Efficient Circulation of Railway Rolling Stock

Arianna Alfieri, Rutger Groot, Leo Kroon, Lex Schrijver

ERIM REPORT SERIES *RESEARCH IN MANAGEMENT*	
ERIM Report Series reference number	ERS-2002-110-LIS
Publication	December 2002
Number of pages	18
Email address corresponding author	l.kroon@fbk.eur.nl
Address	Erasmus Research Institute of Management (ERIM) Rotterdam School of Management / Faculteit Bedrijfskunde Rotterdam School of Economics / Faculteit Economische Wetenschappen Erasmus Universiteit Rotterdam P.O. Box 1738 3000 DR Rotterdam, The Netherlands Phone: +31 10 408 1182 Fax: +31 10 408 9640 Email: info@erim.eur.nl Internet: www.erim.eur.nl

Bibliographic data and classifications of all the ERIM reports are also available on the ERIM website:
www.erim.eur.nl

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT

REPORT SERIES
RESEARCH IN MANAGEMENT

BIBLIOGRAPHIC DATA AND CLASSIFICATIONS		
Abstract	Railway rolling stock (locomotives, carriages, and train units) is one of the most significant cost sources for operatorsof passenger trains, both public and private. Rolling stock costs are due to material acquisition, power supply, and material maintenance. The efficient circulation of rolling stock material is therefore one of the objectives pursued. In this paper we focus on the circulation of train units on a single line. In order to utilize the train units on this line in an efficient way, they are added to or removed from the trains in certain stations, according to the passengers' seat demand. Since adding and removing train units has to respect specific rules, it is important to know the exact order of the train units in the trains. This aspect strongly increases the complexity of the rolling stock circulation problem. In this paper we present an integer programming approach to solve this problem. We also apply this approach to a real life case study based on the 2001-2002 timetable of NS Reizigers, the major Dutch operator of passenger trains.	
Library of Congress Classification (LCC)	5001-6182	Business
	5201-5982	Business Science
	HE 1821	Railways
Journal of Economic Literature (JEL)	M	Business Administration and Business Economics
	M 11 R 4	Production Management Transportation Systems
	R 41	Transportation
European Business Schools Library Group (EBSLG)	85 A	Business General
	260 K 240 B	Logistics Information Systems Management
	25 G	Transport
Gemeenschappelijke Onderwerpsontsluiting (GOO)		
Classification GOO	85.00	Bedrijfskunde, Organisatiekunde: algemeen
	85.34 85.20	Logistiek management Bestuurlijke informatie, informatieverzorging
	55.30	Railverkeer, railvervoer: algemeen
Keywords GOO	Bedrijfskunde / Bedrijfseconomie	
	Bedrijfsprocessen, logistiek, management informatiesystemen	
	Spoorwegen, personenvervoer, materieel, toewijzingsprobleem, operations research	
Free keywords	railways, public transportation, rolling stock circulation, operations research, logistics	

Efficient Circulation of Railway Rolling Stock

Arianna Alfieri
Politecnico di Torino, Dipt. Sistemi di Produzione ed Economia
Corso Duca degli Abruzzi 24, IT–10129, Torino, Italy

Rutger Groot
ORTEC Consultants
P.O. Box 490, NL–2800 AL Gouda, The Netherlands

Leo Kroon
NS Reizigers, Department of Logistics
P.O. Box 2025, NL–3500 HA Utrecht, The Netherlands

Rotterdam School of Management, Erasmus University Rotterdam
P.O. Box 1738, NL–3000 DR Rotterdam, The Netherlands

Lex Schrijver
Centrum voor Wiskunde en Informatica
P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands

Abstract

Railway rolling stock (locomotives, carriages, and train units) is one of the most significant cost sources for operators of passenger trains, both public and private. Rolling stock costs are due to material acquisition, power supply, and material maintenance. The *efficient* circulation of rolling stock material is therefore one of the objectives pursued. In this paper we focus on the circulation of *train units* on a single line. In order to utilize the train units on this line in an efficient way, they are added to or removed from the trains in certain stations, according to the passengers' seat demand. Since adding and removing train units has to respect specific rules, it is important to know the exact order of the train units in the trains. This aspect strongly increases the complexity of the rolling stock circulation problem. In this paper we present an integer programming approach to solve this problem. We also apply this approach to a real life case study based on the 2001-2002 timetable of NS Reizigers, the major Dutch operator of passenger trains.

1 Introduction

The efficient circulation of railway rolling stock is an important problem for operators of passenger trains, since the rolling stock represents a huge capital investment. The rolling stock is also an investment that cannot be changed frequently, because it usually has an economic life cycle of several decades. For these reasons, it has to be decided carefully how many carriages or train units are necessary per scheduled train in order to facilitate a service to the passengers of a certain quality. In this paper we focus on the efficient circulation of train units on a single line of NS Reizigers, the major Dutch operator of passenger trains.

Important objectives that are pursued in the planning process of the rolling stock circulation are to minimize the number of train units that are needed to provide a certain service to the passengers, and to minimize the number of train unit kilometers or wagon kilometers.

An efficient circulation of rolling stock involves an optimal match between the passengers' demand for transportation in each train and the rolling stock capacity that is offered there. In order to achieve this, the compositions of the trains may have to be changed at certain stations during the operations by adding train units to the trains or by removing them from the trains. For example, train units may be uncoupled from a train after the morning rush hours, and they may be coupled again before the afternoon rush, possibly onto another train. In this *shunting* process, which is carried out in the short time interval between the arrival of a train at a station and its subsequent departure, several practical rules related to the feasibility of the transition of the train composition are to be taken into account.

In the planning phase of the rolling stock circulation, one usually assumes the rolling stock circulation over a certain period of time (e.g. one week) to be cyclic. This typical period of one week is then split into periods of single days for which rolling stock circulations are planned. Circulations are usually planned on a line-by-line basis, so that a large part of the rolling stock is dedicated to it's own line. In the Netherlands, a line is usually called a *train series*. Throughout this paper we will use this terminology therefore. The single day circulations may be non-cyclic due to the fact that the timetable or the passengers' seat demand varies over the day and over the days of the week. In a next step, the single day circulations are connected with each other into a cyclic circulation for a complete week. The latter may be accomplished by modifying the initial single day circulations in the late evenings, or by adding repositioning trips during the nights. The weekly rolling stock circulation also contains time that is reserved for maintenance of the rolling stock. In the current paper we describe a model based method for solving the rolling stock circulation problem for a single train series on a single day.

In the literature several related problems have been treated, which testifies the importance of the rolling stock circulation problem. Brucker et al. [4] study the problem of finding a routing of railway carriages through a network, given a certain timetable. They focus on the repositioning trips of carriages from one train to another. Their approach is based on local search techniques like simulated annealing. Also Van Montfort [10] focuses on the efficient circulation of railway carriages. He studies the assignment of carriages to trains, given a timetable and a "core standard structure" for train compositions on a combination of train series in the Netherlands. Lingaya et al. [8] deal with the problem of assigning carriages to trains at VIA Rail in Canada. They allow for coupling and uncoupling of carriages and locomotives at various locations in the network. They explicitly take into account the order of the carriages in the trains. Their solution approach is based on a branch-and-bound method and on column generation.

Ben-Khedher et al. [3] study the problem of allocating identical train units to the French High Speed Trains. Their rolling stock allocation system is directly linked to the seat reservation system. Schrijver [12] considers the problem of minimizing the number of train units of different subtypes for an hourly train series in the Netherlands, given demand satisfaction constraints and certain coupling and uncoupling possibilities. In this paper, the only restrictions on the transition of a train composition to another one are posed by flow conservation. The same train series is dealt with by Groot [6], with the difference that now also several practical restrictions on the transition possibilities of the train compositions are taken into account.

In particular, the last problem gave rise to the current paper. Groot [6] describes an interesting algorithm based on Integer Programming. The presented computational tests show positive results for the train series dealt with (Amsterdam-Vlissingen). His approach is based on modelling the feasible transitions of the compositions of the trains in a number of so-called *transition graphs*. The aim is then to find an appropriate path in each transition graph and at the same time to keep track of the stocks of the train units in the different stations. A key step in this approach is a preprocessing phase in order to reduce the sizes of the transition graphs. However, from the obtained results it is not clear if this approach can be extended to different cases which are larger or more complex, or have different peculiarities.

This subject is investigated in the current paper. In particular, the case study that we consider here is based on the 3000 train series of NS Reizigers, which provides twice per hour an Intercity connection from Den Helder (Hdr) to Nijmegen (Nm) and vice versa (see Figure 1). The fact that this train series has a frequency of twice per hour increases the complexity of the case significantly in comparison with the case dealt with by Groot [6]. In order to be able to deal with this more complex case, we modified the model described in [6], resulting in a more compact model with a lower number of variables and constraints.

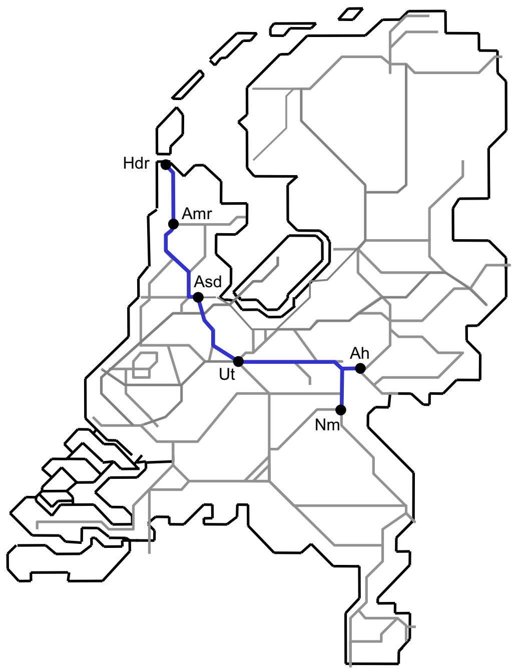

Figure 1: The 3000 train series Den Helder - Nijmegen

The current paper is organized as follows. A detailed description of the problem is given in Section 2. In Section 3, we describe the model formulations of the rolling stock circulation problem neglecting the compositions and the problem taking into account the compositions. Section 4 deals with the solution approach based on a combination of a multi-commodity flow problem and finding paths through transition graphs. The results of our computational experiments are presented in Section 5. Finally, in Section 6, some conclusions are drawn.

2 Problem description

Before we go into the details of the problem, we first describe some background information pertaining to the rolling stock circulation of NS Reizigers.

2.1 Background information

In the Netherlands, three different categories of trains for passenger transportation can be distinguished: Intercity trains, Inter-Regional trains, and Regional trains. The line system contains the train series (lines), each of which connecting an origin station and a destination station with a certain fixed frequency. Each train series belongs to one of the three train categories. The timetable gives the arrival and departure times of the trains at the relevant stations. The timetable of NS Reizigers is quite dense, and it is cyclic with a cycle length of one hour.

NS Reizigers has a large variety of rolling stock available for passenger transportation. The major part of the rolling stock consists of train units of different types and subtypes. Each train unit consists of a certain number of wagons that cannot be split from each other during the daily operations (see Figure 2).

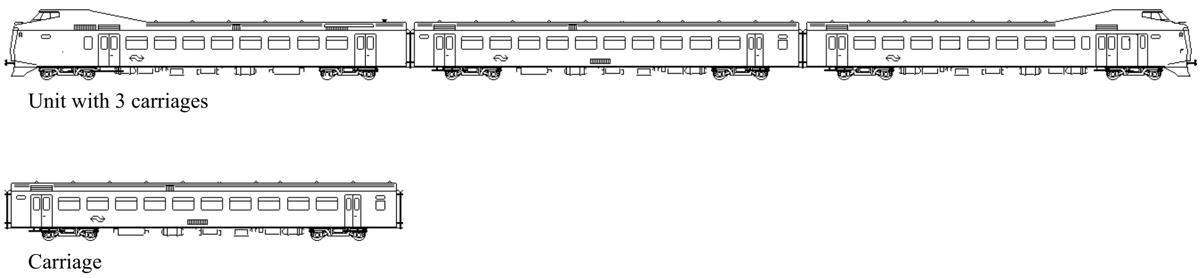

Figure 2: A train unit with 3 wagons and a single carriage

The main difference between train units and carriages is that train units can move individually in both directions without a locomotive, while carriages can not: carriages need a locomotive. Furthermore, each train unit has a fixed mix of first class and second class capacity. The different train types (i.e. single deck or double deck) can be subdivided subsequently into different subtypes. For example, single deck train units can be subdivided into train units with 3 wagons and train units with 4 wagons.

Train units of different train types cannot be combined with each other into one single train. However, train units of the same type, but possibly of different subtypes, may be combined with each other into a train. The latter implies that per subtype not only the number of train units in a train is to be determined, but also their order within the train. Indeed, the transition possibilities from a certain train composition to another one strongly depend on the order of the train units of the different subtypes within the train. In the following, we indicate an ordered sequence of train units in a train as a *composition*.

For example, if X and Y denote train units of the same type but of different subtypes, then XYY and YXY indicate two possible compositions. It is clear that, from a capacity point of view, they are identical. However, these compositions have different transition possibilities, since the train unit of subtype X cannot be uncoupled easily from the train with composition YXY, whereas the latter is not the case for the train with composition XYY. Furthermore, coupling train units onto a train or uncoupling train units from a train is carried out usually at a specific side of the train, depending on the local circumstances: a train unit is usually coupled to the *front* side of an incoming train, and uncoupled from the *rear* side of a leaving train. By coupling or uncoupling train units at the indicated side of the train, the coupled or uncoupled train unit disturbs the movements of the ongoing train as few as possible.

The fact that the train types can be subdivided into different subtypes complicates both the planning process and the daily operations significantly. However, the *gain* is that per train a better match between the expected number of passengers (demand) and the available number of seats (supply) can be realized. Indeed, whereas with train units with, for example, 3 wagons only trains with 3, 6, 9, etc. wagons can be composed, a combination of train units with 3 and 4 wagons may give rise to trains with 3, 4, 6, 7, 8, etc. wagons. Of course, we cannot combine any number of train units, since a train should not be longer than the shortest platform of the stations along its route, and certainly not longer than 12 wagons.

2.2 The 3000 train series

Usually, a train series is served by a fixed number of trains. Each train runs up-and-down between the two end points of the train series. The number of trains on a train series is determined by the circulation time of the train series, including the return times at the endpoints, and its frequency. For example, the Intercity train series 3000, running between Den Helder and Nijmegen contains 12 trains. This is caused by the fact that the circulation time on the train series between Den Helder and Nijmegen and vice versa is six hours and that there are two trips per hour in each direction. Hence every twelfth departure from, say, Nijmegen, can be covered by the same train. Figure 3 shows a time-space diagram for part of the trips of this train series. The numbers at the top of the figure indicate the time axis. The diagonal lines indicate the trips and the adjacent numbers are the train numbers. Each short line, close to a trip line, represents one train unit.

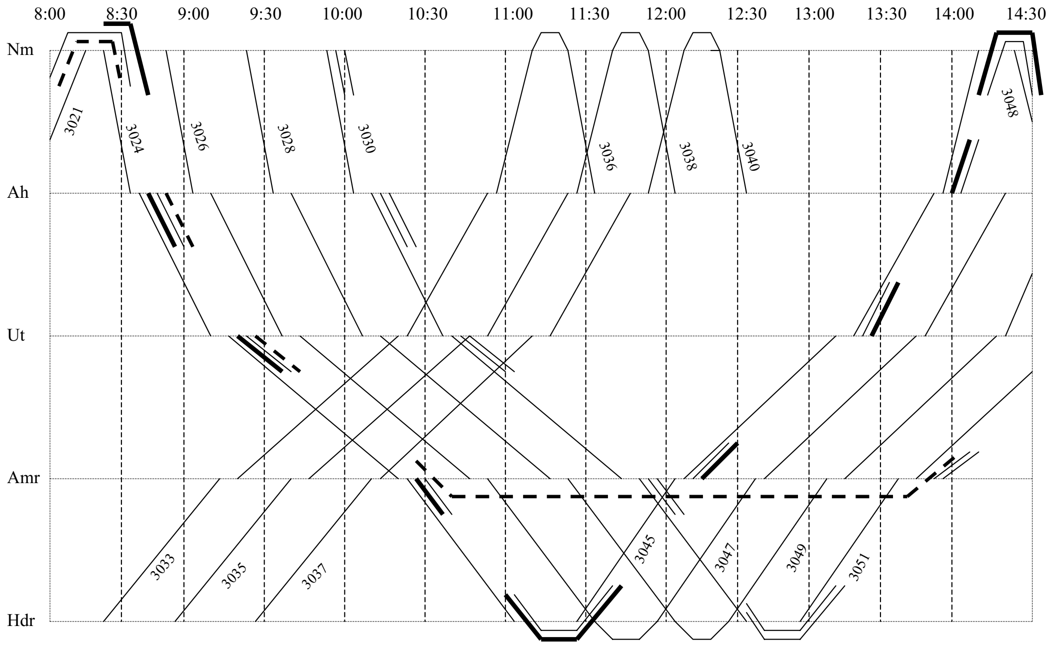

Figure 3: Part of rolling stock circulation for the 3000 train series

In order to keep the graph in Figure 3 as simple as possible, we represent only a subset of the trains, and we show the train units for only some of these trains. For example, the train on trip 3021 is composed of 2 train units (solid thin and dashed lines). It arrives in Nijmegen at 8:11, and at 8:20 it returns to Den Helder on trip 3024. In Nijmegen a train unit (solid thick line) is added, and then the train has 3 train units on trip 3024. The front train unit of a train is represented in the graph by the line that is closer to the trip line.

Train units are usually coupled onto the front of an incoming train. Nijmegen is a terminal station, where front and rear are exchanged. Thus the added train unit becomes the rear of the outgoing train. In Arnhem, the composition of a train is usually not changed, but front and rear are always exchanged, as is shown in Figure 3. The latter is caused by the fact that the trains leave the station Arnhem from the same side as where they entered it. In Alkmaar, the composition of a train can be changed: the rear train unit of the train on trip 3024 is uncoupled and stays there. The remaining train continues to Den Helder where it arrives at 11:06. At 11:27 it returns to Nijmegen on trip 3045 with a reversed composition. This process continues until the end of the day. Train units that have been removed from a certain train can be added to another train. Figure 3 shows how the train unit uncoupled from the rear side of the train on trip 3024 at 10:23 in Alkmaar is coupled to the front side of the train on trip 3051 at 13:37.

2.3 Objectives

Given the system described above, finding an appropriate rolling stock circulation means finding a balance between several objectives, such as minimizing the number of train units required for operating the train series and minimizing the variable rolling stock costs.

NS Reizigers is faced with a shortage of rolling stock capacity currently (2002). Therefore, it is important to use the rolling stock as efficiently as possible, especially during the rush hours. Since the number of passengers on a trip has a stochastic character and since there is no seat reservation system, it is impossible to *guarantee* a seat for all passengers, especially during the rush hours. However, outside the rush hours, the rolling stock capacity is usually sufficient to provide all passengers with a seat.

For forecasting the required capacity per trip, the passengers are counted continuously by the conductors. The translation of the counts of the conductors into the minimally required first and second class capacity per trip is based on a statistical procedure, which falls outside the scope of this paper.

The variable rolling stock costs are obviously related to the power supply (electricity or diesel), but also to the maintenance of the rolling stock: after a certain number of kilometers, each train unit is directed to a maintenance station for a preventive check-up and possibly for a repair. Therefore, the maintenance costs are correlated with the number of train unit or wagon kilometers. However, a decrease in the number of train unit or wagon kilometers does not only lead to a decrease in maintenance *costs*, but also to a decrease in the maintenance *requirements*. Less maintenance may lead to an increase in the effectively available rolling stock capacity. Other variable rolling stock costs are related to the crew: each train requires at least one driver and one conductor, but if the length of a train exceeds a certain threshold, then a second (or third) conductor is required for safety reasons. Thus, wherever this is possible, it is advisable to keep the length of the train below this threshold.

The model described in this paper deals with several of these conflicting objectives. For example, we will minimize the number of train units, given constraints on passengers' demand satisfaction, and we will minimize the number of train unit or wagon kilometers, given certain constraints on the number of available train units.

3 Model formulation

In this section we describe the model used for solving the rolling stock circulation problem. We assume to have a single train series connecting two end stations with a given frequency. The timetable on this train series is fixed and the number of trains that are running on this train series depends on the circulation time of the train series and on its frequency. We assume that all trains are operated by train units of the same type, but possibly of different subtypes. It follows that the problem is a complex case of a multi-commodity flow problem. In particular, the problem could be called an *ordered* multi-commodity flow problem, since the order of the train units in the trains is as important as their number.

For single commodity flow problems there are many algorithms available (Ahuja et al. [1]), but for multi-commodity flow problems this is not the case. Furthermore, the available algorithms almost always assume that variables may have fractional values (Ahuja et al. [1], McBride [9]), which is not true in our case. In case of integrality constraints on the flows, single commodity problems are much easier to solve than multi-commodity flow problems. Solution approaches for integer multi-commodity flow problems are studied by Barnhart et al. [2], based on branch-and-price and branch-and-price-and-cut algorithms, but they do not deal with the *ordered* multi-commodity flow problem.

In the following, a mathematical model for solving the rolling stock circulation problem is presented. The problem clearly consists of a part that is a multi-commodity flow problem (neglecting the compositions), and a part related to the compositions.

3.1 Neglecting the compositions

As was noted earlier, if we neglect the order of the train units in the trains, then the problem can be represented by a multi-commodity flow model, with a number of additional constraints. This model is represented by a *flow graph* (see Figure 4), whose nodes correspond to events (arrival or departure of a train in a station) and arcs are connections between events. These arcs represent trips if the two connected nodes belong to different stations (represented by solid lines in Figure 4), while they are station arcs if the two events belong to the same station (represented by dashed lines in Figure 4).

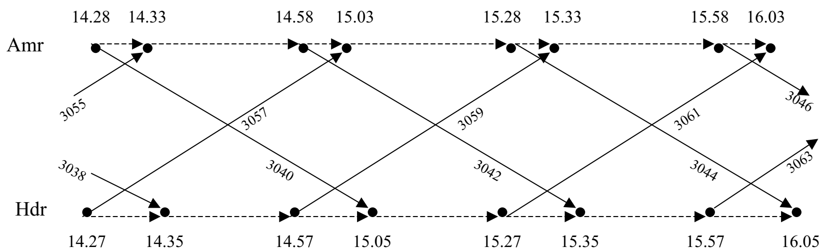

Figure 4: Part of the flow graph for the 3000 train series

A station arc represents a connection between the arrival (or departure) of a train in a station and the next arrival (or departure) of a train in the same station. In each node we have to assure a flow balance. In Figure 4, for example, the number of train units in Alkmaar before the arrival of the train on trip 3059 plus the number of train units arriving in the train on trip

3059 must be equal to the number of train units in Alkmaar before the departure of the train on trip 3046. Note that the network may be aggregated to some extent. For example, an arrival node in a station and the next departure node in the same station may be aggregated into one node, thereby reducing the number of nodes and the number of station arcs.

In mathematical terms, we have a set T of trips, a set S of stations, and a set J of subtypes of train units. Each trip $t \in T$ is represented by an origin station O_t, a destination station D_t, a start time S_t, and an end time E_t. Furthermore, the parameters $d_{c,t}$ ($c = 1,2$) denote the expected number of passengers in class c on trip t. The set T_t^a is the subset of trips t' arriving in O_t before the departure of trip t from O_t. That is, $T_t^a = \{t' \in T | D_{t'} = O_t, E_{t'} < S_t\}$. Note that it is not difficult to take into account a certain minimum re-assignment time between the uncoupling of a train unit from a train and the subsequent coupling of this train unit onto another train. However, we have omitted this re-assignment time here. Similarly, T_t^d is the subset of trips t' departing from station O_t before the departure of trip t. Thus $T_t^d = \{t' \in T | O_{t'} = O_t, S_{t'} < S_t\}$. The shortest platform along trip t is denoted by M_t. The number of wagons in each train unit of subtype j is denoted by N_j. Next, the parameter $C_{j,c}$ represents the capacity in class c ($c = 1,2$) of each train unit of subtype j.

The model is expressed in terms of the decision variables $x_{t,j}$ indicating the number of train units of subtype j that is allocated to trip t. Additional decision variables are the variables $y^0_{s,j}$ denoting the number of train units of subtype j that is stored in station s during the night, and $y_{tot,j}$ denoting the total number of available train units of subtype j. Now the model neglecting the compositions can be expressed as follows:

Model 1

$$\min F(x,y) \qquad \text{s.t.}$$

$$y_{tot,j} = \sum_s y^0_{s,j} \qquad \forall j \in J \tag{1}$$

$$x_{t,j} \leq y^0_{s,j} + \sum_{t' \in T_t^a} x_{t',j} - \sum_{t' \in T_t^d} x_{t',j} \qquad \forall j \in J, t \in T, s = O_t \tag{2}$$

$$\sum_j N_j x_{t,j} \leq M_t \qquad \forall t \in T \tag{3}$$

$$\sum_j C_{j,c} x_{t,j} \geq d_{c,t} \qquad \forall t \in T, c = 1,2 \tag{4}$$

$$x_{t,j} \in \{0, \ldots, M_t\} \qquad \forall t \in T, j \in J \tag{5}$$

$$y_{tot,j}, y^0_{s,j} \in \mathcal{Z}^+ \qquad \forall s \in S, j \in J \tag{6}$$

In this model, $F(x,y)$ is the objective function that we want to minimize. The details of this objective function are provided in Section 5. The total number of required train units of each subtype equals the total number of train units $y^0_{s,j}$ that stay in the various stations during the night, as is represented by constraints (1). Constraints (2) describe that the number of train units of subtype j that is allocated to trip t should not exceed the number of train units of this subtype that is available in station O_t just before the start time of trip t. The latter equals the number of such train units by the start of the day plus the number of train units that have arrived in this station until this time instant, minus the number of train units that have departed from there until then. On each trip t a train should not be longer than the shortest platform M_t along the trip. The latter is guaranteed by constraints (3). Constraints (4) are the demand satisfaction constraints for first class and second class seat demands on each trip. Finally, constraints (5) and (6) specify the integer character of the decision variables.

3.2 Valid Inequalities

In this subsection we describe how the constraints on maximum train length (3) and demand satisfaction (4) can be made more tight by adding certain *valid inequalities*. In fact, due to the integrality of the number of train units per trip, a set of better constraints can be created by taking the convex hull of the integer feasible points in each polyhedron (one for each trip t) described by constraints (3) and (4). For example, the model may contain the following constraints (7) and (8) for a certain trip t:

$$3x_{t,3} + 4x_{t,4} \leq 12 \tag{7}$$

$$166x_{t,3} + 224x_{t,4} \geq 510 \tag{8}$$

These constraints represent the fact that a train should have a length of at most 12 wagons, and that the second class seat demand is to be satisfied. Here the number 510 is the required number of second class seats on trip t, and the numbers 166 and 224 represent the second class capacities of train units with 3 and 4 wagons, respectively. Due to the integrality of the variables $x_{t,3}$ and $x_{t,4}$, the constraint (8) can be sharpened as follows:

$$x_{t,3} + 2x_{t,4} \geq 4$$

$$x_{t,3} + x_{t,4} \geq 3$$

The above example is shown in Figure 5. Here the grey and stripped areas correspond to the feasible region for the train composition on trip t obtained by considering constraints (7) and (8). The black dots represent the feasible combinations of the two subtypes of train units (Unit 3 and Unit 4, respectively). The stripped area represents the convex hull of the feasible region.

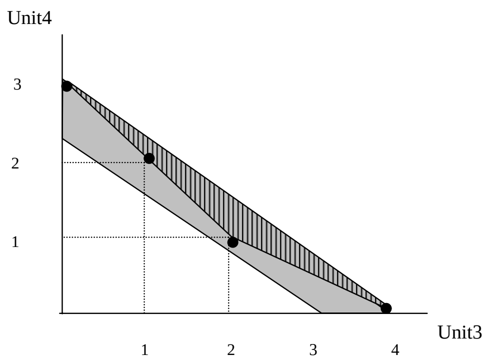

Figure 5: Reduced feasible region

Obviously, the *sum* of all such convex hulls does not correspond to the convex hull for the complete problem. Sometimes the number of valid inequalities is larger than the number of original constraints, as is shown in the example. Anyway, the improvement in the performance of the algorithm is quite high in general, although there are cases for which the addition of these valid inequalities has a detrimental effect on the overall CPU time (see Section 5).

3.3 Taking into account the compositions

In order to take the details of the compositions of the trains into account as well, many *additional* decision variables and constraints are required. Moreover, all the new decision variables are binary, which makes the solution process more complex.

For representing the compositions of a train on subsequent trips, a *transition graph* is used for each train. In such a graph, nodes correspond to feasible compositions for a certain trip, while arcs represent feasible transitions between compositions on consecutive trips (see Figure 6). For example, a transition from composition 33 to composition 334 may be allowed (depending on the location), since it implies only coupling a train unit with 4 wagons on the right side of the train. A transition from composition 33 to composition 34 is not allowed, since this would imply both uncoupling a train unit with 3 wagons and coupling a train unit with 4 wagons. Usually such time consuming transitions are not allowed in practice. Furthermore, each transition graph has a source node, connected to all nodes of the first trip of the corresponding train, and a sink node, connected to all nodes of the last trip of this train.

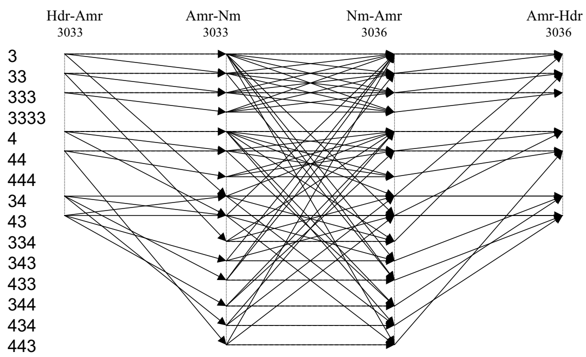

Figure 6: Part of the transition graph for one train of the 3000 train series

There is a transition graph for each train. At first sight, this seems to give a decomposition of the problem across the trains. However, this is not the case, since the trains interact with each other via the stocks of train units in the stations. Train units removed from a train can be added to another one, as was shown in Figure 3.

The set of all feasible compositions is denoted by K, and the set of compositions that are feasible for trip t is denoted by K_t. The parameter $g_{k,j}$ denotes the number of train units of subtype j in composition k. For each trip t, the trip $p(t)$ is the preceding trip of the same train that carries out trip t. Next, for each trip t and composition k, the set $A_{t,k}$ denotes the set of compositions that are feasible on trip $p(t)$ if the train has composition k on trip t. In other words, the elements of the sets $A_{t,k}$ correspond with the arcs of the transition graphs.

In the model, binary decision variables are associated with the *nodes* in the transition graphs. A decision variable $a_{t,k}$ assumes the value 1 if and only if the corresponding node is chosen in the solution. This means that the corresponding composition k is selected for the train on the corresponding trip t.

Now we have to find in each transition graph a path between the source node of the graph to the sink node of the graph, which altogether minimize the objective function $F(x, y)$. Only one node is to be selected for each trip and all nodes in consecutive trips of the path should be compatible with each other. This means that constraints on demand satisfaction and capacity are satisfied as well as the coupling and uncoupling restrictions between two consecutive trips. Now the model taking into account the compositions can be represented as follows:

Model 2

$$\min F(x,y) \qquad \text{s.t.}$$

$$y_{tot,j} = \sum_s y^0_{s,j} \qquad \forall j \in J$$

$$x_{t,j} \leq y^0_{s,j} + \sum_{t' \in T^a_t} x_{t',j} - \sum_{t' \in T^d_t} x_{t',j} \qquad \forall j \in J, t \in T, s = O_t$$

$$\sum_{k \in K_t} a_{t,k} = 1 \qquad \forall t \in T \tag{9}$$

$$a_{t,k} \leq \sum_{k' \in A_{t,k}} a_{p(t),k'} \qquad \forall t \in T, k \in K_t \tag{10}$$

$$x_{t,j} = \sum_{k \in K_t} g_{k,j} a_{t,k} \qquad \forall t \in T, j \in J \tag{11}$$

$$x_{t,j} \in \{0, \dots, M_t\} \qquad \forall t \in T, j \in J$$

$$y_{tot,j}, y^0_{s,j} \in \mathcal{Z}^+ \qquad \forall s \in S, j \in J$$

$$a_{t,k} \in \{0,1\} \qquad \forall t \in T, \forall k \in K_t \tag{12}$$

Note that several constraints of Model 1 also appear in Model 2, except for constraints (3) and (4), which are handled now by constraints (9). These constraints represent the fact that for each trip exactly one *appropriate* composition is to be selected. Constraints (10) guarantee that the composition that is selected for trip t is compatible with the composition that is selected for the preceding trip $p(t)$ of the same train. Note that for trip t we have $a_{t,k} = 1$ for exactly one appropriate composition k, according to constraints (9). Therefore, the *inequality* in constraints (10) cannot cause a problem. Next, constraints (11) link the flow graph and the transition graph: for each subtype, the number of train units on a trip follows directly from the composition that is used on that trip. In fact, these constraints make the flow variables $x_{t,j}$ superfluous, since in all occurrences of these variables they can be removed by substituting constraints (11). If this substitution is applied consequently, then obviously constraints (11) become superfluous as well. Finally, constraints (12) specify the binary character of the composition variables $a_{t,k}$.

4 Solution Approach

For real life cases, the dimensions of the Integer Programs presented in Section 3 are quite large, in particular those of the model taking into account the compositions. A commercial solver may not be able to solve them in an acceptable CPU time. Even tailored methods could have difficulties with efficiently and effectively solving the whole problem. However, the overall problem can obviously be partitioned into a *flow* part and a *composition* part. Therefore, a solution method can be based on this decomposition of the problem into subproblems.

Neglecting the compositions, the problem is an integer multi-commodity flow problem with several additional constraints. Although the solution of the model neglecting the compositions may be infeasible for the real problem, it provides a (usually strong) lower bound for the optimal value of the problem taking into account the compositions. For this reason, the solution approach is based on a *hierarchical decomposition* into subproblems.

The hierarchical decomposition scheme works as follows. After the model neglecting the compositions (Model 1) has been solved, we check if there exists a solution for the problem taking into account the compositions (Model 2) with the same value of the objective function $F(x^*, y^*)$. The latter is done by first solving a series of subproblems. These subproblems correspond to fixing a variable $x_{t,j}$ to all the feasible *values* it can assume. This is done for each trip t and for each subtype j. However, the algorithm does not check all the feasible *compositions*, since this would be an unnecessary amount of overwork. In fact, in Model 1 only the number of train units of each subtype is relevant. Examining all possible compositions would require to solve the problem for a certain value $x_{t,j}$ more than once (i.e. compositions $c_1 = 334$, $c_2 = 433$ and $c_3 = 343$ have $g_{c,3} = 2$, $g_{c,4} = 1$, for $c = c_1, c_2, c_3$).

For each of the subproblems to be solved, we only have to solve its Linear Programming relaxation and not the complete Integer Program. Indeed, we just want to check if a certain value for a variable $x_{t,j}$ is compatible with the initially obtained solution for Model 1. If the Linear Programming relaxation is infeasible, then the same holds for the Integer Program as well. If a certain value for a variable $x_{t,j}$ is infeasible in this elimination phase, then all compositions corresponding to that value are deleted for trip t. Also the corresponding nodes in the transition graph are eliminated. This kind of elimination is called *subproblem elimination*.

After the subproblem elimination phase, some nodes in the transition graph may have become disconnected from the nodes in the adjacent trips. In that case, no path in the transition graph from the source node to the sink node may pass through these nodes. We then perform *disconnection elimination* in order to eliminate all the nodes that have become disconnected in this way. During the disconnection elimination phase, other nodes may become disconnected and so this phase may be carried out until no more nodes are eliminated.

Once the disconnection elimination phase has been terminated, the algorithm could start another round of subproblem elimination. This process could be carried out until the number of nodes left in the transition graph drops below a certain value, depending on the particular case. However, each iteration has a certain cost in terms of CPU time. In our case, this cost turned out to be high, compared with the improvement that we obtained in terms of the number of additionally eliminated nodes. Therefore, we carried out the elimination phase only once.

If at the end of the elimination process at least one connected node for each trip remains, then the feasible compositions are used as input for the complete model taking into account the compositions. Otherwise, if there exists at least one trip whose nodes have all been eliminated, then obviously no feasible solution exists with the objective function value found for the model neglecting the compositions (i.e. $F(x^*, y^*)$). In this case, the constraint

$$F(x, y) \geq F(x^*, y^*) + \varepsilon$$

is added to Model 1 and the solution process is restarted. Here ε depends on the objective function $F(x, y)$, i.e. if we are minimizing the total number of wagons, then $\varepsilon = 1$.

5 Computational Experiments

In this section we present the computational results that we obtained by applying the algorithm described in Section 4 to the case of the 3000 train series, which was shown in Figure 1. Some of the peculiarities of this train series were described in Section 2. The data that we used for our computational results correspond to a standard Tuesday from the 2001-2002 timetable of NS Reizigers. For the solution of the Integer Programming models and their Linear Programming relaxations we used CPLEX (version 6.6) on a RISC 6000 workstation.

5.1 Experiments

We consider one train type only, namely single deck train units, for which two subtypes are available: train units with 3 wagons and train units with 4 wagons (see Figure 2). The maximum train length on this train series is 12 wagons. Each train unit of a certain subtype has a fixed capacity in terms of the numbers of first and second class seats.

In order to evaluate the influence of using the valid inequalities of the capacity constraints instead of the constraints themselves, we experimented with both problem formulations.

We further solved the instances for a whole day, but also for just the trips during the morning rush hours. In fact, the number of train units is mainly determined by the passengers' seat demand during the morning rush hours. Note that the peak in the morning rush hours is usually higher than the peak in the afternoon rush, which lasts longer.

For the station Alkmaar, we also studied further coupling and uncoupling restrictions. In the direction from Nijmegen to Den Helder, usually a train unit is uncoupled from a train in Alkmaar or the train remains of the same length. In the reverse direction, from Den Helder to Nijmegen, usually a train unit is coupled to a train or the train remains of the same length. Hence, if t_1 and t_1' are trips Nijmegen-Alkmaar and Alkmaar-Den Helder on a certain train, and t_2 and t_2' are trips Den Helder-Alkmaar and Alkmaar-Nijmegen on a certain train, then:

$$x_{t_1',j} \leq x_{t_1,j} \qquad \forall j \in J \tag{13}$$

$$x_{t_2',j} \geq x_{t_2,j} \qquad \forall j \in J \tag{14}$$

We experimented with the model both including these additional restrictions in Alkmaar and without them, in order to evaluate their influence on the solution quality and the CPU time.

Furthermore, in order to evaluate the flexibility of the algorithm, we used the following four objective functions:

1. PB1: minimize the total number of wagons ($\min \sum_j N_j y_{tot,j}$).

2. PB2: minimize the total number of wagon kilometers ($\min \sum_t L_t(\sum_j N_j x_{t,j})$, where L_t represents the length of trip t).

3. PB3: minimize the total number of wagon kilometers with an upper bound UB on the total number of wagons ($\min \sum_t L_t(\sum_j N_j x_{t,j})$ s.t. $\sum_j N_j y_{tot,j} \leq UB$).

4. PB4: minimize the total number of train unit kilometers with an upper bound UB on the total number of wagons ($\min \sum_t L_t(\sum_j x_{t,j})$ s.t. $\sum_j N_j y_{tot,j} \leq UB$).

Here UB is the optimal objective function value of problem PB1 with restrictions in Alkmaar. Minimizing the number of wagons and minimizing the total number of wagon kilometers (problems PB1 and PB2, respectively) are opposite objectives, since increasing the number of wagons may reduce the number of wagon kilometers and vice versa. To take both these objectives into consideration, problems PB3 and PB4 have been solved. In practice, one of the two objectives is kept as the objective, while the other is used as a constraint.

5.2 Results

In Table 1, the dimensions of the 3000 train series case study are reported in terms of the numbers of decision variables (*columns*), constraints (*rows*) and *non-zeros* in the constraint matrix. The labels *no-comp* and *comp* refer to the model neglecting the compositions (Model 1) and to the model taking into account the compositions (Model 2). Furthermore, the labels *complete* and *MRH* indicate the problem with all the trips in the timetable and the problem reduced to only trips leaving from each station before 10.30 am (called Morning Rush Hours).

	no-comp, complete	no-comp, MRH
columns	1544	494
rows	1755	544
non-zeros	4132	1290
	comp, complete	comp, MRH
columns	14142	3716
rows	6380	1894
non-zeros	66017	16182

Table 1: Dimensions of the 3000 train series problem

Table 1 shows how the problem characteristics change when passing from Model 1 to Model 2. In Model 1 (*no-comp*) the number of decision variables is smaller than the number of constraints, while in Model 2 (*comp*) the situation is the opposite. In case of Model 2, the numbers of decision variables and constraints, added to represent compositions, depend on the number of nodes eliminated in the transition graph, and thus also on the objective function used. The values in Table 1 are average values observed in our computational experiments.

Tables 2 to 5 report the numerical results of our computational experiments on the data of the 3000 train series. In each of these tables, $y_{tot,3}$ and $y_{tot,4}$ represent the number of train units with 3 wagons and the number of train units with 4 wagons respectively. The number *UB* is an upper bound on the decision variables $y_{tot,3}$ and $y_{tot,4}$, while *iter* represents the number of iterations the algorithm needed in order to find a solution for Model 2 that is compatible with the solution for Model 1 (see Section 4). The columns *res* and *no-res* refer to the problem with and without additional restrictions in Alkmaar. The columns *cut* and *no-cut* refer to the problems with valid inequalities and without. The rows *tot-nodes* and *elim-nodes* contain the total number of nodes in the transition graph and the number of nodes eliminated by subproblem and disconnection elimination, respectively. The difference between these two numbers gives the number of nodes in the reduced graph used to solve the problem.

Using the formulation with additional valid inequalities has the advantage to speed up the solution process of almost all the problems, due to an improvement in the value of the lower bound obtained from the Linear Programming relaxation. In some cases, however, the total CPU time is higher for the case with valid inequalities, due to the larger number of iterations made during the solution process of Model 2 (row *iter*).

In some cases, the solution found by using the formulation with valid inequalities is better also in terms of the value of the objective function. This is due to the upper bound added on the

	no-cut, no-res	cut, no-res	no-cut, res	cut, res
CPU (sec)	4401.1	955.9	812.1	2132.1
objval	110	110	112	112
$y_{tot,3}$	18	22	20	20
$y_{tot,4}$	14	11	13	13
iter	1	1	1	4
UB $y_{tot,3}, y_{tot,4}$	23	100	23	100
tot-nodes	13534	13534	8604	8604
elim-nodes	936	6782	515	1950
CPU (sec)	45.8	31.9	39.3	93.1
objval	110	110	112	112
$y_{tot,3}$	22	18	20	20
$y_{tot,4}$	11	14	13	13
iter	1	1	1	3
UB $y_{tot,3}, y_{tot,4}$	23	100	23	100
tot-nodes	3804	3804	2498	2498
elim-nodes	582	3348	298	1597

Table 2: Results for PB1 (complete data in the upper part, MRH in the lower part)

total number of train units to reduce the CPU time when no valid inequalities are used. If the optimal solution of the unbounded problem does not respect such bounds, then the solutions of the two formulations are different.

For all the objective functions experimented with, most of the CPU time is, proportionally, spent for the solution of Model 1. The presence of many different solutions with the same objective function value (*degeneracy*), combined with a weak continuous relaxation, makes it difficult to certify optimality. Very often, the first integer solution found is the optimum, but then it may take quite a lot of time to close the integrality gap. In this context, using a tighter relaxed feasible region may help, which is demonstrated by the smaller CPU times of most of the problems with valid inequalities. The Linear Programming relaxations solved during the subproblem elimination phase are usually very fast, but the large number of them may make the whole elimination process quite time consuming.

After the elimination phases, the problem taking into account the compositions (Model 2) does not seem to be very difficult to solve, at least not much more difficult than the model neglecting the compositions (Model 1). This is due to the structure of the problem itself. Once a certain composition has been fixed for a certain trip (one of the $a_{t,k}$ variables in Model 2 has been fixed to 1), a path through the transition graph is often relatively easy to find.

Comparing the problems with and without the additional restrictions in Alkmaar, we would expect a reduction in the computational complexity in problems with restrictions in Alkmaar, since these restrictions reduce the number of feasible compositions, thereby reducing the dimensions of the composition graph. However, the restrictions in Alkmaar seem to affect the CPU times only in problems PB1 and PB3.

In all cases, the results obtained when solving the MRH problems are quite similar to the results obtained in the solution of the *complete* problems in terms of the numbers of train units of the different subtypes. This was to be expected, since the number of train units needed to satisfy a certain seat demand is mainly determined by the peak demand during the morning

	no-cut, no-res	cut, no-res	no-cut, res	cut, res
CPU (sec)	4451.7	496.5	4122.8	478.5
objval	1580	1575	1735	1735
$y_{tot,3}$	23	31	23	23
$y_{tot,4}$	15	15	14	14
iter	1	1	1	1
UB $y_{tot,3}, y_{tot,4}$	23	100	23	100
tot-nodes	13534	13534	8604	8604
elim-nodes	721	721	416	416
CPU (sec)	38.4	32.8	36.7	29.7
objval	537	535	617	617
$y_{tot,3}$	22	26	23	23
$y_{tot,4}$	13	9	11	11
iter	1	1	1	1
UB $y_{tot,3}, y_{tot,4}$	23	100	23	100
tot-nodes	3804	3804	2498	2498
elim-nodes	263	263	144	144

Table 3: Results for PB2 (complete data in the upper part, MRH in the lower part)

rush hours. Given a solution for the morning rush hours, in fact, the way the solution can be extended to the complete day is determined for many trips. Therefore, the solutions to the MRH problems provide a useful basis for the complete solutions.

6 Conclusions and further research

In this paper we described an algorithm for solving the rolling stock circulation problem for a single train series on a single day, thereby taking into account the fact that trains can be composed of train units of different subtypes. The latter implies that not only the number of train units of the different subtypes in the trains, but also their order in the trains is to be modelled. We applied the algorithm to a case study of the 3000 train series of NS Reizigers.

The solution approach we experimented with is a powerful solution scheme, which succeeds to find optimal solutions within an acceptable amount of time. However, its implementation (using valid inequalities, iterating the elimination process, finding binding constraints, etc.) is highly dependent on the input data and on the real problem's peculiarities (number of trips, demand pattern, and related feasible compositions).

Although the results for the 3000 case study were quite satisfying, it can be concluded that to deal with more complex cases, for example with several train types and/or with more than 2 subtypes of train units, or with trains splitting and combining underway, different methods may be necessary in order to reduce the computation times. The approach presented in this paper could be used as a starting point for the development of such methods. For example, this solution algorithm could be used to find a starting solution using only the data of the morning rush hours. Thereafter, the solution could be extended by heuristics to the complete day and finally be improved by a local search algorithm.

Another approach for solving the rolling stock circulation problem may be an approach based on column generation. Here the column generation mechanism generates appropriate

	no-cut, no-res	cut, no-res	no-cut, res	cut, res
CPU (sec)	9287.3	543.9	4266.9	483.6
objval	1602	1596	1769	1769
$y_{tot,3}$	21	24	20	20
$y_{tot,4}$	12	10	13	13
iter	1	1	1	1
UB $y_{tot,3}, y_{tot,4}$	23	100	23	100
tot-nodes	13534	13534	8604	8604
elim-nodes	757	3290	449	2766
CPU (sec)	44.6	33.7	38.1	30.3
objval	541	537	621	618
$y_{tot,3}$	23	28	20	24
$y_{tot,4}$	10	7	13	10
iter	1	1	1	1
UB $y_{tot,3}, y_{tot,4}$	23	100	23	100
tot-nodes	3804	3804	2498	2498
elim-nodes	299	2101	177	1597

Table 4: Results for PB3 (complete data in the upper part, MRH in the lower part)

paths through the transition graphs based on shortest path algorithms. This column generation mechanism takes into account *dual cost* information obtained from the master problem. The latter handles the coordination between the paths in the different composition graphs, mainly by taking into account the stocks of train units in the different stations. This approach based on column generation is a subject for further research.

Moreover, in a further study also other objectives may be used to better deal with the complex situation of a real railway system, such as the impossibility to satisfy the passengers' seat demand completely. This situation is due to the fact that acquiring new train units can be infeasible at short notice. In this case, a reasonable objective is to find a balance between the shortages of seats, the number of available train units, and the number of train unit or wagon kilometers. Another objective that may be worth studying is the objective of minimizing the total number of shunting movements in stations. The latter is relevant since shunting movements are potential sources of disruptions of the railway process. Therefore, reducing the number of shunting movements may be beneficial for the punctuality. On the other hand, this objective is conflicting with the objectives of minimizing the shortages of seats or of minimizing the number of train unit or wagon kilometers. These issues are also subjects for further research.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, "Network Flow - Theory, Algorithms, and Applications", Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[2] C. Barnhart, A. Hane, P.H. Vance, "Using Branch-and-Price to Solve Origin-Destination Integer Multi-Commodity Flow Problems", *Operations Research*, 32 (1998), 208-220.

[3] N. Ben-Khedher, J. Kintanar, C. Queille, W. Stripling, "Schedule Optimization at SNCF: From Conception to Day of Departure", *Interfaces*, 28 (1998), 6-23.

	no-cut, no-res	cut, no-res	no-cut, res	cut, res
CPU (sec)	634.4	478.9	594.6	993.9
objval	448	448	518	518
$y_{tot,3}$	12	12	20	20
$y_{tot,4}$	19	19	13	13
iter	1	1	1	2
UB $y_{tot,3}, y_{tot,4}$	23	100	23	100
tot-nodes	13534	13534	8604	8604
elim-nodes	912	7923	503	416
CPU (sec)	36.8	28.6	33.7	57.8
objval	153	153	184	184
$y_{tot,3}$	12	12	20	20
$y_{tot,4}$	19	19	13	13
iter	1	1	1	2
UB $y_{tot,3}, y_{tot,4}$	23	100	23	100
tot-nodes	3804	3804	2498	2498
elim-nodes	454	3064	229	279

Table 5: Results for PB4 (complete data in the upper part, MRH in the lower part)

[4] P. Brucker, J. Hurink, T. Rolfes, "Routing of Railway Carriages: A Case Study", *Osnabrücker Schriften zur Mathematik*, Reihe P, Heft 205 (1998).

[5] J.F. Cordeau, F. Soumis, J. Desrosiers, "Simultaneous Assignment of Locomotives and Cars to Passenger Trains", *Operations Research*, 49 (2001), 531-548.

[6] R. Groot, "Minimum Circulation of Railway Stock: an Integer Programming Algorithm", Master's Thesis, University of Amsterdam, 1996.

[7] ILOG Inc., CPLEX Division, *Using the CPLEX Callable Library.* CPLEX Reference Manual, 1999.

[8] N. Lingaya, J.F. Cordeau, G. Desaulniers, J. Desrosiers, F. Soumis, "Operational Car Assignment at VIA Rail Canada", *Transportation Research B*, 36 (2002), 755-778.

[9] R.D. McBride, "Advances in Solving the Multi-Commodity-Flow Problem", *Interfaces*, 28 (1998), 32-41.

[10] J. van Monfort, "Optimizing Railway Carriage Circulation with Integer Linear Programming", Master's Thesis, University of Amsterdam, 1997.

[11] NS homepage: www.ns.nl

[12] A. Schrijver, "Minimum Circulation of Railway Stock", *CWI Quarterly*, 6 (1993) 205-217.

Publications in the Report Series Research* in Management

ERIM Research Program: "Business Processes, Logistics and Information Systems"

2002

The importance of sociality for understanding knowledge sharing processes in organizational contexts
Niels-Ingvar Boer, Peter J. van Baalen & Kuldeep Kumar
ERS-2002-05-LIS

Crew Rostering for the High Speed Train
Ramon M. Lentink, Michiel A. Odijk & Erwin van Rijn
ERS-2002-07-LIS

Equivalent Results in Minimax Theory
J.B.G. Frenk, G. Kassay & J. Kolumbán
ERS-2002-08-LIS

An Introduction to Paradigm
Saskia C. van der Made-Potuijt & Arie de Bruin
ERS-2002-09-LIS

Airline Revenue Management: An Overview of OR Techniques 1982-2001
Kevin Pak & Nanda Piersma
ERS-2002-12-LIS

Quick Response Practices at the Warehouse of Ankor
R. Dekker, M.B.M. de Koster, H. Van Kalleveen & K.J. Roodbergen
ERS-2002-19-LIS

Harnessing Intellectual Resources in a Collaborative Context to create value
Sajda Qureshi, Vlatka Hlupic, Gert-Jan de Vreede, Robert O. Briggs & Jay Nunamaker
ERS-2002-28-LIS

Version Spaces and Generalized Monotone Boolean Functions
Jan C. Bioch & Toshihide Ibaraki
ERS-2002-34-LIS

Periodic Review, Push Inventory Policies for Remanufacturing
B. Mahadevan, David F. Pyke, Moritz Fleischman
ERS-2002-35-LIS

Modular Decomposition of Boolean Functions
Jan C. Bioch
ERS-2002-37-LIS

Classification Trees for Problems with Monotonicity Constraints
R. Potharst & A.J. Feelders
ERS-2002-45-LIS

---

* A complete overview of the ERIM Report Series Research in Management:
http://www.ers.erim.eur.nl

ERIM Research Programs:
LIS Business Processes, Logistics and Information Systems
ORG Organizing for Performance
MKT Marketing
F&A Finance and Accounting
STR Strategy and Entrepreneurship

Allocation of Railway Rolling Stock for Passenger Trains
Erwin Abbink, Bianca van den Berg, Leo Kroon & Marc Salomon
ERS-2002-47-LIS

Monotone Decision Trees and Noisy Data
Jan C. Bioch and Viara Popova
ERS-2002-53-LIS

Business Modeling Framework For Personalization In Mobile Business Services: a Case and Sociological Analysis
L-F Pau, Jeroen Dits
ERS-2002-56-LIS

Polynomial time algorithms for some multi-level lot-sizing problems with production capacities
Stan van Hoesel, H. Edwin Romeijn, Dolores Romero Morales, Albert P.M. Wagelmans
ERS-2002-59-LIS

A Note on Ending Inventory Valuation in Multiperiod Production Scheduling
Wilco van den Heuvel, Alfred P.M. Wagelmans
ERS-2002-63-LIS

Determining The Optimal Order Picking Batch Size In Single Aisle Warehouses
Tho Le-Duc and René B.M. de Koster
ERS-2002-64-LIS

Solving Variational Inequalities Defined on A Domain with Infinitely Many Linear Constraints
Shu-Cherng Fang, Soonyi Wu, Ş. İlker Birbil
ERS-2002-70-LIS

Entropic Regularization Approach for Mathematical Programs with Equilibrium Constraints
Ş. İlker Birbil, Shu-Cherng Fang, Jiye Han
ERS-2002-71-LIS

On the Finite Termination of An Entropy Function Based Smoothing Newton Method for Vertical Linear Complementarity Problems
Shu-Cherng Fang, Jiye Han, Zhenghai Huang, Ş. İlker Birbil
ERS-2002-72-LIS

The Role Of Product Quality Information, Market State Information And Transaction Costs In Electronic Auctions
Otto Koppius and Eric van Heck
ERS-2002-73-LIS

Shunting of Passenger Train Units in a Railway Station
Richard Freling, Ramon M. Lentink, Leo G. Kroon, Dennis Huisman
ERS-2002-74-LIS

Inventory strategies for systems with fast remanufacturing
Ruud Teunter, Erwin van der Laan, Dimitrios Vlachosz
ERS-2002-77-LIS

A Business Evaluation Of The Next IPv6 Protocol In Fixed And Mobile Communication Services: An Analytical Study And Calculation
L-F Pau
ERS-2002-78-LIS

IPv6 Return on investment (R.O.I) analysis framework at a generic level, and first conclusions
L-F Pau
ERS-2002-79-LIS

On Borel Probability Measures and Noncooperative Game Theory
J.B.G.Frenk, G.Kassay, V.Protassov
ERS-2002-85-LIS

The Newsboy Problem with Resalable Returns
Julien Mostard, Ruud Teunter
ERS-2002-89-LIS

Integrating Closed-loop Supply Chains and Spare Parts Management at IBM
Moritz Fleischmann, Jo van Nunen, Ben Gräve
ERS-2002-107-LIS

Eliciting Expert Knowledge for Fuzzy Evaluation of Agricultural Production Systems
A. M. G. Cornelissen, J. van den Berg, W. J. Koops, U. Kaymak
ERS-2002-108-LIS

Efficient Circulation of Railway Rolling Stock
Arianna Alfieri, Rutger Groot, Leo Kroon, Lex Schrijver
ERS-2002-110-LIS

Direct Mailing Decisions for a Dutch Fundraiser
Jedid-Jah Jonker, Nanda Piersma, Rob Potharst
ERS-2002-111-LIS