Re-scheduling in railways: the Rolling Stock Balancing Problem

Gabriella Budai*‡, Gábor Maróti†‡§¶, Rommert Dekker*
Dennis Huisman*‡§, Leo Kroon†‡§

May 30, 2007

Econometric Institute Report EI2007-21

Abstract

This paper addresses the Rolling Stock Balancing Problem (RSBP). This problem arises at a passenger railway operator when the rolling stock has to be re-scheduled due to changing circumstances. These problems arise both in the planning process and during operations.

The RSBP has as input a timetable and a rolling stock schedule where the allocation of the rolling stock among the stations does not fit to the allocation before and after the planning period. The problem is then to correct these off-balances, leading to a modified schedule that can be implemented in practice.

For practical usage of solution approaches for the RSBP, it is important to solve the problem quickly. Therefore, the focus is on heuristic approaches. In this paper, we describe two heuristics and compare them with

*Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, NL-3000 DR, Rotterdam, The Netherlands. E-mail: {budai,huisman,rdekker}@few.eur.nl.

†Rotterdam School of Management, Erasmus University Rotterdam, P.O. Box 1738, NL-3000 DR, Rotterdam, The Netherlands. E-mail: {gmaroti,lkroon}@rsm.nl.

‡Erasmus Center for Optimization in Public Transport (ECOPT).

§This work was partially supported by the Future and Emerging Technologies Unit of EC (IST priority - 6th FP), under contract no. FP6-021235-2 (project ARRIVAL).

¶Corresponding author.

each other on some (variants of) real-life instances of NS, the main Dutch passenger railway operator. Finally, to get some insight in the quality of the proposed heuristics, we also compare their outcomes with optimal solutions obtained by solving existing rolling stock circulation models.

Keywords: railway planning, rolling stock re-scheduling, integer linear programming, heuristics.

1 Introduction

The rolling stock planning process of most railway operators is commonly divided into several planning phases. Huisman et al. (2005) distinguish four planning phases, namely strategic, tactical, operational and short-term planning. Strategic planning deals with long term decisions such as the acquisition of new rolling stock. At the tactical level, the different types of rolling stock are assigned to the different lines of the network. This is typically done once a year. The main goal of operational planning is to find rolling stock schedules with low operational costs and high service quality; this basic schedule is to be carried out throughout the whole year. However, every day there are minor modifications to the timetable due to some extra trains or maintenance work on some parts of the infrastructure. These exceptions are handled during the short-term planning phase. The time horizon of short-term planning ranges from a couple of days to a couple of weeks. The final plans are carried out (and modified if necessary) in real-time operations.

This paper deals with the *Rolling Stock Balancing Problem (RSBP)*, which is a problem faced in the short-term planning phase as well as during the operations. The input consists of the timetable for a given planning period, the available rolling stock, the desired inventories and an *input rolling stock schedule*, which is feasible except that it may contain some *off-balances*. An off-balance is defined as a deviation from the desired inventory level of a certain type of rolling stock at a certain station. The goal is to construct a new rolling stock schedule that satisfies the balancing constraints as much as possible. Therefore, the primary objective is to minimize the number of off-balances (i.e. the deviation from the desired inventories). As secondary objective, other criteria related to costs and service may be optimized as well.

To better understand the motivation for studying this problem, we first give some background information on rolling stock planning at NS, the main passenger railway operator in the Netherlands.

At NS, most trains are operated by self-propelled train units, and only a few are operated by a locomotive and carriages. Therefore, we will only consider train units in the remainder of this paper. These train units are available in several types. Units of compatible types can be attached to each other to form longer compositions. Units of the same type are fully interchangeable, since individual units are not distinguished during the planning phase.

The rolling stock schedule specifies the train composition of each train trip. That is, how many units of each type are to be used for each timetable service and in which order. From this assignment, one can obtain the *duties*. A duty is the workload of a single rolling stock unit on a single day: it is a chain of tasks where a task is characterized by a trip and by the position of the unit in the composition of this trip, e.g. front or rear. The practical feasibility of a schedule highly depends on the shunting possibilities of the stations. Since shunting is a complex problem on its own (see e.g. Freling et al. (2005), Lentink (2006)), NS uses an iterative approach. First, rolling stock duties are determined. Afterwards, when the whole set of duties leads to infeasibility at certain stations, the duties are modified. This process continues until there is an overall feasible solution. In practice, this may take several rounds. To speed up this process, several key aspects of the shunting process are taken into account during the creation of the rolling stock schedules. Examples are the restrictions on composition changes at certain stations: uncoupling (or coupling) of units can only take place at the appropriate side of the train.

During the whole planning process, i.e. in operational and short-term planning, the process above is applied. The difference between both planning phases is that the time available to come up with a solution is much higher in the operational phase. This throughput time is even more relevant for large-scale re-scheduling problems during the real-time operations, for instance due to major disruptions.

In recent years, NS introduced Operations Research based decision support tools for operational planning (see Alfieri et al. (2006), Fioole et al. (2006), Maróti (2006) and Peeters and Kroon (2003)). However, the running time of these methods may reach several hours on instances that are smaller than typical re-scheduling instances. Thus the restrictive deadlines in re-scheduling raise the need for alternative approaches. Moreover, discussions with planners revealed that it is usually easy to come up with a schedule that fulfills all requirements except that the desired inventories are not realized. This was our first motivation to study the RSBP.

A second motivation finds its background in disruption management (see Jespersen Groth et al. (2007)). During a disruption, the dispatchers try to use all available rolling stock to transport as many passengers as possible in the right direction. As a result, the rolling stock units will not finish their daily duties at the location where they were planned to. This is not a problem if two units of the same type get switched. In many cases, however, the number of units ending up in the evening at a certain station differs from the number of units that has to start their next day's duty there. To prevent expensive deadheading trips, it is attractive to modify the rolling stock schedules such that the rolling stock is balanced before the night. This problem is equivalent to the RSBP in the planning phase, the only difference is that the initial input plan is not constructed by planners but is accidentally born.

In this paper we describe the Rolling Stock Balancing Problem. Although the motivating problems arise at NS, we believe that similar problems are to be solved whenever trains between multiple depots are operated with train units of multiple types. To the best of our knowledge, the RSBP has not been studied before. However, ideas of several related problems can be used, therefore we give a brief literature overview on these related problems.

We analyze the computational complexity of RSBP and we prove that even its simple special cases are NP-hard; this fact, together with the need for fast solution approaches, has decided us to focus on heuristics. We present two heuristics solution approaches. The heuristics are compared with each other on (variants of) real-life problem instances of NS. Moreover, to get some insight in the quality of the heuristics, we also compare the outcomes with the results of an existing model for operational rolling stock planning from scratch.

The remainder of this paper is organized as follows. In Section 2 we give a precise description of the RSBP. Section 3 contains a brief literature overview. Section 4 is devoted to complexity results. The heuristics are described in Sections 5 and 6. The computational results are discussed in Section 7. Finally, in

Section 8 we draw some conclusions and we outline some directions for future work.

2 Problem description

In this section we define the Rolling Stock Balancing Problem in more detail. We are given the timetable for the planning period. The timetable defines a set of *trips*, which are characterized by the train number, departure and arrival times, departure and arrival locations as well as the estimated number of passengers. Moreover, we have a list of available rolling stock types and the number of available units per type.

For each station, the *desired initial inventory* is the number of units per type that arrive there before the planning period. The *desired final inventory* is the number of units per type that are needed there after the planning period. The desired inventories link the output of the balancing problem to what happens before and after the planning period.

Each trip (except for arrivals in the late evening) has a *successor* trip; a trip and its successor trip are to be operated in principle by the same rolling stock units. A trip is followed shortly (within at most an hour, usually within minutes) by its successor trip, this leaves only time for restricted composition changes: One or two units can be coupled to the arriving train or uncoupled from it, before departing again. A general rule states that coupling and uncoupling cannot be performed at the same time. These composition changes may take place on a pre-defined side of the train: either on the front side or on the rear side. Train units uncoupled from a train are not immediately available yet, only a certain *re-allocation time* later; this is to reserve time for necessary shunting operations before using the uncoupled unit in another train.

Next we need the concept of the inventory. The inventory of a station at a given time moment is formed by the units that are currently staying idle at that station. These units can be coupled to a departing train, while uncoupled units go to the inventory. It is vital to see that the order of the units in the inventory is arbitrary; this is in contrast with the trains themselves where the order of the units is essential.

Finally, we are given an *input rolling stock schedule* which assigns a composition to each trip; the input schedule satisfies all technical requirements above. The following requirements must also hold: the length of the composition on a trip must be under a certain limit (determined by the relevant platform lengths). Moreover, the trip has to be assigned at least a given number of carriages in order to cover (at least a large part of) the passenger demand.

The input schedule may not comply with the desired inventories. A station has a *deficit* (or a *surplus*) in the *initial* inventory of a given type if, according to the input plan, the number of units of this type that are located at the beginning of the planning period at that station is higher (or lower) than the desired initial inventory of this type. Similarly, a station has a *deficit* (or a *surplus*) in the *final* inventory of a given type if according to the input plan, the desired final inventory of this type is higher (or lower) than the number of units of this type that are located at the end of the planning period at that station in the input plan. The *number of off-balances* in a rolling stock schedule is obtained by summing the surpluses over all stations and over all types. This number expresses how many units have to be involved in dead-heading trips.

The *Rolling Stock Balancing Problem (RSBP)* can now formally be defined as the problem of modifying the input rolling stock schedule to a new schedule such that (1) the new schedule is still feasible, and (2) it contains a minimum number of off-balances. Next to minimizing the number of off-balances, secondary objectives related to costs and service can be taken into account. In the experiments, we choose for an objective function which is a linear combination of the number of off-balances (with a very high weight), carriage-kilometers, shortage-kilometers and the number of composition changes. Carriage-kilometers express the operational costs of the railway operator. Seat shortage kilometers are computed by taking the expected number of passengers without a seat on a trip, multiplying it by the length of the trip and adding them up over all trips; the obtained value corresponds to the service quality. The number of composition changes counts how many times units are coupled to or uncoupled from a train during a short stop. A schedule with a smaller number of composition changes is expected to be less sensitive to small delays.

The planning period may consist of several days. Then it is common to allow dead-heading trips every night during the planning period. The RSBP and the solution methods suggested in this paper can easily be extended to such cases. In fact, we carried out some computational tests on instances with a 2-day planning period.

3 Literature overview

A large number of publications addressed operational rolling stock planning. We only mention here Peeters and Kroon (2003) and Fioole et al. (2006). Their models have basically the same specifications as those in this paper. In the case when trains are not combined or split, Peeters and Kroon (2003) solve the problem by applying Dantzig-Wolfe decomposition and Branch-and-Price as solution technique. Fioole et al. (2006) extend the model for splitting and combining of trains. They use the commercial MIP software CPLEX to solve the model.

Compared to operational planning, literature on short-term railway rolling stock planning is scarce. Ben-Khedher et al. (1998) study the short-term re-scheduling problem of the French TGV trains from a revenue management's point of view. The rolling stock circulation must be adjusted to the latest demand from the seat reservation system in order to maximize the expected profit. Shunting is not really an issue since trains may consist of at most two units.

Lingaya et al. (2002) deal with the effect of an altered timetable and passenger demand on the rolling stock schedules, focusing on the case of locomotive hauled carriages. They explicitly take the order of the carriages in the trains into account and assume that for each train a successor train has already been specified. Several real-life aspects, such as maintenance, are considered as well.

Substantial research has been carried out on aircraft and bus re-scheduling. Kohl et al. (2004) and Clausen et al. (2005) give overviews of airline disruption management, including a detailed list of aircraft re-scheduling publications and applications. The common solution approaches are based on multicommodity network flows, applying various exact and heuristic methods. Many of the models incorporate maintenance of the aircraft as well.

Recently, Li et al. (2004) introduced the single depot vehicle re-scheduling problem. It is motivated by the problem of updating bus schedules in the case when a single vehicle breaks down. The re-scheduling problem is formulated as a minimization problem over a number of vehicle scheduling problems.

A main distinguishing feature of railway (re-)scheduling is that the order of the train units in the trains is to be regarded when they are attached to each other. In contrast, a single bus or aircraft is to be used for a flight or a bus trip. Also, strict airline maintenance regulations make it necessary to follow the path of each individual aircraft during the entire planning period. In railway re-scheduling, however, preventive maintenance is less binding, therefore train units of the same type can be considered interchangeable.

We conclude that although a large variety of related problems has been described and partly successfully solved, railway rolling stock scheduling – in particular in real-time operations – still lacks the appropriate models and solution methods.

4 Complexity results

In this section we prove that it is NP-complete to decide whether an instance of the Rolling Stock Balancing Problem has a feasible solution, even if only a single station has a surplus in the final inventory and another station has a deficit in the final inventory. Subsequently, we extend the construction in the proof and show that the problem remains NP-complete in the case of an off-balance of a single unit.

In the constructions below, there are two rolling stock types P and Q which can be combined with each other in one train. Each trip must receive one or two units. The compositions are denoted by strings of the characters P and Q, the right hand side of a string corresponding to the front of the train. The shunting side of the stations, either left or right, is indicated by [L] and [R], respectively. Stations with right hand shunting side admit composition changes at the front of a train, while stations with left hand shunting side admit composition changes at the rear of a train.

Throughout the whole section, the railway networks are drawn in time-space diagrams: Stations are represented by horizontal time-lines, the time increases to the right. The trips correspond to diagonal lines between the time-lines. Train stops are indicated by dots. Dotted arcs connect the arrivals of the trips to the departures of their successors.

In the figures, some trips do not have a departure or arrival station. These missing anonymous stations are all different. Trips to or from anonymous stations always have a single unit of a certain type in the input plan. Anonymous stations with a departing trip have initial inventory 1 for this type and 0 for the other type; the final inventory is 0 for both types. The analogous condition holds for anonymous stations with an arriving trip: the initial inventories are 0, the final inventory is 1 for the type of the arriving unit and 0 for the other type.

In the input plan, trips are operated by a single unit of type P (in the figures represented by thick solid lines), by a single unit of type Q (thick dotted lines) or by a two-unit composition PP (thick dashed lines).

4.1 Building blocks for the proofs: the gadgets

A *gadget* is a part of the railway network shown in Figure 1. The figure indicates the 8 named stations, their shunting sides, the trips (among them trips s_1, s_2, t_1 and t_2), and their compositions in the input plan.

The initial and final inventories of stations β, γ, δ and ε are 0. Stations α_1, α_2, ω_1 and ω_2 have undefined initial and final inventories in type P and they have zero initial and final inventory in type Q.

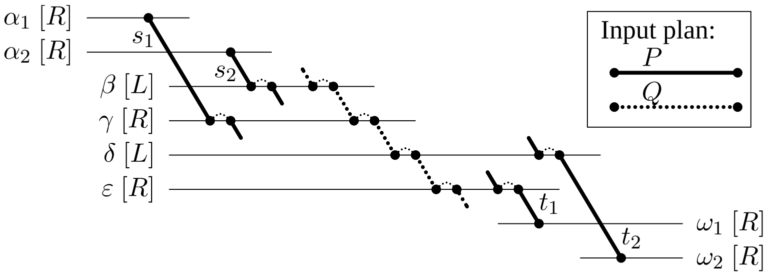


Figure 1: A gadget.

Lemma 4.1. *Consider a rolling stock circulation for a gadget that satisfies the given inventory, shunting and train length constraints. Then the following holds:*

(i) Trip s_1 has composition PP if and only if trip t_1 has composition PP.

(ii) Trip s_2 has composition PP if and only if trip t_2 has composition PP.

(iii) At most one of the trips t_1 and t_2 can have composition PP.

Proof. (*i*) If trip s_1 has composition PP, then a unit can be uncoupled from it at station γ. This unit can be coupled to the right-hand side of the unit that travels from γ towards ε. Then the unit of type P can only be uncoupled at station ε. Actually, this is the only possibility to lead the uncoupled unit to either ω_1 or ω_2. Moreover, this is the only way to get composition PP for trip t_1.
(*ii*) Similar to (*i*).
(*iii*) Two extra units of type P can reach stations ω_1 and ω_2 only if the trip between γ and δ has composition PQP. However, this would violate the upper bound on the train length. □

We use the simplified symbol in Figure 2 for a gadget. The main purpose of a gadget is to bring an additional unit either from α_1 to ω_1, or from α_2 to ω_2, but not both.

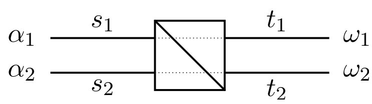


Figure 2: A simple symbol for a gadget.

4.2 Resolving an off-balance of k units

Consider an undirected graph $G = (V, E)$ with $V = \{1, \ldots, n\}$ and let k be a positive integer with $k \leq n$. We build an instance of RSBP that is feasible if and only if G contains a stable set of size k. A *stable set* is a subset of nodes such that no pair of them is joined by an edge. It is well known that deciding whether a graph has a stable set with k nodes is NP-complete (Karp (1972)).

Create two stations A and Z. For every node $v \in V$ with d_v neighbors, we create $d_v + 1$ stations $S^v_1, \ldots, S^v_{d_v+1}$. The shunting side of all these stations is [R].

For each $v \in V$, insert a trip from station A to station S^v_1 and insert a trip from station $S^v_{d_v+1}$ to station Z. For each trip from A to a station S^v_1, create a predecessor trip from an anonymous station to A. For each trip arriving at Z, insert a successor from Z to an anonymous station. All trips so far are operated with a single unit of type P in the input plan.

For each node $v \in V$ with neighbors $u_1, \dots, u_{d_v}$, assign stations $S_1^v, \dots, S_{d_v}^v$ to the edges $u_1v, \dots, u_{d_v}v$, bijectively in an arbitrary way. For each edge $uv \in E$ with $u < v$, add a gadget as follows. Let S_i^u and S_j^v be the stations assigned to edge uv. Create four new stations β, γ, δ and ε, set $\alpha_1 = S_i^u$, $\alpha_2 = S_j^v$, $\omega_1 = S_{i+1}^u$, $\omega_2 = S_{j+1}^v$ and insert all the trips described in the definition of a gadget. A station S_j^v with $1 < j < d_v + 1$ belongs to exactly two gadgets and has one arriving and one departing trip. The departing trip is the successor of the arriving trip.

This completes the railway network. Its size is polynomial in n: it contains $O(n^2)$ trips between $O(n^2)$ stations. The network for a small graph is shown schematically in Figure 3.

The input plan satisfies the following inventory constraints. The initial and final inventories for type Q are 0 (except for some anonymous stations inside the gadgets). For type P, the initial and final inventories of stations S_j^v are 0. Station A has initial and final inventory k, while station Z has initial and final inventory 0. The initial and final inventories of the β-, γ-, δ- and ε-stations of the gadgets are all zero.

The *desired* inventories differ from these at two points. The desired final inventory of station A in type P is 0, the desired final inventory of station Z in type P is k. In some sense, balancing means that k units of type P must be routed from A to Z.

Note that the inventory and train length constraints do not leave much choice for feasible rolling stock circulations. Each trip has either the same composition as in the input plan or it receives the original composition extended by a single unit of type P.

Theorem 4.2. *Graph $G = (V, E)$ contains a stable set of size k if and only if the instance of RSBP constructed above has a feasible solution.*

Proof. Suppose that G contains the stable set $\{v_1, \dots, v_k\}$. A solution of RSBP can be obtained as follows. Couple the k units of type P at station A to the k trips that depart towards stations $S_1^{v_1}, \dots, S_1^{v_k}$.

Consider any gadget that connects stations $S_j^{v_i}$ and $S_{j+1}^{v_i}$ for some indices i and j. We adjust the input plan inside the gadget as follows. The trips of this gadget

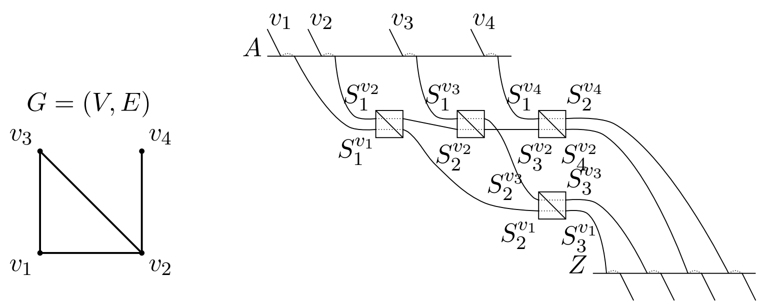


Figure 3: An example of the construction of the network.

that are incident to stations $S_j^{v_i}$ and $S_{j+1}^{v_i}$ get composition PP; we also make all the necessary modifications to route the additional unit through the gadget from $S_j^{v_i}$ to $S_{j+1}^{v_i}$. The adjustment of the gadgets can be done simultaneously since there is no edge between the nodes $v_1, \ldots, v_k$. Then all the k excess units reach station Z where they can be uncoupled. Therefore, RSBP is feasible.

Conversely, consider a solution of the RSBP. At station A, k units of type P are coupled to trips towards stations, say, $S_1^{v_1}, \ldots, S_1^{v_k}$. These units pass through all gadgets that are related to the nodes $v_1, \ldots, v_k$ and end up at station Z. Then the nodes $v_1, \ldots, v_k$ form a stable set in G as otherwise Lemma 4.1 (*iii*) would be violated. □

Corollary 4.3. *The feasibility version of RSBP is NP-complete.*

4.3 Resolving an off-balance of one unit

Here we extend the construction described in the previous section. Thereby we prove that the maximum stable set problem can be reduced to RSBP with an off-balance of one unit.

Let $G = (V, E)$ be an undirected graph with $|V| = n$ and let k be a positive integer with $k \leq n$. Consider the railway network constructed in the previous section. It is represented in Figure 4 by stations A and Z and the gray box between them.

Create $k + 1$ new stations $\alpha_1, \ldots, \alpha_k$ and ω. Insert $4k$ trips as follows (see Figure 4). Create a trip from α_i to α_{i+1} for each $i = 1, \ldots, k$ (where $\alpha_{k+1} = \omega$) and insert their predecessors and successors from and to anonymous stations. Also insert a trip that departs from station α_i and returns to the same station and has no predecessor or successor. All these new trips are operated by a single unit of type P in the input plan.

Insert k additional gadgets $g_1, \dots, g_k$. Let $s_1^{(i)}$, $s_2^{(i)}$, $t_1^{(i)}$ and $t_2^{(i)}$ denote the s_1-, s_2-, t_1- and t_2-trips of gadget g_i. For each $i = 1, \dots, k$, trips $s_1^{(i)}$, $s_2^{(i)}$, $t_1^{(i)}$ and $t_2^{(i)}$ (and their predecessor or successor trips from or to anonymous stations) connect gadget g_i to stations A, Z and α_i as shown in Figure 4. The figure also indicates the compositions on these trips in the input plan.

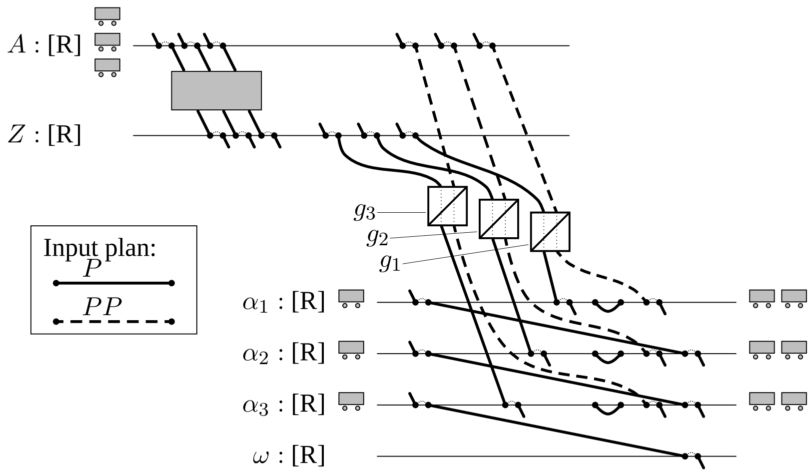


Figure 4: One unit to be routed ($k = 3$). At the beginning and the end of the time-lines, we give the initial and final inventories of type P realized by the input plan.

The railway network we constructed has polynomial size in n since it contains $O(n^2)$ trips and $O(n^2)$ stations.

The initial and final inventories for units of type Q are 0 except for some anonymous stations inside the gadgets. For type P, the initial inventory of station A is k, at stations $\alpha_1, \dots, \alpha_k$ it is 1 and at stations Z and ω it is 0. The final inventory of stations A, Z and ω is 0, while at stations $\alpha_1, \dots, \alpha_k$ it is 2. Anonymous stations have initial and final inventory zero or one. All other stations (i.e. the β-, γ-, ε- and δ-stations of the gadgets) have zero initial and final inventories.

The goal is to decrease the final inventory of station α_1 in type P by one and to increase the final inventory of station ω in type P by one.

Note that, as in the previous section, the shunting, inventory and train length constraints restrict the possible rolling stock circulations a lot. Each trip must be assigned the same composition as in the input plan, eventually with a unit of type P coupled or uncoupled at the appropriate side. In particular, the circulation of the unit of type Q does not change at all.

Theorem 4.4. *Graph G has a stable set of size k if and only if the input plan can be modified to decrease the final inventory of α_1 in type P by one and to increase the final inventory of ω in type P by one.*

Proof. To increase the final inventory at ω by one, the trip from α_k to ω must get a composition PP. Then the trip from α_k returning to α_k has no unit to serve unless an extra unit arrives earlier from gadget g_k. That is, trip $s_1^{(k)}$ from Z to gadget g_k and trip $t_1^{(k)}$ from gadget g_k to α_k must get composition PP, too. Then trip $s_2^{(k)}$ from A to gadget g_k and trip $t_2^{(k)}$ from gadget g_k to α_k must get composition P only. To correct the final inventory at α_k, the trip from α_{k-1} to α_k must get composition PP. Repeating the argument, it follows that RSBP can be solved if and only if all the k units that start at A can reach station Z. Invoking Theorem 4.2 completes the proof. □

Corollary 4.5. *The feasibility version of RSBP is NP-complete in the case of an off-balance of a single unit.*

5 A heuristic based on elementary balancing possibilities

In this section we describe a two-phase heuristic approach for the Rolling Stock Balancing Problem.

In Phase 1 we identify a number of “elementary” balancing possibilities (below abbreviated as BP) in the input plan: how can one unit (or several units) be sent from a station with a surplus to another station with a deficit such that the rolling stock balance remains unchanged for all other stations. We restrict ourselves to two kinds of BPs. In some of the BPs, the excess unit of a certain type is coupled to a train and it will be uncoupled from the train once the station with a deficit in that type is reached. Thus, the train length on some trips is increased. In the second kind of BPs, the train length on some trips is decreased by removing one (or more) unit(s) of a certain type from a sequence of trips; the sequence starts at a station with a deficit and ends at a station with a surplus in that type.

A cost value is assigned to each BP expressing how the weighted sum of the carriage kilometers, shortage kilometers and shunting operations changes if the BP is indeed used. Dead-heading trips are also considered to be BPs. These correspond to unresolved off-balances, so their cost is defined as the penalty of an off-balance.

Phase 2 selects some of the elementary BP computed in Phase 1 such that carrying out these selected BPs leads to a new rolling stock schedule without off-balances. Phase 2 minimizes the cost of the selected BPs and makes sure that they do not interfere with each other. This is done by solving an integer linear program.

A BP may solve off-balances only at the beginning (or at the end) of the planning period, but may also connect a surplus at the beginning of the planning period to a deficit at the end, or vice versa. In the latter cases, implementing that BP only would mean that one unit more or one unit less is used than in the input plan. Constraints in the model of Phase 2 make sure than the output rolling stock plan uses the same number of units as the input plan does.

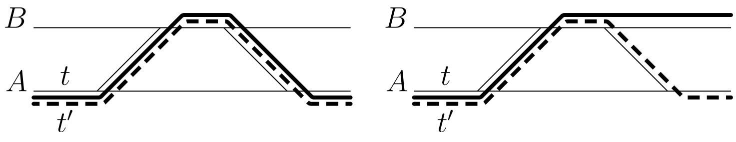
5.1 Phase 1: finding the balancing possibilities

Here we show a couple of examples for BPs; these can be found by applying easy search algorithms on the input rolling stock plan. Many other BPs can be computed using very similar ideas.

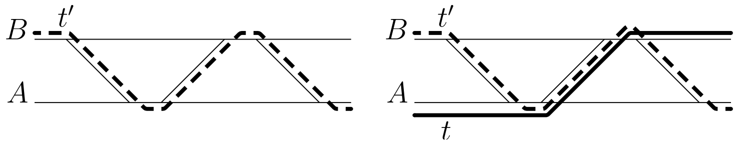
We consider a rolling stock type t, any pair (A, B) of stations where station A has a surplus and station B has a deficit of type t, and any integer number $k \geq 1$. Moreover, we distinguish four cases depending on whether the initial or the final inventories have the deficit and the surplus.

For these values t, A, B, k and for each of these four cases, a BP is an assignment of new compositions to some of the trips such that in this new rolling stock schedule, the surplus of type t at station A in the initial or final inventory is decreased by k, the deficit of type t at station B in the initial or final inventory is decreased by k, and all other stations have unchanged inventories.

Figure 5(a) shows a BP where the train length on some trips is decreased. We assume that station A has a surplus of type t and station B has a deficit of type



(a) Decreasing the train length on some trips



(b) Increasing the train length on some trips

Figure 5: Examples of balancing possibilities. The thick continuous line indicates a duty of a unit of type t, the thick dashed line stays for a duty of a unit of type t'.

t, both in the final inventory. Suppose there is a sequence of trips that starts at A, passes B and returns to A. Then the BP is to uncouple a unit (or eventually several units) of type t from that trip of this sequence that arrives at B and this unit remains idle at station B. This BP is only allowed if all the shunting rules and the train length restrictions are satisfied.

Figure 5(b) shows an example where the BP increases the train length on some trips. We assume that station A has a surplus of type t in the initial inventory and station B has a deficit of type t in the final inventory. Suppose there is a sequence of trips that starts at A and ends at B. Then the BP is to couple a unit (or eventually several units) of type t to the first trip of this sequence and to uncouple it after the last trip of the sequence. Of course, we only allow this BP if it complies with the shunting rules and with the train length restrictions. Note that several sequences of trips may provide ways to bring a unit of type t from A to B. Then each of them corresponds to a distinct BP.

A similar pattern may also be useful if the following two conditions hold. First, station A has a surplus of type t in the final inventory (instead of the initial inventory). Second, a unit of type t stays idle in the input plan from the departure time of the train from A to B till the end of the planning horizon. Under these conditions, the unit of type t can simply be sent from station A to B, giving rise to a BP.

A combination of the previous two BPs is shown in Figure 6. We assume that station A has a surplus of type t and station B has a deficit of type t, both in the initial inventory. Suppose there is a sequence of trips that starts at B, passes A and ends at B. Then a BP is to decrease the train length between B and A by a unit of type t, and to couple a unit of type t at A to the train leaving towards C. Again, this BP is only allowed if all the shunting rules and the train length restrictions are satisfied on the sequence of trips from station B to C.

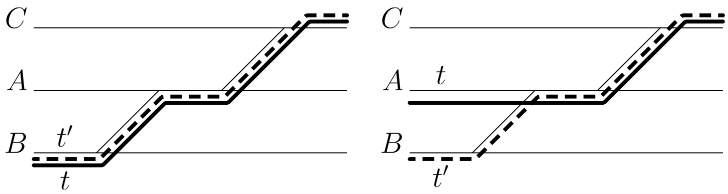


Figure 6: Combined balancing possibility. The thick continuous line indicates a duty of a unit of type t, the thick dashed line represents a duty of a unit of type t'.

More complicated BPs arise when solving two off-balances for two different train unit types at once. This can be done by switching two complete or partial duties of the input plan. Such a BP is shown in Figure 7. Here we assume that station A has a surplus of type t in the final inventory and station B has a deficit of type t in the final inventory. Moreover, we assume that station B has a surplus of type t' and station A has a deficit of type t', both in the final inventory. Suppose there are two sequences of trips: the first from station A to A that is carried out by a unit of type t, and the second sequence from station B to B that is carried out by a unit of type t'. If these sequences have a common idle period at station Z, then the units of type t and t' may take over each other's role from the meeting point till the end of the planning period.

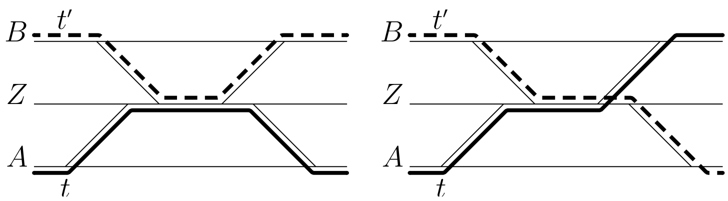


Figure 7: Switching two partial duties. The thick continuous line indicates a duty of a unit of type t, the thick dashed line represents a duty of a unit of type t'.

5.2 Phase 2: combining balancing possibilities

Once all BPs and their costs have been defined in Phase 1, the question remains how to choose those BPs which have the lowest possible total cost and which lead to a feasible solution to the RSBP. The answer to this question is given in Phase 2 of this heuristic algorithm.

Given a set of all BPs that were defined in Phase 1, we choose those BPs that minimize the weighted sum of carriage kilometers, shortage kilometers, shunting movements and number of dead-heading trips. The latter is equivalent with the number of remaining off-balances.

Selecting several BPs may result in a conflict, e.g. by exceeding the maximal allowed train lengths. To stay on the safe side, we allow for BPs to be selected simultaneously only if each trip gets modified at most once. Thereby we make sure that the selected BPs can be implemented in practice. A disadvantage of doing so is that the solution space might be restricted too much: BPs that modify the same trip may be used without conflicts.

The BPs with overall minimum cost are selected with the following integer linear programming model. Let E be the set of all BPs, S be the set of all stations, T be the set of train unit types and $Trip$ be the set of all trips. Let $b^{\text{beg}}_{s,t} \in \{0, \pm 1, \pm 2, \dots\}$ and $b^{\text{end}}_{s,t} \in \{0, \pm 1, \pm 2, \dots\}$ denote the surplus or deficit in the initial and final inventory of type $t \in T$ on station $s \in S$; a positive value indicates a deficit and a negative value indicates a surplus.

Let c_e be the cost of $e \in E$. Furthermore, the BP $e \in E$ increases the initial (or final) inventory of type $t \in T$ on station $s \in S$ with $d^{\text{beg}}_{s,t,e}$ (or $d^{\text{end}}_{s,t,e}$) units. Note that $d^{\text{beg}}_{s,t,e}$ and $d^{\text{end}}_{s,t,e}$ may be negative. Γ_e denotes the list of modified trips by BP $e \in E$.

For each BP $e \in E$, let x_e be a binary decision variable expressing whether or not $e \in E$ is selected. Then the BP selection problem can now be formulated as follows.

$$\text{minimize} \quad \sum_{e \in E} c_e x_e \tag{1}$$

$$\text{s.t.} \quad \sum_{e \in E} d^{\text{beg}}_{s,t,e} x_e = b^{\text{beg}}_{s,t} \qquad \forall s \in S, \forall t \in T \tag{2}$$

$$\sum_{e \in E} d^{\text{end}}_{s,t,e} x_e = b^{\text{end}}_{s,t} \qquad \forall s \in S, \forall t \in T \tag{3}$$

$$\sum_{e \in E:\, v \in \Gamma_e} x_e \le 1, \qquad \forall v \in \mathit{Trip} \tag{4}$$

$$x_e \in \{0,1\} \qquad \forall e \in E \tag{5}$$

The objective minimizes the total costs of the selected BPs. At each station and for each train unit type, the sum of the changes in the initial (or final) inventory is equal to the deficit or surplus in the initial (or final) inventory. This is ensured by constraints (2) (or (3)). Some BPs cannot be chosen together in the solution, since they use the same trip. This is expressed by constraints (4). Finally, constraints (5) state that the decision variables are binary.

In our computations, it turned out to be essential to strengthen the model by adding the following valid inequalities:

$$\sum_{e \in E: d^{\text{beg}}_{s,t,e} > 0} x_e \ge 1 \qquad \forall s \in S, \forall t \in T : b^{\text{beg}}_{s,t} > 0. \tag{6}$$

These constraints state that, if a station has a deficit in the initial inventory in a given type, then at least one BP is selected which increases the initial inventory of that type at that station. Furthermore, inequalities similar to (6) but $b^{\text{beg}}_{s,t} > 0$ replaced by $b^{\text{beg}}_{s,t} < 0$, $b^{\text{end}}_{s,t} > 0$ and $b^{\text{end}}_{s,t} < 0$ are also added to the model.

For each type t, the sum of $b^{\text{beg}}_{s,t}$ over the stations is zero. Therefore if a model chooses some BPs connecting an initial surplus to a final deficit (i.e. where the BP increases the number of train units needed), then the appropriate number of BPs from an initial deficit to a final surplus will also be selected. So at the end, the updated rolling stock plan uses the same number of train units as the input.

The model (1)–(5) is solved by general purpose MIP software. The performance of the heuristic is presented in Section 7.

6 A flow-based heuristic approach

In this section, we describe an iterative heuristic approach for the Rolling Stock Balancing Problem. In each iteration, either a *type switching step* or a *re-routing* *step* is carried out. Both steps are intended to decrease the number of off-balances in a greedy way. The overall algorithm stops if no step can bring any further improvement.

In a *type switching step*, pairs of rolling stock units of different types are considered. The algorithm checks whether exchanging units in such a pair results in a feasible rolling stock schedule and also whether the exchange would decrease the number of off-balances. The two units whose exchange leads to the largest improvement are in fact switched, yielding an updated rolling stock schedule. Thereafter another iteration is launched.

Re-routing steps modify the internal structure of the schedule. Each step attempts to solve a special case of the problem with an off-balance of a single train unit. Having found a solution to this special case, the rolling stock schedule is updated accordingly. The updated plan has one off-balance less. Then another iteration is carried out.

The concept of type switching is rather straightforward. In what follows in this section we describe the re-routing step in detail.

6.1 The re-routing step

Here we assume that station A has an initial surplus of one unit of type t, station B has an initial deficit of one unit of the same type t, and there is no further off-balance in the input plan. (The case when A and B have off-balances in the final inventory is analogous.) The goal is to modify the input rolling stock schedule in order to resolve these off-balances. In Section 4 we have seen that the feasibility variant of this problem is NP-complete.

The re-routing step is intended to be significantly simpler than the general RSBP as we do not expect to change the input plan too deeply just to re-route a single unit. This motivates the basic restriction here: *the input schedule is to be modified in such a way that the circulation of every train unit type differing from* t *must not be changed.* So for example, if a trip has composition 'tab' in the input schedule with train unit types a, b and t, then the output schedule may assign composition 'ab', 'atb', '$attb$', etc. to the trip. However, it may not assign 'taa' or 'tba' since those would change the circulation of types 'a' and 'b'.

The graph representation

We represent the RSBP as a network flow problem. To do so, we build up a graph $G = (V, E)$ which is going to be a variant of the usual time-space networks. Let us start with an empty graph.

A time moment j is *relevant* at station C if a trip departs at j from C or if a trip r arrives at $j - \varrho(r)$ at C where $\varrho(r)$ is the re-allocation time. In addition, the begin and the end of the planning period are also relevant. Create a *station node* for each pair (C, j) where C is a station and j is a relevant time moment at C. For each pair j, j' of consecutive relevant time moments at C, draw a *station arc* from the node associated with (C, j) to the node associated with (C, j'). The flow values on the station arcs shall express the current inventories of type t at the stations. Station nodes at the beginning of the planning period are the *source nodes*, station nodes at the end are the *sink nodes*.

Consider a trip r and suppose that the input plan assigns composition

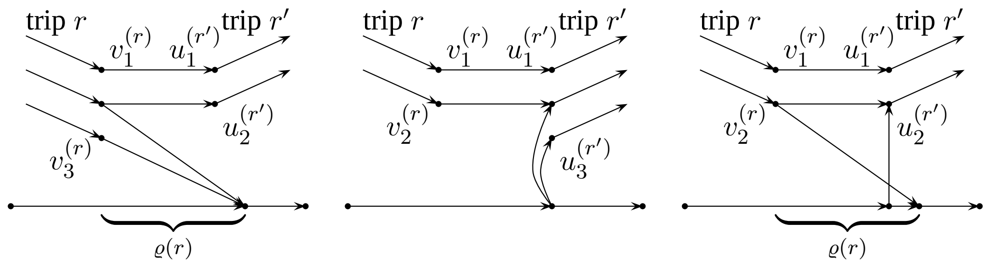
$$\underbrace{t \ldots t}_{k_1^{(r)}} t_1 \underbrace{t \ldots t}_{k_2^{(r)}} \ldots t_{\ell_r - 1} \underbrace{t \ldots t}_{k_{\ell_r}^{(t)}} \tag{7}$$

to r where $t_1, \ldots, t_{\ell_r - 1}$ denote train unit types differing from t. (As usual, we assume that the right-hand side of this string corresponds to the front of the train.) That is, train units of type t are assigned to trip r in ℓ_r possibly empty groups, separated by $\ell_r - 1$ units of other types. The heuristic algorithm shall only modify the integer values $k_1^{(r)}, \ldots, k_{\ell_r}^{(r)}$.

Create ℓ_r new nodes $u_1^{(r)}, \ldots, u_{\ell_r}^{(r)}$ corresponding to the groups of type t at the departure of r, and create ℓ_r new nodes $v_1^{(r)}, \ldots, v_{\ell_r}^{(r)}$ corresponding to the arrival of trip r. Moreover, draw the arcs $u_i^{(r)} v_i^{(r)}$ for each $i = 1, \ldots, \ell_r$. We call these arcs *trip arcs*.

Let r' be the successor trip of trip r and suppose that in the input plan, units are uncoupled form the arriving trip r. (Note that this can cover the case when r has no successor trip at all.) We also assume that the uncoupling takes place at the front side of the train. Then our graph representation does not contain the possibility of coupling any unit to trip r' and we have $\ell_r \geq \ell_{r'}$. Physically, the train is split into two parts at a point that lies in the ${\ell_{r'}}^{\text{th}}$ group of the arriving composition. Then the first (i.e. left-most in (7)) $\ell_{r'} - 1$ groups go over unchanged to become the first $\ell_{r'} - 1$ groups of trip r'. The last (i.e. right-most in (7)) $\ell_r - \ell_{r'}$

groups (if any) are uncoupled. Units in the ${\ell_{r'}}^{\text{th}}$ group of trip r can go over to the ${\ell_{r'}}^{\text{th}}$ group of trip r' or they can be uncoupled. These possibilities are expressed by the arcs shown in Figure 8(a) for the case $\ell_r = 3$ and $\ell_{r'} = 2$. Notice that the reallocation time is respected. The construction can easily be adjusted if uncoupling takes place at the rear side of the arriving trip.



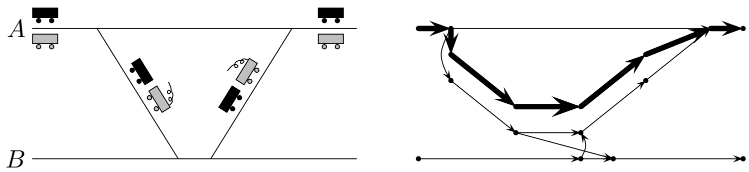
(a) Uncoupling (b) Coupling (c) No change

Figure 8: The graph representation of the cases when in the input plan uncoupling, coupling or no composition change takes place between trips r and r'.

The case when units are added to the departing trip r' in the input plan is modeled similarly. Then the graph does not include the possibility of uncoupling units after the arrival of trip r. An example (where a composition change may take place at the front side) is shown in Figure 8(b).

Finally, if trips r and r' have identical compositions in the input plan, then the graph expresses the possibility of coupling and uncoupling units. Suppose that uncoupling and coupling are possible at the front side (the other cases being analogous). Then the first (i.e. left-most in (7)) $\ell_r - 1$ groups of trip r go unchanged over to the first $\ell_r - 1$ groups of trip r'. However, the last group of trip r can be decreased or the last group of trip r' can be increased by uncoupling or coupling units of type t. Recall that coupling and uncoupling at the same time is not allowed. An example is given in Figure 8(c).

We call an arc from a station node to a node $u_i^{(r)}$ a *coupling arc* and we call an arc from a node $v_i^{(r)}$ to a station node an *uncoupling arc* as they are intended to describe coupling and uncoupling of units. This completes the definition of graph G. Figure 9 indicates the graph representation of a small railway network.



(a) The time-space diagram (b) The graph

Figure 9: The graph representation of a small railway network and the flow of the black unit type: bold arcs have flow value one, other arcs have zero flow value.

Network flows in this graph

The schedule of the units of type t in the input plan corresponds to a network flow x in $G = (V, E)$ as follows. Each trip arc gets the corresponding value $k_i^{(r)}$. The number of coupled or uncoupled units of type t is assigned to the coupling and uncoupling arcs. The flow value on a station arc is the inventory of type t at that station during the time interval indicated by the arc. Then the source nodes have a (possibly zero) net out-flow, the sink nodes have a (possibly zero) net in-flow, and all other nodes satisfy the flow conservation law.

The flow value on each arc is non-negative and, depending on the problem specification, they obey certain upper bounds denoted by $g(a)$ for arc a. For example, bounds on the station arcs may express the storage capacity of the stations. In addition, the following two side constraints (8) – (9) must be satisfied.

First, the train length on each trip r obeys the lower and upper bounds:

$$\mu_r^{\min} - L_r \leq \sum_{i=1}^{\ell_r} x'\left(u_i^{(r)} v_i^{(r)}\right) \leq \mu_r^{\max} - L_r \qquad \text{for each trip } r \tag{8}$$

where $\mu_r^{\min}$ (and $\mu_r^{\max}$) is the minimal (and maximal) length of the train on trip r, respectively, and L_r denotes the number of carriages in those units on trip r whose type differs from t.

Second, coupling and uncoupling may not take place at the same time between a trip r and its successor r':

$$\sum_{i=1}^{\ell_r} \sum_{\substack{a \in \delta^{\text{out}}(v_i^{(r)}): \\ a \text{ uncoupling arc}}} x'(a) = 0 \qquad \text{or} \qquad \sum_{i=1}^{\ell_r} \sum_{\substack{a \in \delta^{\text{in}}(u_i^{(r')}): \\ a \text{ coupling arc}}} x'(a) = 0 \tag{9}$$

where $\delta^{\text{in}}(v)$ (and $\delta^{\text{out}}(v)$) denotes the set of arcs entering (and leaving) node v.

Conversely, if a network flow in G satisfies side constraints (8) – (9), then it corresponds to a feasible rolling stock schedule.

Recall that station A has an initial surplus of one unit and station B has an initial deficit of one unit. That is, the desired final inventories are equal to the net in-flow of the sink nodes; the desired initial inventories are equal to the net out-flow of the source nodes except for A and B. In order to resolve this off-balance, we have to find a network flow x' such that, identifying stations A and B with their source node, $x'(\delta^{\text{out}}(A)) = x(\delta^{\text{out}}(A)) + 1$ and $x'(\delta^{\text{out}}(B)) = x(\delta^{\text{out}}(B)) - 1$. At each other node, the net in- and out-flow of x and x' must be equal. Furthermore, x' must satisfy the side constraints (8) – (9).

It is well-known in network flow theory that, if such a flow x' (without requiring (8) – (9)) exists, then it can be obtained by modifying x along an augmenting path P which is a directed $A - B$ path in the auxiliary graph G_x. The auxiliary graph G_x on node set V is constructed as follows. Let G_x have the *forward arc* uv if $uv \in E$ with $x(uv) < g(uv)$. Let G_x have the *backward arc* vu if $uv \in E$ with $x(uv) > 0$. Then the modification of x is defined as

$$x'(uv) = \begin{cases} x(uv) + 1 & \text{if the forward arc } uv \text{ is used by path } P, \\ x(uv) - 1 & \text{if the backward arc } vu \text{ is used by path } P, \\ x(uv) & \text{otherwise.} \end{cases} \tag{10}$$

An arbitrary augmenting path P may lead to the violation of the side constraints (8) – (9). Actually, the feasibility version of RSBP is NP-complete, therefore an augmenting path satisfying the side constraints cannot be found in polynomial time (unless P=NP). In our heuristic approach, we simply relax the side constraints (8) – (9): we look for an augmenting path and verify afterwards whether the updated network flow x' satisfies the side constraints (8) – (9).

If there is no augmenting path at all then the instance of RSBP is certainly infeasible. If there is an augmenting path and x' fulfills constraints (8) – (9) then the off-balance of stations A and B has been resolved. However, if there exists an augmenting path but the side constraints are violated, the algorithm reports that the off-balance could not be resolved. In the latter case, the answer might be wrong: other augmenting paths might result in satisfied side constraints. Note that in our extensive computational tests, we did not find any augmenting path that lead to violated side constraints (8) – (9).

Variants of the re-routing step

As described above, the re-routing step attempts to find any augmenting path. This reflects that the main objective is to resolve as many off-balances as possible. However, the additional objective criteria (carriage-kilometers, seat shortage kilometers and the number of composition changes) can be taken into account by assigning cost values to the arcs of G. Then, according to classical network flow theory, arc cost in G_x are defined by $c_x(uv) = c(uv)$ if uv is a forward arc and by $c_x(vu) = -c(uv)$ if vu is a backward arc. Now we have to look for a minimum cost augmenting path in G_x. However, in our implementation we do not allow negative arc cost in the auxiliary graph. On one hand, investigation on the trade-off between the objective criteria falls out of our scope. On the other hand, non-negative arc costs admit a very efficient path search method in G_x by using Dijkstra's algorithm with Fibonacci heaps (see Fredman and Tarjan (1987)).

The re-routing step can easily be extended to the case when, instead of a single station A and a single station B, a list of stations with initial off-balances is given. Denoting the set of stations with an initial surplus by $\mathcal{A}$ and the set of stations with an initial deficit by $\mathcal{B}$, we can compute a cheapest $\mathcal{A} - \mathcal{B}$-path in G_x. If it satisfies constraints (8) – (9), then the updated rolling stock schedule resolves one off-balance in the cheapest possible way.

7 Results

In this section we report our computational results. All test instances are based on the generic Saturday and Sunday timetables of the so-called 3000 line of NS connecting Den Helder (Hdr) to Nijmegen (Nm). The stations are indicated in Figure 10. The line is operated twice an hour in both directions. The timetable contains about 500 trips on each day.

Composition changes are possible at the terminals as well as the intermediate stations Alkmaar (Amr) and Arnhem (Ah). Furthermore, units may start and finish their daily duties at Amsterdam (Asd) and Utrecht (Ut), although underway composition changes are not permitted at these two stations. The 3000 line has a closed rolling stock circulation and is currently serviced by 11 units of type VIRM4 and 24 units of type VIRM6; these are double-deck units with 4 or 6 carriages, respectively. The maximally allowed train length is 12 carriages, thus the VIRM types admit no more than 7 possible compositions for a trip, namely 4, 6, 44, 46, 64, 66, and 444.

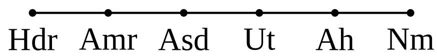


Figure 10: The 3000 line connecting Den Helder (Hdr) to Nijmegen (Nm) via Alkmaar (Amr), Amsterdam (Asd), Utrecht (Ut) and Arnhem (Ah).

In the first computational tests, referred to as V46, we considered the timetable on Sunday. We assumed that a certain part of the trajectory (either Amr-Asd or Asd-Ut or Ah-Nm) is closed either until 14:00 or for the entire Sunday. The reduced timetable has about 400 trips. In each of these six cases, we defined the rolling stock turn-arounds at the boundaries of the closed infrastructure by joining an arrival to the first possible departure. The solutions may have off-balances on Sunday morning and evening.

To illustrate the behavior of the solution methods under different preferences of the decision makers, we considered three different settings for the relative importance of the objective criteria (off-balances, carriage-kilometers, shortage-kilometers, number of composition changes). We refer to these as Obj-A, Obj-B and Obj-C. Table 1 contains the values of the coefficients in the objective functions. Besides heavily penalizing the remaining off-balances, the three cost structures prefer minimizing one of the other three optimization criteria: Obj-A focuses on carriage-kilometers, Obj-B on composition changes, and Obj-C on seat shortages.

Table 1: Objective coefficients in the three cost structures.

Criterion	Obj-A	Obj-B	Obj-C
Off-balance	1,000.0	1,000.0	1,000.0
Carriage-kilometers	0.050	0.005	0.005
Composition changes	0.010	20.000	0.010
Shortage-kilometers	0.015	0.010	0.015

We computed the input rolling stock schedules by the model of Fioole et al. (2006) with the objective functions Obj-A, Obj-B and Obj-C, setting the penalty of off-balances to zero; we used the commercial MIP software CPLEX to solve these integer programs. Then we applied the heuristic algorithms described in Sections 5 and 6 to resolve the off-balances. We also solved the model of Fioole et al. (2006) *with* the high penalties on the off-balances. That is, we computed three solutions for each of the 18 V46-instances.

So far, the planning period was one day. In the case when the infrastructure is blocked for an entire day, it is common in practice to modify the rolling stock schedules on the previous day. This leads to instances with a planning period of two days: Saturday and Sunday. To make the instance similar to practical instances, we allow off-balances at Sunday morning and evening but not at Saturday morning. That is, the off-balances are to be resolved by dead-heading at Saturday night and Sunday night. As input plan, we used the unchanged operational rolling stock schedules for Saturday; for Sunday, we took the same input plan as in the one-day instances. The 2-day test problems concern about 900 trips.

The VIRM4 and VIRM6 units have a limited number of possibilities to be attached to one another. Therefore in further artificial experiments, referred to as V23, we changed the rolling stock types used. We split each VIRM4 and VIRM6 unit into two identical parts (i.e. VIRM2 and VIRM3). This results in as much as 48 possible compositions for each trip, increasing the complexity of the problem significantly. For these artificial rolling stock types, we considered the same one- and two-day instances and the same solution methods as for the original rolling stock types.

Throughout this section, Heur-1 denotes the two-phase approach described in Section 5 as well as its results, while Heur-2 denotes the iterative approach described in Section 6 as well as its results.

For Phase 1 of Heur-1, we identified about 20 classes of balancing possibilities and collected about 10,000–30,000 possibilities. Thus the integer program in Phase 2 of Heur-1 has about 10,000–30,000 variables and 500–1,000 constraints. The graphs for Heur-2 have up to 5,800 nodes and up to 6,500 arcs.

The computations have been carried out on a PC equipped with a Pentium IV 3.0GHz processor and 1GB internal memory. We used CPLEX 9.0 with the modeling software ILOG Opl Studio 3.7. The heuristic algorithms have been implemented in the C language (Heur-1) and in the Perl language (Heur-2).

7.1 The quality of the solutions

We present in Tables 3 and 4 the number of unresolved off-balances and the objective values for all our solutions. It turns out that Heur-2 significantly outperforms Heur-1. This can be explained partially by the fact that in Heur-1 two selected balancing possibilities may not touch the same trip, even if the technical and market requirements would allow using both of them. Yet, in some cases the greedy method in Heur-2 terminates with a higher number of off-balances than Heur-1. However, Heur-1 appears to be able to balance the four optimization criteria better than Heur-2. Indeed, the contribution of carriage-kilometers, shortage-kilometers and shunting movements to the objective function (this is the 'Rest' column in Table 3 and 4) is often much higher in Heur-2 than in Heur-1. This is particularly true for the test problems with Obj-C.

The tables also show the numbers of unresolved off-balances and the objective values in the optimal solutions. Actually, these optimal solutions have the smallest possible numbers of off-balances.

We can observe that the quality of the solutions highly depends on the structure of the input plan (see Table 2). In Obj-B, the number of composition changes has the largest weight. Therefore the input plan (which was computed by CPLEX) contains a very small number of couplings and uncouplings. Then, the heuristic methods find ways to resolve many off-balances. The input plans for Obj-A and Obj-C are obtained by penalizing the carriage-kilometers and shortage-kilometers more heavily. The resulting larger number of composition changes is apparently disadvantageous for both heuristic methods.

The input plans of the V23 instances have much more off-balances than those of the V46 instances (almost twice as high on average). However, if one considers how much off-balances the heuristic solutions leave compared to the optimal solutions, it turns out that these differences are very similar in the V46 and V23 instances. For Heur-1, the average difference in the number of off-balances is 5.11 and 5.93 for V46 and V23, respectively, while for Heur-2 these averages are 2.78 and 2.89. That is, the heuristic algorithms perform relatively better on the V23 instances than on the V46 instances. This is not surprising: the shorter units give much more possibilities for adjustments without violating the constraints on the minimal and maximal lengths of the trains.

Table 2: The average difference between the off-balances in the heuristic solutions (OB_1 for Heur-1 and OB_2 for Heur-2) and in the optimal solution ($\mathrm{OB}_{\mathrm{opt}}$).

		$\mathrm{OB}_1 - \mathrm{OB}_{\mathrm{opt}}$	$\mathrm{OB}_2 - \mathrm{OB}_{\mathrm{opt}}$
V46	Obj-A	6.00	3.78
V46	Obj-B	4.00	1.00
V46	Obj-C	5.33	3.56
V46	total	5.11	2.78
V23	Obj-A	6.33	3.67
V23	Obj-B	3.00	0.89
V23	Obj-C	8.44	4.11
V23	total	5.93	2.89

7.2 Computation times

Recall that the main motivation for using heuristic algorithms is the need for a quick (suboptimal) solution process. The algorithm Heur-1 has a running time of 1–2 minutes for each of the test problem instances. Most of this time is spent on identifying the balancing possibilities, the CPLEX model in the second phase is solved within seconds. The algorithm Heur-2 also has a running time of 1–2 minutes.

In Tables 3 and 4 we give the computation times of the exact optimization method of CPLEX. Although this involves a relatively small instance of NS, it already shows how unpredictably the solution times grow when increasing the problem size. The V46 instances are easily solved for one-day planning horizons (these instances are denoted by 46-C*-H-* and 46-C*-F-* where * represents an arbitrary character) within 10–20 seconds. The two-day instances (46-C*-W-*) require more than 5 times more CPU time on average. The V23 instances have a much more complex combinatorial structure due to the higher number of possible compositions. The solution time ranges from 2 minutes to 45 minutes, and a particular two-day instance requires nearly 2 hours.

Table 3: Results for the V46 instances. The instance names are composed of the types used (V46), the closed infrastructure (C1 for Ah-Nm, C2 for Asd-Ut and C3 for Amr-Asd), the duration of the blockage (H for half of Sunday, F for the full Sunday, W for the whole weekend) and the objective function (A for Obj-A, etc.). 'Optimal' stands for the exact solutions by CPLEX. 'IOB' denotes the off-balance in the input plan, 'OB' the remaining off-balance, 'Rest' the contribution of carriage-kilometers, seat shortages and shunting movements, 'Obj' the objective value, and 'ST' the solution time (in seconds).

Instance		Heur-1			Heur-2			Optimal			
Name	IOB	OB	Rest	Obj	OB	Rest	Obj	OB	Rest	Obj	ST
46-C1-H-A	13	9	4,086	13,086	4	4,048	8,048	0	3,907	3,907	20
46-C1-H-B	8	3	607	3,607	2	624	2,624	0	545	545	24
46-C1-H-C	8	4	408	4,408	2	1,252	3,252	0	395	395	21
46-C1-F-A	11	8	4,106	12,106	8	3,707	11,707	3	3,660	6,660	15
46-C1-F-B	9	4	559	4,559	3	590	3,590	3	490	3,490	18
46-C1-F-C	13	7	563	7,563	5	3,772	8,772	3	379	3,379	15
46-C1-W-A	11	5	8,365	13,364	6	8,451	14,451	0	7,840	7,840	73
46-C1-W-B	9	4	1,686	5,686	1	1,875	2,875	0	1,165	1,165	74
46-C1-W-C	13	5	2,490	7,490	3	5,713	8,713	0	814	814	80
46-C2-H-A	11	5	4,032	9,032	3	3,747	6,747	0	3,654	3,654	19
46-C2-H-B	13	5	642	5,642	0	865	865	0	583	538	16
46-C2-H-C	12	5	390	5,390	3	1,766	4,766	0	371	371	19
46-C2-F-A	11	10	2,734	12,734	6	2,850	8,850	3	2,832	5,832	12
46-C2-F-B	11	6	419	6,419	4	516	4,516	3	428	3,428	12
46-C2-F-C	11	10	277	10,277	9	278	9,278	3	286	3,286	12
46-C2-W-A	11	8	7,551	15,551	3	7,917	10,917	0	7,088	7,088	103
46-C2-W-B	11	5	1,596	6,596	1	1,809	2,809	0	1,121	1,121	79
46-C2-W-C	11	8	2,392	10,392	6	3,268	9,268	0	728	728	86
46-C3-H-A	11	5	3,950	8,950	3	3,790	6,790	0	3,753	3,753	17
46-C3-H-B	12	4	661	4,661	0	849	849	0	557	557	17
46-C3-H-C	13	5	1,018	6,018	2	1,991	3,991	0	384	384	16
46-C3-F-A	11	8	3,029	11,029	6	2,935	8,935	4	2,961	6,961	12
46-C3-F-B	12	7	483	7,483	6	703	6,703	4	424	4,424	11
46-C3-F-C	11	8	308	8,308	7	774	7,774	4	302	4,302	11
46-C3-W-A	11	6	8,018	14,018	5	7,741	12,741	0	7,342	7,342	97
46-C3-W-B	12	8	1,671	9,671	2	2,142	4,142	0	1,178	1,178	71
46-C3-W-C	11	6	2,429	8,429	5	2,878	7,878	0	757	757	194

Table 4: Results for the V23 instances. Here we use the same notations as in Table 3.

Instance		Heur-1			Heur-2			Optimal			
Name	IOB	OB	Rest	Obj	OB	Rest	Obj	OB	Rest	Obj	ST
23-C1-H-A	23	7	3,556	10,556	2	3,928	5,928	0	2,961	2,961	730
23-C1-H-B	16	3	677	3,677	0	964	964	0	521	521	2,619
23-C1-H-C	23	11	570	11,570	5	6,568	11,586	0	302	302	310
23-C1-F-A	23	10	3,561	13,561	9	3,101	12,101	6	2,808	8,808	104
23-C1-F-B	22	7	646	7,646	6	762	6,762	6	451	6,451	133
23-C1-F-C	25	13	1,498	14,498	10	4,473	14,473	6	289	6,289	116
23-C1-W-A	23	5	7,488	12,488	3	7,894	10,894	0	5,824	5,824	715
23-C1-W-B	22	2	2,108	4,108	1	2,077	3,077	0	878	878	367
23-C1-W-C	25	6	3,604	9,604	4	5,805	9,805	0	595	595	1,173
23-C2-H-A	21	8	2, 906	10,906	4	3,345	7,345	0	2,782	2,782	2,758
23-C2-H-B	20	5	730	5,730	2	870	2,870	0	542	542	247
23-C2-H-C	21	8	1,512	9,512	4	5,144	9,144	0	284	284	2,096
23-C2-F-A	24	10	3,029	13,029	10	2,171	12,171	6	2,128	8,128	114
23-C2-F-B	17	6	1,274	7,374	7	909	7,909	6	391	6,391	253
23-C2-F-C	28	14	220	14,220	10	408	10,408	6	217	6,217	114
23-C2-W-A	24	8	6,836	14,836	6	6,918	12,918	0	5,167	5,167	1,878
23-C2-W-B	17	4	2,171	6,171	2	2,299	4,299	0	782	782	333
23-C2-W-C	28	9	2,308	11,308	6	2,505	8,505	0	528	528	6,810
23-C3-H-A	20	4	3,929	7,929	5	3,224	8,224	0	2,917	2,917	348
23-C3-H-B	21	6	726	6,726	0	1,337	1,337	0	555	555	272
23-C3-H-C	24	8	282	8,282	4	2,265	6,265	0	301	301	392
23-C3-F-A	22	16	2,369	18,369	9	2,669	11,669	8	2,363	10,363	132
23-C3-F-B	18	9	607	9,607	8	990	8,990	8	831	8,371	111
23-C3-F-C	24	17	241	17,241	9	3,868	12,868	8	240	8,240	204
23-C3-W-A	22	9	6,909	15,909	5	7,628	12,628	0	5,404	5,404	597
23-C3-W-B	18	5	2,722	7,722	2	2,181	4,181	0	802	802	367
23-C3-W-C	24	10	2,330	12,330	5	8,186	13,186	0	551	551	1,264

8 Conclusions

In this paper we formulated the Rolling Stock Balancing Problem (RSBP). This problem arises at various stages of the planning process of a passenger railway operator: from the short-term planning phase (i.e. planning some days or weeks ahead) till the real-time operations. Due to changes in the timetable (e.g. planned maintenance or unplanned disruptions) the previously created rolling stock schedules for a certain time period have to be adjusted.

Two heuristics have been developed to solve the RSBP. The performance of these algorithms are compared with the performance of the exact solution method used at NS, the main Dutch passenger railway operator. The comparison of the results is done on some (variants of) real-life instances of NS. These instances varied in size and complexity.

From the results presented in Section 7 we can conclude that both heuristics are very fast, even if the problem size is increased (two-day variants). The results also show that both heuristics can be effectively used not only for solving larger size problems, but they can be also used as a basis for solving real-time rescheduling problems.

References

A. Alfieri, R. Groot, L.G. Kroon, and A. Schrijver. Efficient Circulation of Railway Rolling Stock. *Transportation Science*, 40:378–391, 2006.

N. Ben-Khedher, J. Kintanar, C. Queille, and W. Stripling. Schedule Optimization at SNCF: From Conception to Day of Departure. *Interfaces*, 28:6–23, 1998.

J. Clausen, A. Larsen, and J. Larsen. Disruption Management in the Arline Industry: Concepts, Models and Methods. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, Richard Petersens Plads, Building 321, DK–2800 Kgs. Lyngby, 2005. URL `http://www2.imm.dtu.dk/pubdb/p.php?3763`.

P.J. Fioole, L.G. Kroon, G. Maróti, and A. Schrijver. A Rolling Stock Circulation Model for Combining and Splitting of Passenger Trains. *European Journal of Operational Research*, 174:1281–1297, 2006.

M.L. Fredman and R.E. Tarjan. Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms. *Journal of the ACM*, 34(3):596–615, 1987.

R. Freling, R.M. Lentink, L.G. Kroon, and D. Huisman. Shunting of Passenger Train Units in a Railway Station. *Transportation Science*, 39:261–272, 2005.

D. Huisman, L.G. Kroon, R.M. Lentink, and M.J.C.M. Vromans. Operations Research in Passenger Railway Transportation. *Statistica Neerlandica*, 59(4): 467–497, 2005.

J. Jespersen Groth, D. Potthoff, J. Clausen, D. Huisman, L. Kroon, G. Maróti, and M. Nyhave Nielsen. Disruption Management in Passenger Railway Transportation. Econometric Institute Report EI 2007-05, Erasmus University Rotterdam, The Netherlands, 2007. URL `https://ep.eur.nl/handle/1765/8527`.

R.M. Karp. Reducibility among Combinatorial Problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

N. Kohl, A. Larsen, J. Larsen, A. Ross, and S. Tiourine. Airline Disruption Management: Perspectives, Experiences and Outlook. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, Richard Petersens Plads, Building 321, DK–2800 Kgs. Lyngby, 2004. URL `http://www2.imm.dtu.dk/pubdb/p.php?3311`.

R.M. Lentink. *Algorithmic Decision Support for Shunt Planning*. PhD thesis, Erasmus University Rotterdam, The Netherlands, 2006.

J. Li, P.B. Mirchandani, and D. Borenstein. Vehicle Rescheduling Problem: Model and Algorithms. 2004. To appear in *Networks*.

N. Lingaya, J.F. Cordeau, G. Desaulniers, J. Desrosiers, and F. Soumis. Operational Car Assignment at VIA Rail Canada. *Transportation Research B*, 36: 755–778, 2002.

G. Maróti. *Operations Research Models for Railway Rolling Stock Planning*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2006.

M. Peeters and L.G. Kroon. Circulation of Railway Rolling Stock: a Branch-and-Price Approach. ERIM Research Report ERS-2003-055-LIS, Erasmus University Rotterdam, The Netherlands, 2003. To appear in *Computers and Operations Research*.