

Shunting of Passenger Train Units in a Railway Station

Richard Freling, Ramon M. Lentink, Leo G. Kroon, Dennis Huisman

ERIM REPORT SERIES <i>RESEARCH IN MANAGEMENT</i>	
ERIM Report Series reference number	ERS-2002-74-LIS
Publication	September 2002
Number of pages	23
Email address corresponding author	rlentink@fbk.eur.nl
Address	Erasmus Research Institute of Management (ERIM) Rotterdam School of Management / Faculteit Bedrijfskunde 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	In this paper we introduce the problem of shunting passenger train units in a railway station. Shunting occurs whenever train units are temporarily not necessary to operate a given timetable. We discuss several aspects of this problem and focus on two subproblems. We propose mathematical models for these subproblems together with a solution method based on column generation. Furthermore, a new efficient and speedy solution technique for pricing problems in column generation algorithms is introduced. Finally, we present computational results based on real life instances from Netherlands Railways.	
Library of Congress Classification (LCC)	5001-6182	Business
	5201-5982	Business Science
	HB 143.7	Optimization Techniques
Journal of Economic Literature (JEL)	M	Business Administration and Business Economics
	M 11	Production Management
	R 4	Transportation Systems
	C 61	Optimization Techniques
European Business Schools Library Group (EBSLG)	85 A	Business General
	260 K	Logistics
	240 B	Information Systems Management
	255 B	Decisions under constraints
Gemeenschappelijke Onderwerpsontsluiting (GOO)		
Classification GOO	85.00	Bedrijfskunde, Organiseatiekunde: algemeen
	85.34	Logistiek management
	85.20	Bestuurlijke informatie, informatieverzorging
	31.80	Toepassingen van de wiskunde
Keywords GOO	Bedrijfskunde / Bedrijfseconomie	
	Bedrijfsprocessen, logistiek, management informatiesystemen	
	Spoorwegvervoer, optimaliseren, logistiek, wiskundige modellen	
Free keywords	railway optimization, column generation, real world application, logistics	

Shunting of Passenger Train Units in a Railway Station

Richard Freling¹, Ramon M. Lentink^{1,*}, Leo G. Kroon^{1,2},
Dennis Huisman¹

¹ *Erasmus Center for Optimization in Public Transport (ECOPT),
Erasmus University Rotterdam, the Netherlands*

² *NS Reizigers, Department of Logistics, the Netherlands*

* *Corresponding author, email: rlentink@fbk.eur.nl*

Abstract

In this paper we introduce the problem of shunting passenger train units in a railway station. Shunting occurs whenever train units are temporarily not necessary to operate a given timetable. We discuss several aspects of this problem and focus on two subproblems. We propose mathematical models for these subproblems together with a solution method based on column generation. Furthermore, a new efficient and speedy solution technique for pricing problems in column generation algorithms is introduced. Finally, we present computational results based on real life instances from Netherlands Railways.

In memoriam of Richard Freling, who passed away too early.

1 Introduction

Within the rush hours, the rolling stock of a passenger railway operator is typically operating the timetable or it is in maintenance. However, outside the rush hours, an operator usually has a surplus of rolling stock. These surplus train units can be parked at a shunt yard in order to be able to fully exploit the main railway infrastructure. Especially during the night, many passenger train units have to be parked, since usually there are just a few night trains. In the Netherlands, mainly freight trains operate at night.

The process of parking train units together with several related processes is called *shunting*. A major complicating issue is the fact that train units are strongly restricted in their movements by the railway infrastructure. In addition, time is also a restrictive resource for shunting. For example, for safety reasons it is mandatory to respect a certain minimum headway time between two train movements on the same track.

We define an arriving shunt unit as a train unit that has to be parked at a shunt yard and, similarly, a departing shunt unit as a train unit that has to be supplied from the shunt yard. Arriving shunt units are uncoupled from through trains, or come from complete ending trains, while departing shunt units are units that are coupled onto through trains, or form complete starting trains. The related shunting processes include the matching of arriving and departing shunt units, the routing of shunt units on the station infrastructure, cleaning and maintenance of shunt units, and crew planning for the crew that carry out the activities that result from the shunting processes.

At night, the goal of shunting is to park the units in such a way that the operations in the next morning can start up as smoothly as possible, while certain restrictions with respect to parking, routing, cleaning, maintenance, and crew planning are met.

In the Netherlands, most train units can move by themselves in two directions and do not need locomotives. In addition, Dutch train units are classified according to types and subtypes. Only train units of the same type can be combined to form trains, while subtypes of a type are discerned from each other by their numbers of carriages per train unit. Figure 1 depicts an example of a Dutch train unit with 3 carriages. This particular type of train unit (ICM) consists of subtypes with 3 or 4 carriages.



Figure 1: An example of an ICM train unit with 3 carriages (ICM.3).

In general, train units of the same subtype can be used interchangeably. This flexibility implies that a planner has to determine a matching of arriving and departing shunt units.

Figure 2 shows the layout of the station Zwolle, which is a station in the northeastern part of the Netherlands. The black areas in the figure represent the platforms, while next to the platform tracks and around those tracks several shunt tracks are located.

The parking of train units is far from trivial because space is usually scarce. In addition, the choice to park a shunt unit on a particular shunt track

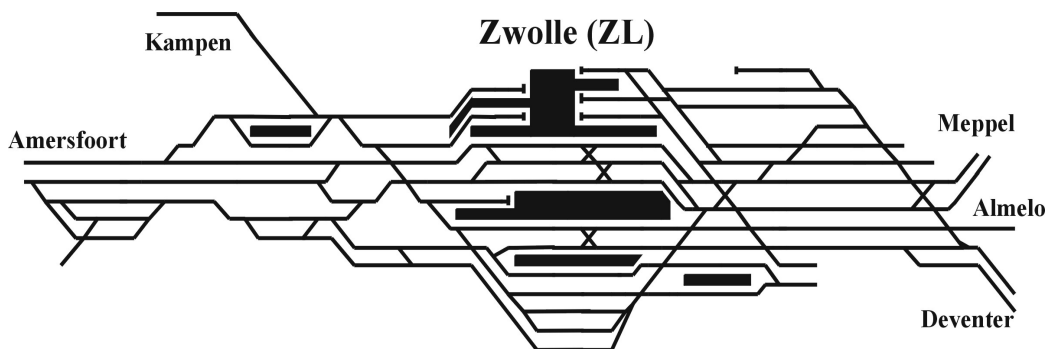


Figure 2: The layout of station Zwolle.

has several implications. Firstly, when train units are of different subtypes, the order of the train units at a shunt track has influence on the amount of required shunting effort. Secondly, the possible routes between platforms and shunt tracks are restricted by this decision. Thirdly, crew has to be available to carry out the resulting shunt activities within certain desired time intervals. Finally, in practice, certain routes and shunt tracks are preferred over others. Here, a track is less desired if it is used frequently for other purposes, e.g. for through trains or for temporary parking. Both of these preferences could be taken into account.

In the remainder of this paper, we provide an in-depth introduction to the shunting problem in Section 2, containing an overview of the relevant literature. Furthermore, we propose a solution approach consisting of two steps in Section 3. One step is related to the matching of arriving and departing shunt units, while the other step is related to the assignment of shunt units to shunt tracks. We will focus on the matching step in Section 4 and on the assignment step in Section 5. In addition, we present computational results of the solution approach at station Zwolle in Section 6. Finally, we end with some conclusions and suggestions for further research.

2 Problem description

In this section, we introduce the *train unit shunting problem* and we describe the most important aspects in more detail. Furthermore, we provide an overview of the relevant literature.

Given

- a railway station,
- a shunt yard,

- a timetable, with the compositions of the trains, and for each train the arrival and / or departure time and platform at the involved station,
- estimates of route costs from platform tracks to shunt tracks and vice versa,

the *train unit shunting problem* consists of (i) matching the arriving and departing shunt units and (ii) parking these on a shunt track. In (ii), the route costs differ in time, because the claims on the station infrastructure by through trains are to be considered. If the arrival time of an arriving shunt unit and the departure time of the matched departing shunt unit are close enough, it is not necessary to park the unit at a shunt track. In this case, the shunt unit can stay at the arrival or departure platform.

Issues with respect to routing, cleaning, maintenance and crew planning are not considered in this paper. These issues are outside the scope of this paper and will be described in forthcoming papers.

We minimize the costs of a shunt plan, such that the capacity of the shunt tracks is never exceeded. The costs of a shunt plan consist of e.g. route preferences, preferences for shunt tracks, and penalties for not parking shunt units, while it is necessary to park these.

Some characteristics of the general problem are the following:

- Arrivals and departures of train units may be mixed in time. This implies that, within the planning horizon, the first departure may take place before the last arrival.
- Shunt units may have different subtypes (and thus lengths). Its type may restrict the set of shunt tracks where a shunt unit can be parked. For example, electrical train units can only be parked at tracks with catenary.
- Shunt tracks may have different types and lengths. The type of a track determines how a unit can approach the track. Some tracks can be approached from one side only and these tracks will be called *Last In First Out (LIFO) tracks*. Other tracks can be approached from both sides and these tracks will be called *free tracks*.
- Trains have fixed arrival and departure times, but flexible arrival and departure times from the shunt tracks. For example, the departure time of an arriving shunt unit from a platform to a shunt track can be flexible within a time interval starting at the arrival time of the unit at the platform and ending before the next arrival of a train at the same platform.

A restriction in the matching of arriving and departing shunt units is that the subtypes of the units in the matching are the same. This is a hard restriction in the Dutch rolling stock deployment. Furthermore, if several units of different subtypes of one arriving train are matched to one departing train, the subtypes of the units in both trains have to be in the right order.

Table 1 describes a shunt plan for 5 shunt units on a *free* shunt track. Each row indicates the matching of arriving shunt units to departing shunt units. In this example, the first train, with ID 3628, arrives at 11:09 on Monday at platform 5A. One unit of subtype MAT64.4 is parked at the shunt track and this unit leaves the station in train 3629 at 07:49 on Tuesday from platform 5A. Furthermore, the third row describes two units of subtype MAT64.2, arriving in train 3678, that also depart in train 3629 on Tuesday. Thus, these units are coupled on the shunt track.

Arriving Train			Departing Train			Subtypes of train units
Train ID	Platform	Time	Train ID	Platform	Time	
3628	5A	mo 11:09	3629	5A	tu 07:49	MAT64.4
561	3B	mo 21:02	520	1A	tu 07:18	ICM.4 ICM.3
3678	5B	mo 23:43	3629	5A	tu 07:49	MAT64.2 MAT64.2

Table 1: An example of a shunt plan with 5 train units.

Figure 3 represents the situation at the track on Tuesday morning at 6 o'clock. The arriving units of trains 561 and 3678 cannot arrive via the same side of the track. Suppose the units of these trains would do so. Then the train units of train 520 that have to leave at 07:18 on Tuesday are blocked from both sides of the track. This is called a *crossing* and is not allowed because it results in high additional shunting effort. Furthermore, in that case the units of train 3629 would not be parked next to each other, which would also lead to additional shunting effort in the early morning. Note that, if the track would have been a LIFO track, a crossing can only be avoided by parking some of the units at another track.

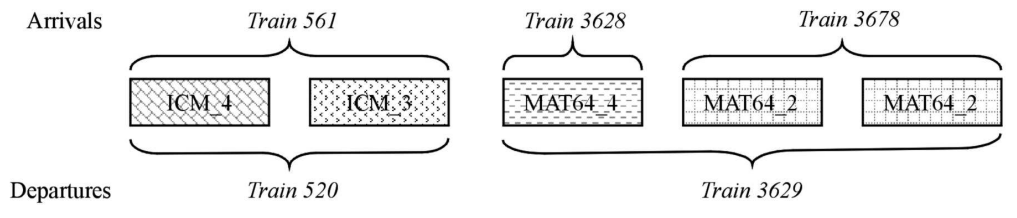


Figure 3: The situation at the track, on Tuesday at 6 o'clock.

Literature on this topic is very scarce, but recently some special cases of the train unit shunting problem have been dealt with by Winter and

Zimmermann [2000] and Blasum et al. [2000] for dispatching trams in a depot, and by Gallo and Di Miele [2001] for dispatching buses in a depot. In both problems all arrivals take place before the first departure, although Gallo and Di Miele [2001] discuss an extension of their model to take into account mixed arrivals and departures.

We choose to use the term *shunting* instead of dispatching because it is more commonly used in the railway context, and because, in contrast to parking trams and buses, routing and several other processes are also involved. Complicating factors for our problem are the different subtypes and lengths of the units, the different approach types of the tracks, and the mixed arrivals and departures.

Complicating matters that may occur in practice, but that are not considered in this paper, are, for example, routing, maintenance planning and shunt personnel planning. Tomii and Zhou [2000] and Tomii et al. [1999] propose a genetic algorithm that takes into account some of these matters in a railway setting. However, their problem is of a different nature, since in their context at most one train unit can be parked on a shunt track at the same time. Finally, the subject of the paper by He et al. [2000] is the separation of train units from arriving trains, sorting these according to their destination, and finally combining them to form new departing trains, which resembles the matching of arriving and departing shunt units.

3 Solution approach

Winter and Zimmermann [2000] and Blasum et al. [2000] showed that several variants of their shunting problems are NP-hard. Although our problem is slightly different, we consider their complexity results as sufficient evidence that our shunting problem is also NP-hard. Therefore, we have decided to focus immediately on the development of a solution approach.

Before we introduce this approach, we give several useful definitions:

- The *supply* is the set of arriving shunt units.
- The *demand* is the set of departing shunt units.
- A *block* is an entity of one or more train units that remain together for the entire planning period. For example, the train units of train 561 in Figure 3 form a block.

The solution approach is based on a decomposition of the problem into 2 steps, where one step is related to the matching of arriving to departing shunt units, and the other one deals with parking shunt units on the shunt

yard. Below, the two steps of our solution approach are described in more detail.

Step 1 *Match supply to demand*. Each arriving shunt unit is assigned to a departing shunt unit. The objective is to minimize the shunting effort by keeping units together as much as possible. This step results in a set of blocks.

Step 2 *Parking*. Each block is assigned to a shunt track, such that the capacity is never exceeded and crossings do not occur.

The motivation for this 2-step solution approach is twofold. First, the problem would be intractable if we would consider it completely integrated. This is even more of interest bearing in mind possible future extensions of the approach, i.e. crew planning and maintenance issues. Second, the planner can interact more easily with the planning process. Already after Step 1, the planner can make modifications to the obtained results if desired.

A drawback of this approach may be that global optimality is lost. However, the objective is not only to improve the current manual solutions, but also to speed up the planning process. Moreover, we may not be able to find an overall feasible solution if there exists one. This occurs, for example, when it is impossible to park all the blocks created in Step 1.

4 Matching supply to demand

In this section, we formulate the matching problem (Step 1) as a mathematical model. Therefore, we give some useful definitions first. Recall that in Step 1, each arriving shunt unit is assigned to a departing shunt unit.

For each arriving and each departing train we define a network, with as nodes the units of the train and a dummy node (with index 0). The nodes are places where the train can be divided into parts. The arcs represent feasible configurations of possible *parts* of the train, where a part is a subset of adjacent train units of the train. Thus, a block represents a combination of a part of an arriving train and an identical part of a departing train.

Figure 4 depicts the network for departing train 3629 of Table 1. Note that each arc corresponds with one part and thus 6 different parts are possible in this example. Furthermore, notice that a path from the first to the last node in this network corresponds to a division of the train into parts. For example, the path $0 \rightarrow 2 \rightarrow 3$ in Figure 4 implies that train units 1 and 2 form one part, and train unit 3 forms another part.

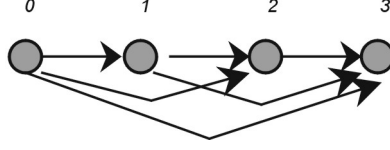


Figure 4: The network of train 3629 of Table 1.

Let I be the set of all possible parts for all arriving trains and let J be the set of all possible parts for all departing trains. Formally, we introduce a mathematical formulation with the following binary variables:

$$\begin{aligned}
 u_i &= \begin{cases} 1 & \text{if part } i \in I \text{ is used;} \\ 0 & \text{otherwise.} \end{cases} \\
 v_j &= \begin{cases} 1 & \text{if part } j \in J \text{ is used;} \\ 0 & \text{otherwise.} \end{cases} \\
 z_{ij} &= \begin{cases} 1 & \text{if part } i \in I \text{ is assigned to part } j \in J; \\ 0 & \text{otherwise.} \end{cases}
 \end{aligned}$$

In addition, we define T^a as the set of arriving trains with shunt units, and T^d as the set of departing trains with shunt units. The arcs in a network of train t are given in the set A^t . We additionally use A_h^{t+} as the set of arcs out of node h for train t and A_h^{t-} as the set of arcs into node h for train t . Furthermore, the set C_t^- is the set of all intermediate nodes in the network of train t . Finally, we use J_i (I_j) as the set of departing (arriving) parts with the same matching configuration as arriving part i (departing part j).

The parameter Q models a penalty for each arriving part that is used. The parameters w_{ij} model the costs of assigning arriving part i to departing part j . An element of these costs is, for example, the difference between the arrival and departure times of parts i and j .

The model is:

$$\text{minimize } Q \sum_{i \in I} u_i + \sum_{i \in I} \sum_{j \in J_i} w_{ij} z_{ij} \quad (1)$$

$$\text{subject to } \sum_{i \in A_0^{t+}} u_i = 1 \quad \forall t \in T^a \quad (2)$$

$$\sum_{i \in A_h^{t+}} u_i - \sum_{i \in A_h^{t-}} u_i = 0 \quad \forall t \in T^a, \forall h \in C_t^- \quad (3)$$

$$\sum_{j \in A_0^{t+}} v_j = 1 \quad \forall t \in T^d \quad (4)$$

$$\sum_{j \in A_h^{t+}} v_j - \sum_{j \in A_h^{t-}} v_j = 0 \quad \forall t \in T^d, \forall h \in C_t^- \quad (5)$$

$$\sum_{j \in J_i} z_{ij} = u_i \quad \forall i \in I \quad (6)$$

$$\sum_{i \in I_j} z_{ij} = v_j \quad \forall j \in J \quad (7)$$

$$z_{ij}, u_i, v_j \in \{0, 1\} \quad \forall i \in I, \forall j \in J \quad (8)$$

The objective (1) is to minimize the weighted sum of the number of parts and the assignment costs, since minimizing the number of parts results in assignments where train units are kept together as much as possible. Path flow constraints (2) and (3) assure the covering of each arriving shunt unit by a part, while constraints (4) and (5) assure this for each departing shunt unit. Assignment constraints (6) guarantee that each arriving part is assigned to a departing part if and only if the arriving part is a result of the train decomposition. Constraints (7) model this for the departing parts. The outcome of this step is a set of arriving parts assigned to departing parts, which we earlier called *blocks*. These blocks are used in Step 2.

In practice, it is possible that an arriving shunt unit is not assigned to a departing shunt unit in the planning period. In this case, the unit will remain parked at a shunt track until the end of the planning period. This can be incorporated easily by allowing dummy assignments. Of course, a similar argument holds for departing shunt units that are not assigned to arriving shunt units.

In Section 6 we show that this formulation can be solved quite efficiently by using the standard MIP solver of CPLEX 6.5.

5 Assigning units to shunt tracks

As we described in the previous section, the matching of supply and demand of shunt units results in a set of blocks B . Blocks that do not need to be parked at a shunt track are not included in the set B . For each block we know the arrival and departure times and the arrival and departure platforms. In addition, we also have the route costs between these platforms and all feasible shunt tracks for each block. Notice for free tracks these costs can differ between the left and right side of such a track. Given this information, the second step in our solution approach consists of assigning blocks to shunt tracks at minimum costs, such that the capacity of the shunt tracks is never exceeded and that there are no crossings among the blocks. Recall that a crossing occurs if one block obstructs the arrival or departure of another

block. We call this subproblem of assigning blocks to shunt tracks the *Track Assignment Problem* (TAP).

5.1 Mathematical formulation

We formulate the TAP as a Set Partitioning Problem with side constraints. The main reason is that modeling the problem directly will lead to a very complex model with considerable variations for different special cases. This approach is thus not very robust with respect to future extensions and solution strategies. For our formulation, we define a *track assignment*, or shortly an *assignment*, as a feasible assignment of a certain subset of blocks to a particular shunt track during the planning period. Here, an assignment is *feasible* if the following conditions hold:

1. it does not contain crossings;
2. the total length of the units on the track never exceeds the length of the track;
3. all blocks of the subset are allowed to park at the track.

Table 1 and Figure 3 (see Section 2) contain an example of a feasible assignment of blocks to a shunt track.

Let S be the set of shunt tracks, furthermore, let K^s be the set of assignments on track $s \in S$, and K_b^s the set of assignments on track $s \in S$ containing block $b \in B$. We define the following decision variables:

$$x_k^s = \begin{cases} 1 & \text{if assignment } k \in K^s \text{ is used on shunt track } s \in S; \\ 0 & \text{otherwise.} \end{cases}$$

$$y_b = \begin{cases} 1 & \text{if block } b \in B \text{ is not parked on any shunt track } s \in S; \\ 0 & \text{otherwise.} \end{cases}$$

Note that the number of variables x_k^s grows exponentially in the number of blocks. The parameters c_k^s model the costs of assignment k on track s . These costs consist of the following aspects:

1. The sum of the route costs of the different blocks in the assignment.
2. A penalty if a less desired track is used. This penalty is related to the duration of the occupation by the blocks of the track.

In practice, there are also other elements of these costs, like penalties if units are in the wrong order on the track, that we do not consider in this paper.

In addition, the parameter d models a penalty if a block is not assigned to a track. The TAP is then formulated as follows:

$$\text{minimize } \sum_{s \in S} \sum_{k \in K^s} c_k^s x_k^s + d \sum_{b \in B} y_b \quad (9)$$

$$\text{subject to } \sum_{s \in S} \sum_{k \in K_b^s} x_k^s + y_b = 1 \quad \forall b \in B \quad (10)$$

$$\sum_{k \in K^s} x_k^s \leq 1 \quad \forall s \in S \quad (11)$$

$$x_k^s \in \{0, 1\} \quad \forall s \in S, \forall k \in K^s \quad (12)$$

$$y_b \in \{0, 1\} \quad \forall b \in B \quad (13)$$

We aim at minimizing the costs of a shunt plan, such that as many blocks as possible are assigned to a shunt track. Constraints (10) state that each block is covered by exactly one assignment on one shunt track or it is not parked at all. Furthermore, constraints (11) model that each shunt track can have at most one assignment.

A major advantage of the proposed formulation is that difficult constraints with respect to the feasibility of an assignment are taken into account implicitly. A way to overcome the disadvantage of exponentially many assignments to tracks in the number of blocks is discussed next.

5.2 A column generation algorithm for assigning blocks to shunt tracks

The general Set Partitioning structure allows for the application of solution techniques that are efficient for several special cases. Because the number of assignments grows exponentially in the number of blocks, we propose a column generation approach. We refer to Barnhart et al. [1998] for a general discussion on column generation in an integer programming context.

In our setting, the *master problem* consists of selecting a set of track assignments according to the model (9) - (13). More detailed, we propose a heuristic algorithm, where columns are generated dynamically to solve the LP relaxation of TAP, and we solve the integer problem without generating additional columns, as opposed to an exact branch-and-price algorithm.

In the *pricing problem*, assignments for individual shunt tracks are generated implicitly and independently. Here, new columns are generated based

on dual information obtained from the master problem. Therefore, we introduce dual variables λ_b ($b \in B$) and μ_s ($s \in S$) for constraints (10) and (11), respectively. The structure of the pricing problem can vary with the specific characteristics of the TAP. However, the structure of the master problem does not change.

Now, we first discuss a network representation for the pricing problem and then give a dynamic programming algorithm for solving the resource constrained shortest path problem in this network. This algorithm results in a set of promising columns or corresponding assignments for a shunt track.

5.2.1 The network representation

In order to generate assignments for one shunt track, we use a network representation of the problem. Let F be the set of different approach types to and from the shunt track. For a LIFO track, F contains only one element, namely arriving from and departing to the same direction, either left or right. However, for a free track, F contains four elements:

- Arriving from the left and departing to the left (LL).
- Arriving from the left and departing to the right (LR).
- Arriving from the right and departing to the left (RL).
- Arriving from the right and departing to the right (RR).

We assume that the set of blocks B that need to be parked is ordered increasingly in the arrival time. Each block $b \in B$ is represented by a layer L_b consisting of nodes n_f^b with $f \in F$ and a node n_{not}^b , which corresponds to not parking block b on the track currently under consideration. The nodes of the network contain a source o , a sink t and for each block b the layer L_b . The arcs in this network are directed from the source to every node in the first layer, from every node in the last layer to the sink, and between two nodes in consecutive layers if the corresponding blocks and approach types can be assigned to the current track without introducing crossings. If an arc would introduce a crossing we call it infeasible. If we define B^- as the set of blocks excluding the last one, we can define our network $G = (N, A)$ as:

$$N = \bigcup_{b \in B} L_b \cup \{o, t\} \quad (14)$$

$$A = \{(n_i^b, n_j^{b+1}) : b \in B^-, i \in L_b, j \in L_{b+1}, \text{ and } (n_i^b, n_j^{b+1}) \text{ is feasible}\} \cup \quad (15)$$

$$\{(o, n_i^1) : i \in L_1\} \cup \{(n_j^{|B|}, t) : j \in L_{|B|}\} \quad (16)$$

We illustrate this network in Figure 5 with the example shunt plan of Table 1 (see Section 2) for a single free shunt track. The bold arcs in this figure represent a path through the network. This path corresponds with an assignment where all blocks are parked on the track. Furthermore, the first block arrives from the left and departs from the right side of the shunt track, while the second block arrives and departs from the left side and the third block arrives and departs from the right side. Note that there are no arcs from e.g. node 'LL' in layer 1 to the nodes 'LR' and 'RL' in layer 2, because these would introduce crossings, which are not allowed.

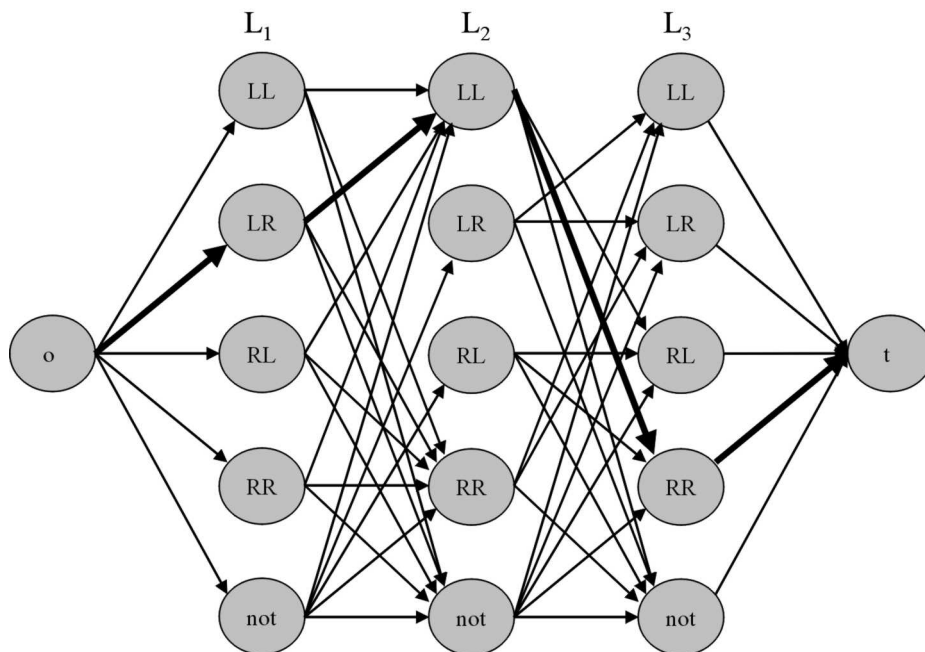


Figure 5: An example of a network for a free shunt track with three blocks.

Costs are defined on the arcs of the network. The costs of an arc represent the costs of the node to which the arc is directed, which are the costs of assigning a certain block to a certain shunt track, arriving from and departing to this track via its specified sides. Furthermore, the costs on the arcs to the sink model the usage of the shunt track.

The reduced costs of an arc (n_i^b, n_j^{b+1}) are equal to the costs of this arc minus the dual costs λ_{b+1} of block $b+1$. In addition, the reduced costs of an arc (o, n_i^1) are defined as its costs minus the dual costs λ_1 and, moreover, the reduced costs of an arc $(n_i^{|B|}, t)$ are equal to its costs minus the dual costs μ_s of the corresponding shunt track s . Using the reduced costs on the arcs,

the reduced costs of a path represent the reduced costs of the corresponding assignment k on shunt track s , which we denote by \overline{c}_k^s . Finally, let B_k^s be the set of blocks of assignment k parked on track s , then

$$\overline{c}_k^s = c_k^s - \sum_{b \in B_k^s} \lambda_b - \mu_s \quad \forall s \in S, k \in K^s. \quad (17)$$

A path in the network represents a feasible assignment of the blocks to the track if several additional constraints are satisfied. These constraints deal with the length of the blocks on the track and their arrivals and departures.

5.2.2 The dynamic programming algorithm

The general dynamic programming procedure, where we define P_i as a set of feasible $(o - i)$ -paths in G , can be described as (see also Freling [1997]):

```

For each node  $i \in N$ 
  For each path  $p \in P_i$ 
    For each arc  $(i, j) \in A$ 
       $p_j^{new} = \text{CheckRules}(p, (i, j));$ 
      If  $p_j^{new}$  exists, then  $\text{UpdatePathList}(P_j, p_j^{new});$ 
    Endfor
  Endfor
Endfor

```

In the function $\text{CheckRules}(p, (i, j))$ several rules are checked to determine whether the path obtained by extending path p with arc (i, j) is feasible, and, if so, the new path p_j^{new} is returned. At the same time, several key values (called *resource consumptions*, see Desrosiers et al. [1995]) of variables necessary for efficiently checking the rules are updated for the new path. We purposely loop first over the paths and then over the arcs. Since a path is considered for several arcs, the removal of blocks that have departed and the corresponding updating of resource variables is done once for each path only.

In case of a LIFO track, the resource variables are the costs of the path, the total length of the blocks currently on the track, and the earliest departure time of the blocks currently on the track. This earliest departure time is the departure time of the first block on the departure side of the track. In case of a free track, instead of the earliest departure time, we register the earliest and latest departure times at both sides of the track. For both track types, feasibility can be checked by verifying that the total length does not exceed the track length, and by comparing departure times of the assigned blocks in order to avoid crossings.

In $\text{UpdatePathList}(P_j, p_j^{new})$, the list of paths P_j is updated for the new path p_j^{new} . In order to facilitate dominance checking, the paths in P_j are kept in lexicographical order of the values of the resource variables, including the reduced costs of the path. A path p_i is dominated by a path p_j if all the resource variables of p_i are dominated by the resource variables of p_j . For example, p_i is dominated on the resource variable ‘length’ if the total length of the blocks of p_i currently on the track is larger than the total length of the blocks of p_j currently on the track. The list can be updated in one loop: as long as the new path is lexicographically smaller than the paths in the list, we need to check if the new path is dominated. Otherwise, we can insert the new path and loop through the remainder of the list, thereby removing paths that are dominated by the new path. The benefit of the lexicographical order is that we do not need to check the first resource variable, whilst looping through the remainder of the list.

The concept of the ‘not node’ can be used in many other applications using column generation, such as e.g. crew scheduling with fixed costs for each duty. The ‘not node’ has two effects: it greatly reduces the number of arcs in the network, because arcs are only present from one layer to the next. Furthermore, it helps to concentrate path lists at a restricted number of nodes, because extending a path with the ‘not node’ of the next layer is always feasible. Because of the layered structure of the network and the usage of the ‘not nodes’, the number of arcs $|A|$ remains small:

$$|A| \leq |L_b|^2(|B| - 1) + 2|L_b| \quad (18)$$

Note that in (18), $|L_b|$ is independent of b (5 for free tracks and 2 for LIFO tracks). Furthermore, because of the concentration of paths at these ‘not nodes’, dominance has more effect and more paths are deleted as compared to having a larger network without the ‘not nodes’, where the ‘smaller’ path lists are divided over much more nodes.

6 Application to a railway station in the Netherlands

The solution method proposed in this paper has been applied to the railway station Zwolle in the northeastern part of the Netherlands (see Figure 2). This station is chosen because relatively many shunting processes take place there and the capacity in terms of the total length of the shunt tracks is scarce. We considered a period from 8 o’clock in the morning to 8 o’clock in the next morning. During this period, between 550 and 600 trains, which

consist of between 800 and 1100 train units, arrive at and depart from Zwolle. These numbers differ for each day of the week. Furthermore, many trains are through trains and do not need shunting. Typically, approximately 80 units need to be parked. The train units range in length from 44 to 124 meters, while the 19 shunt tracks of Zwolle vary in length from 114 to 390 meters. Of these 19 shunt tracks, 15 tracks are free tracks and 4 tracks are LIFO tracks.

The scenarios we used for our computational results are differentiated from each other by the following aspects:

1. The objective function in Step 1. We consider the following three objective functions:
 - (a) Minimization of the number of blocks. Here, we only introduce a high penalty Q in the objective (1) and set w_{ij} equal to 0 for all i and j .
 - (b) Minimization of the number of blocks and minimization of the deviation of the average length of stay. We model this by choosing w_{ij} as the square of the time difference between the arrival of part i and the departure of part j in addition to the penalty of (1a).
 - (c) Minimization of the number of blocks and maximization of the deviation of the average length of stay. Therefore, we model w_{ij} as a penalty, which is imposed when the difference in length of stay between parts i and j lies between 2 and 10 hours. Again, this cost structure is added to the penalty of (1a).
2. The approach type of the free tracks. In the realistic case we consider the 15 free tracks as free tracks, while in the LIFO case we consider them as LIFO tracks. In practice, each free shunt track has a preferred side of approach, which becomes compulsory in the LIFO case.
3. The day of the week. Here we considered a Tuesday / Wednesday in the summer of 2001 as well as the Saturday / Sunday in the same week.

The scenarios are labelled in Table 2. In Table 3 we describe the computational results for the scenarios on Tuesday, while Table 4 gives these results for the scenarios on Saturday.

All computations were carried out on a PC with an Intel Pentium 4 1.6 Ghz. processor and 256 Mb. of internal memory. In addition, we imposed a maximum time of 40 minutes to CPLEX 6.5 for solving the MIP problem.

In Table 3, we see that, although it greatly complicates the problem and free tracks may have a preferred approach side, modelling the tracks as free tracks pays off. Furthermore, the scenarios TD and TF have large gaps. The

	1a	1b	1c
Realistic Tuesday	TA	TB	TC
LIFO Tuesday	TD	TE	TF
Realistic Saturday	SA	SB	SC
LIFO Saturday	SD	SE	SF

Table 2: Labelling of the scenarios.

Scenario	TA	TB	TC	TD	TE	TF
Runtime (in sec.) for Step 1	2.2	2.2	2.0	2.0	2.1	2.0
Number of blocks	67	68	67	67	68	67
Runtime (in sec.) for Step 2	3327	1851	1563	2499	1589	2481
LP solution value	5734.00	5901.50	5691	6704.10	6826.50	6625.50
IP solution value	5928	5971	5691	8435	6925	7573
Gap	3.27%	1.16%	0.00%	20.52%	1.42%	12.51%
# Columns generated	4815	4664	4538	3349	3528	3318
# Iterations of column generation	95	80	95	73	71	70

Table 3: Computational results for Tuesday at Zwolle.

reason is that our heuristic does not generate columns after solving the LP relaxation. Therefore, in both cases one block cannot be parked on any shunt track, while this is possible for the LP relaxation. Thus, a good choice of the objective function in Step 1 may result in parking one or more units more in Step 2. Increasing the maximum time for solving the MIP problem did not improve the solution, which confirms our conclusion. The runtime of the solution approach lies roughly between 20 and 40 minutes with scenario TA as an exception. This is acceptable in practice, since it may take planners several days to create these plans.

Scenario	SA	SB	SC	SD	SE	SF
Runtime (in sec.) for Step 1	1.2	1.1	1.1	1.2	1.1	1.1
Number of blocks	48	48	48	48	48	48
Runtime (in sec.) for Step 2	131	90	67	55	200	47
LP solution value	3612.00	3680.83	3497.33	4686.33	4611.83	4551.75
IP solution value	3662	3699	3528	4740	4748	4601
Gap	1.37%	0.50%	0.87%	1.13%	1.29%	1.07%
# Columns generated	26488	27070	24469	19274	17836	14838
# Iterations of column generation	41	45	44	38	32	31

Table 4: Computational results for Saturday at Zwolle.

For this Saturday, Step 1 resulted in 66 blocks, but since 18 arriving parts are already parked at a shunt track, there are only 48 blocks to be parked in Step 2. The results for the Saturday show that a lot more columns

are generated, in less iterations. The reason is that, since many train units at Zwolle remain there the entire weekend, dominance, based on length of stay, during the generation of columns has less effect. Furthermore, this also results in a large reduction in the runtime as compared to the Tuesday. In addition, the gaps are typically around 1%, which implies that our solution approach works very well for these scenarios. The structure of the Saturday ensures that objective function (1c) results in the best solutions. Again, we see that restricting the free tracks to LIFO tracks deteriorates the solution considerably.

7 Conclusions and future research

In this paper, we described the *train unit shunting problem* in a railway station. The problem is split into two steps, where one step matches arriving and departing train units, and the other step assigns train units to shunt tracks (the *Track Assignment Problem*, or TAP).

The computational results show that the matching step can be solved quite fast by the MIP solver of CPLEX 6.5. For the TAP, we propose a column generation heuristic. This heuristic uses dynamic programming for generating the columns. The underlying network structure is a novelty and speeds up the heuristic. Here, the heuristic results in near optimal solutions in an acceptable computation time in most cases.

In future research, we will extend the presented models and solution approaches to support planners for several related processes, such as crew planning, cleaning, and maintenance. Furthermore, we will also focus on extending the application of our solution procedure to other stations.

Acknowledgements

This research was partly supported by NS Reizigers, the Netherlands. The second and third authors were partly sponsored by the Human Potential Program of the European Union under contract no. HPRN-CT-1999-00104 (AMORE).

References

Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-price: Column generation

- for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.
- Ulrich Blasum, Michael R. Bussieck, Winfried Hochstättler, Christoph Moll, Hans-Helmut Scheel, and Thomas Winter. Scheduling trams in the morning. *Mathematical Methods of Operations Research*, 49(1):137–148, 2000.
- Jacques Desrosiers, Yvan Dumas, Marius M. Solomon, and François Soumis. Time constrained routing and scheduling. In *Handbooks in Operations Research and Management Science*, volume 8, pages 35–139. North-Holland, 1995.
- Richard Freling. *Models and Techniques for Integrating Vehicle and Crew Scheduling*. PhD thesis, Erasmus University Rotterdam, 1997.
- Giorgio Gallo and Federico Di Miele. Dispatching buses in parking depots. *Transportation Science*, 35(3):322–330, 2001.
- Shiwei He, Rui Song, and Sohail S. Chaudhry. Fuzzy dispatching model and genetic algorithms for railyards operations. *European Journal of Operations Research* 124, pages 307–331, 2000.
- Norio Tomii and Li J. Zhou. Depot shunting scheduling using combined genetic algorithm and PERT. In *Proceedings COMPRAIL 2000*, pages 437–446, 2000.
- Norio Tomii, Li J. Zhou, and Naoto Fukumara. Shunting scheduling problem at railway stations. *Lecture Notes in Artificial Intelligence 1611*, pages 790–797, 1999.
- Thomas Winter and Uwe T. Zimmermann. Real-time dispatch of trams in storage yards. *Annals of Operations Research*, 96:287–315, 2000.

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 Vlachos
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