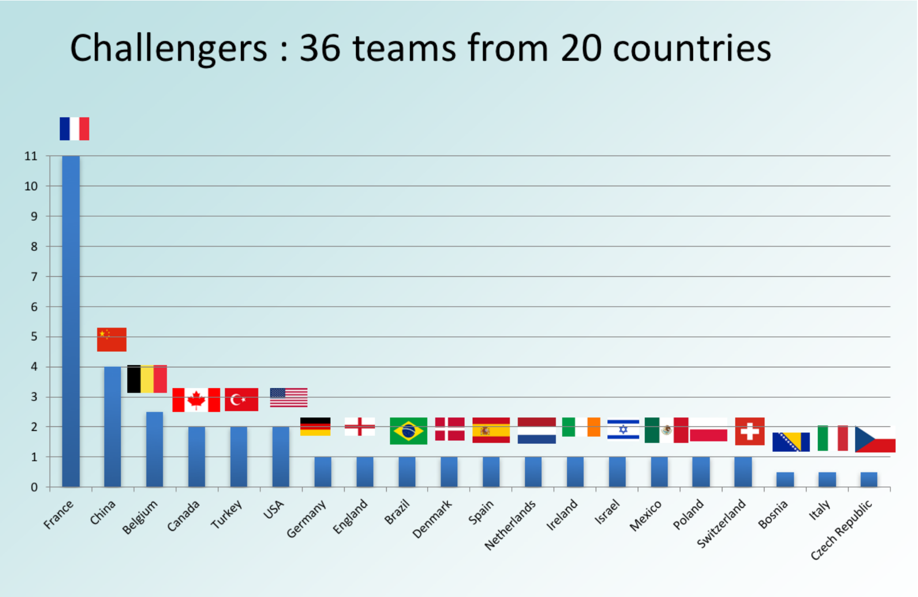

Figure 4: Origin of the competitors.

During the one year process, SNCF provided several sets of problem instances to the candidates throughout the challenge. These data were published to the candidates according to the following timeline: a first set of instances (set A) has been available to the candidates at the beginning of the challenge in September, 2013 until the end of the enrollment/qualification period in January, 2014. During this period, a sprint contest was launched during the first months to give the opportunity to the candidates (and organizers) to have an anticipated estimation on the solution quality. Solution files could be sent by the candidates on a voluntary basis and the team having the best score won the sprint prize. The sprint phase ended on November, 2013. In January 2014, the candidates had to send an abstract describing their method and runtime application. A first evaluation based on the set instances A was performed to select a shortlist of qualified teams for the final round. The second set of instances (set B) was available after the announcement of the qualified teams in February, 2014. These instances are harder than the ones of the previous set to provide to the selected candidates an opportunity of fine-tuning their methods. Moreover, some modifications were introduced in the subject, particularly in the objective function. This so-called final phase, ended by June, 2014, with a deadline for sending the second and last version of the program and an extended abstract describing the final method. A third and last set of instances, that is set X, not revealed to the candidates, was used in conjunction with the set B to determine the ranking of all the finalists. The results were announced during the 2014 IFORS conference. This last set was made available to the candidates only after the announcement. Finally, a last round in July 2014 was performed to award a scientific prize based on a technical report and an oral presentation at IFORS meeting.

The instances provided by SNCF for the competition were built using real data obtained separately from different entities. The goal was to constitute realistic datasets, with a particular attention to the global practical relevance. This was challenging and required an important effort. In some cases, approximations had to be made. Altogether, the main objective of having coherent datasets, i.e. without inherent inconsistencies, with a certain diversity, was achieved. Only a few corrections were made after the publication of instances. Still, we believe that this approach did not allow to generate fully feasible instances, for which it was possible to obtain solutions where all arrivals and departures could be covered without conflicts on the track group.

A solution checker was developed by SNCF and made publicly available. For any instance and any given solution to this instance, it enabled anyone to compute its objective value and to check whether all constraints were respected. For the sake of transparency, this checker was open-source, so that the source of any violated constraint or any question in the objective value could be directly explained to the competitors.

We describe in what follows the characteristics of the instances. Globally, the time horizon varies from one day to a full week. There are up to 2000 trains to schedule on the largest instances, which correspond to realistic situations in large French railway stations. Table 1 gives the characteristics of instances B and X, respectively, in terms of number of days, number of arrivals and departures, number of resources, as well as some features of the best solutions obtained for each instance: percentage of covered arrivals, percentage of covered departures, number of maintenance operations and number of junctions/disjunctions to perform.

	Instance characteristics			Best obtained solutions				
	Days	Trains	Res.	Cov. A.	Cov. D.	Maint.	Junc.	Disj.
B1	7	1235	56	81%	80,5%	202	0	0
B2	7	1235	56	81%	80,5%	202	0	0
B3	7	1235	56	86,3%	85,9%	341	0	0
B4	7	1780	68	76,1%	73,5%	197	0	0
B5	7	2153	62	74,7%	73,4%	222	0	0
B6	7	1780	62	80,8%	78,5%	195	0	0
B7	1	304	88	80,3%	75,3%	24	0	0
B8	1	304	88	73%	68,8%	24	0	0
B9	7	1967	88	72,2%	71,3%	287	0	0
B10	1	196	35	67,3%	67,3%	11	2	0
B11	7	1122	35	55,7%	56,8%	119	36	30
B12	3	570	35	56,5%	58,1%	55	16	16
X1	7	1235	61	76,5%	76,2%	197	0	0
X2	7	1499	56	66,3%	70,2%	208	0	0
X3	7	1235	56	80%	79,6%	202	0	0
X4	7	1780	62	80,6%	78,1%	192	0	0
X5	7	1780	62	80,6%	78,1%	192	0	0
X6	7	1780	62	83,5%	80,7%	258	0	0
X7	7	1967	88	64,3%	63,7%	241	0	0
X8	3	905	88	77,6%	75%	91	0	0
X9	3	905	88	66,2%	65%	75	0	0
X10	1	196	35	57,7%	60,7%	10	0	0
X11	7	1122	35	62,9%	63,9%	130	28	36
X12	3	570	35	66,1%	67,4%	54	20	26

Table 1: Statistics related to final instances (B and X set).

5 Methods overview and results

5.1 Overview of the proposed methods

In this section we propose an overview of the methods used by the competitors. Table 2 presents a summary of these methods as well as their main components. Almost all the participating teams propose a decomposition approach based on two or three phases. The first one aims at assigning trains to departures requests subject to maintenance requirement and linked departure constraints. The objective of the second phase is to build feasible routes inside the station without exceeding resource capacities. A final post optimization procedure aims at improving the obtained feasible solution by local search. In Table 2, the components present in each of these phases are displayed. The following abbreviations are used: CP stands for Constraint Programming, TS for Tabu Search, SA for simulated annealing, LNS for large neighborhood search. As one can notice, two teams tackle globally the problem. One team provides an extended MILP formulation, integrating the two above mentioned aspects, which is solved using a double column generation method. Finally, another team uses a HyperHeuristic framework to solve the problem.

	Arrival to Departure		Route and Schedule		Post Opt.
TeamId	Greedy	MILP	Greedy	CP	
S18	✓	Assignment	✓		TS
S19	✓	Weighted Matching	✓		Add Cut
S1	✓		✓		∅
J9	✓			✓	∅
J10		GenCol (RCSPP)	✓		SA
S9	✓		✓		LNS
J11	HyperHeuristic				
S22	✓(flow)		✓		LS
S20	Double Column Generation				
J4	✓		✓		∅

Table 2: Finalists methods overview.

We provide in what follows a brief description of the proposed methods. The length of our description depends mainly on the material that was sent by the competitors to describe their method.

First, **Buljubašić, Gavranović, Hanafi and Vasquez** (team S18, [1] in this special issue) propose a two phase heuristic. During the first phase, a greedy heuristic associated with a MILP assigns trains to departures. The priority rules aim at avoiding maintenance for trains having small remaining time/distance before maintenance, favor maintenance in the other cases and avoid long waiting times. Linked departures assignments are then fixed by the greedy heuristic. A MILP is used to assign the remaining departures to trains while satisfying a maintenance constraint for each pair of ordered days, so as to determine the number of maintenance activities in the corresponding interval, but not the precise maintenance days. A fast procedure determines afterwards the maintenance days according to the MILP results. A theorem proves that this procedure ensures the maintenance constraint satisfaction. A scheduling algorithm constitutes the second phase. First, a constructive algorithm is based on a set of possible elementary moves and on the pre-enumeration of a subset of shortest paths (a succession of resources) from each pair of locations in the network of connected resources. Resource consumption and feasibility are checked all along the construction process and a specific algorithm determines the exit time of resources that can be dependent on the next visited resource. An ideal starting time for each travel is determined by a set of rules and possible starting times are enumerated between the ideal starting time and a parametrized latest start time. For track groups, the search of feasible entry/exit gates is performed by a depth first search procedure starting with gates that are preferred according to a criterion that reflects their criticality. To anticipate future decisions that impact the overall availability, virtual visits that may occur on the track groups are defined. The number of conflicts with such virtual visits is minimized. The constructive method also orders the trains that have to be be scheduled according to their assigned departures. Second, an iterative improvement procedure is carried out. Infeasible solutions with more scheduled trains but having conflicts are first generated as an input. Then, a local search procedure is launched to remove the generated conflicts. It is based on a tabu search operating on a partial feasible solution extracted from the infeasible solution, that aims at adding gates to partial visits in an iterative fashion. Note that this method won the first prize of the competition on B and X instances, obtaining more than 75% of the best known solutions.

Teypaz (team S19) finishes in the second position and finds the best known solutions on the remaining 25% instances. In a first step, as for the previous team, shortest paths on the station graph connecting all the available resources (yard, platform, gate) are computed for each train category. Additionally, all paths with the same optimal cost are generated to allow alternative routes if there is a lack of resources. In a second phase, the problem is solved iteratively day by day with four different matching models: a minimum cost flow to maximize the linked departures, a minimum cost flow to maximize matching between arrivals and departures, a minimum weighted matching for trains having junction, disjunction and maintenance operations and a minimum weighted matching for the remaining trains. Finally, for each train, a valid time schedule is tentatively generated by instantiating different patterns of pre-computed generic schedules originated from the shortest paths. In case of failure in the scheduling phase, the current associations are banned via combinatorial Benders-like cuts and the matchings are recomputed. All the proposed algorithms are coded in C++ with the graph algorithms LEMON Library from COIN-OR and with GLPK to solve the matchings with IP modeling.

Geiger, Huber, Langton, Leschik, Lindorf and Tüshaus (team S1, see [5] in this special issue) apply also an iterative procedure with a pre-processing phase. They decompose the problem resources into two categories: the first one is devoted to the track groups, on which trains cannot stop. They are modeled as edges of a graph linking nodes corresponding to other resource categories. A preprocessing phase aims at computing all paths between resources (or resources sets) in this graph that do not exceed a threshold length above the shortest one. Gathering resources into resources sets that have the same neighbor structure yields memory and time savings. A yard manager acts as a global constraint ensuring the satisfaction of yard capacity constraints. Managing the conflicts on track groups is also carried out. Given a convoy and its time window usage on a track group, they compute earliest usage time of the track group gates after the convoy and latest usage time before the convoy. The blocking ratio of each track for each path from an entry gate to an exit gate is maintained. The procedure can be sketched as follows. The first step aims at finding a feasible scheduling solution for the trains. A platform is assigned to each arrival and movements into a yard are planned without covering any departure. Then some iterations based on the following improvement procedures are performed until a feasible solution is found. A train-to-departure assignment is done using a multi-attribute priority rule. An attribute gives priority to departures having less possible trains. Another priority attribute is based on a regret. The regret increases if trains with high time or distance before maintenance are assigned to departures with low time and distance before maintenance. Minimum slack between departure time and train availability time is also considered while maintenance is avoided in some cases. Moreover, a so-called transaction model is defined. It consists in defining different local operators (such as "park a convoy") and in verifying their feasibility given the current state. A specific data structure is used to memorize the resource usage and availability. The operators are chained within a high level heuristic based on simple rules according to different assignment scenarios. A transaction is a chaining of elementary operators. Success rates of implemented transactions are analyzed. Let us remark that this team finished at first place during the first phase (qualification) but was ranked third in the final phase. The source code is published [6]

Joudrier and Thiard (team J9, ranked first in junior category, see [9] in this special issue) use a simple constructive heuristic without post optimization. As the previous team, they start by grouping equivalent resources to simplify the graph model and compute shortest paths between pairs of resource by train category. Then, they use a three level routing table to assign trains by checking feasibility with a dedicated constraint propagation technique based on multi-interval bounded variables using contractor propagation.

Haahr and Bull (team J10, second junior prize) propose a matheuristic, which classically mixes an exact method with heuristics. They decompose the problem into three steps: Matching Arrival Departure (MAD), Platform Assignment (PA) and Route Assignment (RA). MAD is achieved by solving a MIP with an exponential number of variables through Column Generation and branch-and-bound approaches. A first easy subproblem consists in generating one train to departure matchings while a second subproblem (one per train category) is defined as a Resource Constrained Shortest Path Problem (RCSPP) in a graph where nodes represent linked arrival/departures and the resource constraints are necessary for maintenance issues. PA is also solved using a MIP that minimizes the cancellation cost due to a lack of available platforms. Finally, RA is achieved through a Simulated Annealing scheme based on assigning randomly generated paths to trains. A path is randomly generated for each train but if it is blocked, a removal of neighbor trains is carried out until feasibility is reached. Then re-insertions of removed trains are randomly performed. Interestingly, the MAP MIP model allows to obtain lower bounds on uncovered departures or on the minimum cost. A description of a part of this work is available in [7].

Spier (team S9) uses a two-stage search: a constructive heuristic followed by a post optimization phase. First, for each available train, train paths on a network graph describing physical and temporal connexions are found via an hitting set model. Post optimization is achieved by removing and then rebuilding paths related to uncovered trains.

Kheiri and Elsayed (team J11) develop a Multi Stage Hyper Heuristic based on low level heuristics applying neighborhood operators on multiple layers of the solution. Solutions are indeed represented in three layers: the planning layer for each train/movement, the layer related to platforms and the layer describing routes and the fine-grain schedule. They propose an iterative framework aiming at exploiting a large set of constructive/destructive operators, each of which attempts to enhance an aspect of the quality of a solution in hand during the search process.

Wasik, Zurkowski and Jaskowski (team S22) also propose a greedy constructive heuristic with post optimization local search phase. First, a heuristic based on network flows determines the initial assignment of arrival trains to departure requests. Then, a resource-time graph is used to find a schedule by a depth first search. Finally, a random local search is applied in a post-optimization phase.

Letocart, Casazza, Rozenknop, Traversi and Wolfler-Calvo (team S20) propose a double column generation scheme based on an exponential set of variables for the schedules (assignment of trains to departures) and an exponential set of variables for the routes (a timed path, including waiting times on authorized areas represented by edges). The Master problem involves the two families of variables, hence yielding two pricing problems. These correspond more or less to the two phases that most all participants have, but they are integrated and coordinated through the Master Problem. The first pricing problem find a schedule of negative reduced cost by solving a resource-constrained multi commodity flow problem on an auxiliary bipartite graph where nodes are trains plus departures and arcs connect trains to compatible departures (plus source and sink nodes). A compact MILP model is presented to solve the pricing problem with constraints involving big-M coefficients for maintenance restrictions. To get rid of these constraints, a surrogate model considers the same set of nodes with additional hyper arcs to represent joint trains. Maintenance constraints are replaced by clique constraints preventing sets of couples that are incompatible with regard to maintenance activities to be selected. These clique constraints are added on the fly as weak feasibility cuts once an integer solution is found.

The second pricing problem finds a route of negative reduced cost by solving a rather standard resource-constrained shortest path problem. Incompatibility constraints between routes are also added on the fly to the master problem by scanning the set of already generated routes as soon as a new route is generated.
Arnaud, Phouratsamay, Pataut and German (team J4) are the last finalist able to solve at least part of instances of the X set. They use a constructive heuristic by ordering trains using the maximum number of covered departures criterion.
During the qualification phase, **Herman, Sessions and Hankins** (team S14) use the DEXPert Optimisation library. In a two-stage approach, they first use a constructive heuristic, then they apply a post optimization procedure in the Large Neighborhood Search fashion. The greedy assignment for arrival trains is made according to their "desirability", i.e.: the decreasing number of departures per time. For each train, a schedule and routing is built to satisfy a departure request. In the second part, the solution is incrementally improved with the "Ruin and Recreate" home made metaheuristic (select worst scoring schedule, remove it and try to rebuild it randomly). Empirically, a careful balance between exploration and exploitation of the search space is done by varying the number of schedules to be ruined, generally in proportion of the number of departures available or missed. Additional artificial constraints restrict the schedule possibilities, such as a junction that is only authorized in a yard or a small upper bound defined for the maintenance storage time.
Elbar (team J3) proposes a two-stage method using IBM OPL and CPLEX: the matching between arrivals and departures is solved first, and a platform assignment model is subsequently considered.
Finally, **Cambazard and Catusse** (team S11) were qualified for the final phase but they did not submit an executable. However they won the scientific price thanks to the following contributions. They propose an efficient matching-based MIP model for the arrival/departure assignment (without linked departures) model integrating linearly maintenance constraints. As team S18, the precise maintenance days are not determined by the MIP and they prove that feasible maintenance days can be derived by solving a bipartite flow model. They also consider the basic matching model with linked departures but without side constraints and they show that the problem is NP-complete by reduction from SAT. They suggest an extended formulation and a greedy algorithm to solve this problem. In a second stage, routings are computed by solving a Resource Constraint Shortest Path Problem with Time Windows model. A constraint programming model making use of the so-called AllDifferent, Element and NoCycle constraints is proposed. The paper is published in [2].

5.2 Analysis of Results

On such a complex problem solved with methods that are generally non trivial associations of several components, it is highly hazardous to draw general conclusions on the method category that would be of choice for the problem. However, by analyzing the similarities and differences of the proposed methods, we can underline a few invariants in the successful approaches and make a few hypotheses on some ingredients that can have made some methods better than others. Among the registered teams, thirteen were qualified for the final phase (see Figure 5).

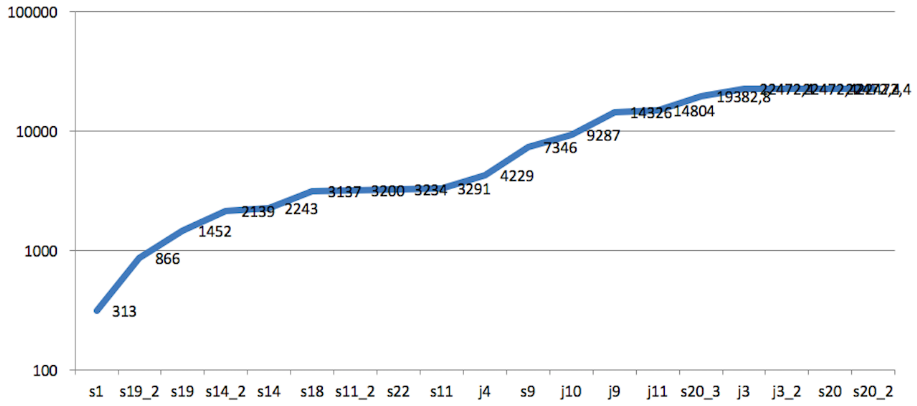

Figure 5: Final ranking of the qualification phase.

Focusing somewhat arbitrarily on the first four teams of the qualification phase, one can see that team S1 obtains the best results, which shows the contribution of multi-attribute constructive heuristics for the problem as well as precomputation of the paths and fast local search. The second team, S19, shares also some successful ingredients with the previous team, especially the shortest path precomputation. But the success of S19 also emphasizes the importance of the underlying matching model that was intensively used in different variants by this team. The potential of large neighborhood search was exhibited by S14 that ranked third on this phase, which can be explained by the close interconnection of different elements of a schedule via resource usages and track groups that may yield the need to destroy and rebuild rather large parts of the solution. However, as the team did not submit a program for the final phase involving larger instances, we have unfortunately no information on the scalability of this approach. In what follows, we analyze the method proposed by the winners of the final phase of the challenge.
Some qualified teams did not sent a program for final evaluation. The results of the 10 teams that submitted a program are summarized in Table 3.

For the final phase of the challenge, the method of team S18 significantly outperformed all other methods on the X and B sets. It turns out that due to the extensions proposed for the problem between the two phases of the competition and the increase on the instance sizes, joining as many forces as possible to address the problem was a necessity. The S18 method cleverly combines almost all ingredients that were included (but not all together) in the concurrent methods: the two matching then scheduling phases, the matching problem variants solved via a greedy algorithm and a MIP with smart maintenance constraints as

#	Team	Team members	Score	
1	S18	Mirsad Buljubašić, Haris Gavranović, Sad Hanafi and Michel Vasquez	19 992 021	Winner
2	S19	Nicolas Teypaz	25 730 692	
3	S1	Martin Geiger, Sandra Huber Sebastian Langton, Marius Leschik, Christian Lindorf, Ulrich Tüshaus	41 173 480	
4	J9	Hugo Joudrier and Florence Thiard	41 842 837	1st Junior
5	J10	Jørgen Thorlund Haahr and Simon Henry Bull	42 947 905	2nd Junior
6	S9	Grégoire Spiers	52 124 835	
7	J11	Ahmed Kheiri and Mohamed Elsayed	53 146 957	3rd Junior
8	S22	Szymon Wasik, Piotr Zurkowski, Wojciech Jaskowski	53 412 077	
9	S20	Lucas Letocart, Emiliano Traversi, Antoine Rozenknop, Marco Casazza, Roberto Wolfler-Calvo	55 094 611	
10	J4	Luc Arnaud, Siao-Leu Phouratsamay, Guillaume Pataut, Grigori German	57 199 217	

Table 3: Final scores.

well as the precomputation of shortest paths and a fast local search procedure. In addition, specific contributions that probably helped to make the difference, such as the concept of virtual visits for a better control of the resource conflicts and the use of local search on partial solutions, that could in short be seen as a scalable variant of the large neighborhood search method. Ranked second as in the qualification phase, the S19 method thereby demonstrates its robustness. We underline that it also includes original components compared to other proposed methods that could explain its success: the Benders-like cuts that enforce the generation of a new matching when the scheduling phase fails and the use of predefined schedule patterns. The winners of the qualification phase, team S1, were ranked third, with a much higher score than the first two teams. According to their comments, they experienced difficulties in handling the modifications brought to the subject between the two phases. Especially, the disjunction management in yards yielded difficulties for the yard overload management that, as we already pointed out, was finely handled by the S18 team through the virtual visits and by the S19 team via the schedule patterns. The junior team J9 was ranked fourth (and first in the junior competition) with results that were really close to the S1 team. The use of constraint propagation algorithms could be credited for guiding the search towards feasible solutions. The junior team J10 was ranked fifth (second of the junior teams) with results very close to J9 and S1. We believe that the extended formulation and column generation they proposed for handling the matching of linked arrivals is a promising research direction.

To evaluate the instance difficulty with regard to the results obtained by the proposed methods, Figure 6 displays the score relative to best known solutions found on each instance of the B and X datasets.

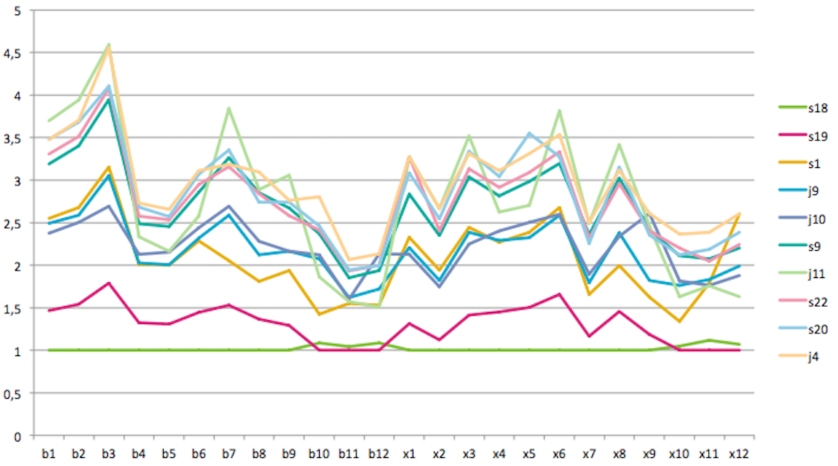

Figure 6: Scores relative to the best solutions.

6 Conclusion

As already mentioned the ROADEF/EURO challenge 2014 was a successful scientific event. We believe that the numerous proposed methods that include graph algorithms, local search, heuristics, hybrid constraint-programming and integer programming components made a significant advance of the state of the art for complex train management problems.

Acknowledgements

The authors warmly thank the SNCF expert group for their high involvement in the challenge subject definition, as well as the IT department who implemented the solution checker. Special thanks are addressed to Nicolas Marcos, who brought a major contribution to this challenge.

References

[1] Mirsand Bujubašić, Michel Vasquez, and Haris Gavranocić. Two-phase heuristic for SNCF rolling stock problem. *Annals of Operations Research*, 2018. this volume.

[2] Nicolas Catusse and Hadrien Cambazard. Roadef challenge 2014: A modeling approach, rolling stock unit management on railway sites. Technical Report hal-01056605, GSCOP-ROSP. `https://hal.archives-ouvertes.fr/hal-01056605`.

[3] Jean-François Cordeau, Paolo Toth, and Daniele Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32(4):380–404, April 1998.

[4] Richard Freling, Ramon M Lentink, Leo G Kroon, and Dennis Huisman. Shunting of passenger train units in a railway station. *Transportation Science*, 39(2):261–272, 2005.

[5] Martin Josef Geiger, Sandra Huber, Sebastian Langton, Marius Leschik, Christian Lindorf, and Ulrich Tüshaus. Multi-attribute assignment of trains to departures in rolling stock management. *Annals of Operations Research*, 2018. this volume.

[6] Martin Jozef Geiger. Roadef/euro challenge 2014 - source code of the qualification phase, mendeley data, v1. http://dx.doi.org/10.17632/nc642wfw2k.1, 2017.

[7] Jørgen Thorlund Haahr and Simon Henry Bull. Exact methods for solving the train departure matching problem. Technical report, DTU Management Engineering, 2015.

[8] Dennis Huisman, Leo G Kroon, Ramon M Lentink, and Michiel JCM Vromans. Operations research in passenger railway transportation. *Statistica Neerlandica*, 59(4):467–497, 2005.

[9] Hugo Joudrier and Florence Thiard. A greedy approach for a rolling stock management problem using multi-interval constraint propagation. *Annals of Operations Research*, 2018. this volume.

[10] Leo G Kroon, Ramon M Lentink, and Alexander Schrijver. Shunting of passenger train units: An integrated approach. *Transportation Science*, 42(4):436–449, November 2008.

[11] Ramon M Lentink, Pieter-Jan Fioole, Leo G Kroon, and Cor vant Woudt. Applying operations research techniques to planning of train shunting. *Planning in Intelligent Systems: Aspects, Motivations, and Methods*, pages 415–436, 2006.

[12] Richard M. Lusby, Jesper Larsen, Matthias Ehrgott, and David Ryan. Railway track allocation: models and methods. *OR Spectrum*, 33(4):843–883, Oct 2011.

[13] François Ramond and Marcos Nicolas. Trains don't vanish ! ROADEF EURO 2014 Challenge Problem Description. https://hal.archives-ouvertes.fr/hal-01057324, February 2014.