A Hypergraph Model for the Rolling Stock Rotation Planning and Train Selection

Mohamed Benkirane, François Clautiaux, Jean Damay, Boris Detienne

A Hypergraph Model for the Rolling Stock Rotation Planning and Train Selection

Mohamed Benkirane[1,2,3], François Clautiaux[2,3], Jean Damay[1], and Boris Detienne[2,3]

[1]Pôle Études, Direction Régionale TER Nouvelle-Aquitaine, SNCF, Bordeaux, France
[2]Univ Bordeaux, CNRS, IMB, UMR 5251, F-33400 Talence, France,
firstname.lastname@math.u-bordeaux.fr
[3]Inria Bordeaux-Sud-Ouest

Abstract

This paper is about an integrated optimization approach for timetabling and rolling stock rotation planning in the context of passenger railway traffic. Given a set of possible passenger trips, service requirement constraints, and a fleet of multiple heterogeneous self-powered railcars, our method aims at producing a timetable and solving the rolling stock problem in such a way that the use of railcars and the operational costs are minimized. To solve this hard optimization problem, we design a mixed-integer linear programming model based on network-flow in an hypergraph. We use this models to handle effectively constraints related to coupling and decoupling railcars. To reduce the size of the model, we use an aggregation and disaggregation technique combined with reduced-cost filtering. We present computational experiments based on several French regional railway traffic case studies to show that our method scales successfully to real-life problems.

Keywords— rolling stock rotation planning, train selection, mixed integer programming, hypergraph model

1 Introduction

Railway production is based on the use of several resources such as infrastructure, staff, and rolling stock. A service planning assembles all these resources while respecting technical, organizational and legal constraints. Many working rules also have to be followed. Interactions between these different resources and constraints complicate the process of planning.
This process is generally decomposed into sequential subproblems in the following order: *line planning*, *timetable planning*, *rolling stock rotation*, and *crew scheduling*. The process of *line planning* corresponds to the definition of lines based on the rail network. The line planning includes the number and the routes of lines. It also includes the choice of the line frequencies (see [12]). In the timetable planning, the decision maker chooses when trips will be operated, using the lines designed in the previous step. In the *rolling stock rotation problem*, one defines the train composition for each trip defined in the timetable, *i.e.*, one associates to each train a set of railcars and specifies when coupling and decoupling occurs. Finally, *crew scheduling* corresponds to the process of assigning crews of working agents to trains.
Rolling stock optimization is key in our problem. This problem was largely discussed in the past recent years. Several different versions of the problem have been studied, depending on whether maintenance constraints, order and orientation of the railcars are considered or not. There are two main types of formulations in the literature: path-based and flow-based formulations. Path based models integrate easily unit specific constraints (e.g maintenance constraints), but produce exponentially large formulations that have to be generated dynamically. Flow-based models are generally not able to follow individual railcars, but can be solved directly by a general-purpose MIP

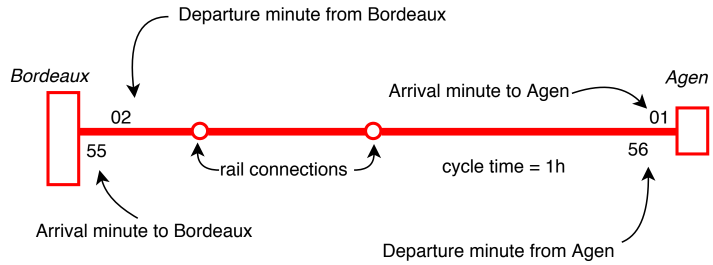

Figure 1: Train pattern in the netgraph.
In this example we can operate a train from Bordeaux to Agen each hour, the defined cycle time is equal to one hour. For instance 00h02, 01h02, ... , 22h02 with an arrival at 01h01, 02h01, ..., 23h01 respectively

solver. For a recent survey of the literature, see [8], where the authors compared the two types of models using testbeds from different countries. Among the most recent approaches, the path based model presented by [8] considers the order of train units in a composition. It extends the model presented in [9]. A branch-and-price algorithm generates the path variables dynamically by solving a sub-problem that corresponds to a shortest path problem with resource constraints. Many network-flow models are extensions of the seminal work of [7]: [11], [3] and [10] for example. One of the most recent flow-based model was presented by [11], based on the work described in [7]. The authors present a multi-commodity flow model where each commodity is a train unit type, which takes into account railcar inventory constraints at the end of the day as well as changes in the composition of trains.
Network-flow models in hypergraphs natively embed train composition constraints within the hypergraph, without adding additional linking constraints. In addition one can associate to multiple train units a cost that is non proportional to the cost of a train unit. [2] solve the rolling sock problem based on such an hypergraph model. They integrate in their models regularity constraints, maintenance constraints and train composition. Each node of the hypergraph represents a departure (or arrival) train station, a departure (or arrival) time, a composition and a train unit appearing in this composition and its orientation. The hyperarcs represent either trips or coupling/decoupling operations. Hyperarcs representing trains link a set of departure nodes and a set of arrival nodes, those nodes represent the same trip and the same composition. Hyperarcs representing train re-compositions link a set of arrival nodes of a trip and a set of departure nodes of another trip such that the number of train units for each type is conserved. The latter hyperarcs integrate also the regularity of the rolling stock. The mixed integer program integrates maintenance constraints through another flow commodity which is absorbed by trip hyperarcs and generated by service nodes.
At the national French railway company (SNCF) the *timetable planning* is decomposed in two phases: *netgraph design* and *train selection*. The concept of *netgraph* is used to describe all possible trips that can be produced. The netgraph is made available by the infrastructure manager for all train operators, which justifies how the train selection problem arises at SNCF Mobility, which represents the train operator, and at all its potential competitors. The netgraph defines a set of *train patterns* (see Figure 1), *i.e.*, an origin and a destination train stations, and a set of stops. For each train pattern, a set of possible departure and arrival minutes are given. Those minutes are translated to possible starting times through a given cycle time of the pattern (typically every one or two hours).
To operate a train, an operator needs to purchase a train path on a specific departure time from the infrastructure manager. The *netgraph design* focuses on the "clock-face" scheduling, which is known in the literature as the *cyclic train timetabling problem* (see e.g. [13, 6, 4]). The *train selection* selects in the netgraph a set of trains that satisfies the passenger demand and public authorities requests. In this work, the term *train* stands for a trip defined by specific stopping points between origin and destination at precise times; the term railcar stands for a self-propelled passenger railway vehicle, also called train unit.
Given a netgraph and train frequency requirements, our approach aims at solving the train selection and rolling stock problems in an integrated manner. This problem occurs at the strategic level, when the company decides which trains will be operated. The objective is to minimize the cost of the railcars used, and some penalties for not satisfying demand constraints. An originality of our approach is to integrate the choice of the selected trains within the process of optimizing the rolling stock. Since we deal with a strategic problem, precise operational constraints

and maintenance are not considered, although an estimation of production costs is accounted for. Trip demand is also defined in an original way, which arises from the opening up to competition and from the netgraph formalism. We use a network-flow model on a hypergraph to model and solve our problem. This model allows to effectively deal with coupling/decoupling constraints and costs. Our hypergraph formulation differs from [2] for several reasons: one flow unit can be related to a single train unit, but also to several combined units. Likewise, nodes are associated with combined train units. We do not integrate maintenance constraints.

In terms of solving approach, we use a variant of the aggregation technique presented in [2]. In our method, we first solve with an MIP solver a simplified problem where all railcars are considered equivalent. We use this first phase to filter out many variables from the initial problem and produce a first feasible solution. Both techniques are used to speed up the solution of the initial problem.

We describe our problem in section 2, and propose a hypergraph-based mixed integer model in section 3. We present in section 4 our solving approach. Finally, we present our computational experiments in section 5.

2 Problem description

In the present work, we focus on combining the train selection and the rolling stock problems, *i.e.* we take the line planning and the netgraph as inputs, and do not consider constraints related to crew scheduling. The goal of our optimization problem is to evaluate for a specific set of lines and in a rapid manner the cost of the involved rolling stock. Note that *the cyclic timetabling problem* and the *netgraph design* are not addressed by our approach.

2.1 Physical network and rolling stock

Let $\mathcal{S}$ be the set of train stations. The infrastructure resources enable to operate train units from one station of $\mathcal{S}$ to another one at given times of the horizon.

Definition 1 (train pattern)**.** *A* train pattern *p is a tuple (s_p^-, s_p^+, S_p), where $s_p^-, s_p^+ \in \mathcal{S}$ respectively represent departure and arrival stations, and S_p the set of stations where stops occur.*

The set of train patterns is denoted as P. Train patterns can only be operated at certain times, which are allocated by the infrastructure manager. A train pattern assigned to a certain time is called *train* in the remainder of this paper. The set of possible trains is denoted as $\mathcal{V}$.

Definition 2 (train)**.** *A* train *i is a tuple (p_i, t_i^-, t_i^+), where $p_i \in P$ is a train pattern, t_i^- and t_i^+ the departure and arrival times for p_i.*

For the rolling stock rotation problem, we consider that we have at our disposal a set of heterogeneous railcars. Each railcar has a *railcar type* r, which is characterized by a length, a cost $\psi(r)$, and a payload $\mu(r)$. We denote by $\mathcal{R}$ the set of railcar types.

Railcars are distributed over a set of warehouses, which are located at some given stations. For $s \in \mathcal{S}$, $r \in \mathcal{R}$, we denote respectively by $n_{r,s}^{\min}$ and $n_{r,s}^{\max}$ the minimum and maximum number of railcars of type r that begin the service from s in a feasible solution. The number of railcars beginning the service from a train station must be equal to the number of railcars ending the service at the same train station. This is required to allow a cyclic planning.

A *railcar composition* c is an ordered list of railcar types. We denote by $\mathcal{C}$ the set of feasible railcar compositions. The number of elements in the list is denoted $\text{len}(c)$. When two compositions c_1 and c_2 are concatenated, we use the classical notation $c_1 + c_2$. With a slight abuse of the notation, we will denote $\mu(c) = \sum_{r \in c} \mu(r)$ the payload of composition c.

Individual railcars are considered as simple compositions. Compositions are created from other compositions that are coupled or decoupled. A composition can be used at a given station only if it is already present at this station, or the stock of compositions at this station is sufficient to create the composition. Operational constraints limit the creation of a new configuration to only one coupling or decoupling operation at train station platforms. Station-dependent constraints may also forbid such operations. However, any composition can be created from single units at the depots, and vice-versa. Each coupling (resp. decoupling) operation incurs a cost denoted by δ^+ and δ^-. A composition is not allowed to stay idle at a station platform more than a given duration τ_s, $s \in \mathcal{S}$: after that time period, it is moved to the depot, as a set of individual railcars.

2.2 Constraints on train selection

Deciding whether each train is operated by a vehicle or not is the main decision in our problem. To be operated, a train has to be assigned a composition. A railcar composition can only be used to operate a train if feasibility conditions are met, *i.e* the length of the composition does not exceed the length of the train station platforms; tracks and powering types must be compatible to railcar types composing the train composition.

As part of the problem definition, some pairs of trains are mutually exclusive. For each $i \in \mathcal{V}$, we denote by $\mathcal{E}_i$ the set of trains that cannot be operated if train i is (independently of the composition used).

If two trains share a same portion of the trip on departure or on arrival, they can be operated together, which reduces the operational costs. In this case, they are coupled (or decoupled) at an intermediate station. When a coupling operation is performed, it means that two trains are first operated each with a given composition. After the intermediate station, the concatenation of the two compositions is used to operate both trains over their common portion. Pairs of trains that can be coupled or decoupled are specified in the data. We denote by $\mathcal{V}^+$ the set of ordered pairs of trains sharing the arrival portion; $\mathcal{V}^-$ represents the set of ordered pairs of trains sharing the departure portion.

A railcar, appearing in two different railcar compositions, can only be used for two successive trains if the trains and their compositions are *compatible*, *i.e.* it is practically feasible. Some pairs of trains are not compatible in any composition. Some are only compatible in some compositions. Compatibility depends on the station where the successive trains are operated. Several practical constraints are taken into account (arrival times, time of coupling or decoupling railcars, stub-end stations etc.). Moreover, operating consecutive (set of) trains with some given compositions has a cost.

The cost for operating one or two trains together with one or two compositions (in the case of coupling/decoupling within the trip) depends on several factors. To simplify the exposition, we introduce $\beta((I_k, c_k)_k)$ as the cost of operating trains in sets I_k with their respective compositions c_k. If this association is not possible, then $\beta((I_k, c_k)_k) = +\infty$. This notation is later used to express the cost of hyperarcs in our model.

2.3 Mission demands

The selection of trains is subject to passenger demand constraints coming from issues and public authorities requests. The frequency distribution of trains and the planned capacity have to meet demand requirements, which are expressed as *missions*.

Definition 3 (mission)**.** *A* mission m *is a tuple* $(p_m, \nu_m^{\min}, \theta_m, [t_m^{\min}, t_m^{\max}])$ *where* $p \in P$ *is a train pattern,* $[t_m^{\min}, t_m^{\max}]$ *a time interval,* $\nu_m^{\min} \in \mathbb{N}$ *the minimum payload capacity for* m *and* θ_m *is a function determining the cost of under- or overcovering the mission demand compared to the desired frequency.*

We denote by $\mathcal{M}$ the set of missions. Since the possible trains are known a priori, for each $m \in \mathcal{M}$, the set $\mathcal{V}_m$ of trains which can cover the demand of mission m is given as an input. Train i covers mission m if $p_i = p_m$ and $t_i^+ \in [t_m^{\min}, t_m^{\max}]$. Mission demands are expressed as follows. During interval $[t_m^{\min}, t_m^{\max}]$, the number of chosen trains that cover m must be in $[f_m^{\min}, f_m^{\max}]$. Additionaly, the sum of the payload capacities of the train compositions used to operate them must be at least $\nu_m^{\min}$.

For each mission $m \in \mathcal{M}$, the following piecewise linear cost function $\theta_m : \mathbb{N} \to \mathbb{R}_+$ expresses the penalty associated to not covering m with the desired frequency f_m^c.

$$\theta_m(f) = \begin{cases} \pi_m^{+\infty}(f - f_m^{\max}) + \pi_m^+(f_m^{\max} - f_m^c) & \text{if } f \geq f_m^{\max} \\ \pi_m^+(f - f_m^c) & \text{if } f_m^c \leq f \leq f_m^{\max} \\ \pi_m^-(f_m^c - f) & \text{if } f_m^{\min} \leq f < f_m^c \\ \pi_m^{-\infty}(f_m^{\min} - f) + \pi_m^-(f_m^c - f_m^{\min}) & \text{if } f \leq f_m^{\min} \end{cases}$$

where $f_m^{\min} \in \mathbb{N}$, $f_m^c \in \mathbb{N}$, $f_m^{\max} \in \mathbb{N}$ are respectively the minimum, ideal, and maximum frequency for the mission, $\pi_m^+ \in \mathbb{R}_+$ and $\pi_m^- \in \mathbb{R}_+$ respectively the unit costs of over- and undercovering the demand, and $\pi_m^{-\infty} \in \mathbb{R}_+$, $\pi_m^{+\infty} \in \mathbb{R}_+$ respectively the unit costs of critical over- and undercovering the demand. The first and last cases respectively happen when the frequency is higher than the maximum or lower than the minimum allowed. In both

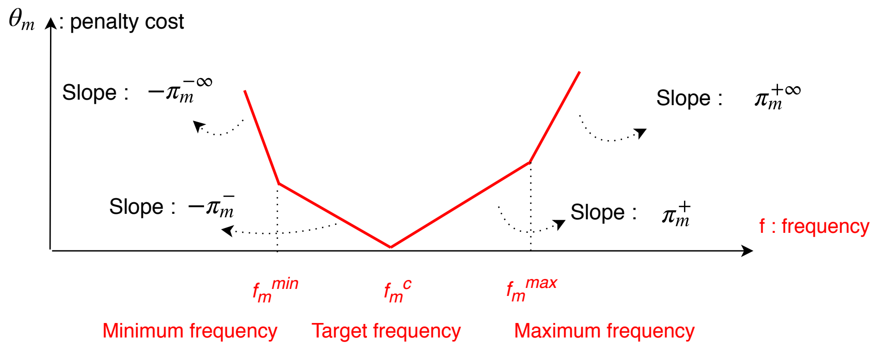

Figure 2: Representation of function θ_m, $m \in \mathcal{M}$

cases, a large cost is paid. The two middle cases correspond respectively to standard over- and under-covering of the demand. Such a function is depicted in Figure 2.
In some cases and for some political issues it is mandatory to operate some specific trains. The set of mandatory trains is denoted by $\mathcal{V}_f$.

3 A network flow formulation

3.1 Hypergraph description

In [2], the authors show that using hyperarcs allows to represent train compositions, coupling and decoupling operations in an efficient manner. We also use hypergraphs in our model: vertices correspond with logical/space-time states where decisions can be made, while hyperarcs correspond with railcar operations (technical moves or operating trains). One unit of flow corresponds with one or several train units from one place to another, possibly after rearrangement of their configuration.
Our directed hypergraph is defined by a couple of sets (V, A) where V is the set of nodes and A the set of hyperarcs. Each hyperarc $a \in A$ is defined by two multisets: $\mathcal{T}(a)$ is the tail of a and $\mathcal{H}(a)$ its head. In this paper, we denote by $\mathbf{card}(u, \mathcal{N})$ the multiplicity of element u in multiset $\mathcal{N}$, and by $\mathbf{support}(\mathcal{N})$ the set of unique elements in $\mathcal{N}$. The sum over a set Ξ of multisets is the multiset $\xi = \sum_{\mathcal{N} \in \Xi} \mathcal{N}$ such that $\mathbf{support}(\xi) = \cup_{\mathcal{N} \in \Xi} \mathbf{support}(\mathcal{N})$ and $\mathbf{card}(u, \xi) = \sum_{\mathcal{N} \in \Xi} \mathbf{card}(u, \mathcal{N})$. Each hyperarc a has a cost $q(a)$. The reader is referred to [5] for more definitions on hypergraphs.
A node is characterized by a tuple $(s, t, c, \mathcal{I})$, where s is a station, t a time, c a railcar composition, and $\mathcal{I}$ a set of trains. Basically, such a node corresponds with being in station s at time t with composition c, with the additional information that it corresponds with operating trains in $\mathcal{I}$. For each train $i \in V$, two train nodes are created for each possible composition c that can operate i: $(s_i^-, t_i^-, c, \{i\})$ and $(s_i^+, t_i^+, c', \{i\})$, where c' is the resulting composition if the train was operated with composition c. For each pair of trains (i, j) in $\mathcal{V}^-$, and each compatible composition c, a node $(s_i^+, t_i^+, c, \{i, j\})$ is created. Similarly, for each pair of trains (i, j) in V^+, and each compatible composition c, a node $(s_i^-, t_i^-, c, \{i, j\})$ is created.
There are also *depot nodes*, which are related to technical stays. For these special nodes, $c = (r)$ is composed of a single railcar type r, and $\mathcal{I} = \emptyset$. There is one depot node for each element of $\mathcal{S} \times \mathcal{T} \times \mathcal{R}$. In addition, two artificial vertices α and Ω are added to V to represent respectively the source and the sink of the network.
Hyperarcs are used to represent the decisions that can be made. We distinguish three types of hyperarcs: *train hyperarcs*, which represent the possibility of selecting trains with a specific composition, *connection hyperarcs*, to connect consecutive trains and/or rolling stock in the technical stations, and *technical hyperarcs*, to ensure flow conservation for each railcar type in the technical stations. Depending on the number of railcars used in the corresponding composition, each hyperarc can have one, two, or three elements in its tail/head. All hyperarcs have unit capacity except technical arcs, which are bounded by the quantity of available rolling stock.
Train hyperarcs are related to the main decision variables (*i.e.* train selection). A hyperarc can represent operating one train, or two trains when a coupling/decoupling operation is performed during the travel. When one

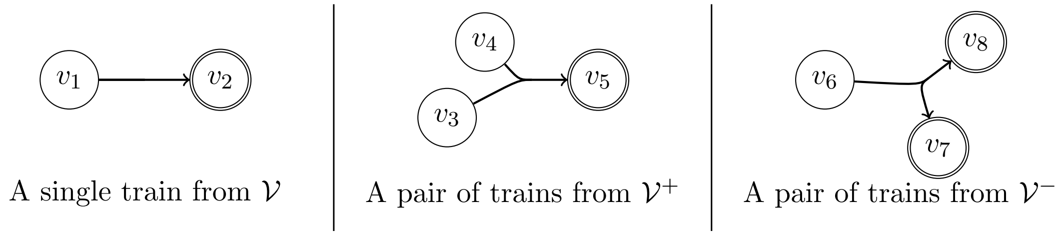

Figure 3: Double circled nodes are arrivals nodes and unique circled nodes are departure ones

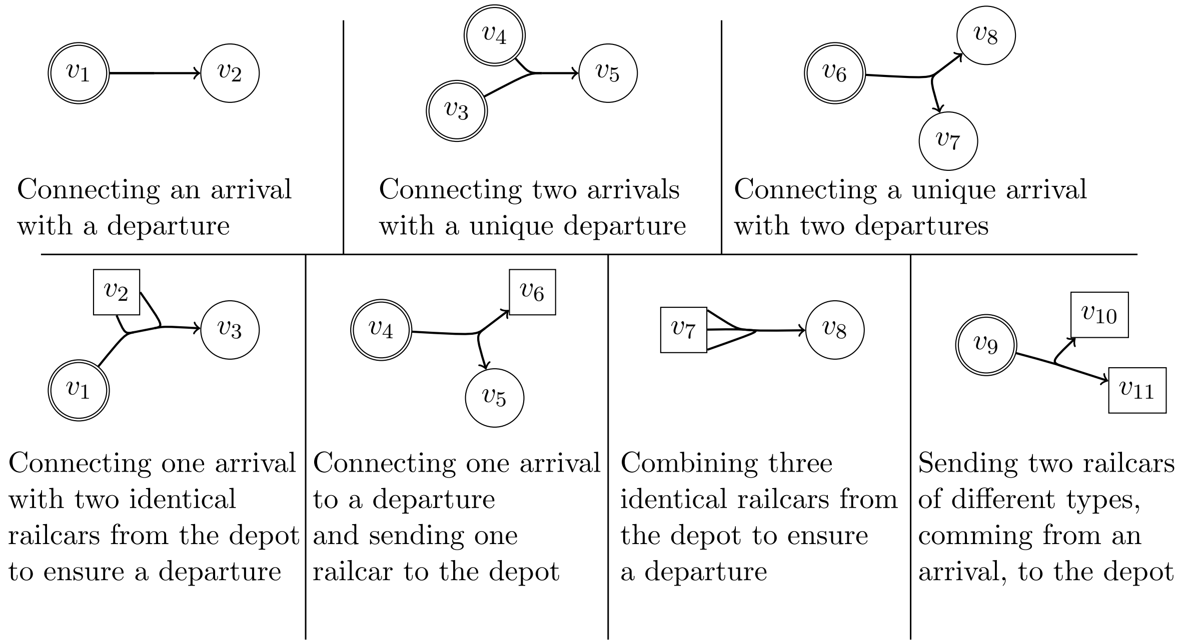

Figure 4: Square nodes are depots nodes, double circled nodes are arrivals nodes and unique circled nodes are departure ones

train is involved, train hyperarcs are created as follows. For each $i \in \mathcal{V}$, and each compatible composition c, there is a hyperarc a such that $\mathcal{T}(a) = \{\!\{v_i\}\!\}$ and $\mathcal{H}(a) = \{\!\{w_i\}\!\}$, where $v_i = (s^-_{p_i}, t^-_i, c, \{i\})$, $w_i = (s^+_{p_i}, t^+_i, c', \{i\})$, and c' is the arrival composition obtained if c is used to cover i. Note that c' can be either c or its symmetric in case of stub-end intermediate stops. The cost $q(a)$ is $\beta((\{i\}, c))$. Train hyperarcs also represent the selection of two trains i and j that are either coupled or decoupled. For all $(i, j) \in \mathcal{V}^+$ and each possible combination c' of compositions c_i and c_j, a hyperarc a is created, such that $\mathcal{T}(a) = \{\!\{v_i, v_j\}\!\}$, $\mathcal{H}(a) = \{\!\{w_{ij}\}\!\}$, where $v_i = (s^-_{p_i}, t^-_i, c_i, \{i\})$, $v_j = (s^-_{p_j}, t^-_j, c_j, \{j\})$, and $w_{ij} = (s^+_{p_i}, t^+_i, c', \{i, j\})$. Similarly, for each pair of trains $(i, j) \in \mathcal{V}^-$, a hyperarc a is created, with $\mathcal{T}(a) = \{\!\{v_{ij}\}\!\}$, $\mathcal{H}(a) = \{\!\{w_i, w_j\}\!\}$, $w_i = (s^+_{p_i}, t^+_i, c_i, \{i\})$, $w_j = (s^+_{p_j}, t^+_j, c_j, \{j\})$, $v_{ij} = (s^-_{p_i}, t^-_i, c', \{i, j\})$. In the two latter cases, $q(a) = \beta((\{i\}, c_i), (\{j\}, c_j), (\{i, j\}, c'))$. Figure 3 illustrates the different types of train hyperarcs.

Connection hyperarcs are used to connect consecutive trains and/or rolling stock. We aggregate all technical requirements for connection between operations in a single function, which is defined as follows. Possible connections are pairs of multisets (e_1, e_2) where e_1 and e_2 are multisets of V. We define a function γ, which takes a pair (e_1, e_2) in parameter and produces a positive real value if the connection is possible, $+\infty$ otherwise. Various technical constraints are embedded in the definition of γ, such as shunting (move to depot), maximum stay time at platforms τ_s, coupling/decoupling and minimum required time for connection. For each pair of multisets (e_1, e_2) (each of cardinality one, two, or three), there exists $a \in A$ with $\mathcal{H}(a) = e_1$ and $\mathcal{T}(a) = e_2$ if and only if $\gamma(e_1, e_2) < +\infty$. Figure 4 illustrates different types of connection hyperarcs.

Finally, **technical hyperarcs** are added to take into account the rolling stock constraints. For each depot node

v, there is a hyperarc connecting α to v. This arc is used to position vehicles at each technical station s at the beginning of the time horizon. Its cost is equal to $\psi(r)$. Additionally, for each $t = 0, \ldots, T-1$, there is a hyperarc of cost zero connecting $(s, t, (r), \emptyset)$ to $(s, t+1, (r), \emptyset)$ This arc represents the fact that railcars of type r stay at technical station s between time t and $t+1$. Finally, there is a hyperarc of cost zero connecting each depot node v to Ω. This arc is used to measure the quantity of vehicles of type r at each station at the end of the time horizon.

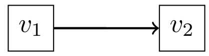

Figure 5: A technical hyperarc

To reduce the number of nodes in the model, we apply the following preprocessing method. Two nodes related to trains arriving at the same time, in the same station entrance, with the same composition, are merged. Moreover, if a vertex has only one predecessor and one successor, then it is removed, and the two corresponding hyperarcs are merged.

3.2 Mixed integer programming formulation

Based on hypergraph (V, A), we define the following MILP model of the problem, which will be referred to as **TSRP** in the sequel. For $i \in \mathcal{V}$, we denote by $\mathcal{A}(i) \subset A$ the subset of hyperarcs that are related to operating train i: a hyperarc $a \in \mathcal{A}(i)$ represents either i or a pair of trains belonging to $\mathcal{V}^- \cup \mathcal{V}^+$ and including i. For $i \in \mathcal{V}$, $a \in \mathcal{A}(i)$, we denote by $\mathcal{N}(a, i) = \mu(c)$, where c is the composition related to train i in hyperarc a, the number of seats represented by the train composition modeled by hyperarc a to cover train i. Also, we denote by $a^{\alpha}_{r,s}$ the hyperarc linking the source node with the first[1] depot node representing the train station $s \in \mathcal{S}$ and the railcar type $r \in \mathcal{R}$; likewise $a^{\Omega}_{r,s}$ is the hyperarc linking the last[1] depot node with the sink node.

We associate to each hyperarc $a \in A$ an integer variable denoted by z_a representing the flow handled by a. For all $v \in V$, we respectively denote by $\Gamma^+(v)$ and $\Gamma^-(v)$ multisets of outgoing and ingoing hyperarcs. For each mission $m \in \mathcal{M}$, we define four continuous variables y^-_m, y^+_m, $y^{-\infty}_m$ and $y^{+\infty}_m$ where $(y^-_m + y^{-\infty}_m)$ represents under frequency coverage and $(y^+_m + y^{+\infty}_m)$ represents over frequency coverage.

We report below our mixed-integer programming formulation for the **TSRP** problem.

[1]The first and the last depot nodes are seen with respect to an increasing hourly order of depot nodes for each train station and for each railcar type.

$$(\mathbf{TSRP}) : \min \ \gamma(z,y) = \sum_{a\in A} q(a) z_a + \sum_{m\in\mathcal{M}} (\pi_m^+ y_m^+ + \pi_m^{+\infty} y_m^{+\infty} + \pi_m^- y_m^- + \pi_m^{-\infty} y_m^{-\infty}) \quad (1)$$

$$\sum_{a\in\Gamma^+(v)} z_a = \sum_{a\in\Gamma^-(v)} z_a \qquad \forall v \in V \setminus \{\alpha\}, \{\Omega\} \quad (2)$$

$$\sum_{i\in\mathcal{V}_m} \sum_{a\in\mathcal{A}(i)} z_a + y_m^- + y_m^{-\infty} - y_m^+ - y_m^{+\infty} = f_m^c \qquad \forall m \in \mathcal{M} \quad (3)$$

$$y_m^+ \le f_m^{\max} - f_m^c \qquad \forall m \in \mathcal{M} \quad (4)$$

$$y_m^- \le f_m^c - f_m^{\min} \qquad \forall m \in \mathcal{M} \quad (5)$$

$$\sum_{a\in\mathcal{A}(i)} z_a = 1 \qquad \forall i \in \mathcal{V}_f \quad (6)$$

$$\sum_{i\in\mathcal{V}_m} \sum_{a\in\mathcal{A}(i)} \mathcal{N}(a,i) z_a \ge \nu_m^{\min} \qquad \forall m \in \mathcal{M} \quad (7)$$

$$\sum_{a\in\mathcal{A}(i)} z_a \le 1 \qquad \forall i \in \mathcal{V} \quad (8)$$

$$\sum_{a\in\mathcal{A}(i)\cup\mathcal{A}(j)} z_a \le 1 \qquad \forall i \in \mathcal{V}, \forall j \in \mathcal{E}_i \quad (9)$$

$$z_{a^\alpha_{r,s}} = z_{a^\Omega_{r,s}} \qquad \forall r \in \mathcal{R}, \forall s \in \mathcal{S} \quad (10)$$

$$n_{r,s}^{\min} \le z_{a^\alpha_{r,s}} \le n_{r,s}^{\max} \qquad \forall r \in \mathcal{R}, \forall s \in \mathcal{S} \quad (11)$$

$$y_m^+, y_m^-, y_m^{+\infty}, y_m^{-\infty} \in \mathbb{R}^+ \qquad \forall m \in \mathcal{M} \quad (12)$$

$$z_a \in \{0,1\} \qquad \forall a \in \cup_{i\in\mathcal{V}} \mathcal{A}(i) \quad (13)$$

$$z_a \in \mathbb{N} \qquad \forall a \in A \setminus \cup_{i\in\mathcal{V}} \mathcal{A}(i) \quad (14)$$

Objective function (1) minimizes the production cost and the penalty related to deviating from the desired demand covering (see figure 2). Constraints (2) represent the flow conservation. Constraints (3) express the demand covering, $(y_m^- + y_m^{-\infty})$ and $(y_m^+ + y_m^{+\infty})$ are respectively equal to the amount of frequency under- and over-covering. Constraints (6) impose that one hyperarc must be activated from a set of hyperarcs referring to the same train within $\mathcal{V}_f$. Constraints (7) impose a minimum number of seats that must be available for each mission. Constraints (4) and (5) bound the value of (y_m^-) and (y_m^+). Constraints (8) impose that at most one hyperarc can be activated from a set of hyperarcs referring to the same train. Constraints (9) avoid activating two hyperarcs representing two conflicting trains. Constraints (10) impose for each train station s and for each railcar type r the number of railcars starting the service at s is equal to the number of railcars ending the service at s. Constraints (11) impose that the number of railcars, for each type, beginning the service in a specific station s must not exceed $n_{r,s}^{\max}$ and must not be lesser than $n_{r,s}^{\min}$.

4 Solving approach

Our preliminary computational experiments have shown that model (1)–(14) can be solved efficiently for real-life instances when only one type of railcars is considered. However real instances include several types (typically three), and including them in the model increases by a wide range the computing time needed to solve the model.

Our method relies on an initial relaxation of the problem, which is refined in a second step to include the totality of the problem. In the relaxation we use, we consider a smaller number of railcar types. The goal of this first relaxation is twofold: it allows us to eliminate a large number of hyperarcs using reduced-cost fixing techniques, and it is used in a heuristic method to compute a first feasible solution.

Our relaxation is defined by an aggregation of the network: all nodes related with the same station, time, number and class of railcars are aggregated.

This type of aggregation is tighly related to the method used in [1]. The authors use simultaneously three models with different levels of precision, called *layers*. In the most precise model, which solves the complete problem, arcs and hyperarcs are considered, and the orientation and position of each railcar in the configuration is recorded. The intermediate model disregard the position and the orientation of the railcars. The less precise model does not consider hyperarcs anymore, which leads to a minimum-cost flow problem. The three layers are used in a column-generation method, where constraints of the fine level are kept in the master problem, and coarse levels are used to price out new columns. This decomposition into layers is well adapted to the version of the problem of [2]: the coarse problem is easy to solve and produces a good lower bound on the cost of operating railcars. The intermediate layer allows to produce feasible parts of solutions to be added to the master problem.
The same idea is used differently in our work. We also identified the fact that solving the precise model directly is computationally challenging, and removing fine grained constraints does not hurt too much in terms of objective function. However, two main differences between our problem and the one solved by [2] explain why we used another variant of aggregation. First, since we have to solve the train selection and the rolling stock problems simultaneously, the analog of the coarser relaxation of [1] would remain an NP-hard minimum cost flow problem with additional demand covering constraints. Second, we do not consider maintenance constraints and thus our flow problem is easier to solve even when hyperarcs are considered.

4.1 The relaxed problem

Our relaxed problem is defined by an aggregation of the set of railcar types, which cascades down to an aggregation of the vertices and the hyperarcs. In the model, each set of aggregated vertices/hyperarcs is represented by a representative, single element. Figure 6 illustrates the successive aggregations.
Let $\mathcal{P}$ be a partition of set $\mathcal{R}$, with $\mathcal{P} = \{\mathcal{R}_1, \ldots, \mathcal{R}_{|\mathcal{P}|}\}$. For all railcar types $r_1 \in \mathcal{R}$, we denote its part – or *aggregate* – by $\mathbf{agg}^r_{\mathcal{P}}(r_1) = \mathcal{R}_i$ such that $r_1 \in \mathcal{R}_i$.

- **Train configurations.** The aggregate of a train configuration is the set of configurations with the same aggregated railcar types. The aggregate of $c = (r_i)_{i \in [\![1,\mathrm{len}(c)]\!]} \in \mathcal{C}$ is $\mathbf{agg}^c_{\mathcal{P}}(c) = \{(r'_i)_{i \in [\![1,\mathrm{len}(c)]\!]} \in \mathcal{C} : (\mathbf{agg}^r_{\mathcal{P}}(r_i))_i = (\mathbf{agg}^r_{\mathcal{P}}(r'_i))_i\}$.
- **Vertices.** The aggregate of a vertex $u = (s, t, c, \mathcal{I}) \in V$ is defined as the set of vertices sharing the same characteristics with u except for their configurations, which share the same aggregate: $\mathbf{agg}^v_{\mathcal{P}}(u) = \{(s, t, c', \mathcal{I}) \in V : c' \in \mathbf{agg}^c_{\mathcal{P}}(c)\} \subseteq V$. The aggregate of α and Ω are defined respectively by $\mathbf{agg}^v_{\mathcal{P}}(\alpha) = \{\alpha\}$ and $\mathbf{agg}^v_{\mathcal{P}}(\Omega) = \{\Omega\}$.
- **Hyperarcs.** In order to define the aggregation of the hyperarcs, we need to introduce the minimal set of vertex aggregates that covers a set U of vertices: $\mathbf{agg}^V_{\mathcal{P}}(U) = \{\mathbf{agg}^v_{\mathcal{P}}(u) : u \in U\}$. We have to define an aggregate cardinality function as well: given a multiset $\mathcal{N}$ and a set of vertices U, $\overline{\mathbf{card}}_{\mathcal{N}}(U) = \sum_{u \in U} \mathbf{card}(u, \mathcal{N})$ is the cumulated cardinality over elements of U in multiset $\mathcal{N}$. The aggregate of a multiset is then: $\mathbf{agg}^{\mathsf{M}}_{\mathcal{P}}(\mathcal{N}) = (\mathbf{agg}^V_{\mathcal{P}}(\mathbf{support}(\mathcal{N})), \overline{\mathbf{card}}_{\mathcal{N}})$. The aggregate of a hyperarc $b \in A$ is thus $\mathbf{agg}^a_{\mathcal{P}}(b) = \{b' \in A : \mathbf{agg}^{\mathsf{M}}_{\mathcal{P}}(\mathcal{T}(b')) = \mathbf{agg}^{\mathsf{M}}_{\mathcal{P}}(\mathcal{T}(b)), \mathbf{agg}^{\mathsf{M}}_{\mathcal{P}}(\mathcal{H}(b')) = \mathbf{agg}^{\mathsf{M}}_{\mathcal{P}}(\mathcal{H}(b))\}$.

We produce a relaxation of our problem by replacing the vertices of our hypergraph model with vertex aggregates, and the hyperarcs with hyperarc aggregates. Let $\mathcal{P}_V = \{\mathbf{agg}^v_{\mathcal{P}}(u) : u \in V\}$ and $\mathcal{P}_A = \{\mathbf{agg}^a_{\mathcal{P}}(a) : a \in A\}$ be the aggregates of V and A. This results in a smaller size hypergraph model, but requires to relax some constraints and the objective function. Formally, let us introduce new variables $X_F = \sum_{a \in F} z_a$, $F \in \mathcal{P}_A$, which correspond with the total flow on hyperarcs in each aggregate F. For each railcar aggregate $\mathcal{R} \in \mathcal{P}$ and each train station $s \in \mathcal{S}$, we denote by $F^{\alpha}_{R,s} = \mathbf{agg}^a_{\mathcal{P}}(a^{\alpha}_{r,s})$ and $F^{\Omega}_{R,s} = \mathbf{agg}^a_{\mathcal{P}}(a^{\Omega}_{r,s})$ with r an arbitrary element of $\mathcal{R}$. A mixed integer linear programming model for the relaxed problem is obtained from (1)-(14) by replacing z-variables with X-variables, partially relaxing some constraints in the process. This yields the following MILP model of our relaxed problem, which we call **(ATSRP)**.

$$\begin{aligned}\textbf{(ATSRP)} : \min\ \bar{\gamma}(X, y) = &\sum_{F \in P_A} \min_{b \in F}\{q(b)\} X_F \\ &+ \sum_{m \in \mathcal{M}} (\pi^+_m y^+_m + \pi^{+\infty}_m y^{+\infty}_m + \pi^-_m y^-_m + \pi^{-\infty}_m y^{-\infty}_m) \qquad (15)\end{aligned}$$

$$\sum_{F\in\mathcal{P}_A}\max_{a\in F}\left\{\overline{\mathbf{card}}_{\mathcal{T}(a)}(W)\right\}X_F=\sum_{F\in\mathcal{P}_A}\max_{a\in F}\left\{\overline{\mathbf{card}}_{\mathcal{H}(a)}(W)\right\}X_F \qquad \forall W\in\mathcal{P}_V \tag{16}$$

$$-y_m^+-y_m^{+\infty}+y_m^-+y_m^{-\infty}+\sum_{i\in\mathcal{V}_m}\sum_{F\in\mathcal{P}_A:\mathcal{A}(i)\cap F\neq\emptyset}X_F=f_m^{c_m} \qquad \forall m\in\mathcal{M} \tag{17}$$

$$y_m^+\leq f_m^{\max}-f_m^c \qquad \forall m\in\mathcal{M} \tag{18}$$

$$y_m^-\leq f_m^c-f_m^{\min} \qquad \forall m\in\mathcal{M} \tag{19}$$

$$\sum_{F\in\mathcal{P}_A:\mathcal{A}(i)\cap F\neq\emptyset}X_F=1 \qquad i\in\mathcal{V}_f \tag{20}$$

$$\sum_{F\in\mathcal{P}_A:\mathcal{A}(i)\cap F\neq\emptyset}\max_{b\in F}\left\{\mathcal{N}(b,i)\right\}X_F\geq\nu_m^{\min} \qquad \forall m\in\mathcal{M} \tag{21}$$

$$\sum_{F\in\mathcal{P}_A:\mathcal{A}(i)\cap F\neq\emptyset}X_F\leq 1 \qquad i\in\mathcal{V} \tag{22}$$

$$\sum_{F\in\mathcal{P}_A:(\mathcal{A}(i)\cup\mathcal{A}(j))\cap F\neq\emptyset}X_F\leq 1 \qquad \forall i\in\mathcal{V},\forall j\in\mathcal{E}_i \tag{23}$$

$$X_{F^{\alpha}_{R,s}}=X_{F^{\Omega}_{R,s}} \qquad \forall R\in\mathcal{P},s\in\mathcal{S} \tag{24}$$

$$\sum_{r\in R}n_{r,s}^{\min}\leq X_{F^{\alpha}_{R,s}}\leq\sum_{r\in R}n_{r,s}^{\max} \qquad \forall R\in\mathcal{P} \tag{25}$$

$$y_m^+,y_m^-,y_m^{+\infty},y_m^{-\infty}\in\mathbb{R}^+ \qquad \forall m\in\mathcal{M} \tag{26}$$

$$X_F\in\{0,1\}\ \forall F\in\mathcal{P}_A:F\cap A^{\mathcal{V}}\neq\emptyset,\quad X_F\in\mathbb{N}\ \forall F\in\mathcal{P}_A:F\cap A^{\mathcal{V}}=\emptyset \tag{27}$$

Proposition 1. *Model **(ATSRP)** is a relaxation of Model **(TSRP)**.*

Proof. We first need an intermediate result to perform a variable substitution in constraints (3), (6), (8) and (9). Since $\mathcal{P}_A$ is a partition of A, $\sum_{a\in\mathcal{A}(i)}z_a=\sum_{F\in\mathcal{P}_A}\sum_{a\in\mathcal{A}(i)\cap F}z_a$. Besides, for all $F\in\mathcal{P}_A$ and $a,b\in F$, we have by definition of $\mathcal{P}_A$ that $\mathbf{agg}^V_{\mathcal{P}}(\mathcal{H}(a))=\mathbf{agg}^V_{\mathcal{P}}(\mathcal{H}(b))$ and $\mathbf{agg}^V_{\mathcal{P}}(\mathcal{T}(a))=\mathbf{agg}^V_{\mathcal{P}}(\mathcal{T}(b))$. Hence, $a\in\mathcal{A}(i)$ if and only if $b\in\mathcal{A}(i)$, so that either $\mathcal{A}(i)\cap F=F$ or $\mathcal{A}(i)\cap F=\emptyset$ and (28) follows.

$$\text{For all } i\in\mathcal{V},\ \sum_{a\in\mathcal{A}(i)}z_a=\sum_{F\in\mathcal{P}_A:\mathcal{A}(i)\cap F\neq\emptyset}X_F \tag{28}$$

We also need to express the flow on hyperarcs going out of or into a vertex aggregate in terms of those new variables.

$$\sum_{v\in W}\sum_{a\in\Gamma^+(v)}z_a=\sum_{v\in W}\sum_{a\in A}\mathbf{card}(v,\mathcal{T}(a))z_a \tag{29}$$

$$=\sum_{F\in P_A}\sum_{a\in F}\sum_{v\in W}\mathbf{card}(v,\mathcal{T}(a))z_a \tag{30}$$

$$=\sum_{F\in\mathcal{P}_A}\max_{a\in F}\left\{\overline{\mathbf{card}}_{\mathcal{T}(a)}(W)\right\}X_F \tag{31}$$

Equality (29) follows from the definition of Γ^+. The second line (30) is deduced from the fact that $\mathcal{P}_A$ is a partition of A, while (31) comes from the definitions of X_F and $\overline{\mathbf{card}}_{\mathcal{T}(a)}$. Remark that for all $a\in F$, the values of $\overline{\mathbf{card}}_{\mathcal{T}(a)}(W)$ are the same and are thus equal to the maximum of them. Note that the choice of the maximum over other possibilities such as the minimum is arbitrary.
Similarly, we can prove the following relation:

$$\text{For all } W\in\mathcal{P}_V,\ \sum_{v\in W}\sum_{a\in\Gamma^-(v)}z_a=\sum_{F\in\mathcal{P}_A}\overline{\mathbf{card}}_{\mathcal{H}(a),a\in F}(W)X_F \tag{32}$$

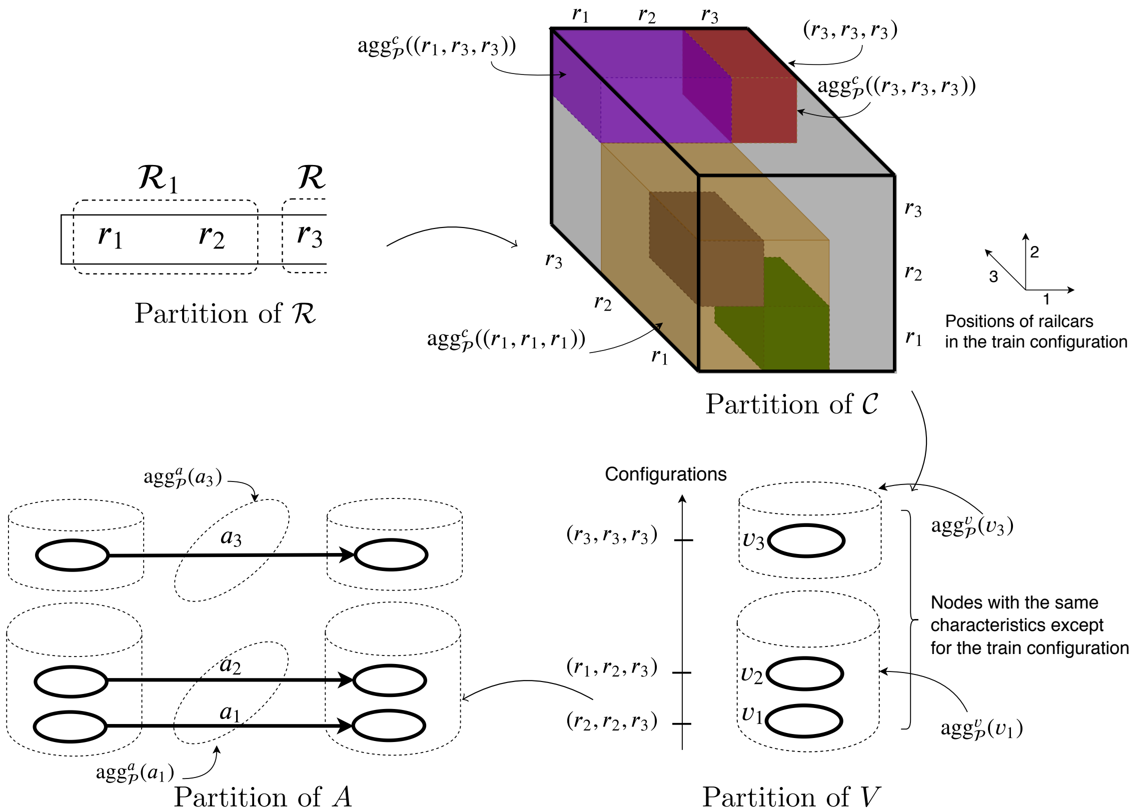

Figure 6: Steps leading to the hypergraph aggregation

Constraints (2) are relaxed into surrogate constraints; more precisely, they are replaced with their sums over each vertex aggregate:

$$(2) \Rightarrow \sum_{v \in W} \sum_{a \in \Gamma^+(v)} z_a = \sum_{v \in W} \sum_{a \in \Gamma^-(v)} z_a \qquad \forall W \in \mathcal{P}_V \setminus \{\{\alpha\}, \{\Omega\}\}$$

$$\Leftrightarrow \sum_{F \in \mathcal{P}_A} \overline{\mathbf{card}}_{\mathcal{T}(a), a \in F}(W) X_F = \sum_{F \in \mathcal{P}_A} \overline{\mathbf{card}}_{\mathcal{H}(a), a \in F}(W) X_F \qquad \forall W \in \mathcal{P}_V \setminus \{\{\alpha\}, \{\Omega\}\} \text{ (from (31)–(32))}$$

Based on the definition of $\mathcal{P}_A$, we can also make the following observation.

$$\text{For all } R' \in \mathcal{P}, \text{ for all } s \in \mathcal{S}, \ \{a^\alpha_{r,s}, r \in R'\} \in \mathcal{P}_A \text{ and } \{a^\Omega_{r,s}, r \in R'\} \in \mathcal{P}_A. \tag{33}$$

Following the same idea, constraints (10) and (11) are summed up over each railcar type aggregate:

$$(10) \Rightarrow \sum_{r \in R} z_{a^\alpha_{r,s}} = \sum_{r \in R} z_{a^\Omega_{r,s}} \qquad \forall R \in \mathcal{P}, s \in \mathcal{S}$$

$$\Leftrightarrow X_{F^\alpha_{R,s}} = X_{F^\Omega_{R,s}} \qquad \forall R \in \mathcal{P}, s \in \mathcal{S}$$

$$(11) \Rightarrow \sum_{r \in R} n^{\min}_{r,s} \leq \sum_{r \in R} z_{a^\alpha_{r,s}} \leq \sum_{r \in R} n^{\max}_{r,s} \qquad \forall R \in \mathcal{P}$$

$$\Leftrightarrow \sum_{r \in R} n^{\min}_{r,s} \leq X_{F^\alpha_{R,s}} \leq \sum_{r \in R} n^{\max}_{r,s} \qquad \forall R \in \mathcal{P}$$

$$\text{with } F^\alpha_{R,s} = \mathbf{agg}^a_{\mathcal{P}}(a^\alpha_{r,s}) \text{ and } F^\Omega_{R,s} = \mathbf{agg}^a_{\mathcal{P}}(a^\Omega_{r,s}) \qquad \text{for some arbitrary } r \in R, \text{ for all } R \in \mathcal{P} \text{ (from (33))}$$

Constraints (7) are relaxed in the following way:

$$(7) \Rightarrow \sum_{i \in \mathcal{V}_m} \sum_{e \in \mathcal{A}(i)} \left\{ \max \mathcal{N}(b,i) : b \in \mathbf{agg}^a_{\mathcal{P}}(e) \right\} z_e \geq \nu_m^{\min} \qquad \forall m \in \mathcal{M}$$

$$\Leftrightarrow \sum_{F \in \mathcal{P}_A : \mathcal{A}(i) \cap F \neq \emptyset} \left\{ \max \mathcal{N}(b,i) : b \in F \right\} X_F \geq \nu_m^{\min} \qquad \forall m \in \mathcal{M} \text{ (from (28))}$$

Note that disjunctive constraints (22) induce binary bounds on variables X_F with $F \in \mathcal{P}_A$ such that $F \cap A^{\mathcal{V}} \neq \emptyset$ in constraints (27).
We now consider the objective function.

$$\sum_{a \in A} q(a) z_a \geq \sum_{F \in P_A} \sum_{a \in F} \min\{q(b) : b \in F\} z_a = \sum_{F \in P_A} \min\{q(b) : b \in F\} X_F$$

We have shown that any solution for the initial model is a solution of the new model, and that for any solution, the new objective function is less than the initial one. Therefore we obtain an relaxation. □ □

The **ATSRP** and **TSRP** models have the same types of constraints. Therefore, the **ATSRP** problem is seen as a **TSRP** problem based on different data inputs. We show that it is the case by defining from the **ATSRP** the new artificial railcar types (denoted by $\mathcal{R}^a$) and train configurations (denoted by $\mathcal{C}^a$). Each $\mathcal{R}' \in \mathcal{P}_{\mathcal{R}}$ defines a new railcar type $r' \in \mathcal{R}^a$ equipped with the most permissive characteristics among the types of $\mathcal{R}'$; the set $\mathcal{C}^a$ is defined likewise based on $\mathcal{P}_{\mathcal{C}}$. Solving the **ATSRP** is then equivalent to solving a **TSRP** defined by $\mathcal{R}^a$ and $\mathcal{C}^a$.

4.2 Our iterative algorithm

Our solving approach, sketched in Algorithm 1 has three main steps: solving a relaxation, computing a feasible solution, and solving exactly the problem using information obtained in the two first phases. Lines 1, 2 and 3 correspond with defining and solving the linear relaxation of the **TSRP** model with inputs $(\mathcal{R}^a, \mathcal{C}^a, \mathcal{V}, \mathcal{V}^+, \mathcal{V}^-, \mathcal{M})$; λ^* is the value of the linear relaxation. We then compute in line 4 and 5 the value σ of a feasible solution by first solving the relaxed model to optimality. We extract from this solution the subset of trains used $(\mathcal{J}, \mathcal{J}^+, \mathcal{J}^-)$ (such that $\mathcal{J} \subseteq \mathcal{V}$, $\mathcal{J}^+ \subseteq \mathcal{V}^+$ and $\mathcal{J}^- \subseteq \mathcal{V}^-$). Second, we solve in line 5 the **TSRP** restricted to that subset of trains, *i.e.* with inputs $(\mathcal{R}, \mathcal{C}, \mathcal{J}, \mathcal{J}^+, \mathcal{J}^-, \mathcal{M})$. We then remove all hyperarcs of the relaxed model by applying the reduced cost-based variable fixing technique described in Proposition 2. We finally solve the initial **TSRP** model restricted to the remaining hyperarcs.

Algorithm 1: The iterative algorithm based on the reduced cost variable fixing procedure

Data: $(\mathcal{R}, \mathcal{C}, \mathcal{V}, \mathcal{V}^+, \mathcal{V}^-, \mathcal{M})$

1 $\mathcal{P} \leftarrow$ Determine an initial aggregation of the railcar types ;
2 $(\mathcal{R}^a, \mathcal{C}^a) \leftarrow$ Compute the artificial railcar types and train configurations based on $\mathcal{P}$;
3 $\lambda^* \leftarrow$ Compute the value of the linear relaxation of the **TSRP**$(\mathcal{R}^a, \mathcal{C}^a, \mathcal{V}, \mathcal{V}^+, \mathcal{V}^-, \mathcal{M})$;
4 $(\mathcal{J}, \mathcal{J}^+, \mathcal{J}^-) \leftarrow$ Solve the **TSRP**$(\mathcal{R}^a, \mathcal{C}^a, \mathcal{V}, \mathcal{V}^+, \mathcal{V}^-, \mathcal{M})$ and compute the selected trains;
5 $\sigma \leftarrow$ Compute an upper bound by solving the **TSRP**$(\mathcal{R}, \mathcal{C}, \mathcal{J}, \mathcal{J}^+, \mathcal{J}^-, \mathcal{M})$;
6 Remove all hyperarcs of the **TSRP**$(\mathcal{R}^a, \mathcal{C}^a, \mathcal{V}, \mathcal{V}^+, \mathcal{V}^-, \mathcal{M})$ whose reduced cost is larger than $(\sigma - \lambda^*)$;
7 Generate the filtered original **TSRP**$(\mathcal{R}, \mathcal{C}, \mathcal{V}, \mathcal{V}^+, \mathcal{V}^-, \mathcal{M})$;
8 Solve the filtered original **TSRP**$(\mathcal{R}, \mathcal{C}, \mathcal{V}, \mathcal{V}^+, \mathcal{V}^-, \mathcal{M})$;

We now show that hyperarcs of the original model whose variable has been removed by the reduced-cost fixing (line 6) in the aggregated model can be removed in the original model. By construction, model **(ATSRP)** is a relaxation of model **(TSRP)** and the model variables are linked by the following property, which is ensured by the constraint-wise process used in the previous section to build **(ATSRP)**.

Proposition 2. *Let **(RATSRP)** be the linear relaxation of **(ATSRP)**, and let σ be an upper bound on the optimum of **(TSRP)**, (X^*, y^*) be an optimal solution of **(RATSRP)**, λ^* its cost, and $\tilde{c}_F$ be the reduced cost of a variable X_F such that $X_F^* = 0$. If $\lambda^* + \tilde{c}_F > \sigma$ then for all solutions of **(TSRP)** with a cost smaller than or equal σ, we have $z_a = 0$ for all $a \in F$.*

Proof. Let $F \in \mathcal{P}_A$ such that $\lambda^* + \tilde{c}_F > \sigma$ and consider a feasible solution (z, y) of **TSRP** such that $z_a \geq 1$ for some $a \in F$.
From the proof of Proposition 1, (X, y) with $X_G = \sum_{a \in G} z_a$ for all $G \in \mathcal{P}_A$ is a feasible solution to **ATSRP** and $\gamma(z, y) \geq \bar{\gamma}(X, y)$. Moreover, $X_F = \sum_{a \in F} z_a \geq 1$, so that $\bar{\gamma}(X, y) > \sigma$ from standard reduced cost-based variable fixing result. It follows that $\gamma(z, y) > \sigma$. Finally, $\hat{z}_a = 0$ for any solution $(\hat{z}, \hat{y})$ of **TSRP** of cost less than σ. □

5 Computational experiments

We present in this section our experimental results carried out on real life instances. The objective of our experiments is twofold: *(i)* to evaluate the quality of our formulation and *(ii)* to compare the filtering based algorithm and the initial one. All experiments were run using a 2.5 Ghz Haswell Intel Xeon E5-2680 with 128Go of RAM. The MIPs are solved using the commercial MILP solver IBM Ilog Cplex 12.6.

Instances are distinguished by the size of the set of trains, the set of mission demands, the set of railcar types and the set of compositions. We also study parameter $\tau_s, s \in \mathcal{S}$, which reflects the maximum stay duration alongside a train station platform. This last one impacts the size of the hypergraph. We recall that flow units are merged in depot nodes depicting a move of railcars to depot areas. τ_s is set to around an hour for instances with major train stations having several train departures and arrivals. It is set to around two hours for small train stations with restrained trips. The time period of the study is equal to one day for all instances and the size of $\mathcal{R}$ is bounded by three, in accordance with current practice of the French operator over a specific part of the network. Some trains have a huge demand of payload capacity, we therefore allow train compositions with up to three combined railcars. These compositions are not desired as they are practically difficult to manage, so we penalize them in the objective function. We measure the percentage of these compositions in $\mathcal{C}$ and denote it by %MU3. Table 1 summarizes the instances in our testbed. For each instance, we generated some variants, named in column *Data*, by changing parameters $\mathcal{R}$, $\mathcal{C}$ and τ. Column *TP* represents the number of train patterns and $\tau = \max_{s \in \mathcal{S}}(\tau_s)$ is expressed in minutes.

In the following, the gap between a lower bound l and an upper bound p is denoted by Gap(l, p) and is equal to $100 * \frac{|l-p|}{|p|}$. The speedup between an initial CPU time i and a new one n is equal to $100 * \frac{i-n}{i}$ and is denoted by S_{CPU}, the speedup for the hypergraph generation phase is denoted by S_{build}. To evaluate the impact of operating train compositions with three railcars, we vary the fraction ξ of the cost of those train compositions over the penalty π_m^- related to the demand covering.

5.1 Solving TSRP directly

Table 2 shows the results of model **TSRP** (subsection 3.2) using using a general purpose commercial MILP solver. In columns **#Ctrs** and **#Vars**, we respectively report the number of constraints and variables in the model. Column **CPU1** represents the overall amount of time in seconds needed to solve the **TSRP** and **%CPU Build** represents the percentage amount of time needed to generate the hypergraph model. Columns **Opt** and **LP** respectively contain the optimum and the linear relaxation value for the problem. Here, the value of ξ is set to 0.7.
Although the models have up to 2 000 000 variables, the quality of the linear relaxation to allow a fast resolution. All instances are solved to optimality (within the solver's default tolerance equal to 10^{-4}). The hardest instance is solved in 32 hours, while 15 out of 20 of them are solved within one hour. Compared to the overall computing time, the percentage of hypergraph generation time is small except for small instance I7 and instance I5.

Table 1: Characteristics of instances

Instance	Data	TP	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{V}^{+}\rvert + \lvert\mathcal{V}^{-}\rvert$	τ	$\lvert\mathcal{M}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{C}\rvert$	%MU3
I1	D1	30	426	0	50	63	2	14	57
	D2	30	426	0	50	63	3	47	70
I2	D3	31	620	0	50	175	2	14	57
	D4	31	620	0	50	175	3	47	70
I3	D5	40	576	118	50	93	2	14	57
	D6	40	576	118	70	93	2	14	57
	D7	40	576	118	50	93	3	47	70
I4	D8	44	644	152	30	129	2	14	57
	D9	44	644	152	70	129	2	14	57
	D10	44	644	152	70	129	3	47	70
I5	D11	24	480	0	70	64	2	22	64
	D12	24	480	0	70	64	3	47	70
I6	D13	23	287	0	60	107	2	22	64
	D14	23	287	0	90	107	2	22	64
	D15	23	287	0	90	107	3	47	70
I7	D16	10	200	0	120	60	2	14	57
	D17	10	200	0	120	60	3	47	70
I8	D18	86	1462	0	50	120	3	47	70
	D19	86	1462	0	50	120	3	9	33
	D20	86	1462	0	50	120	3	6	0

Table 2: Solving directly the initial **TSRP**

Instance	Data	**#Ctrs**	**#Vars**	**CPU1**	**%CPU Build**	**Gap(LP, Opt)**
I1	D1	13211	139255	924	0,54	4,57
	D2	38814	530663	12161	0,62	4,38
I2	D3	19175	463959	29361	0,17	5,31
	D4	44655	1190217	79316	0,41	5,32
I3	D5	17557	196889	1024	0,98	42,58
	D6	17557	310065	3787	0,61	42,71
	D7	52088	733240	15939	0,80	42,58
I4	D8	19759	114598	821	0,49	1,51
	D9	19759	387908	3968	0,91	1,49
	D10	58795	1518731	112967	0,47	1,50
I5	D11	20904	846961	425	39,06	2,03
	D12	43204	2102872	2984	34,01	6,73
I6	D13	10178	145421	80	7,50	13,23
	D14	10178	247017	200	8,00	13,06
	D15	22214	690238	2284	4,90	9,64
I7	D16	4648	80661	9	22,22	4,63
	D17	19888	729407	621	24,32	0,98
I8	D18	70309	2245103	3474	36,87	0,45
	D19	22501	426573	1165	4,12	0,65
	D20	16633	168505	540	1,48	0,65

Table 3: Comparing the direct solving and the filtering procedure with ξ from $\{6, 8, 10, 15\}$

Data	$\overline{\text{Gap(Opt, Primal)}}$	$\overline{\text{Gap(LPA, Primal)}}$	$\overline{\text{Gap(LPA, Opt)}}$	$\overline{\%\text{Gen}}$	$\overline{\text{CPU2}}$	$\overline{S_{\text{build}}(d_{\text{build}})}$	$\overline{S_{\text{CPU}}}$
D1	0,19	5,22	5,04	49,35	111,25	61,16 (3,22)	84,35
D2	0,16	5,00	4,85	41,60	1024,00	71,17 (49,09)	90,07
D3	0,26	6,40	6,16	58,46	5185,25	51,07 (26,15)	72,18
D4	0,25	6,40	6,16	55,75	15152,75	52,02 (171,85)	83,21
D5	**0,01**	43,01	43,00	**99,90**	241,75	-5,25 (-0,51)	**77,05**
D6	**0,02**	43,15	43,14	**100,00**	525,75	-1,95 (-0,45)	**76,31**
D7	**0,01**	43,01	43,00	**99,90**	1660,75	-1,13 (-1,48)	**82,69**
D8	0,02	1,82	1,80	27,79	123,50	79,78 (2,93)	79,47
D9	0,02	1,79	1,77	24,81	129,25	92,01 (32,71)	97,39
D10	0,02	1,79	1,76	17,59	858,50	96,37 (530,70)	99,24
D11	16,95	54,98	45,79	18,26	154,00	96,17 (160,10)	71,57
D12	24,44	45,21	27,48	14,67	1133,50	97,68 (990,99)	82,25
D13	13,65	44,35	35,55	37,72	56,00	75,78 (4,49)	48,88
D14	8,48	41,35	35,91	20,77	58,75	94,17 (15,02)	86,43
D15	7,57	25,97	19,90	16,49	317,50	96,84 (126,89)	87,37
D16	**33,57**	57,77	**36,42**	26,88	8,50	77,04 (1,41)	**-3,47**
D17	0,86	11,99	11,22	15,19	136,50	97,35 (124,40)	81,86
D18	2,12	17,33	15,54	18,93	2692,75	96,25 (1244,08)	24,66
D19	1,48	5,93	4,52	39,33	703,00	82,67 (41,00)	27,76
D20	1,45	5,90	4,52	99,64	974,75	-3,84 (-0,31)	**-2,53**

5.2 Reduced cost filtering procedure

We now compare the reduced cost based filtering procedure with the direct solution. This time we tested instances of table 2 with different values of ξ between 0.7 and 15. Columns of table 3 and table 4 represent average values obtained by fixing ξ from the set $\{6, 8, 10, 15\}$ for table 3 and fixing it from the set $\{0.7, 1, 2, 4\}$ for table 4; the over-lines stands for averages. Basically, table 3 represents the results where we significantly penalize the use of train compositions with three railcars. In both tables, **CPU2** represents the overall computing time in seconds needed to solve **TSRP** using algorithm 1. **%Gen** represents the percentage of the generated hyperarcs compared with **#Vars**. **LPA** is the value of the linear relaxation of the aggregated problem, **Primal** is the value obtained using the primal heuristic presented in the algorithm 1, S_{CPU} and S_{build} are the speedup of the overall computing time and the build time. d_{build} is the time difference between the initial build and the build of the filtered hypergraph.

Overall, we managed to speed up consequently the solution time, with quite better results when compositions with three railcars are more penalized (Table 3). This is explained by the large part of hyperarcs that are filtered in this case. For instance I3 (D5, D6 and D7) we speed up the solution time even if we generated the whole initial hypergraph. This is explained by the tight value of the primal solution. However, almost no variable is filtered because of the poor linear relaxation obtained for in this case. Also, data sets D16 and D20 draw no benefit from our algorithm. For D16, the quality of both the primal solution and the linear relaxation are poor. For D20, the number of possible configurations is very small, leading to a small hypergraph, which is not reduced further by the filtering algorithm. Note that the amount of time needed to solve them is already small, 8 seconds for D16 and 406 seconds for D20. Considering the primal heuristic, good solutions are obtained for most of the instances except for D16 where the fixed set of trains obtained in the aggregated model leads to a surplus of two railcars.

Table 4: Comparing the direct solving and filtering procedure with ξ from $\{0.7, 1, 2, 4\}$

Data	$\overline{\text{Gap(Opt, Primal)}}$	$\overline{\text{Gap(LPA, Primal)}}$	$\overline{\text{Gap(LPA, Opt)}}$	$\overline{\%\text{Gen}}$	$\overline{\text{CPU2}}$	$\overline{S_{\mathsf{build}}(d_{\mathsf{build}})}$	$\overline{S_{\mathsf{CPU}}}$
D1	0,19	5,21	5,04	98,39	221,75	-3,82 (-0,20)	74,89
D2	0,14	4,98	4,85	98,41	980,50	0,05 (0,04)	90,33
D3	0,25	6,65	6,41	97,33	2812,50	4,01 (2,06)	85,79
D4	0,25	6,65	6,41	98,24	11157,00	3,06 (10,07)	76,56
D5	**0,01**	43,01	43,00	**99,90**	248,00	-4,01 (-0,39)	**76,41**
D6	**0,02**	43,15	43,14	**100,00**	601,50	-2,06 (-0,47)	**76,30**
D7	**0,01**	43,01	43,00	**99,90**	1764,25	-1,18 (-1,51)	**87,11**
D8	0,02	1,82	1,80	98,56	143,50	-17,59 (-0,65)	78,28
D9	0,02	1,79	1,77	98,70	294,00	1,01 (0,36)	92,41
D10	0,02	1,79	1,76	99,48	4057,00	1,88 (10,30)	95,82
D11	16,92	54,97	45,79	78,59	455,00	25,76 (42,90)	-1,20
D12	23,69	44,66	27,48	60,56	2594,75	47,50 (482,46)	16,82
D13	15,03	45,24	35,55	100,00	114,75	-3,64 (-0,22)	**-15,29**
D14	8,48	41,35	35,91	99,79	164,50	-0,08 (-0,01)	7,37
D15	7,68	26,05	19,90	78,85	603,00	24,08 (31,15)	70,94
D16	**33,61**	57,79	**36,42**	100,00	12,75	-8,41 (-0,15)	**-45,49**
D17	0,86	11,99	11,22	19,26	146,75	96,69 (136,00)	78,71
D18	2,06	17,28	15,54	78,82	3654,50	25,87 (332,17)	22,48
D19	1,39	5,85	4,52	54,23	800,00	63,28 (31,97)	50,34
D20	1,48	5,93	4,52	99,41	1032,25	-4,72 (-0,39)	**-54,63**

6 Conclusion

In this work, we have developed an optimization approach to deal with a new problem integrating the train selection problem and the rolling stock rotation problem (**TSRP**). The proposed algorithm is based on a network flow in a hypergraph model which made it possible to easily apprehend coupling and decoupling multiple units. An originality of our model is that flow units represent both railcar units in the technical hyperarcs and train configurations in the other hyperarcs. Also, merging flow units in depots nodes reduces the size of the generated hyperarc set. To reduce the size of the model, we implemented an algorithm based on an aggregation and disaggregation technique combined with reduced-cost filtering. The computational experiments show that this algorithm is numerically efficient for real size instances.

Future works may include a broader integration in the railway production problem, including netgraph design decisions, or maintenance.

Acknowledgments

This work was partly funded by an ANRT (`http://www.anrt.asso.fr/`) CIFRE scholarship between University of Bordeaux and SNCF. Experiments presented in this paper were carried out using the PLAFRIM experimental testbed, being developed under the Inria PlaFRIM development action with support from Bordeaux INP, LABRI and IMB and other entities: Conseil Régional d'Aquitaine, Université de Bordeaux, CNRS and ANR in accordance to the programme d'investissements d'Avenir (see `https://www.plafrim.fr/`).

References

[1] Ralf Borndörfer, Markus Reuther, and Thomas Schlechte. A Coarse-To-Fine Approach to the Railway Rolling Stock Rotation Problem. In Stefan Funke and Matúš Mihalák, editors, *14th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 42 of *OpenAccess Series in Informatics (OASIcs)*, pages 79–91, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[2] Ralf Borndörfer, Markus Reuther, Thomas Schlechte, Kerstin Waas, and Steffen Weider. Integrated optimization of rolling stock rotations for intercity railways. *Transportation Science*, 50(3):863–877, 2015.

[3] Valentina Cacchiani, Alberto Caprara, Laura Galli, Leo Kroon, Gábor Maróti, and Paolo Toth. Railway rolling stock planning: Robustness against large disruptions. *Transportation Science*, 46(2):217–232, 2012.

[4] Valentina Cacchiani and Paolo Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737, 2012.

[5] Riccardo Cambini, Giorgio Gallo, and Maria Grazia Scutellà. Flows on hypergraphs. *Mathematical Programming*, 78(2):195–217, 1997.

[6] Alberto Caprara, Matteo Fischetti, and Paolo Toth. Modeling and solving the train timetabling problem. *Operations research*, 50(5):851–861, 2002.

[7] Pieter-Jan Fioole, Leo Kroon, Gábor Maróti, and Alexander Schrijver. A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, 174(2):1281–1297, 2006.

[8] Jørgen T Haahr, Joris C Wagenaar, Lucas P Veelenturf, and Leo G Kroon. A comparison of two exact methods for passenger railway rolling stock (re) scheduling. *Transportation Research Part E: Logistics and Transportation Review*, 91:15–32, 2016.

[9] Jørgen Thorlund Haahr, Richard Martin Lusby, Jesper Larsen, and David Pisinger. A branch-and-price framework for railway rolling stock rescheduling during disruptions. *DTU Management Engineering*, 2014.

[10] Leo Kroon, Gábor Maróti, and Lars Nielsen. Rescheduling of railway rolling stock with dynamic passenger flows. *Transportation Science*, 49(2):165–184, 2014.

[11] Lars Kjær Nielsen, Leo Kroon, and Gábor Maróti. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220(2):496–509, 2012.

[12] Anita Schöbel. Line planning in public transportation: models and methods. *OR spectrum*, 34(3):491–510, 2012.

[13] Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989.