

Hump Yard Track Allocation with Temporary Car Storage*

Markus Bohlin¹ Holger Flier² Jens Maue² Matúš Mihalák²

¹ Swedish Institute of Computer Science, Kista, Sweden
markus.bohlin@sics.se

² Institute of Theoretical Computer Science, ETH Zürich, Switzerland
firstname.lastname@inf.ethz.ch

Abstract

In rail freight operation, freight cars need to be separated and reformed into new trains at hump yards. The classification procedure is complex and hump yards constitute bottlenecks in the rail freight network, often causing outbound trains to be delayed. One of the problems is that planning for the allocation of tracks at hump yards is difficult, given that the planner has limited resources (tracks, shunting engines, etc.) and needs to foresee the future capacity requirements when planning for the current inbound trains. In this paper, we consider the problem of allocating classification tracks in a rail freight hump yard for arriving and departing trains with predetermined arrival and departure times. The core problem can be formulated as a special list coloring problem. We focus on an extension where individual cars can temporarily be stored on a special subset of the tracks. An extension where individual cars can temporarily be stored on a special subset of the tracks is also considered. We model the problem using mixed integer programming, and also propose several heuristics that can quickly give feasible track allocations. As a case study, we consider a real-world problem instance from the Hallsberg Rangerbangård hump yard in Sweden. Planning over horizons over two to four days, we obtain feasible solutions from both the exact and heuristic approaches that allow all outgoing trains to leave on time.

1 Introduction

A rail-freight transportation network is used to transport goods between two points in the network. Described in a very simplified way, such a network consists of terminals and tracks connecting these stations. Besides freight system trains that consist only of cars from one origin to one destination (e.g., trains transporting coal or ore), most freight trains consist of cars with various origins and destinations. These freight trains arrive at intermediate stops of the network, where the cars are disassembled and new trains are formed, together with other cars of other freight trains, that continue to deliver the goods closer to the final destination. Delivering goods in an efficient way requires careful planning of which cars should form which trains, which route should the planned trains take, and at what time the planned train should depart and arrive at the respective intermediate stops. Naturally, this planning has to be made with respect to time, space, and logistics capacities of intermediate stops. In this paper we assume that such planning of trains has been made and look at

*This work was supported by Trafikverket under FUD grant F 09-11546/AL50 and by the Swiss National Science Foundation (SNF) under grant 200021-125033/1.

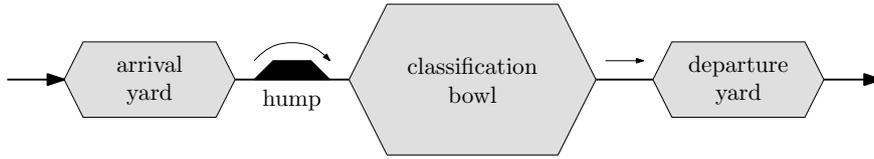


Figure 1: Layout of a typical shunting yard with a receiving yard, a classification bowl, and a departure yard.

computational aspects of the process of forming departing trains from the cars of arriving trains. Efficient train formation is important as this process is usually the bottleneck of the freight transportation and naturally limits the throughput in the network.

An intermediate stop in the network is called a *shunting yard*, which is a central distribution point where the cars of arriving freight trains are formed into new departing freight trains in a process called *shunting*. A typical layout of a shunting yard is illustrated in Figure 1. It consists of an *arrival yard*, a *hump*, a *classification bowl*, and a *departure yard*. The *arrival yard* is a composition of tracks used for storing cars of arriving trains, where all cars of an arriving train are always stored on one single track. The cars of an arriving train are inspected and decoupled in the arrival yard. After the inspection the cars of one track in the arrival yard are ready to be pushed over the *hump* to the classification bowl. The hump track is a long track connecting the arrival yard with the classification bowl. At its latter end, there is the hump, i.e. a small hill, so that uncoupled cars pushed from the hump track over the hump by a shunting locomotive can roll in to the classification bowl alone by the means of gravity. The *classification bowl* is a collection of parallel tracks called *classification tracks* connected via a set of switches to the hump such that a car pushed over the hump can reach any classification track by an appropriate setting of switches. (The set of switches can be seen as a binary tree rooted at the hump track.) There are two main operations that can be performed inside the classification bowl: *roll-in* operations and *pull-out* operations. A *roll-in* operation is the process of pushing all cars at the hump track over the hump to the classification tracks, and switching the switches during this process such that every car is routed to the desired classification track. In a *pull-out* operation all cars on a selected classification track are coupled and pulled by the shunting engine to the hump track, where the cars are decoupled again and ready for a subsequent roll-in operation. In a shunting yard, a pull-out operation is always followed by a roll-in operation. In particular, it is not possible to perform two subsequent pull-out operations without a roll-in operation in between. Pull-out and roll-in operations are used to form desired sequences of cars on classification tracks. Once a desired sequence of cars is formed on a classification track, the cars on the track can be coupled and an engine brings them to a track on the *departure yard*, which is a composition of tracks used to store departing freight trains before their timetabled departure from the yard.

In this paper we study operational questions of a shunting yard with a hump. In general, we are given the arrival and departure time of every arriving and departing train, respectively, as well as which cars form these trains. A general operational plan needs to decide the movements of every car in the shunting yard by the means of roll-ins and pull-outs in order to achieve the desired formation of every departing train before its departure time. This also includes deciding the time points when cars enter the classification bowl and the time point they leave. There may be various constraints on the formation of departing freight

trains. In particular, there are cases where the order of cars within a train is important. In this paper we only consider the case in which no particular order of cars within a departing train is required. This is usually the case for freight trains which are not delivering goods to the final destination but to an intermediate stop.

Currently, the operational plans are made by hand at the shunting yards, with only little help from computers. When computer-aided planning do exist, it is often limited to visualization and monitoring. As the number of cars passing through shunting yards approaches the yard capacity, preparing operational plans manually becomes very challenging and often results in unfeasible operational plans that cause delayed departures. Obviously there is a need for a computer-driven operation of shunting yards.

In this paper we restrict ourselves to a certain class of operational plans which is a generalization of current customs that are used in several shunting yards in Europe. An operational plan that we consider reserves a fixed set of classification tracks for building departing trains, where there is at most one train on each track at the same time. A few classification tracks (usually one or two) called *mixing tracks* are used to increase capacity by temporarily storing a mix of early cars. At some fixed time points per day, the mixing tracks are pulled-out to again distribute its cars onto the classification tracks (or the mixed tracks). We aim to develop tools for computing such operational plans. In this paper we present preliminary results concerning our aim, including our modeling approach, relation to algorithmic theory, and initial computational results which suggest that such operational plans can be found in real-time.

Outline The paper is structured as follows. We first define the problem formally in Section 1.1, review the related work and best practices in Section 1.2, and give some complexity results on variants of our problem in Section 2. In the following, we consider the problem of forming departing trains inside the classification bowl in Section 3, model the problem as a mixed-integer program in Section 3.1, and present heuristics for this subproblem in Sections 3.2–3.4. The problem of operating the arrival yard is treated in Section 4. We give experimental results on a real world case study for both the implemented heuristics and the mixed-integer programs in Section 5. A nomenclature is given at the end of the paper.

1.1 Problem Definition

We are given a hump yard consisting of k^{arr} receiving tracks, a single hump followed by k^{class} classification tracks and k^{dep} departure tracks. There are n^{in} arriving trains, which are to be formed into n^{out} departing trains. Train formation is carried out on the classification tracks, which have non-uniform length. Each arriving train i has a timetabled arrival time T_i^{arr} and a set of cars C_i ; each departing train q has a departure time T_q^{dep} and a set of cars C_q . Each unique car c has a physical length w_c . The goal is to plan freight operations at the yard for a fixed time period. Arrival tracks and departure tracks can accommodate any train, while classification tracks θ can only accommodate trains shorter than the track length ℓ_θ , where the length of a train q is computed as $\sum_{c \in C_q} w_c$.

The flow of cars at the yard proceeds as described in the following. An incoming train i is pushed over the hump at time $T_i^{\text{roll}} \geq T_i^{\text{arr}} + S^{\text{arr}} + S^{\text{hump}}$, where S^{arr} is the inspection duration and S^{hump} is the time to prepare and decouple the train prior to the roll-in. Every car c of train i has a time label $T_c^{\text{roll}} = T_i^{\text{roll}}$. The time point T_q^{start} determines when the assigned classification track is allocated by this train. An outbound train q is moved from

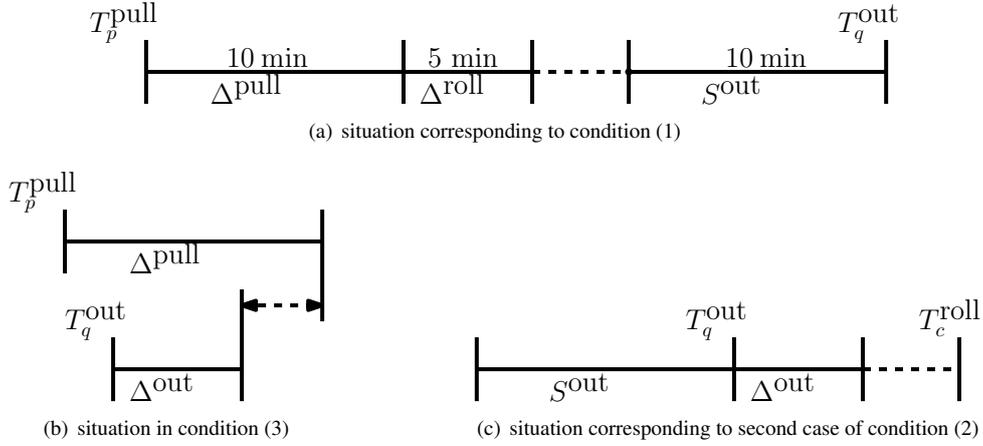


Figure 2: The points of time and time intervals included in a hump step, in rolling in a car to a classification track, and the departure of outbound trains from the classification tracks.

its classification track to the departure yard at time $T_q^{\text{out}} \leq T_q^{\text{dep}} - \Delta^{\text{out}}$, where the duration of preparing an outbound train (including brake testing) is given by S^{out} , and the amount of time which the outbound train needs to leave its track by Δ^{out} . The track is then deallocated at time $T_q^{\text{out}} + \Delta^{\text{out}}$ and can be used to collect the next train. Let s denote the number of cars flowing from an inbound train i to outbound train q . Then, for any outbound train q and all inbound trains i where $s \geq 1$, the inequality $T_i^{\text{roll}} + \Delta^{\text{roll}} + S^{\text{out}} \leq T_q^{\text{out}}$ must be satisfied.

Temporary Car Storage Furthermore, it is possible to temporarily store cars on one of the *mixed tracks*, which are only used for this purpose. Storage of cars on the mixed tracks is simply done by initial roll-ins of cars to the mixing tracks. All cars on the mixed tracks are then pulled back over the hump at time points T_p^{pull} where $p \in \mathcal{P}$ and $\mathcal{P} = \{1, \dots, m\}$, after which all cars are rolled in again. Δ^{pull} denotes the time required to drive to the mixed track, pull back the cars over the hump on the hump track, and prepare the cars for being pushed over the hump. This does not include the time for the subsequent roll-in. A car may then be sent either to the classification track of the departing train, or to a mixed track again for further temporary storage. The time it requires for rolling is denoted by Δ^{roll} . The other classification tracks are never used for temporary storage of cars.

For a certain hump step p at some time T_p^{pull} , an outbound train q may receive cars from this hump step depending on the departure of q as depicted in Figure 2(a). The condition is formalized as follows:

$$T_p^{\text{pull}} + \Delta^{\text{pull}} + \Delta^{\text{roll}} + S^{\text{out}} \leq T_q^{\text{out}} \quad (1)$$

When assigning trains q and r to the same track and both trains leave between the same pair of hump steps, the trains must not overlap. (Otherwise, there must be some hump step between the two departures, which is dealt with further below.) More precisely, the arrival time T_c^{roll} of the first car c of train r must be greater than the time of deallocation by q :

$$T_q^{\text{out}} + \Delta^{\text{out}} \leq T_c^{\text{roll}} \quad (2)$$

Additionally, for a departing train r some cars of which must be sent to the mixing track, a car c of this train needs no mixing if it arrives later than the previous train q deallocates this track. This corresponds to Figure 2(c) and is also formalized by condition (2). Note that in this situation, car c denotes the first of q to arrive.

For a departing train r some cars of which are humped, we want to estimate the earliest hump step T_p^{pull} in which humped cars can finally be sent to the track of this train r . We have to compare the time when the track is deallocated by the previous train q departing from it with the time the first car from the hump is about to allocate the track at time $T_p^{\text{pull}} + \Delta^{\text{pull}}$ (see also Figure 2(b)):

$$T_q^{\text{out}} + \Delta^{\text{out}} \leq T_p^{\text{pull}} + \Delta^{\text{pull}} \quad (3)$$

Problem Solution A problem solution comprises three parts: a roll-in time T_i^{roll} for every inbound train i , a roll-out time T_q^{out} for every departing train q , and an assignment of departing trains to classification tracks. Our objective is to find a feasible solution, i.e. a solution that satisfies all the feasibility conditions just described.

The roll-in time of inbound trains from the arrival yard over the hump to the classification bowl must make sure there is enough time between two trains on the the same arrival track. Also, there must be no collisions in time for two trains on the hump, and the roll-in times of all inbound trains must not interfere with any predetermined pull-out step. All these conditions are formalized in Section 4. For the roll-out time of an outbound train from its classification to some departure track, there must be a free track in the departure yard at this point of time. Finally, between all roll-ins of cars for some outbound train and that train's roll-out time, there must be a sufficient gap for preparation works.

Moreover, every pair q and r of trains consecutively departing from the same classification track must satisfy one of the following two conditions. Either, q deallocates the track before the first inbound train with cars for r is rolled in, i.e. condition (2) is satisfied. Or, q and r overlap in time but they are separated by some pull step, i.e. there exists a pull p which satisfies both the following conditions:

$$T_q^{\text{out}} - S^{\text{out}} < T_p^{\text{pull}} + \Delta^{\text{pull}} \quad (4)$$

$$T_p^{\text{pull}} + \Delta^{\text{pull}} + \Delta^{\text{roll}} \leq T_r^{\text{out}} - S^{\text{out}} \quad (5)$$

Finally, we have to respect the mixing capacity of the mixing tracks at any time, so the total length of all cars on the mixing tracks must not exceed ℓ^{mix} for any pull $t \in \mathcal{P}$. For two departing trains q and r leaving consecutively from the same classification track, a car $c \in C_r$ of r will be humped in pull step p if the second case of condition (3) is not satisfied (i.e. $T_c^{\text{roll}} < T_q^{\text{out}} + \Delta^{\text{out}}$) and either condition (3) does not hold (i.e. $T_q^{\text{out}} + \Delta^{\text{out}} > T_p^{\text{pull}} + \Delta^{\text{pull}}$) or it does not hold for $p-1$ (i.e. $T_q^{\text{out}} + \Delta^{\text{out}} \leq T_p^{\text{pull}} + \Delta^{\text{pull}}$ but $T_q^{\text{out}} + \Delta^{\text{out}} \leq T_{p-1}^{\text{pull}} + \Delta^{\text{pull}}$).

1.2 Related Work

The particular problem that we consider in this paper has, as far as we know, not been studied before. There are various papers that are related to the problem of shunting, but the solutions techniques are not applicable to our problem.

Main research efforts related to operation of shunting yards have been put in *sorting schemes* for sorting cars inside a classification bowl. Given a sequence of n cars labeled

from 1 to n , the general goal of a sorting scheme is to form (by roll-in and pull-out operations) a sorted sequence (on a classification track). Early literature considers sorting schemes that transform the input sequence into a sorted sequence regardless of what the input looks like [10], whereas newer literature looked at how to use the “pre-sortedness” of the input in order to minimize the number of pull-out operations [5, 9], and also studied variants of the problem [4]. A recent survey by Gatto et al. [7] gives an overview of this topic.

A related problem to the shunting of cars in shunting yards is the parking of trams to tracks in depots in the evening such that the trams can leave the depot in the morning without any shunting operation [1, 11, 6].

Another related problem is the train scheduling at yards – the problem of assigning trains and train times for a set of rail lines and station stops. This problem was considered together with some operational planning at shunting yards, under different considerations than ours, by He et al. [8] for the shunting yards in China.

In our problem we have to decide for each departing train the classification track and time at which the departing train is going to be build on this track. A related problem for passenger trains has been considered [3], in which more than one arriving train can be assigned to a track of a train station. The problem asks for an assignment of tracks to arriving trains such that the trains do not block each other when departing the train station.

Best Practice Today, the planning of hump yard operation is to a large extent done manually. In the Hallsberg yard in Sweden, detailed planning is done by the hump yard staff one day at a time, usually during the morning when fewer trains arrive than in the afternoon. The allocation of tracks at the arrival yard and departure yard is performed manually, separately and in advance by traffic planning personnel, who are not directly involved with hump yard operation and detailed planning. However, there is in practice frequent communication between the different groups, since the allocation of arrival and departure yards and the yard operation planning is interdependent and cannot be done in full isolation.

The normal state of practice is to use the same roll-in order as the arrival time order (T_i^{arr}) and the same roll-out order (onto the departure yard) as the departure time order (T_q^{dep}). However, in Section 4 we will discuss an extension for our approach with the goal of determining a suitable roll-in order. Just before a train is pushed over the hump, it is decoupled into car groups going to separate tracks. When an incoming train is moved from the arrival yard to the classification bowl, each of its cars may either be sent to the train formation track of its corresponding outgoing train, or it may be sent to the mixing tracks for temporary storage. Pull-backs of the mixing tracks are done at a few pre-specified time points each day, usually when traffic and operation is slow. The yard planner must therefore be careful not to put cars on the mixing tracks which are about to leave the yard shortly. If the cars furthest from the hump on a mixing track do not need to leave the yard immediately, a pull-back can be performed on only the first part of the mixed track.

Formation of a train is usually done on a single classification track only. However, it is possible to form a single train on two (or more) classification tracks at the same time, and then perform the final coupling of the train at the departure yard. Similarly, it is possible to form two short trains on a single classification track if the cars arrive in the correct order. However, the procedures above are exceptions and we therefore do not take them into account in this paper.

2 Theoretical Complexity

In this section we briefly discuss the complexity of the mixing-problem, which asks us to determine for each outbound train a track and a time interval, during which the track is reserved exclusively for the formation of that train. As a result of the roll-in problem we are given for every outbound train q a time interval $I = (arr, dep)$ in which cars of the outbound train arrive to the classification bowl, i.e., $arr = \min_{c \in C_q} T_c^{\text{roll}}$ (where T_c^{roll} is the roll-in time of car c and C_q is the set of cars of train q), and $dep = T_q^{\text{out}}$ is the time when the train q leaves from its classification track to the departure track. If we do not send any car of the outbound train to a mixing track then we need to reserve a classification track for the whole time period I . Sending cars to the mixed tracks allows us to shorten the time interval for which the classification track is reserved. Equivalently, we can see the shortening of an interval I by cutting-off a prefix of I as sending the cars that arrive in the cut-off period of time to mixed tracks. As at time dep the outbound train has to leave the classification track, there is a time period – a suffix of I – in which all arriving cars of the outbound train cannot be sent to mixed tracks. We call this suffix of I the *uncuttable part* of I .

If we assume unlimited capacity of mixed tracks then naturally we can cut-off every interval I of an outbound train such that only the uncuttable part of I remains. The problem is then to assign each outbound train to a classification track of sufficient capacity such that no two uncuttable intervals of trains assigned to the same track overlap. This problem can be modeled as a list-coloring problem: given a graph $G = (V, E)$ and a function $\mu : V \rightarrow \mathbb{N}$, find a proper node coloring $c : V \rightarrow \mathbb{N}$ of G that minimizes $\max_{v \in V} c(v)$ such that $c(v) \geq \mu(v)$ for every $v \in V$. In our case, the graph G is an interval graph, defined as follows: nodes correspond to outbound trains, edges correspond to pairs of trains with overlapping time intervals, and colors correspond to tracks. Assume w.l.o.g. that the classification tracks $1, \dots, k^{\text{class}}$ are ordered in a decreasing order of their length. For each outbound train q , let $\mu(q)$ indicate the smallest track on which q can fit (i.e. we assume that every train fits on the first track with the largest length, and the last track the train fits is the track $\mu(q)$). Then, there is a valid track allocation if and only if there is a μ -coloring for G . It has been shown that the μ -coloring problem for interval graphs is NP-complete, see [2]. As a consequence we immediately obtain the following theorem:

Theorem 1. *Finding a feasible track allocation for the mixing-problem is NP-complete even for instances where the capacity of the mixed tracks ℓ^{mix} is zero, or where ℓ^{mix} is unlimited and all intervals may have arbitrary uncutted parts.*

Proof. Observe first that if there is no capacity on mixed tracks then no car can be sent to a mixed track and thus no interval can be cut-off. It is now easy to see by the above discussion how to transform any instance of the μ -coloring problem for interval graphs to a corresponding instance of the mixing-problem. \square

Despite it's NP-complete core, the practical complexity of the mixing-problem strongly depends on the distribution of the length of both the classification tracks and the outbound trains. If, for example, each train fits on all tracks, and the capacity of mixed tracks is zero, then our problem reduces to the problem of coloring an interval graph, which is well-known to be polynomially solvable by a simple greedy algorithm.

If we assume sufficient track length of both mixed and classification tracks, then the problem of minimizing the total number of cars sent to the mixed tracks that allows a feasible track allocation can be solved in polynomial time.

Theorem 2. *In case of uniform and sufficient track lengths, the problem of finding a feasible track allocation that minimizes the sum of all cars sent to the mixed tracks over all time periods is solvable in polynomial time.*

Proof. Assume there are k tracks and n trains, each with a time interval as described above. Observe that if the trains are assigned to tracks then computing the minimum number of cars that need to be sent to the mixing tracks in order to make this assignment feasible is a trivial task. To see this, consider a classification track θ and train q with interval (arr_q, dep_q) such that the train is last to leave the track, i.e., for any other train q' with time interval $(arr_{q'}, dep_{q'})$ assigned to track θ we have $dep_{q'} < dep_q$. The minimum number of cars of train q that we need to send to the mixing tracks are the cars that arrive in time period $(arr_q, dep_{q'})$ where $dep_{q'}$ is the departure time of train q' that departs from the classification track second to last, i.e., just before q . We can proceed similarly with the cars of train q' by sending to the mixing tracks all cars of train q' that arrive in time interval $(arr_{q'}, dep_{q''})$, where q'' is the train that leaves the classification track just before q' . We can proceed recursively to determine the minimum number of cars that need to be sent to classification tracks.

We can therefore see our problem as finding for every train q its direct predecessor q' on a classification track. The actual assignment of a track to trains is done by introducing a *phantom* train q_i for every track θ . Thus, if a train q is assigned a phantom train q_θ as the direct predecessor of q , then we interpret this as assigning train q to the classification track θ .

In this modified setting where every train is asked to have a predecessor (a real train or a phantom train), our problem can be solved by computing a minimum-weight matching in the following complete bipartite graph: the (real) trains form one part of the bipartition, and the real trains together with the phantom trains form the other part of the bipartition; the weight of the edge connecting train q from the first part with (phantom) train q' is the minimum number of cars that need to be sent from q to the mixed tracks in order to allow train q' to be an immediate predecessor of train q on a classification track. \square

3 Track Allocation and Temporary Car Storage

In this section, we discuss solutions to the track allocation problem with temporary car storage, from now on referred to as the *mixing problem*. Let $\mathcal{O} = \{q_1, \dots, q_{n^{\text{out}}}\}$ be a set of outbound trains, where each train q consists of a set of cars \mathcal{C}_q , each car c having a length of w_c meters, $\Theta = \{\theta_1, \dots, \theta_{k^{\text{class}}}\}$ a set of classification tracks, each track θ_i having a length of ℓ_i meters, and ℓ^{mix} a maximum mixing capacity (total length of the cars stored on the mixing tracks). All inbound trains i have a predetermined roll-in time T_i^{roll} , and all outbound trains q have a predetermined roll-out time T_q^{out} . All cars on the mixing tracks are periodically pulled back (in a pull-out operation) over the hump at time T_p^{pull} , $1 \leq p \leq m$, after which the cars are rolled in again. If the train formation of the corresponding outgoing train q of car c has not started yet, the car is sent to the mixing tracks again. We assume that at any pull-out time T_p^{pull} , all mixed tracks are pulled out consecutively without interruption. We can then treat all mixed tracks as a single mixed track with length ℓ^{mix} equal to the sum of the individual track lengths.

The goal of the mixing problem is to find 1) an assignment of long-enough classification tracks to the outbound trains, and 2) a car mixing schedule (i.e., an assignment of a subset

of the cars to the mixing tracks over time), which is conflict-free in the sense that two trains never occupy a track at the same time, and does not exceed the available mixing capacity ℓ^{mix} .

This section is structured as follows. In Section 3.1, a mixed integer programming formulation is given for the mixing problem. Further, in Sections 3.2 and 3.3, two heuristics are given for the problem. Finally, in Section 3.4, an improvement heuristic is given, that takes any feasible solution and finds an improved solution, i.e., a certain local minimum.

3.1 A MIP Model for the Mixing Problem

In this section, we formulate a mixed integer programming model for the problem which captures the requirements given above. To begin with, we focus on feasibility. Since it is possible that the problem is infeasible (which can be hard to prove) we first settle for minimizing a penalty cost v , which is proportional to the violation of the available mixed track capacity. If $v = 0$, then the problem is feasible.

$$\text{minimize } v \tag{6}$$

$$\text{s.t. } T_c^{\text{roll}} x_{c0} + \sum_{p \in \mathcal{P}_c} (T_p^{\text{pull}} + \Delta^{\text{pull}}) x_{cp} = T_c^{\text{start}} \quad c \in \mathcal{C} \tag{7}$$

$$x_{c0} + \sum_{p \in \mathcal{P}_c} x_{cp} = 1 \quad c \in \mathcal{C} \tag{8}$$

$$\sum_{p \in \mathcal{P}, T_p^{\text{pull}} \geq T_o^{\text{pull}}} x_{cp} \leq z_{co} \quad c \in \mathcal{C}, o \in \mathcal{P}_c \tag{9}$$

$$T_q^{\text{start}} \leq T_c^{\text{start}} \quad q \in \mathcal{O}, c \in \mathcal{C}_q \tag{10}$$

$$T_c^{\text{start}} + \Delta^{\text{roll}} + \Delta^{\text{p+r}} \sum_{p \in \mathcal{P}} x_{cp} + S_q^{\text{out}} \leq T_q^{\text{out}} \quad q \in \mathcal{O}, c \in \mathcal{C}_q \tag{11}$$

$$\sum_{\theta \in \Theta} y_{q\theta} = 1 \quad q \in \mathcal{O} \tag{12}$$

$$T_q^{\text{out}} + \Delta^{\text{out}} - M(2 - y_{q\theta} - y_{r\theta}) \leq T_r^{\text{start}} \quad q, r \in \mathcal{O}, \theta \in \Theta : q < r \tag{13}$$

$$T_\theta^{\text{free}} y_{q\theta} \leq T_q^{\text{start}} \quad q \in \mathcal{O}, \theta \in \Theta \tag{14}$$

$$\sum_{c \in \mathcal{C}_q} w_c \leq \sum_{\theta \in \Theta} \ell_\theta y_{q\theta} \quad q \in \mathcal{O} \tag{15}$$

$$\sum_{c \in \mathcal{C}: T_c^{\text{roll}} \leq T_p^{\text{pull}}} w_c z_{cp} \leq \ell^{\text{mix}} + v \quad p \in \mathcal{P} \tag{16}$$

$$T_c^{\text{start}} \geq 0 \quad c \in \mathcal{C}$$

$$T_q^{\text{start}} \geq 0 \quad q \in \mathcal{O}$$

$$x_{cp} \text{ binary} \quad c \in \mathcal{C}, p \in 0, \dots, m$$

$$y_{q\theta} \text{ binary} \quad q \in \mathcal{O}, \theta \in \Theta$$

$$z_{cp} \geq 0, z_{cp} \leq 1 \quad c \in \mathcal{C}, p \in \mathcal{P}$$

$$v \geq 0$$

The objective (6) is to minimize the violation v of the mixed-track capacity. Constraints

(7) and (8) define the earliest arrival times (T_c^{start}) for car c on its classification track, as controlled by the x_{cp} binary variable: if $x_{cp} = 1$ then the car c is sent from the mixing tracks to its classification track in pull-out p . Note that if $x_{c0} = 1$, then car c is not mixed at all, but instead goes directly from the arrival yard to its classification track. The set \mathcal{P}_c denotes the feasible pull-outs for car c , that is, the pull-outs which occur after car c has been rolled in from the arrival yard, or formally:

$$\mathcal{P}_c = \{p \mid p \in \mathcal{P} \wedge T_c^{\text{roll}} + \Delta^{\text{roll}} \leq T_p^{\text{pull}}\}. \quad (17)$$

To penalize or enforce mixed track capacity constraints for a pull-out p , we need to know whether car c is on a mixed track at pull-out p or not. To this end, we add an extra binary variable z_{cp} . The associated constraint (9) then ensures that for any car c and pull-outs o and p where $T_c^{\text{roll}} + \Delta^{\text{roll}} \leq T_o^{\text{pull}} \leq T_p^{\text{pull}}$, if $x_{cp} = 1$ then $z_{co} = 1$ as well. In other words, $z_{co} = 1$ whenever car c is on a mixing track. Finally, $y_{q\theta}$ is a binary variable indicating whether departing train q is on the classification track θ (when $y_{q\theta} = 1$) or not.

Constraint (10) ensures that the track allocation for all outbound train begins no later than when the first car arrives. Constraint (11) makes sure that there is enough time on the classification track to prepare the train for departure, as measured after the last car has arrived. This includes enough time for pull-outs if the cas has been mixed. Constraint (12) ensures that an outbound train q is allocated to exactly one track; constraint (13) then guarantees that allocations on the same track do not overlap. Since the roll-out and departure order are equal, we also know which allocation will come first. Constraint (14) ensures that allocation does not start until the track is free from historic allocations (at time point T_t^{free}), i.e., due to trains that were planned before the current planning horizon. Constraint (15) ensures maximum-length compliance on classification tracks, while Constraint (16) defines the non-negative penalty v in relation to the total length of the mixed tracks and their utilization in between pull-outs.

3.2 Heuristic A: A Heuristic Based on Cliques

The problem of finding a feasible assignment of reservation intervals of trains to tracks when the tracks have different length is NP-complete, as noted in Section 2. We are, however, allowed to cut off a part of the beginning of the intervals. If we had mixed tracks of unlimited capacity, we could cut off as much as technically possible, resulting in an instance with substantially smaller intervals. The following heuristic is based on the observation that these smallest possible intervals can be μ -colored greedily in practice, as described below. If this is the case, then it is possible to find a track assignment that respects the lengths of the classification tracks by the heuristic below. The capacity of the mixed tracks may, however, be exceeded.

The heuristic is based on a greedy μ -coloring as described in Algorithm 1. There, the intervals are sorted non-descendingly according to their starting time point and are assigned the smallest fitting track that is free in that order.

The heuristic starts by applying Algorithm 1 on the original intervals, i.e., for each train i there is an interval $I(i) = (T_i^{\text{roll}}, T_i^{\text{out}})$ with $\mu_{I(i)}$ being the index of the smallest track on which train i fits. If the largest color used in the resulting coloring is less or equal to the index of the largest classification track, we are done. Otherwise, there must exist a clique K in which at least one interval was assigned a color $c > k^{\text{class}}$. The heuristic then calculates the intersection of all intervals $I \in K$. Let t denote the end of this intersection. Let there be

<p>Algorithm 1: Greedy_coloring(S, μ)</p> <p>input : A set of intervals S and a minimum color $\mu(I)$ for each interval $I \in S$</p> <p>output: A coloring c of S, such that $c(I) \geq \mu_I$ for all $I \in S$</p> <p>while S not empty do</p> <p style="padding-left: 20px;">$I \leftarrow$ an $I' \in S$ with smallest start time, tie break on largest end time;</p> <p style="padding-left: 20px;">$S \leftarrow S - I$</p> <p style="padding-left: 20px;">$c \leftarrow \mu_I$</p> <p style="padding-left: 20px;">while $I \cap \{J \in S \mid c(J) = c\} \neq \emptyset$ do</p> <p style="padding-left: 40px;">$c \leftarrow c + 1$</p> <p style="padding-left: 20px;">end</p> <p style="padding-left: 20px;">$c(I) \leftarrow c$</p> <p>end</p>

q intervals in K using a color $c > k^{\text{class}}$. The heuristic tries to eliminate the clique by cutting of q of its intervals at t . If possible, it chooses the q intervals that increases the maximum usage of the mixing tracks over all periods the least. The resulting set of intervals is then colored again greedily, repeating the steps above until a feasible coloring is found. If at some iteration no interval gets cut off, the heuristic returns a greedy coloring of the smallest possible intervals, which, as mentioned above, is feasible for practical instances.

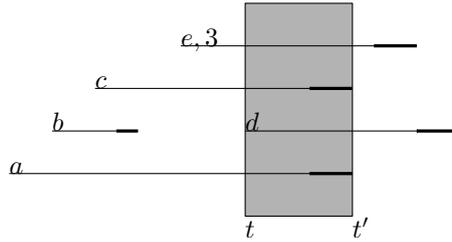


Figure 3: Cutting of parts of intervals. The intersection (t, t') of the 3-clique is shown in grey. Tails of the intervals that cannot be cut off are depicted in bold. Assuming that there are only three tracks available, that $\mu(e) = 3$, and that $\mu(I) = 0$ for all $I \neq e$. Interval e will be cut off at t' .

3.3 Heuristic B: A Greedy Minimum Mixing Heuristic

Another heuristic greedily assigns one outbound train after the other to the classification tracks. We iterate over the outbound trains in the order they depart from the classification bowl. In every step, all those tracks are considered the capacity of which suffices to hold the train q to be assigned.

As a special case, there might be a track among them for which no cars of q have to be mixed since the first car c for q arrives on the hump later than this track is deallocated by the train p leaving from this track before q . This corresponds to condition (2) and Figure 2(c). This also applies if no train has been assigned to the track before q , which occurs frequently in the first iteration steps of the heuristic.

Otherwise, what is the general case, some cars of the train must be mixed. A candidate track occupied with some train p can of course only be chosen if there exists a hump step that satisfies both condition (1) and condition (3). We define the *local cutoff* of q for this track as the sum over the lengths of all cars of q arriving earlier than the deallocation time $T_p^{\text{dep}} + \Delta^{\text{out}}$ of p , i.e. all cars of q not satisfying condition (2). We simply choose the track which minimizes the local cutoff of q . Note that, for the special case above, the local cutoff just equals zero, so that case actually needs no separate treatment.

In order to break ties, we follow a best-fit strategy: when assigning some outbound train q , if the minimum local cutoff value is obtained for two or more candidate tracks, we choose the one with the smallest capacity. This must be applied particularly often in the beginning of the assignment, when most classification tracks are still completely empty.

3.4 Heuristic I: An improvement Heuristic

Finally, we present a heuristic that takes a feasible assignment of trains to tracks, e.g. obtained by the heuristics above, and returns a solution which is optimal in a certain neighborhood of the original solution, as described below. The heuristic seeks to minimize the sum of car-meters sent to the mixing tracks over all periods. It is based on two observations.

First, when looking at the data, one observes that both tracks and trains can be partitioned into subsets, called *buckets*, such that each train in a bucket fits on all tracks in the same bucket. Second, given a feasible solution, one can minimize the sum of the mixing track usage for each bucket of tracks independently. Because within a bucket, all trains fit on all tracks, it suffices to solve an assignment problem for each bucket, as detailed in Theorem 2, in order to find an optimal reassignment of trains to tracks within that bucket.

4 Arrival and Departure Yard Planning

In Section 3 we assumed that the time of roll-in T_i^{roll} to the classification tracks and the time of roll-out T_q^{out} to the departure yard had been pre-determined for each inbound train i and each outbound train q , respectively. In this section, we develop a model to actually determine these parameters using a separate integer programming model. In the process we will also get arrival and departure yard track allocations and a schedule for the hump track. The goal is to decide the times the cars spend on the classification tracks, i.e., T_i^{roll} and T_q^{out} , which should be suitable for further processing in the mixing problem of Section 3.

4.1 A mixed-integer programming model for the Roll-In Problem

We solve the roll-in problem using an integer programming model as described in the following. The underlying idea is that because of the scarce capacities of both the classification tracks and mixing tracks, we would like to minimize the amount of time that cars spend in the classification bowl. As it turned out, the number of car-meter-days spent in the classification bowl, i.e., the sum of time differences between a roll-in and roll-out of a car, taken over all cars and weighted by the length of each car, is a good choice for a proxy-measure.

Assuming that each arriving train fits on each arrival track, we make use of Theorem 2 in modeling the roll-in problem as an augmented bipartite matching problem. We assign outbound trains to departure tracks in a round-robin fashion and assume that the roll-out order is the same as the departure order, which is the current normal practice in Hallsberg.

Let $\mathcal{I} = \{i_1, \dots, i_{n^{\text{in}}}\}$ be the set of inbound trains, and $\mathcal{A} = \{\alpha_1, \dots, \alpha_{k^{\text{arr}}}\}$ the set of tracks on the arrival yard. The binary variable u_{ij} indicates whether train i immediately precedes train j on some arrival track, where $u_{\alpha i}$ indicates whether train i is the first train allocated to arrival track α , and similarly $u_{i\alpha}$ indicates whether train i is the last train allocated to arrival track α in the considered planning horizon. We also schedule the hump track by separating roll-in i and j using the binary variables v_{ij} , where $v_{ij} = 1$ indicates that i is humped before j . To handle separation of a roll-in of a train i and a pull-out p , we use the binary variable w_{ip} , where $w_{ip} = 1$ indicates that i is rolled in before pull-out p .

The set $V = V_1 \uplus V_2$ where $V_1 = V_2 = \mathcal{I} \cup \mathcal{A}$ form the vertices in the graph, while the set of edges E connect 1) all trains and arrival tracks, 2) all arrival tracks and all trains, and 3) pairs of unique inbound trains which can arrive to the same arrival track. More formally, the set of edges is

$$E = (\mathcal{A} \times \mathcal{I}) \cup (\mathcal{I} \times \mathcal{A}) \cup \mathcal{T},$$

where

$$\mathcal{T} = \{(i, j) \mid i, j \in \mathcal{I} \wedge T_i^{\text{arr}} + S^{\text{arr}} < T_j^{\text{arr}}\}$$

and \times denote the cross product operator. We can now formulate the integer problem.

$$\text{minimize} \quad \sum_{i \in \mathcal{I}, q \in \mathcal{O}} \sum_{c \in \mathcal{C}_i \cap \mathcal{C}_q} w_c (T_q^{\text{out}} - T_i^{\text{roll}}) \quad (18)$$

$$\text{subject to} \quad \sum_{(i,j) \in E} u_{ij} = 1 \quad i \in \mathcal{I} \cup \mathcal{A} \quad (19)$$

$$\sum_{(i,j) \in E} u_{ij} = 1 \quad j \in \mathcal{I} \cup \mathcal{A} \quad (20)$$

$$T_i^{\text{roll}} + \Delta^{\text{roll}} \leq T_j^{\text{arr}} + M(1 - u_{ij}) \quad (i, j) \in \mathcal{T} \quad (21)$$

$$T_i^{\text{roll}} + \Delta^{\text{roll}} \leq T_j^{\text{roll}} + M(1 - v_{ij}) \quad i, j \in \mathcal{I} \quad (22)$$

$$T_j^{\text{roll}} + \Delta^{\text{roll}} \leq T_i^{\text{roll}} + Mv_{ij} \quad i, j \in \mathcal{I} \quad (23)$$

$$T_i^{\text{roll}} + \Delta^{\text{roll}} \leq T_p^{\text{pull}} + M(1 - w_{ip}) \quad i \in \mathcal{I}, p \in \mathcal{P} \quad (24)$$

$$T_p^{\text{pull}} + k^{\text{mix}}(\Delta^{\text{pull}} + \Delta^{\text{roll}}) \leq T_i^{\text{roll}} + Mw_{ip} \quad i \in \mathcal{I}, p \in \mathcal{P} \quad (25)$$

$$T_i^{\text{roll}} + \Delta^{\text{roll}} + S^{\text{out}} \leq T_q^{\text{out}} \quad i \in \mathcal{I}, q \in \mathcal{O}, \mathcal{C}_i \cap \mathcal{C}_q \geq 1 \quad (26)$$

$$T_q^{\text{out}} + \Delta^{\text{out}} \leq T_r^{\text{out}} \quad q, r \in \mathcal{O}, q < r \quad (27)$$

$$T_{q-k^{\text{dep}}}^{\text{dep}} + \Delta^{\text{out}} \leq T_q^{\text{out}} \quad q \in \mathcal{O}, q > k^{\text{dep}} \quad (28)$$

$$u_{ij} \leq v_{ij} \quad (i, j) \in \mathcal{T} \quad (29)$$

$$w_{ip} \leq w_{ip'} \quad i \in \mathcal{C}, 1 \leq p < p' \leq m \quad (30)$$

$$u_{ij}, v_{ij} \quad \text{binary} \quad i, j \in \mathcal{I}$$

$$w_{ip} \quad \text{binary} \quad i \in \mathcal{I}, p \in \mathcal{P}$$

$$T_i^{\text{roll}} \geq 0 \quad i \in \mathcal{I}$$

$$T_q^{\text{out}} \geq 0 \quad q \in \mathcal{O}$$

The objective (18) minimizes, for all pairs (i, q) of inbound and outbound trains and all cars c flowing from i to q , the car length w_c multiplied by the time the car spends in the classification bowl. Constraints (19) and (20) ensure that each train and track is

matched exactly once. Constraint (21) specifies the temporal relationship between two trains immediately following each other on a single arrival track. In this constraint, M is a constant large enough to trivially satisfy the constraint when $u_{ij} = 0$. Note that constraint (21) ensures a total temporal order for all trains allocated to a single track. Constraints (22) and (23) ensure that roll-ins are scheduled properly on the single hump track, while Constraints (24) and (25) ensure that roll-ins are separated from the fixed pull-outs. Constraint (26) ensures that enough time is spent for roll-in, coupling and brake testing before the train rolls out to the departure yard. Constraint (27) applies the same roll-out order as the departure order, and constraint (28) ensures that trains can roll-out only when there are free tracks on the departure yard.

Finally, the two constraints (29) and (30) ensure the following: (29) states that if two trains directly follow each other, then they will also be humped in that order, and that two trains cannot follow each other unless they are also humped in that order. (30) enforces that as soon as a roll-in happens before a certain pull-back, it will also happen before all later pull-outs.

5 Case Study

The Swedish traffic administration authority has provided us with historic data for the Hallsberg Rangerbangård hump yard in central Sweden for validation of our approach. Hallsberg has 8 tracks of length 595 to 693 meters on the arrival yard, two parallel humps (of which only one is in use), 32 available classification tracks of varying length (between 374 and 760 meters), and 12 tracks with length 562 to 886 meters on the departure yard. Although there are several other tracks on the yard (most notably tracks going to light and heavy repair facilities) they are not normally used for shunting, and we are therefore not considering them in the model. The layout of Hallsberg is shown in Figure 4.

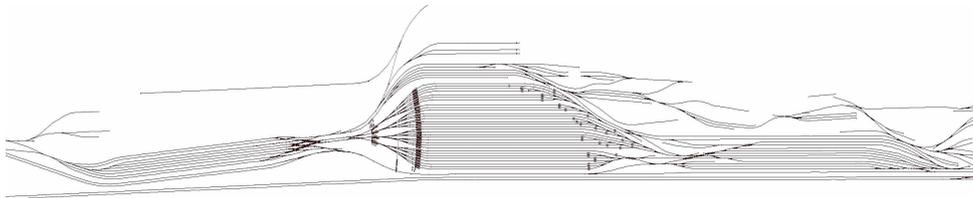


Figure 4: Layout of the Hallsberg classification yard in Sweden. The arrival yard is on the left, followed by the hump, the switching system, classification tracks, and finally the departure yard on the right. Image from Handbok BRÖH 313.00001, Anläggningsbeskrivning Hallsbergs rangerbangård. The image is scaled to emphasize details.

5.1 Test Instances

We collected one week of historic traffic data for the yard, including arriving trains, departing trains, and the set of cars going from each train to each outbound train. Thursday is the busiest day of the week, and was therefore chosen as the start day of the case study. We performed planning for periods of two, three and four days.

We first tried planning for track allocation of the classification bowl only, without using any mixing tracks. However, the problem turned out to be infeasible unless mixed tracks were allowed. We used the following working process to determine a suitable shunting plan:

1. Determine suitable roll-in (T_i^{roll}) and roll-out times (T_q^{out}) for all arriving and departing trains by solving the roll-in problem using the MIP model in Section 4.
2. Solve the feasibility mixing problem in Section 3 using the MIP model in Section 3.1.
3. Improve the MIP mixing solution by fixing v at zero, and replacing the objective (6) with the following:

$$\text{minimize} \quad \sum_{q \in \mathcal{O}, c \in \mathcal{C}_q, p \in \mathcal{P}} z_{cp} \quad (31)$$

This new objective minimizes the number of extra roll-ins needed due to mixing, since a the number of roll-ins affects both how much manual work is performed, and how much the infrastructure is worn. We also added a small penalty term which discourages track allocations starting before any cars have arrived, since this was preferred by Trafikverket. In our experiments, we set $\varepsilon = 1e - 4$.

4. In addition, we ran heuristics A and B from Section 3.2 and 3.3. The heuristic solutions were also improved using the heuristic from Section 3.4.

In the planning, we considered all cars passing through Hallsberg within the planning period, and all trains which contain at least one such car. Table 1 contains information on the separate problem instances we considered. The number of inbound and outbound trains and cars increase slowly when going from two to four days, because Thursday and Friday are by far more busy than both Saturday and Sunday. However, the problem instances are still larger than what is needed for operational planning purposes.

Table 1: Key numbers for the experiments.

	<i>Horizon (days)</i>		
	Two	Three	Four
Inbound trains	126	130	132
Outbound trains	105	108	109
Freight cars	1773	1831	1854

The longest inbound and outbound trains for all three scenarios were 597.4 and 613.5 meters long, respectively. In our sample data, 80% of the trains arrived between 12:00 and 23:59. The following parameter values were used:

- An arrival setup time S^{arr} of 30 minutes.
- A roll-in duration Δ^{roll} (from arrival yard to classification bowl) of 14 minutes.
- A pull-back duration Δ^{pull} of 15 minutes.
- A roll-out duration Δ^{out} (from classification bowl to the departure yard) of 5 minutes.
- A departure preparation time S^{out} of 10 minutes.

- Track 26 and 27 used for mixing, with a total capacity ℓ^{mix} of 1484 meters for temporary car storage.
- A time limit of 20 minutes for the roll-in problem, and 30 minutes each for the two mixing problems.
- Pull-outs at 12:00 and 00:00 every day.

To determine a suitable planning problem we removed cars which

1. spent less than 2 hours and 20 minutes on the yard,
2. spent more than two days at the yard, or
3. did not have a departure train before Monday

Rule (1.) is consistent with standard practice. In addition, many cars in our historic data which spend long time periods on the yard are believed to have been sent to repair, and should therefore not be planned for. We therefore implemented rule (2.) after discussions with Trafikverket. Rule (3.) removes cars for which shunting is impossible.

5.2 Results

The following MIP computations were carried out using AMPL and Gurobi 3.0.2 on a laptop computer. For the problem instances in this paper, the preprocessing using the roll-in MIP model terminated with a feasible solution which was not optimal within 20 minutes; the MIP gap was between 28.0 and 29.6 %. However, the solutions turned out to be good enough for the following steps.

Table 2 shows the results obtained for the problem instances using the tried approaches. In the table, MIP F is the feasibility mixing MIP model, MIP F+M includes running the additional optimization phase with Equation (31) as objective once a feasible solution (respecting mixed track capacity) is found. Heur A and B are the two heuristics (Section 3.2 and 3.3 respectively), while Heur A+I and B+I indicate that the improvement heuristic from Section 3.4 has been applied.

Table 2: Experimental results in solution time, maximum mixed track usage (MTU) with a limit of 1484 m, and the extra car-roll-ins needed due to mixing (ER).

<i>Algorithm</i>	<i>Planning horizon</i>								
	Two days			Three days			Four days		
	<i>Time</i> (s)	<i>MTU</i> (m)	<i>ER</i> (#)	<i>Time</i> (s)	<i>MTU</i> (m)	<i>ER</i> (#)	<i>Time</i> (s)	<i>MTU</i> (m)	<i>ER</i> (#)
MIP F	34	1483.2	389	22	1470.6	476	46	1481.9	497
MIP F+M	1834	1340.2	176	1822	1159.4	191	1846	1415.6	190
Heur.A		2045.8	284		1725.3	270		1487.1	237
Heur.A+I		1454.8	195		1460.7	212		1155.2	204
Heur.B		2016.4	296		1750.6	262		1734.2	265
Heur.B+I		1506.2	228		1534.6	229		1574.7	236

As can be seen in Table 2, MIP F+M yields the overall lowest number of car roll-ins. However, it should be noted that the runtime of the MIP solver was terminated after

30 minutes without finding an optimal solution. Indeed, the lower bound for the model minimizing the number of car roll-ins is far from tight, ranging from 90.4 to 96.0 %. On the other hand, the heuristics, in particular the combination of Heuristic A and the improvement heuristic, consistently yielded feasible plans which also were competitive in terms of the number of rolled-in cars. At the same time, the run-time of the heuristics is negligible, as it is in the order of a fraction of a second, compared to the 30 minutes needed for the MIP model. Note that Heuristic B, which seems to be a plausible first approach if planning manually, does not lead to feasible solutions, even when using the improvement heuristic on top. This indicates that it is not at all an easy task to solve the problem manually. We remark that the quality of all solution approaches for the mixing problem depends strongly on the model chosen for the roll-in problem.

6 Conclusions

In this paper, we studied the track allocation problem for hump yards where temporary storage (mixing) of cars is allowed. We showed that a simplified version of the problem, which is a new coloring problem on interval graphs, is NP-complete. We modeled the mixing problem using integer programming and implemented two construction heuristics and one improvement heuristic. We also modeled the arrival track allocation and hump sequence problem to be able to pre-process timetable and freight connection data.

In a real-world case study of the Hallsberg yard in Sweden, we were able to solve problem instances ranging from two to four days of traffic. Feasible solutions were obtained already by heuristics that run within a fraction of a second. Thus, the problem can be solved in a real-time setting. Regarding the minimization of mixing track usage, the results in the case study indicated that the integer programming solutions were, although better than the heuristic solutions, far from optimal, given a maximum computation time of 30 minutes. We will therefore continue our research in order to refine exact optimization methods. From a practical perspective, we believe that further developing heuristics, perhaps extended with some form of limited backtracking, is relevant for a real-world implementation where system response-time is important.

From a modeling perspective, our approach is limited in that the roll-in problem is solved separately from the mixing problem, when in reality the two problems are interdependent. A possible future research direction is to lift this restriction by integrating the two problems. In addition, the chosen pull-out times (constant in our approach) also affects the classification procedure, and future work should investigate how suitable pull-out times can be found. A further natural question is to incorporate the non-deterministic nature of delays of incoming trains into the model, in order to find plans that are in some sense robust against such delays.

Acknowledgements

We are grateful to Per Danielsson for providing us with the visualization tool for the classification yard occupation. We would also like to thank Pertti Kuusisto and Patrik Fritzing at Trafikverket for providing the yard layout and details on shunt yard operation, and Stefan Huss at Green Cargo AB for verifying timing properties in the model.

Nomenclature

Generic Notation

α	Arrival yard track.
θ	Classification yard track.
Δ^x	Duration of train movement. x specifies activity.
δ	Departure yard track.
c	Car.
i, j	Arriving trains.
p	Pull-out number.
q, r	Departing trains.
S^x	Setup time, work performed on stationary train. x specifies the activity.
T_y^x	Time point of type x for object y .

Setups, Durations, etc.

v	Additional “virtual” capacity on the mixed tracks; should be zero for a feasible solution.
S^{arr}	Arrival setup time, including inspection.
S^{hump}	Roll-in setup time.
S^{out}	Setup time for departure, including coupling and brake testing, carried out on classification tracks.
Δ^{out}	Duration for train movement from classification bowl to departure yard.
Δ^{pull}	Pull-out duration for a single mixed track.
Δ^{roll}	Roll-in duration for a single train.
$\Delta^{\text{p+r}}$	Pull-out and roll-in duration for $k^{\text{mix}} - 1$ trains; $\Delta^{\text{p+r}} = (k^{\text{mix}} - 1)(\Delta^{\text{pull}} + \Delta^{\text{roll}})$.

Binary Variables

x_{cp}	1 if car c is on a mixing track until time period p , 0 otherwise.
y_{qt}	1 if departing train q is allocated to classification track t , 0 otherwise.
z_{cp}	1 if car c is on a mixing track in period p , 0 otherwise.
u_{ij}	1 if arriving trains i precedes arriving train j on the same arrival track, 0 otherwise.
v_{ij}	1 if arriving train i is rolled in before train j , 0 otherwise.
w_{ij}	1 if arriving train i is rolled in before pull-out p , 0 otherwise.

Tracks

\mathcal{A}	Set of arrival tracks.
Θ	Set of classification tracks.
T_t^{free}	End of previous allocation of track t .
ℓ^{mix}	Total capacity of the mixing tracks, in meters.
k_v^{arr}	Number of arrival tracks.
k_v^{class}	Number of classification tracks.
k_v^{dep}	Number of departure tracks.
k_v^{mix}	Number of mixing tracks.
ℓ_t	Length of track t , in meter.

Trains

T_i^{arr}	Timetabled yard arrival time for arriving train i .
T_q^{dep}	Timetabled yard departure time for departing train q .
n^{in}	Number of arriving trains.
n^{out}	Number of departing trains.
T_q^{out}	End time of classification track allocation for train q .
T_i^{roll}	Roll-in time for arriving train i .
T_q^{start}	Start time of classification track allocation for train q .
\mathcal{I}	Set of arriving trains.
\mathcal{O}	Set of departing trains, ordered according to departure time.

Freight Cars

T_c^{start}	Time point of arrival of car c at the classification track of its departing train.
w_c	Physical length of car c , in meter.
T_c^{roll}	Roll-in time for the arriving train i carrying car c .
\mathcal{C}	Set of freight cars.
\mathcal{C}_i	Set of cars on arriving train i .
\mathcal{C}_q	Set of cars on departing train q .

Pull-outs

m	Number of pull-outs.
-----	----------------------

\mathcal{P}	Set of pull-outs.
\mathcal{P}_c	Set of pull-outs where $T_c^{\text{roll}} + \Delta^{\text{roll}} \leq T_p^{\text{pull}}$.
T_p^{pull}	Pull-out time number p .

References

- [1] 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, March 1999.
- [2] Flavia Bonomo, Guillermo Durán, and Javier Marenco. Exploring the complexity boundary between coloring and list-coloring. *Annals OR*, 169(1):3–16, 2009.
- [3] Sabine Cornelsen and Gabriele Di Stefano. Track assignment. *J. Discrete Algorithms*, 5(2):250–261, 2007.
- [4] Elias Dahlhaus, Peter Horák, Mirka Miller, and Joseph F. Ryan. The train marshalling problem. *Discrete Applied Mathematics*, 103(1–3):41–54, 2000.
- [5] Elias Dahlhaus, Fredrik Manne, Mirka Miller, and Joe Ryan. Algorithms for combinatorial problems related to train marshalling. In *Proceedings of the Eleventh Australasian Workshop on Combinatorial Algorithms (AWOCA)*, pages 7–16, 2000.
- [6] Gabriele Di Stefano and Magnus Love Koči. A graph theoretical approach to the shunting problem. *Electronic Notes in Theoretical Computer Science*, 92:16–33, February 2004.
- [7] Michael Gatto, Jens Maue, Matus Mihalak, and Peter Widmayer. Shunting for dummies: An introductory algorithmic survey. In *Robust and Online Large-Scale Optimization*, volume 5868 of *LNCS*, pages 310–337. Springer, 2009.
- [8] Shiwei He, Rui Song, and Sohail S. Chaudhry. An integrated dispatching model for rail yards operations. *Computers & OR*, 30(7):939–966, 2003.
- [9] Riko Jacob, Peter Marton, Jens Maue, and Marc Nunkesser. Multistage methods for freight train classification. *Networks*, 2010. to appear.
- [10] M. W. Siddiquee. Investigation of sorting and train formation schemes for a railroad hump yard. In *Proceedings of the 5th International Symposium on the Theory of Traffic Flow and Transportation*, pages 377–387, 1972.
- [11] Thomas Winter and Uwe T. Zimmermann. Real-time dispatch in storage yards. *Annals of Operations Research*, 96(1-4):287–315, November 2000.