

## Applying Operations Research techniques to planning of train shunting

Ramon M. Lentink, Pieter-Jan Fioole, Leo G. Kroon and Cor van 't Woudt

ERIM REPORT SERIES <i>RESEARCH IN MANAGEMENT</i>	
ERIM Report Series reference number	ERS-2003-094-LIS
Publication	2003
Number of pages	19
Email address corresponding author	rlentink@fbk.eur.nl
Address	Erasmus Research Institute of Management (ERIM) Rotterdam School of Management / Rotterdam School of Economics Erasmus Universiteit Rotterdam P.O. Box 1738 3000 DR Rotterdam, The Netherlands Phone: +31 10 408 1182 Fax: +31 10 408 9640 Email: <a href="mailto:info@erim.eur.nl">info@erim.eur.nl</a> Internet: <a href="http://www.erim.eur.nl">www.erim.eur.nl</a>

Bibliographic data and classifications of all the ERIM reports are also available on the ERIM website:  
[www.erim.eur.nl](http://www.erim.eur.nl)

# ERASMUS RESEARCH INSTITUTE OF MANAGEMENT

## REPORT SERIES *RESEARCH IN MANAGEMENT*

BIBLIOGRAPHIC DATA AND CLASSIFICATIONS		
Abstract	In this paper, we discuss a model-based algorithmic approach for supporting planners in the creation of shunt plans for passenger trains. The approach provides an example of a mathematical model and a corresponding solution approach for model based support. We introduce a four-step solution approach and we discuss how the planners are supported by this approach. Finally, we present computational results for these steps and give some suggestions for further research.	
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 R 4	Production Management Transportation Systems
	C 61	Optimization Techniques
European Business Schools Library Group (EBSLG)	85 A	Business General
	260 K 240 B	Logistics Information Systems Management
	255 A	Decision theory
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, optimalisatie, modellen, routeren	
Free keywords	Railway optimization, A* search, real world application, routing	

# Applying Operations Research techniques to planning of train shunting

Ramon M. Lentink<sup>\*, 1</sup>  
Pieter-Jan Fioole<sup>2</sup>  
Leo G. Kroon<sup>1, 2</sup>  
Cor van 't Woudt<sup>2</sup>

<sup>1</sup> Erasmus University Rotterdam, the Netherlands

<sup>2</sup> NS Reizigers, Department of Logistics, the Netherlands

\* Corresponding author, email: [rlentink@fbk.eur.nl](mailto:rlentink@fbk.eur.nl)

## Abstract

In this paper, we discuss a model-based algorithmic approach for supporting planners in the creation of shunt plans for passenger trains. The approach provides an example of a mathematical model and a corresponding solution approach for model based support. We introduce a four-step solution approach and we discuss how the planners are supported by this approach. Finally, we present computational results for these steps and give some suggestions for further research.

## 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 by other trains. 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 at a shunt yard together with several related processes is called *shunting*. The corresponding planning problem is called the *Train Unit Shunting Problem* (TUSP). 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. Finally, arrivals and departures are typically mixed in time. It is common to solve the TUSP for a 24 hour period, and station by station.

During the night, the goal of shunting is to select the positions and compositions of the trains at the shunt yard in such a way that the operations in the next morning can start up as smoothly as possible. This selection is subject to certain restrictions with respect to several related processes, which are introduced below. Furthermore, the created plans should be robust for it is almost sure that disruptions will occur in real time. When this happens, the plans should need as little changes as possible.

The Dutch railway network consists of approximately 3,000 kilometers of railway track and some 400 stations. This infrastructure is used on a daily basis by a total of 4,500 scheduled trains on lines ranging in length between 15 and over 200 kilometers. Approximately 1 million passengers travel more than 40 million kilometers each workday,

which makes the Dutch network one of the most heavily utilized networks in the world. During the weekend, the infrastructure is used less intensively.

In the Netherlands most trains are operated by train units, which are classified according to types and subtypes. Train units can move bi-directionally without the need for locomotives. Only train units of the same type can be combined to form trains, which contain at most 15 carriages. Subtypes belonging to the same type are discerned from each other by their numbers of carriages per train unit. The different subtypes of train units have different characteristics such as seating capacity and length. 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, which are typically used for intercity services.



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, given a timetable with times and exact compositions of the arriving and departing trains, a planner has to determine a matching of arriving and departing units at a station. A large part of this matching is already prescribed by the timetable.

For an appropriate introduction of the shunting process, we will start with the description of some relevant terms. 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, which continue their service after a short dwell time or come from complete ending trains. Departing shunt units are units that are coupled onto through trains or form complete starting trains. The related shunting processes are depicted in Figure 2. These related processes make the train units “flow” through the shunt yard between arrival at and departure from the station.

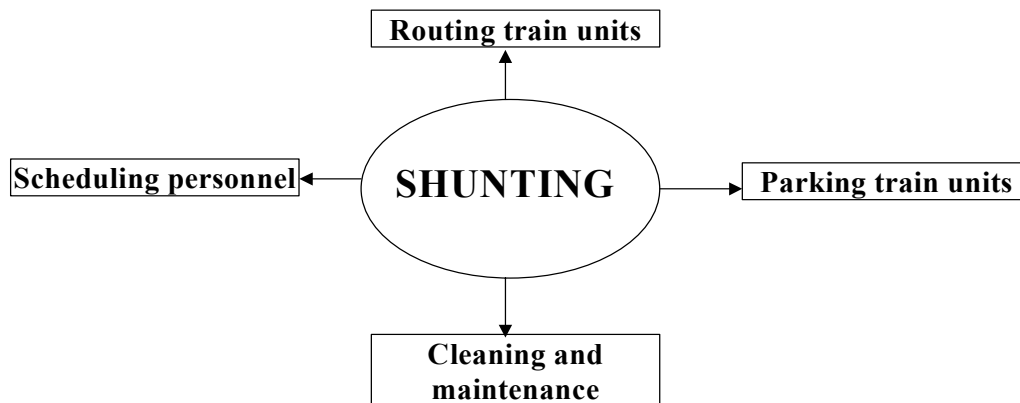


Figure 2: Elements of the shunting problem

It is mandatory that train units be checked every 48 hours for defects. This check can be done locally at the stations. Furthermore, all train units that lay over at a station with internal cleaning facilities should be cleaned internally. Finally, train units should be cleaned externally on a regular basis, which takes place at a so-called train-wash, which is available at some stations.

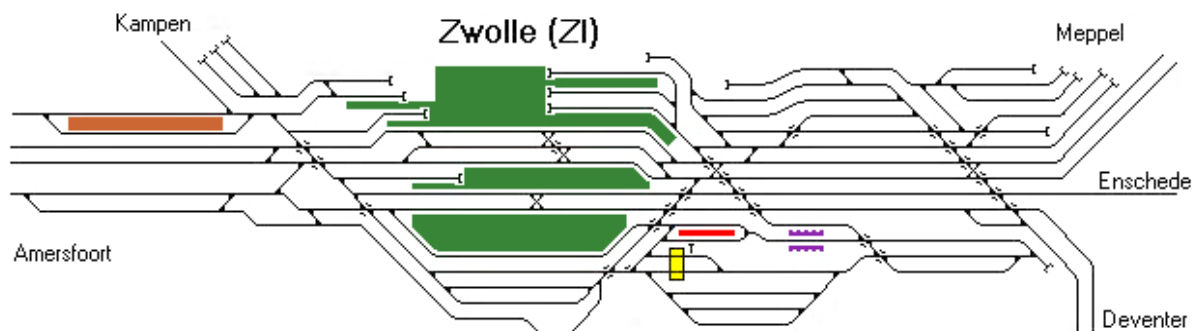
Routing of the train units takes place from the platform to the shunt yard and back. Sometimes it is possible to leave a train unit for a certain period of time on a platform track or to park it there some time before it is actually necessary. This introduces some flexibility with respect to the timing of the routing. Additional routing could be necessary for internal or

external cleaning operations. Of course, the routes of the different train units should neither conflict with each other nor with the routes of the through trains or other infrastructure reservations, such as track maintenance.

The tasks that result from the routing of train units, the coupling and decoupling of them, the cleaning, and the maintenance checks have to be assigned to shunting personnel. This personnel is local shunting personnel at the station under consideration. Since the different shunt tracks can be located quite far from each other, it is important to incorporate sufficient slack time between two tasks. In addition, there is uncertainty with respect to the exact arrival times and the process times of the activities. Finally, the schedules need to comply with union regulations as well as several laws. This particular crew scheduling problem has been analysed in Hoekert [12].

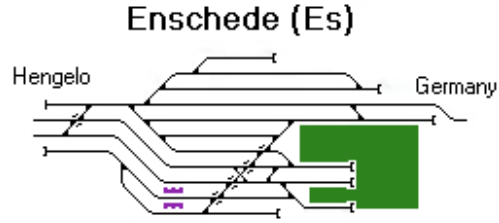
The parking of train units is far from trivial because space is usually scarce. In addition, the choice to park a shunt unit on a particular shunt track has several implications. First, when train units are of different subtypes, then the order of the train units on a shunt track is important. Second, the possible routes between the platforms and shunt tracks are restricted by this decision. Third, crew has to be available to carry out the resulting shunt activities within certain time intervals. Finally, planners prefer certain routes and shunt tracks 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, or if it is located far away from the platform tracks. A robust solution would be to park only units of the same subtype on each shunt track. This way, the planner can pick the first train unit if he desires a unit from this track. The disadvantage is that it may require additional activities in the morning for coupling units of different subtypes to form departing trains. Moreover, if the number of subtypes of train units is larger than the number of available shunt tracks at a certain point in time, mixing of subtypes is unavoidable.

Figure 3 shows the layout of the station Zwolle, which is a station in the northeastern part of the Netherlands. The rectangles in the figure represent the platforms, while next to the platform tracks and below those tracks several shunt tracks are located.



**Figure 3: The layout of station Zwolle**

Station Zwolle is a challenging station since the capacity is rather scarce and it has facilities for cleaning and washing train units. A smaller station is station Enschede, which is located in the eastern part of the country, near the German border. At station Enschede, less train units arrive and depart, and also less different subtypes of train units arrive and depart, as compared to station Zwolle. The layout of station Enschede is given in Figure 4.



**Figure 4: The layout of station Enschede**

An important characteristic of a shunt track is its *approach type*. Some of the tracks have a dead end side. We will call these *LIFO tracks* (Last In First Out). Tracks that do not have a dead end side are called *free tracks*. Especially these free tracks greatly complicate the parking problem, even though planners typically have a preferred side for each track. This is caused by the fact that one needs to decide the arrival and departure side for each train unit that one wants to park on a certain shunt track, which implies a huge increase in the set of feasible solutions for the parking subproblem.

In this paper, we will give a short literature review and we will present our solution approach for decision support for the routing and parking processes. Paragraph 4 describes how our solution approach supports the planners in creating shunt plans. In Paragraphs 5 to 8 we discuss the four parts of our solution approach in more detail. Hereafter, we will give some computational results of the steps of the solution approach before the conclusions and suggestions for further research are given in Paragraph 10.

## **2 A short literature review**

Cordeau et al. [3] provide a recent overview of the use of Operations Research in railway systems, focusing on train routing and scheduling problems. In addition, a good introduction to the TUSP and a detailed introduction to the main part of our solution approach including computational results can be found in Freling et al. [8].

Some special cases of the TUSP have recently been dealt with by Winter and Zimmermann [19] and Blasum et al. [2] for dispatching trams in a depot. Winter [20] extends this approach with length restrictions, mixed arrivals and departures and an application in a bus depot.

Di Stefano and Koci [16] study theoretical properties of several variants of subproblems of the TUSP. Furthermore, they also present algorithms for solving some of these subproblems.

Furthermore, Gallo and Di Miele [9] discuss an application for dispatching buses in a depot, with an extension of their models to take into account mixed arrivals and departures.

Tomii and Zhou [17] and Tomii et al. [18] propose a genetic algorithm that takes into account some of the processes of Figure 2. However, their problem is of a less complex nature, since in their context at most one train unit can be parked on a shunt track at the same time.

Lübbecke and Zimmermann [14] discuss a problem similar to the TUSP that arises at an in-plant private railroad. In this problem, one assigns transportation requests to certain regions of the in-plan railroad and selects cars of specific types from a shunt track in this region. However, the authors only discuss LIFO tracks and assume that there is no prescribed sequence of different types of cars in a train. In addition, it is assumed that there are no limitations for this temporary parking.

In addition, the subject of the paper by He et al. [11] 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.

Dahlhaus et al. [4] discuss the rearranging of carriages in one train to group them by destination, which is a similar problem. Their goal is to use a minimum number of tracks for this rearrangement and to show that this problem belongs to the class of most difficult optimisation problems.

Zwaneveld [22] also studied a routing problem for train units over railway infrastructure. In this problem, one is looking for a set of routes for trains in a one hour period where arrival and departure times are fixed. He develops a solution approach based on state of the art techniques from the field of Operations Research. This approach is applied to a number of railway stations in the Netherlands. Compared to the problem of Zwaneveld, the flexible start and end times, the fact that infrastructure can already be reserved for certain time intervals for e.g. through trains, the splitting and recombining of train units, and a different planning period complicate our problem (typically 24 hours instead of 1 hour).

### **3 Solution approach**

For practical instances, the TUSP becomes far too large to be solved as one integrated optimisation problem. Therefore, the problem is decomposed in 4 parts, and we developed a four-step algorithmic solution approach that assists planners in creating parts of the shunt plan. The decomposed approach enables planners to view and possibly modify partial solutions between the steps. The four steps are:

- Step 1. Matching arriving to departing train units
- Step 2. Estimating routing costs of train units
- Step 3. Parking of train units on shunt tracks
- Step 4. Routing of train units

In the first step, we match arriving shunt units to departing shunt units. In this problem, the order of the subtypes of train units in the train, prescribed by the timetable, is considered fixed and has to be respected. Furthermore, type mismatches between the desired subtype and the supplied subtype are not allowed. This step results in a set of so-called *blocks*. A block is a set of shunt units that remain together from their arrival at the station until their departure from the station. The main objective in this step is to create a minimum number of blocks since this implies a minimum number of shunt activities.

The second step computes a lower bound on the routing costs of each block from its arrival platform to each shunt track and back to its departure platform. These lower bounds are required as input for Step 3, where we choose a shunt track for each train unit that needs parking. Here, we take the routes of through trains and freight trains and other known infrastructure reservations into account. Elements of the routing costs of one block are costs for the travelled distance and costs for the number of reverses in direction. A reverse in direction requires the driver to walk to the other end of the train because he needs to be upfront. This takes additional time. This step is based on work by Van 't Woudt [21] and Fioole [5]. Note that the costs can differ for different approach sides of free shunt tracks.

Step 3 decides for each block on which shunt track it should be parked, taking into account certain preferences of the planner. Furthermore, the capacity of the shunt tracks should never be exceeded and it is not allowed for a block to obstruct another block upon arrival or departure, which we will call a *crossing*. Note that, for free tracks, the side of arrival or departure on a shunt track also influences whether or not a crossing occurs.

The fourth step determines the final routes of shunt units, where we also incorporate the relations among the routes of the blocks. This will result in a proposal for the final routes of the blocks to the planner. Again, this step is based on work by Van 't Woudt [21], with extensions from Fioole [5].

The steps are chosen in such a way that the scope of the presented models can easily be extended. An example of such an extension is to incorporate the planning of the internal cleaning of rolling stock. Of course, the described decomposition can result in solutions of inferior quality. This effect is studied by Fodor Birtalan [6], which provides sufficient support for the proposed decomposition.

#### **4 How does the solution approach support the planner?**

We have been developing a prototype of a decision support system for the TUSP. Besides algorithms that are able to solve parts of the problem, this system also contains an intuitive interface for operating these algorithms, as well as several graphical representations of the solutions, and facilities for interactively changing the solutions.

In the first step, the planner is able to fix a part of the matching of arriving shunt units to departing shunt units a priori. This fixation would reflect considerations of the planner that are not taken into account in the underlying mathematical model. Moreover, the planner is also able to change the solution of the first step afterwards to some extent.

In the third step, the system supports the planner in determining the tracks where the blocks will be parked. Again, the planner has a lot of flexibility for creating a solution. He can determine a subset of the shunt tracks where a block should be parked beforehand. For example, blocks containing diesel units need to be parked on specific tracks with fuelling facilities. Furthermore, he can also prohibit to park blocks on a certain track. In casu, blocks containing electrical train units cannot be parked on shunt tracks without catenary. Finally, the planner also has the possibility to guide the optimisation into certain directions by attaching certain weights to elements of the objective function. These elements are:

1. Penalties for not parking blocks. Blocks that cannot be parked need special attention of the planner, since either he has to plan the parking himself, or he needs to check if the units can remain at the arrival or departure platform.
2. Penalties for parking blocks with different subtypes of train units on a shunt track. With this element the planner can favour solutions where units of the same subtype are parked on the same shunt track.
3. A bonus for corresponding blocks that are parked on the same shunt track next to each other in the right order. This occurs when the shunt units of the blocks leave in the same departing train but come from different arriving trains. When these blocks are parked next to each other on the same track and in the right order, they can be combined on the shunt track and considered as one block when the train units need to be routed to the departure platform, which results in less work. When these blocks are not parked next to each other on the same track or are parked in the wrong order, we call the departing train a *broken departure*.
4. Penalties for using certain shunt tracks. Sometimes, shunt tracks can be of strategic importance for related processes, such as cleaning, or they may be used for freight trains at night. With this penalty a planner can try to use such undesired tracks as little as possible. Furthermore, with this penalty the planner can also indicate that a certain track is out of service due to maintenance.

Related to the routing steps 2 and 4, the user can configure 15 different parameters in total. Six of these deal with the amount of time infrastructure should be reserved for a train



movement. Obviously, if these parameters are set relatively low, then more movements can be planned, but the solution will also be less robust. Moreover, the planner can specify a number of minutes he prefers a shunting movement to take place after arrival of a train (or before departure) and set a penalty for each minute the movement deviates from the specified time. Moreover, for a reverse in direction, the required time can be specified by a constant time and additional time dependent of the length of the train. In addition, a penalty for reversing the direction of the train can be set. This way, the planner can influence the number of reverses in direction in all routes. Finally, a penalty can be set for movements that are planned simultaneously. Here, the occurrence of less simultaneous movements implies that fewer drivers are necessary at the same time.

As mentioned before, during the weekends fewer trains are in service, as a result of decreased demand for transportation. Therefore, typically on Friday several units arrive at a shunt yard, remain there for the entire weekend, and leave again on the next Monday. In order to incorporate this situation, the solution approach can handle more arriving shunt units than departing ones (e.g. on Friday at certain yards), such that the yard is not empty at the end of the planning period. Also, the solution approach can handle a non empty shunt yard at the start of the planning period.

## 5 Matching of arrivals to departures

In the first step, we focus on efficiently matching arriving shunt units to departing ones. This means that we want to keep units together as much as possible since this implies less work. The matching should exactly respect the required subtypes, i.e. it is not allowed to provide a train with a different train unit. Furthermore, if a train consists of several subtypes, the prescribed order of the subtypes of train units in the train has to be respected.

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*, where a part is a subset of adjacent train units of a train. The arcs represent feasible configurations of possible parts of the train. Given the parts, a block represents a combination of a part of an arriving train and an identical part of a departing train.

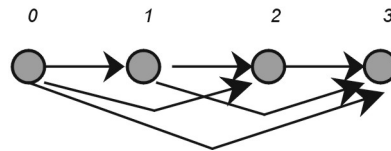


Figure 5: The network of a train with 3 units

Figure 5 depicts the network for a train consisting of 3 units. Note that each arc corresponds to 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 5 implies that train units 1 and 2 form one part, and train unit 3 forms another part.

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. We use  $I_j$  ( $J_i$ ) as the set of arriving (departing) parts with the same matching configuration as departing part  $j$  (arriving part  $i$ ) that arrive before departing part  $j$  (depart after arriving part  $i$ ). Formally, we introduce a mathematical formulation with the following binary decision variables:

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

A matching of an arriving part to a departing part is feasible if the units in both parts are the same, these units are in the same order, and the departing part leaves the station after the arrival of the arriving part. We define  $T^a$  ( $T^d$ ) as the set of trains with arriving (departing) shunt units. The arcs in a network of train  $t$  are given in the set  $A^t$ . We additionally use  $A_h^{t+}$  ( $A_h^{t-}$ ) as the set of arcs out of (into) node  $h$  for train  $t$ . Finally, the set  $C_t^-$  is the set of all intermediate nodes in the network of train  $t$ .

The model now becomes:

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

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

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

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

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

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

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

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

The objective (1) is to minimize the weighted sum of the number of parts, with penalty  $Q$ , and the matching costs, since minimizing the number of parts results in a matching where train units are kept together as much as possible. The matching costs contain e.g. cost for the time difference between the arrival time of an arriving part and the departure time of a departing part. If these times are close, routing the unit(s) from the arrival platform to the departure platform suffices and it is not necessary to park the unit(s) at the shunt yard. Flow conservation 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. Constraints (6) guarantee that each arriving part is matched 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.

As mentioned before, it is possible that an arriving shunt unit is not matched to a departing shunt unit in the planning period. In this case, the unit will remain parked on a shunt track until the end of the planning period. This can be modelled easily by incorporating dummy train units, which depart after the end of the planning period. Of course, a similar argument holds for departing shunt units that are not matched to arriving shunt units.

Typically, each free track has one side that planners prefer to use, although both sides can be used. Therefore, we would like algorithms that take into account this preference. Together

with the LIFO tracks, this implies that we would prefer matchings that adhere to the LIFO principle. Fodor Birtalan [6] discusses extensions of the presented solution approach to take into account these preferences.

In Paragraph 9 we will show that this formulation can be solved quite efficiently by available standard software packages from the field of Operations Research, like Cplex.

## 6 Parking train units on shunt tracks

As we described in the previous paragraph, the matching of supply and demand of shunt units results in a set of blocks  $B$ . Blocks that do not need to be parked on a shunt track because their arrival and departure times are sufficiently close are not included in  $B$ . For each block we know the arrival and departure times and the arrival and departure platforms. In addition, in the second step we estimated the route costs between these platforms and all feasible shunt tracks for each block, which is described in the next two paragraphs. Given this information, the third step of our solution approach consists of assigning blocks to shunt tracks at minimum costs. We call this subproblem the *Track Assignment Problem* (TAP).

We formulate the TAP as a Set Partitioning Problem with additional constraints. The Set Partitioning Problem is a well-known problem in Operations Research and is defined in Garey and Johnson [10]. 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:

- it does not contain crossings. Recall that this occurs when a train unit is obstructing the arrival or departure of another train unit;
- the total length of the units on the track never exceeds the length of the track;
- all blocks of the subset are allowed to park on the track.

Let  $S$  be the set of shunt tracks, 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  $b \in B$ . Given these sets, we define the following decision variables:

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

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

The parameter  $c_k^s$  models the costs of assignment  $k$  on track  $s$ . These costs consist of the sum of the estimated route costs of the different blocks in the assignment as well as the penalties and bonuses described in Paragraph 4 regarding this assignment. In addition, the parameter  $d$  models a penalty if a block is not assigned to any track. The TAP is then formulated as follows:

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

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

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

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

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

We aim at minimizing the costs of a shunt plan, such that as many blocks as possible are assigned to shunt tracks, which is modelled in (9). Constraints (10) state that each block is covered by exactly one assignment for one shunt track or it is not parked at all. Constraints (11) model that each shunt track can have at most one assignment.

Note that the solution space, measured by the number of variables or feasible assignments of blocks to shunt tracks  $x_k^s$ , grows exponentially in the number of blocks. A way to overcome this disadvantage of exponentially many decision variables is the use of *column generation* techniques. Column generation is a state of the art technique from Operations Research. It is becoming more and more one of the preferred solution approaches to deal with a huge number of variables. Although the basic theory goes back to the 1960's, the popularity of this approach started to rise only in the 1980's, stimulated by the advances in computer hardware. Lübbecke and Desrosiers [13] provide a recent overview of applications as well as advances in theory. A standard reference on this subject is Barnhart et al. [1], while Freling [7] uses column generation to solve vehicle and crew scheduling problems in an integrated manner. More information on the algorithm for this application can be found in Freling et al.[8].

## 7 Modelling the infrastructure and its use for the routing steps

Before discussing Steps 2 and 4 of our solution approach related to routing problems, we will introduce a model of the infrastructure, which is suitable for solving these problems. Attention is paid to this issue because a straightforward model can be incapable of detecting all possible conflicts.

Recall that we solve routing problems in Steps 2 and 4 of our algorithmic solution approach. In Step 2 we know the configuration of the blocks. Given these blocks and the reservation of the infrastructure by the through trains, we estimate the costs of routing a block from its arrival platform to each shunt track, and from each shunt track to its departure platform. These estimated costs are part of the input of Step 3, where the actual shunt track for each block is selected. When this shunt track has been selected for each block, Step 4 tries to find routes for each block to and from it. An additional complication in this fourth step is that infrastructure reservations for the routes of the other blocks need to be respected.

In Figure 6, the standard representation of a part of a station, as used by Dutch railway organizations, and the translation of that representation into a graph are given. In the left part, the relevant points of the infrastructure are given, such as tracks and switches. Here, the tracks are encircled. The resulting graph is given in the right part, where all tracks and relevant points are nodes in the graph. If two nodes are directly connected in the left part then an arc in the corresponding graph connects the nodes.

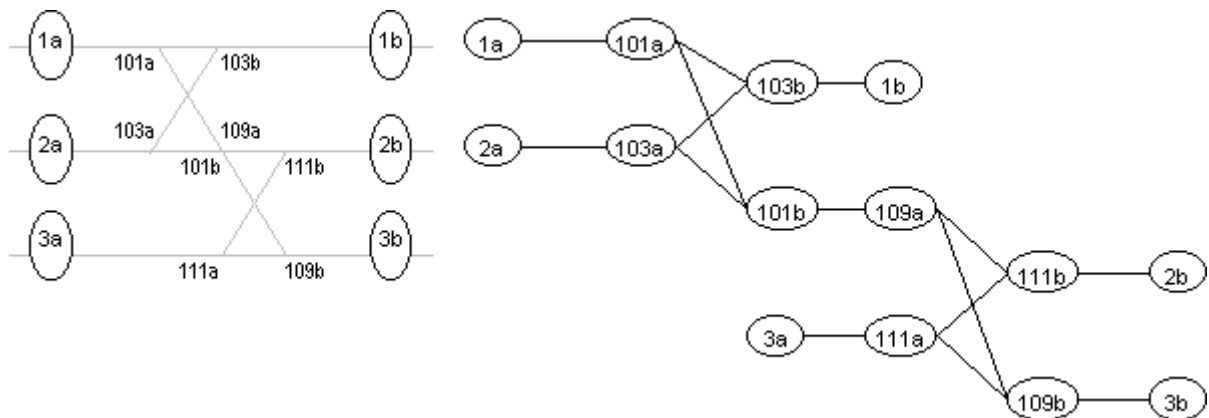
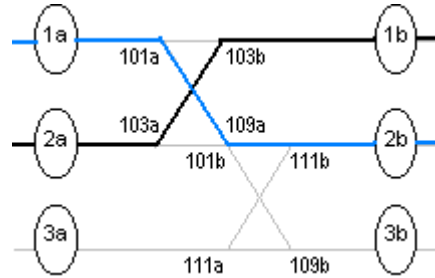


Figure 6: An instance of railway infrastructure and its typical representation

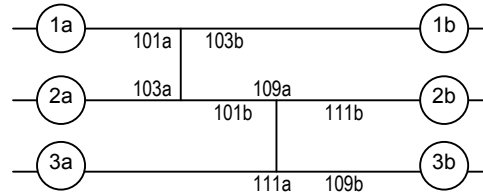
Each node has several attributes: a cost (or weight), an indicator whether or not it is possible to reverse directions at this node, at most two arcs on the left side, and at most two arcs on the right side. Suppose that in the example reversing directions is only allowed at nodes 1b, 2b and 3b. If, in this case, we are looking for a route from track 3a to track 2a, we need to reverse direction either at node 2b or 3b.

Unfortunately, this model is not suitable for solving the routing problems studied in this paper, since it does not detect all possible conflicts. Consider for example the routes of the trains in Figure 7 from track 1a to 2b and from track 2a to 1b. From a practical point of view, it is obvious that the routes in this example conflict because they simultaneously use the same switch. However, the model does not detect the conflict since the routes do not share any node in the graph.



**Figure 7: An example of conflicting routes**

In order to detect the conflict of Figure 7, we need to use a little more sophisticated model. The example infrastructure results in the model of Figure 8. In this representation, a route from track 1a to 2b passes three nodes using switch 101 (101a, 103a and 101b). This representation enables us to detect all conflicts on the infrastructure.



**Figure 8: An example of the final model of the infrastructure**

We use the model of Figure 8 for the reservations of the infrastructure. Such a reservation by a train movement is split into reservations of route-elements of this movement. This enables us to release a reservation a short period after the train has left the route element. This period of time depends on the safety system.

## 8 The planning algorithm for finding routes

Given the model of the infrastructure of the previous paragraph, the reservation of the infrastructure by through trains, and possibly other reservations of the infrastructure, we need to find routes for the shunt units. Algorithms for searching routes can be partitioned into two classes:

1. Algorithms that search routes simultaneously.
2. Algorithms that search routes sequentially.

An obvious advantage of algorithms belonging to the first class over algorithms of the second class, is the possibility to compute an overall optimal solution for routing all shunt units over the infrastructure, while taking into account all restrictions and interdependencies.

However, a disadvantage of such an approach is typically the amount of runtime that is needed for finding an optimal solution.

Algorithms of the second class can be partitioned further based on the characteristics *completeness* and *optimality*. A complete algorithm of the second class will find a feasible route if it exists. Optimality for such an algorithm implies that it will find the best route, given the reservations of already planned routes.

The methodology of Zwaneveld [22] belongs to the first class of algorithms. Since practical instances typically consist of approximately 70 blocks and 20 shunt tracks, it is clear that algorithms of the first class will not be able to produce optimal solutions in reasonable computation time, typically several minutes, for the considered real world problems. This conclusion is supported by computational results in Zwaneveld [22]. Therefore, methodology of this class is not useful for supporting planners in the TUSP. Another advantage of a sequential algorithm is that it better resembles the current practice of planners, which comforts them.

The class of *informed search* algorithms, a subset of the search algorithms, contains algorithms that are complete. Furthermore, informed search algorithms have some prior knowledge of the state space, which they are searching. This knowledge is used when extending a partial solution. For example, suppose you are interested in the shortest path from Amsterdam to Paris in a certain network and assume you have found a partial shortest path that brings you from Amsterdam to Brussels. In this case, the additional knowledge is the estimated length of the remaining shortest path from Brussels to Paris, which can be e.g. the geometric distance between Brussels and Paris.

Before discussing the general structure of informed search algorithms, we need to introduce some notation first. We define the *costs* of a partial path  $p$ , denoted with  $g(p)$ , as the sum of the costs of the nodes in the path plus additional costs based on the characteristics of it (e.g. penalties for reversing directions). Furthermore, the function  $h(p)$  estimates the remaining costs for completing partial path  $p$ . In fact, this function models the knowledge of the state space. There is a wide range of possibilities for the functions  $g(p)$  and  $h(p)$ . For instance, if one is interested in finding a solution as fast as possible, one can choose  $g(p) \equiv 0$  for all  $p$ , and  $h(p)$  as the expected number of remaining steps. Finally, we define  $n^*$  as the last node in path  $p$  and  $E(n)$  as the set of edges incident to node  $n$ . The general structure of informed search algorithms is given in Figure 9.

**Informed Search(problem, network)**

```

Step 1: Initialise the list of partial solutions  $P$  with an empty partial
        solution  $p$ ,  $P = \{p\}$ 
Step 2: If  $P$  is empty,  $P = \emptyset$ , then stop
        Else let  $p^* = \operatorname{argmin}_{p \in P} f(p)$  and remove  $p^*$  from  $P$ ,  $P = P \setminus \{p^*\}$ .
Step 3: If  $p^*$  is a feasible solution then stop
        Else for all  $m \in E(n^*)$  do
            If  $q = p^* \cup \{m\}$  is a feasible partial solution then
                 $P = P \cup \{q\}$  and return to Step 2

```

**Figure 9: The general structure of informed search algorithms**

A well known and popular informed search algorithm from the field of Artificial Intelligence is A\* search, which is optimal if the function  $h(p)$  is a lower bound to the actual costs of the remaining route. Typically, A\* searches for paths from a source node  $s$  to a sink node  $t$ . More information on A\* search can be found in Pearl [15]. The algorithm is suitable

for solving the routing problems in this paper, since we are able to give good lower bounds on the costs of the remaining route, which makes the algorithm fast. These lower bounds are determined by the cheapest paths on the infrastructure between all pairs of nodes, when through trains, the other shunt movements as well as track maintenance are neglected.

However, the particular characteristics of the routing problems in this paper require special attention. In these problems, infinite routes are possible, due to the possibility of reversing directions. Furthermore, the infrastructure is well known and stable, but the nodes are not always available. This occurs for example when a track goes out of service due to maintenance for a certain period of time.

Incorporating additional stop criteria for the algorithm can easily solve the first characteristic. Due to these stop criteria, the algorithm loses optimality and completeness. However, these stop criteria exclude undesired solutions from the planners' point of view. The stop criteria we implemented are a maximum number of iterations for the overall problem, maximum costs for a route, and a maximum number of reverses in direction in a route.

The second characteristic implies that the feasibility of extending a partial path is also time dependent. Since the route will have to take a detour when a certain node is temporarily unavailable, the function  $h(\cdot)$  remains a lower bound and optimality remains guaranteed.

The routing problems in our solution approach are solved by applying the so-called Occupied Network A\* search (ONA\* search) algorithm. Before discussing the ONA\* algorithm, we introduce some additional notation. The function  $h(x, y)$  gives a lower bound on the costs of the cheapest path from node  $x$  to node  $y$ , which is computed in an empty network. In addition,  $c(x)$  represents the costs of node  $x$  excluding possible penalties  $C(x, p)$ , for a partial path  $p$  at a node  $x$ . In addition,  $d(p)$  returns the number of changes in direction in path  $p$ . Finally,  $N$  is the maximum number of iterations,  $M$  is the maximum costs of a route, and  $K$  is the maximum number of reverses in direction in one route. Given this additional notation, the ONA\* search algorithm is sketched in Figure 10.

```

ONA* Search(node s, node t, network)

Step 1: Initialise the list of partial solutions  $P$  with  $s$ ,  $P = \{s\}$ , set
 $g(s) = c(s) + C(s, s)$  and  $f(s) = g(s) + h(s, t)$ . Finally,  $t = 0$ .
Step 2: If  $P = \emptyset$  or  $t = N$  then stop
      Else let  $p^* = \operatorname{argmin}_{p \in P} f(p)$  and remove  $p^*$  from  $P$ ,  $P = P \setminus \{p^*\}$ .
      If  $f(p^*) \geq M$  then stop
      Else if  $d(p^*) \geq K$  then
        set  $t=t+1$  and return to Step 2
Step 3: For all  $m \in E(n^*)$  do
      If  $q = p^* \cup \{m\}$  is a feasible partial solution then
         $g(q) = g(p) + c(m) + C(m, q)$  and  $f(q) = g(q) + h(m, t)$ 
      If  $m = t$  then stop ( $q$  is the best route between  $s$  and  $t$ )
      Else  $P = P \cup \{q\}$ , set  $t=t+1$  and return to Step 2

```

Figure 10: The ONA\* search algorithm

The ONA\* search algorithm can be extended to allow for dynamic start-times of routes for shunt units. This is implemented by initializing the list  $P$  in Step 1 with a set of partial routes, one for each potential start-time. Furthermore, a variable penalty for deviations from the ideal start-time of the route will be accounted for. This feature is very helpful for planners, since it

resembles their current practice: the start-times of the routes for shunt units are flexible to some extent.

As mentioned before, the sequential search approach implies that the order of selecting the routes for the blocks matters for the overall solution, since it does not necessarily find the overall optimal solution. To reduce this setback, a 2-opt improvement strategy has been implemented.

The 2-opt improvement strategy assumes that the route requests are ordered in some fashion, typically on the start-time of the request. Now, the strategy loops over all requests. For each selected request, it performs a second loop to select a second request with a higher index and tries to swap the order of these two requests. The requests should overlap in time and should be distinct, because otherwise no improvement will occur. If this swapping results in a better overall solution then the routes are changed. Note that for an individual route the solution might get worse but the overall solution will improve. Furthermore, note that this is a local search strategy, which does not guarantee to find an optimum solution and therefore it can be beneficial to apply the 2-opt improvement strategy more than once. This will be a subject of our computational results in Paragraph 9. After introducing  $I$  as the number of requests,  $R$  as the total route costs,  $r_i$  as the costs of route request  $i$ , and  $b_i$  as the best route for request  $i$ , the framework for the 2-opt improvement strategy is described in Figure 11:

```

2-opt strategy(list of route requests, network)

Step 1: Set  $R = \sum_i g(r_i)$ , and for all  $i = 1, \dots, I$  set  $b_i = r_i$ 
Step 2: For all  $i = 1, \dots, I-1$  do
    For all  $j = i+1, \dots, I$  do
        Release the reservations for routes  $b_i$  and  $b_j$ 
        Call ONA* Search (start( $b_j$ ), end( $b_j$ ), network), save  $r_j'$ 
        Call ONA* Search (start( $b_i$ ), end( $b_i$ ), network), save  $r_i'$ 
         $R' = R - g(r_i) - g(r_j) + g(r_i') + g(r_j')$ 
        If  $R' < R$  then set  $R = R'$ ,  $b_i = r_i'$ , and  $b_j = r_j'$ 

```

**Figure 11: An extension of ONA\* search to improve the order of route finding**

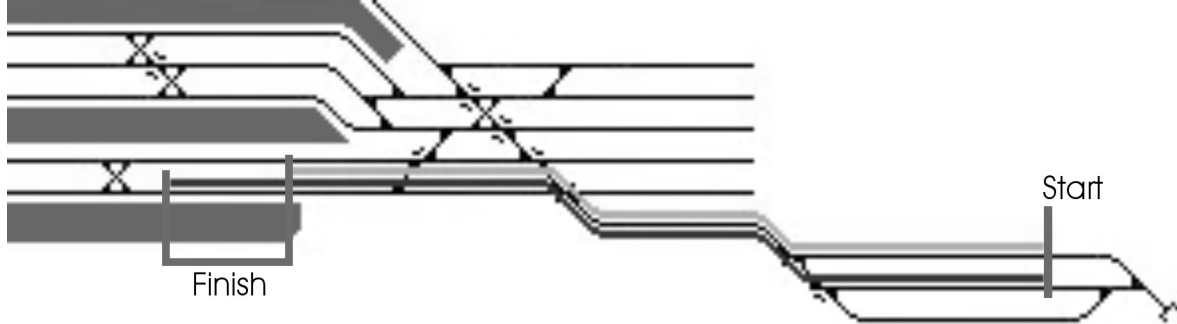
To illustrate the effect of the 2-opt improvement strategy, we consider a small example with only 2 route requests. Both requests have to end on a specific platform track  $t$ , where the order of the blocks on this track is important. Furthermore, they also originate from a track on the right side of platform  $t$ . Suppose that the route that has to end on the right side of platform track  $t$  is planned first. Finally, we assume that the timing of the reservations of the infrastructure for the first route imply that it is impossible to plan the second route before the start of the first route or after the block of the first route has left track  $t$ . Therefore the second route has to go around the first block. This situation is sketched in Figure 12.



**Figure 12: The solution to our example without the 2-opt improvement strategy**



However, if we change the order of planning these routes, we find an improved solution because there is no occupation for platform  $t$  for the route destined for the right side of the platform. This improved solution is depicted in Figure 13. Note that both routes share a major part of the infrastructure they use. This is no problem because the routes take place sequentially with enough buffer time between them.



**Figure 13: The solution to our example with the 2-opt improvement strategy**

A third extension is the focus on reducing the number of simultaneous shunt routes. Currently, personnel planning and rolling stock planning for shunting are performed consecutively. Therefore, the number of simultaneous routes is taken as a proxy for the required number of shunt personnel in a certain shunt plan. After the morning and evening rush hours, the peaks for routing shunt units occur. In order to reduce the required number of employees per time interval, and thereby to increase the possibility of a feasible solution, the algorithm can be guided towards a solution where a minimum number of concurrent routes are necessary. This is implemented by penalizing routes that are planned at the same time interval. This interval contains the actual start and endtimes of the routes as well as an additional 10 minutes before the start. These 10 minutes are a proxy for the time required to walk to the start location of a route.

## 9 Computational results

Currently, the automated support for the planners, described in this paper, is in a research phase. This research is conducted at stations Zwolle and Enschede, whose layouts are given in Figure 3 resp. Figure 4. Some more details on both stations are presented in Table 1.

	Enschede	Zwolle
# Trains (daily)	138-160	550-600
# Train units (daily)	212-293	800-1100
# Types of train units	4	12
# Platforms	4	17
# LIFO tracks	8	4
# Free tracks	5	15

**Table 1: Some statistics on stations Enschede and Zwolle**

The dataset for station Enschede represents a weekend from 08:00 AM on a typical Friday until 08:00 AM on the subsequent Monday. The dataset for station Zwolle represents a 24-hour period starting at 08:00 AM on a typical Wednesday. First, we will discuss the results for station Enschede, followed by the results for station Zwolle. For each dataset, we discern 4 scenarios for Step 3 of our algorithmic solution approach:

1. Focus on parking as many blocks as possible.
2. Primary focus on parking as many blocks as possible and secondary focus on minimising the number of shunt tracks with more than one subtype parked on it.

3. Primary focus on parking as many blocks as possible and secondary focus on minimising the number of broken departures.
4. Primary focus on parking as many blocks as possible and secondary focus on both secondary items of the previous two scenarios.

### 9.1 Results for station Enschede

For station Enschede, Step 1 requires a few seconds of computation time and results in 57 blocks that need to be parked at the shunt yard. The estimates for the routing costs are computed within a minute as well. With the results of the first two steps we are able to compute the results for the different scenarios of Step 3, which are given in Table 2:

	1	2	3	4
# Blocks not parked	0	0	0	0
# Adjacent different subtypes	10	0	10	1
# Combinations of arriving blocks into one departing train	3	3	10	6
Computation time (in sec.)	34	134	22	1660

**Table 2: Results for the different scenarios of Step 3 applied to station Enschede**

The results show that the algorithm is always able to park all the blocks on the shunt yard. Furthermore, we see that the secondary objectives have a desired effect on the respective performance indicators. The computation times differ substantially for the scenarios, especially the runtime of outlier Scenario 4 is worth mentioning here.

Given the results of these 4 scenarios, we can analyse Step 4 in more detail. The results for these scenarios are given in Table 3:

	1	2	3	4
# Route requests	545	547	540	544
<i>No 2-opt improvement strategy</i>				
# Requests not found	16	20	19	35
Relative gap with minimal route costs	25,8%	29,5%	29,9%	46,7%
Computation time (in sec.)	12	14	13	17
<i>Apply 2-opt improvement strategy once</i>				
# Requests not found	10	10	11	28
Relative gap with minimal route costs	17,4%	16,9%	19,4%	36,7%
Computation time (in sec.)	18	25	23	50
<i>Apply 2-opt improvement strategy twice</i>				
# Requests not found	9	10	11	28
Relative gap with minimal route costs	16,8%	16,9%	19,3%	36,5%
Computation time (in sec.)	27	36	32	78

**Table 3: Results for routing blocks in station Enschede**

The relative gap with the minimal route costs is determined by the relative difference between the total route costs of Step 4 and the total estimated route costs from Step 2 for the selected tracks for the blocks in the third step. We see that the computations for this step are quite fast since the computation time exceeds one minute in only one case. Moreover, the 2-opt improvement strategies improve the solution by saving on secondary objectives as well as by granting more requests.

## 9.2 Results for station Zwolle

Executing the first step for station Zwolle requires 29 seconds and results in 69 blocks that need parking. The second step estimates the costs for the potential routes of these 69 blocks in approximately 50 seconds, which is acceptable. Again, we have enough information to execute Step 3, which results in the following table:

	1	2	3	4
# Blocks not parked	1	0	0	0
# Adjacent different subtypes	21	0	20	0
# Combinations of arriving blocks into one departing train	0	4	10	10
Computation time (in sec.)	4206	738	2154	906

**Table 4: Results for the different scenarios of Step 3 applied to station Zwolle**

For station Zwolle, the third step is unable to park one block in one scenario. We see that the required computation time varies between 12 and over 40 minutes, thereby making this step rather intensive for station Zwolle from a computational point of view. Moreover, it is interesting to see that applying the solution approach to Scenario 4 yields very good results and requires the smallest amount of computation time.

Finally, we studied the effect of these scenarios for station Zwolle on Step 4. Our findings are summarised in Table 5.

	1	2	3	4
# Route requests	740	736	734	734
<i>No 2-opt improvement strategy</i>				
# Requests not found	11	4	6	7
Relative gap with minimal route costs	34,4%	17,5%	21,4%	23,2%
Computation time (in sec.)	13	12	13	12
<i>Apply 2-opt improvement strategy once</i>				
# Requests not found	4	1	1	1
Relative gap with minimal route costs	14,1%	7,4%	7,8%	6,8%
Computation time (in sec.)	25	19	22	24
<i>Apply 2-opt improvement strategy twice</i>				
# Requests not found	4	1	1	1
Relative gap with minimal route costs	14,1%	7,4%	7,6%	6,7%
Computation time (in sec.)	36	27	31	36

**Table 5: Results for routing blocks in station Zwolle**

From this table, we can conclude that the second 2-opt improvement strategy has little added value, while the first strategy does remarkably well.

## 10 Conclusions and further research

In this paper, we introduced a model based algorithmic solution approach for creating shunt plans in a railway station. These shunt plans prescribe what should happen with train units when they are temporarily not necessary to operate a timetable. Elements of such plans are matching of arriving and departing train units, decisions where to park blocks, and determining detailed routes of train units in a station.

Our solution approach provides support for the planners, who are creating plans for these elements as well as the shunt plans as a whole. The elements are discussed in detail in this paper. The proposed solution approach consists of a decomposition of the problem into 4

subproblems, and solving each subproblem separately. Furthermore, we provide computational results for our solution approach based on real life data of NS Reizigers for the stations Zwolle and Enschede.

The plans generated by our approach have been validated extensively with planners at NS Reizigers. Although this lead to several improvements, planners feels that this approach indeed supports them in creating shunt plans. Planners are still required to check the generated plans, make modifications, and finalize the plan. One reason for this is the fact that our proposed solution approach may not be able to find routes for all blocks in the last step.

In our future research, we will extend this solution approach with modules for internal and external cleaning, and for crew planning. This will result in additional steps in our approach and maybe in minor changes in these steps. Furthermore, we will also look at other stations to apply this solution approach to.

## **Acknowledgements**

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

The authors want to thank several students who have implemented parts of the presented algorithms.

## **References**

- [1] 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.
- [2] 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.
- [3] Jean-François Cordeau, Paolo Toth and Daniele Vigo. A survey of optimisation models for train routing and scheduling. *Transportation Science*, 32(4):380-404, 1998.
- [4] Elias Dahlhaus, Peter Horak, Mirka Miller, and Joseph F. Ryan. The train marshalling problem. *Discrete Applied Mathematics*, 103:41-54, 2000.
- [5] Pieter-Jan Fioole. *Rintel, het automatisch genereren en verbeteren van een rangeerplanning* (in Dutch). Masters thesis, Erasmus University Rotterdam, 2003.
- [6] Ildikó Fodor Birtalan. *Matching arrivals to departures in the train unit shunting problem*. Masters thesis, Erasmus University Rotterdam, 2003.
- [7] Richard Freling. *Models and techniques for integrating vehicle and crew scheduling*. PhD thesis, Erasmus University Rotterdam, 1997.
- [8] Richard Freling, Ramon M. Lentink, Leo G. Kroon and Dennis Huisman. Shunting of passenger train units in a railway station (accepted for publication in *Transportation Science* in 2003).
- [9] Giorgio Gallo and Frederico Di Miele. Dispatching buses in parking depots. *Transportation Science*, 35(3):322-330, 2001.
- [10] Michael R. Garey and David S. Johnson. *Computers and intractability, a guide to the theory of NP-completeness*. Freeman, 1979.
- [11] Shiwei He, Rui Song, and Sohail S. Chaudhry. Fuzzy dispatching model and genetic algorithms for railyard operations. *European Journal of Operations Research*, 124:307-331, 2000.
- [12] Wichert Hoekert. *Het maken van diensten voor rangeerpersoneel* (in Dutch). Masters thesis, Erasmus University Rotterdam, 2001.

- [13] Marco E. Lübbecke and Jacques Desrosiers. *Selected topics in column generation*. Technical Report G-2002-64, GERAD.
- [14] Marco E. Lübbecke and Uwe T. Zimmermann. *Shunting minimal rail car allocation*. Technical report, Braunschweig University of Technology, 2003.
- [15] Judea Pearl. *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley, 1984.
- [16] Gabriele Di Stefano and Magnus L. Koci. A graph theoretical approach to the shunting problem. *Electronic Notes in Theoretical Computer Science*, 92(1), 2003.
- [17] Norio Tomii and Li J. Zhou. Depot shunting scheduling using combined genetic algorithm and PERT. In *Proceedings COMPRAIL 2000*, pages 437-446, 2000.
- [18] Norio Tomii, Li J. Zhou, and Naoto Fukumara. Shunting scheduling problem at railway stations. In *Lecture Notes in Artificial Intelligence*, 1611:790-797, 1999.
- [19] Thomas Winter and Uwe T. Zimmermann. Real-time dispatch of trams in storage yards. *Annals of Operations Research*, 96:287-315, 2000.
- [20] Thomas Winter. *Online and Real-time Dispatching Problems*. PhD thesis, Technical University of Braunschweig, 1999.
- [21] Cor van 't Woudt. *Zoeken naar rangeerroutes op een station* (in Dutch). Masters thesis, Erasmus University Rotterdam, 2001.
- [22] Peter J. Zwaneveld. *Railway Planning - Routing of Trains and Allocation of Passenger Lines*. PhD thesis, Erasmus University Rotterdam, 1997.

## Publications in the Report Series Research\* in Management

ERIM Research Program: “Business Processes, Logistics and Information Systems”

2003

*Project Selection Directed By Intellectual Capital Scorecards*

Hennie Daniels and Bram de Jonge

ERS-2003-001-LIS

<http://hdl.handle.net/1765/265>

*Combining expert knowledge and databases for risk management*

Hennie Daniels and Han van Dissel

ERS-2003-002-LIS

<http://hdl.handle.net/1765/266>

*Recursive Approximation of the High Dimensional max Function*

Ş. İl. Birbil, S.-C. Fang, J.B.G. Frenk and S. Zhang

ERS-2003-003-LIS

<http://hdl.handle.net/1765/267>

*Auctioning Bulk Mobile Messages*

S.Meij, L-F.Pau, E.van Heck

ERS-2003-006-LIS

<http://hdl.handle.net/1765/274>

*Induction of Ordinal Decision Trees: An MCDA Approach*

Jan C. Bioch, Viara Popova

ERS-2003-008-LIS

<http://hdl.handle.net/1765/271>

*A New Dantzig-Wolfe Reformulation And Branch-And-Price Algorithm For The Capacitated Lot Sizing Problem With Set Up Times*

Zeger Degraeve, Raf Jans

ERS-2003-010-LIS

<http://hdl.handle.net/1765/275>

*Reverse Logistics – a review of case studies*

Marisa P. de Brito, Rommert Dekker, Simme D.P. Flapper

ERS-2003-012-LIS

<http://hdl.handle.net/1765/277>

*Product Return Handling: decision-making and quantitative support*

Marisa P. de Brito, M. (René) B. M. de Koster

ERS-2003-013-LIS

<http://hdl.handle.net/1765/278>

---

\* A complete overview of the ERIM Report Series Research in Management:  
<http://www.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

*Managing Product Returns: The Role of Forecasting*  
Beril Toktay, Erwin A. van der Laan, Marisa P. de Brito  
ERS-2003-023-LIS  
<http://hdl.handle.net/1765/316>

*Improved Lower Bounds For The Capacitated Lot Sizing Problem With Set Up Times*  
Zeger Degraeve, Raf Jans  
ERS-2003-026-LIS  
<http://hdl.handle.net/1765/326>

*In Chains? Automotive Suppliers and Their Product Development Activities*  
Fredrik von Corswant, Finn Wynstra, Martin Wetzels  
ERS-2003-027-LIS  
<http://hdl.handle.net/1765/363>

*Mathematical models for planning support*  
Leo G. Kroon, Rob A. Zuidwijk  
ERS-2003-032-LIS  
<http://hdl.handle.net/1765/332>

*How and why communications industry suppliers get “squeezed out” now, and the next phase*  
L-F Pau  
ERS-2003-033-LIS  
<http://hdl.handle.net/1765/317>

*Financial Markets Analysis by Probabilistic Fuzzy Modelling*  
Jan van den Berg, Uzay Kaymak, Willem-Max van den Bergh  
ERS-2003-036-LIS  
<http://hdl.handle.net/1765/323>

*WLAN Hot Spot services for the automotive and oil industries :a business analysis or : “Refuel the car with petrol and information , both ways at the gas station “*  
L-F Pau, M.H.P.Oremus  
ERS-2003-039-LIS  
<http://hdl.handle.net/1765/318>

*A Lotting Method for Electronic Reverse Auctions*  
U. Kaymak, J.P. Verkade and H.A.B. te Braake  
ERS-2003-042-LIS  
<http://hdl.handle.net/1765/337>

*Supply Chain Optimisation in Animal Husbandry*  
J.M. Bloemhof, C.M. Smeets, J.A.E.E. van Nunen  
ERS-2003-043-LIS  
<http://hdl.handle.net/1765/353>

*A Framework for Reverse Logistics*  
Marisa P. de Brito and Rommert Dekker  
ERS-2003-045-LIS  
<http://hdl.handle.net/1765/354>

*An assessment system for rating scientific journals in the field of ergonomics and human factors*  
Jan Dul and Waldemar Karwowski  
ERS-2003-048-LIS  
<http://hdl.handle.net/1765/432>

*Circulation of Railway Rolling Stock: A Branch-and-Price Approach*

Marc Peeters and Leo Kroon

ERS-2003-055-LIS

<http://hdl.handle.net/1765/902>

*Emerging Multiple Issue e-Auctions*

Jeffrey E. Teich, Hannele Wallenius, Jyrki Wallenius and Otto R. Koppius

ERS-2003-058-LIS

<http://hdl.handle.net/1765/922>

*Inventory Management with product returns: the value of information*

Marisa P. de Brito and E. A. van der Laan

ERS-2003-060-LIS

<http://hdl.handle.net/1765/925>

*Promising Areas for Future Research on Reverse Logistics: an exploratory study*

Marisa P. de Brito

ERS-2003-061-LIS

<http://hdl.handle.net/1765/926>

*A Polynomial Time Algorithm for a Deterministic Joint Pricing and Inventory Model*

Wilco van den Heuvel and Albert P.M. Wagelmans

ERS-2003-065-LIS

<http://hdl.handle.net/1765/929>

*A geometric algorithm to solve the  $n/g/n/nd$  capacitated lot-sizing problem in  $o(t^2)$  time*

Wilco van den Heuvel and Albert P.M. Wagelmans

ERS-2003-066-LIS

<http://hdl.handle.net/1765/930>

*Arrival Processes for Vessels in a Port Simulation*

Eelco van Asperen, Rommert Dekker, Mark Polman, Henk de Swaan Arons & Ludo Waltman

ERS-2003-067-LIS

<http://hdl.handle.net/1765/973>

*The distribution-free newsboy problem with resalable returns*

Julien Mostard, Rene de Koster and Ruud Teunter

ERS-2003-068-LIS

<http://hdl.handle.net/1765/975>

*A note on a multi-period profit maximizing model for retail supply chain management*

Wilco van den Heuvel and Albert P.M. Wagelmans

ERS-2003-072-LIS

<http://hdl.handle.net/1765/991>

*The Impact of Complexity, Rate of Change and Information Availability on the Production Planning and Control Structure: Evidence from Medium-Sized Dutch Discrete Manufacturing Firms*

Marcel van Assen and Steef van de Velde

ERS-2003-083-LIS

<http://hdl.handle.net/1765/1033>

*The impact of innovation and organizational factors on APS adoption: Evidence from the Dutch discrete parts industry*

Bart van Hezewijk, Marcel van Assen and Steef van de Velde

ERS-2003-084-LIS

<http://hdl.handle.net/1765/1067>



*Equilibrium Constrained Optimization Problems*

S. Birbil, G. Bouza, J.B.G. Frenk and G. Still

ERS-2003-085-LIS

<http://hdl.handle.net/1765/1068>

*Network-Based Business Process Management: Embedding Business Logic In Communications Networks*

L-F Pau and P.H.M. Vervest

ERS-2003-086-LIS

<http://hdl.handle.net/1765/1070>

*Airline Revenue Management with Shifting Capacity*

Kevin Pak, Rommert Dekker and Gerard Kindervater

ERS-2003-091-LIS

<http://hdl.handle.net/1765/1074>

*Combining Column Generation and Lagrangian Relaxation*

Dennis Huisman, Raf Jans, Marc Peeters and Albert P.M. Wagelmans

ERS-2003-092-LIS

<http://hdl.handle.net/1765/1098>