CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Improving Performance of Rail Yards through Dynamic Reassignments of Empty Cars

Tolga Bektas
Teodor Gabriel Crainic
Vincent Morency

June 2007

CIRRELT-2007-19

Bureaux de Montréal:
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone: 514 343-7575
Télécopie: 514 343-7121

Bureaux de Québec:
Université Laval
Pavillon Palasis-Prince, local 2642
Québec (Québec)
Canada G1K 7P4
Téléphone: 418 656-2073
Télécopie: 418 656-2624

www.cirrelt.ca

Improving Performance of Rail Yards through Dynamic Reassignments of Empty Cars

Tolga Bektas[1], Teodor Gabriel Crainic[1,*] Vincent Morency[2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 and Chaire de recherche industrielle du CRSNG en management logistique, ESG, Université du Québec à Montréal, C.P. 8888, succursale Centre-ville, Montréal, Canada H3C 3P8

[2] CN, 935 de La Gauchetière Street West, Montréal, Canada, G3B 2M9

Abstract. This paper considers the problem of reducing the time that empty cars spend in rail yards. For this purpose, a methodology based on dynamic reassignments of empty cars is proposed. To solve the associated reassignment problem, a fast and an efficient solution procedure based on the assignment algorithm is described. The procedure has been tested on real-life data gathered from one of the largest railroads in North America. Computational results show that the procedure runs fast and yields notable savings in the time that the empty cars spend in the yard.

Keywords. Rail freight transport, rail yard, empty car, dynamic reassignment.

Acknowledgements. This research has been financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Chairs and Faculty Support programs. The authors gratefully acknowledge the support by CN, a sponsoring partner of the NSERC Industrial Research Chair in Logistics Management held by Teodor G. Crainic. We also wish to express our special thanks to Mack Barker, Assistant Vice-President, Network Optimization at CN, for the original idea on empty car redistribution within a yard.

---

* Corresponding author: Teodor-Gabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2007

1 Introduction

In today's world, freight transportation constitutes one of the most important economic activities. Railroads share a significant portion of such operations. Despite the fact that rail freight transportation is known for its ability to offer cost-effective long-haul transportation service, its share of all commercial freight activity in the United States has only exhibited a slight change from 26.5% (in ton-miles) in 1993 to 27.8% in 2002 (Bureau of Transportation Statistics, 2007) (http://www.bts.gov/programs/freight_transportation/html/rail.html). The situation is even more striking in Europe. As reported by the European Conference of Ministers of Transport (ECMT, 2005), the market share of rail freight transport in Western European countries (in ton-km) has decreased from 23.3% in 1993 to 14.8% in 2003, where the latter figure is less than 50% of its share in 1970 (31.1%). The same report by ECMT states the reason for the decreasing performance of rail freight transport to be the inadequate quality of the services provided. This is most probably the main factor nowadays driving railroads to improve the efficiency of their operations. The goal is to increase profitability and competitiveness relative to other modes of transportation, especially trucking. Railroads therefore face significant challenges in achieving improved timeliness and flexibility, while, at the same time, still providing an economically efficient long-haul transportation service.

A rail system is a network that is composed of large and small classification yards, pick-up and delivery stations and junction points, which are all connected by rail tracks. These networks are very large and complex organizations, which require accurate, efficient and flexible tools to plan the operations. It is clear that the performance of rail freight transportation as a whole depends on the performance of the individual elements that make up the network. Rail yards (terminals) are important components of this network, where incoming trains are disassembled, cars are sorted and classified for their next destination, and other trains are formed. With the traditional operational policies of railroads, it used to be the case that cars spent "most of their lifetime in yards: being loaded and unloaded, being classified, waiting for an operation to be performed or for a train to come or, simply, sitting idle on a side track" (Crainic, 1987). Railroads, faced with the challenge of fiercely competing with other modes of transportation, are now going through a significant restructuring (Bektaş and Crainic, 2007), with the goal of increasing their profitability and competitiveness. One of the fundamental efforts in reaching towards this goal is focused on efficient rail yard management by processing the cars in the yard as rapidly as possible to provide for an efficient utilization of the assets and the timeliness of connections.

Designing the rail service network addresses issues at the tactical planning level, including but not restricted to route selection and terminal policies, services frequencies, types and capacities, load consolidation (see Crainic, 2000). Issues related to asset management (such as crews, power units, and vehicles) are of concern at the operational level planning, based on a predetermined service network configuration. Particular to the planning decisions of this level include (re)assigning and repositioning of these assets, in order to ensure that the demand is met by using the resources in an efficient manner.

Repositioning or reassigning part of the assets in rail yards as a part of operational level activities may reveal opportunities to optimize asset utilization and to speed up the rail yard operations. In this paper, we contribute to this goal by focusing on a particular case inspired by an actual application. More specifically, we consider the problem of accelerating the flow of empty cars through rail yards. Our goal is to identify opportunities to reduce the idle time empty cars spend in yards by dynamically changing their train assignments.

We summarize the fundamental contributions of this paper as follows:

- This paper shows that there are opportunities to reduce the idle time that cars spend in a rail yard by dynamically reassigning empty cars and investigates the problem of identifying these opportunities. To the best of our knowledge, the empty car reassignment problem in rail yards is hereby introduced and studied in the literature for the first time.
- A dynamic approach to the empty car reassignment problem in rail yards is proposed. An integer programming formulation for the problem is developed, which gives way to a fast and very efficient solution procedure based on the well-known assignment algorithm. Furthermore,

the results of the computational experiments confirm that the procedure can be efficiently used in practice.

- The proposed solution procedure is tested on real data that is obtained from one of the major North American railroads. The computational results show that it is possible to obtain noteworthy savings in time spent by the cars in the yard through reassignments.
- The proposed methodology is not restricted to railroads and can be generalized to contexts such as intermodal container terminals (e.g., seaports or inland depots). In these terminals, (empty) containers are stored while waiting to be transferred from one transportation mode to the other. More information on this is provided in Section 3.1.2.

The rest of the paper is organized as follows. The following section provides background on the main operations performed in freight railroad transportation. Section 3 is dedicated to providing a definition of the problem considered in this paper and a detailed description of the setting in which it arises. We briefly review the related literature in Section 4. Section 5 presents the proposed procedure, along with mathematical notation, the integer programming formulation, and the algorithm. We illustrate the application of the procedure on a real life data set in Section 6 and present the results of the computational experiments. The paper concludes in Section 7 with suggestions for further research.

2 Description of freight railroad operations

In this section, we provide background information on how a rail freight transportation system functions and the fundamental operations that take place at rail yards. This section is further divided into two parts, where the first part discusses railroad planning activities and the second part focuses on rail yard operations.

2.1 Railroad planning

Railway planning activities can be examined according to the classical hierarchy of planning decisions, namely strategic (long-term), tactical (medium-term), and operational (short-term). At any point in time, the network traffic is characterized by the cars and trains that are travelling on the rail tracks and those that are processed at yards. A train is characterized by its route, origin, destination, intermediate stops and physical path, as well as the departure and arrival times at each station of the rail network at which it stops. A *timetable*, constructed as a part of tactical planning activities, includes all the aforementioned information related to the trains, in addition to the transportation time on each route between yards and the processing times therein.

Each car in the network has also associated with it an itinerary that specifies, but is not limited to, an origin and a destination. A car does not necessarily have the same destination as that of the train it is assigned to, as cars may travel on multiple trains during their journey. Cars in the network do not travel alone, but as a part of groups of cars, called *block*s. Blocks are formed at rail yards through a process called classification, and the general rules as to which cars should form each block are referred to as *blocking policies*. A block is characterized by an origin-destination pair, although individual cars in a single block may have different origins and destinations. A block is treated as a single unit for handling purposes, hence once it is formed at its origin yard, it will not be classified again until it arrives at its destination yard. As the classification process is a major source for delays in a rail network, developing efficient railroad blocking plans may reduce the delays in rail yards. The problem of determining such plans is known as the *railroad blocking* problem, with an overall goal of designing the blocks such that all the traffic is transported while minimizing the number of classifications and satisfying blocking capacity constraints.

Once the blocks are formed, their assignments to trains has to be determined. These decisions, named *make-up policies*, are reached through the solution of the *block-to-train assignment problem*, which consists of transporting the blocks at a minimum cost and delay, while obeying train capacity and other operational constraints.

Both the blocking and the make-up policies fall into a larger class of decisions called terminal policies, which are usually decisions of the tactical planning level.

2.2 Rail yard operations

There exist two types of rail terminals in rail networks, *flat* and *hump* yards, depending on how the classification process is carried out. More details on how these two yards differ will be provided shortly. In Figure 1, we depict the general structure of a hump yard. A flat yard exhibits a similar structure, except there is no hump.

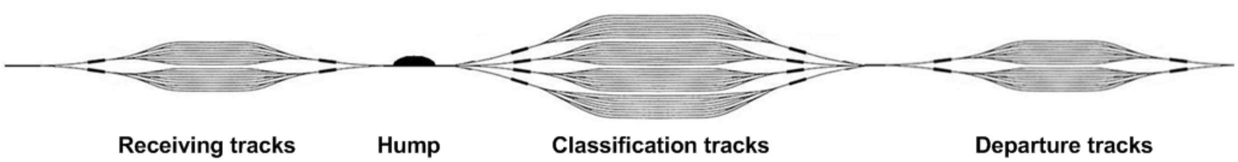


Figure 1: The generic structure of a hump yard

A train arriving at a yard first enters a *receiving area*, where the engines are taken out and taken away for inspection and maintenance, blocks are separated and cars are inspected. The cars are then grouped into what is called a *cut* (more on this concept will be provided in Section 3), and each cut is usually positioned on one of the receiving tracks. The classification operation begins from this point on, and it can be performed in two ways, depending on the type of the rail yard. In flat yards, a switching engine is used to pull the cuts out of the receiving tracks and pull them onto one of the classification tracks, usually one car at a time. In hump yards, classification is performed by using an artificially built hill, called the *hump*. An engine pushes a cut out of the receiving tracks and up the ramp until it reaches the top of the hill. Then, with the help of gravitational force, the cars in the cut roll down the incline, usually one car at a time, and are directed onto one of the classification tracks.

Following this operation, each classification track is thus occupied by a group of cars that form the blocks. Each block then waits until the departure time of its outbound train. When the time comes, the blocks are pulled out of the classification tracks onto the departure tracks and are attached to the train. Following one last inspection of the whole train, the train and blocks leave the yard.

Classification is not the only operation that is performed at a rail yard. Other types of operations include inspection, crew change, refuelling the trains, and dropping and picking up blocks of cars. Among these, however, classification is known to be the major time-consuming operation.

In the next section, we define the problem we address, describe in detail the setting in which it arises, and provide discussions on the possible variants.

3 Problem Description and Setting

The time that a car spends in the yard is usually referred to as the *dwell time*, which is perhaps the primary performance measure in determining a yard's efficiency. Our interest therefore lies in reducing the dwell times of empty cars to improve the performance of rail yards in the following manner.

Cars that are processed at a rail yard are either loaded or empty. Due to their very own nature of carrying specific freight, loaded cars do not permit any kind of a change in their itinerary. This is, however, not the case for empty cars, and their itineraries can be subject to a change if such an operation results in a reduction of their dwell times. The problem undertaken in this paper can then be stated as performing dynamic reassignments of empty cars to outbound trains such that the total dwell time of the cars processed at the yard is reduced as much as possible, while satisfying the following constraints:

- The block compositions with respect to the number and the type of empty cars should stay the same as specified in the timetable,
- The outbound train capacities (length, number of cars) should be respected,
- Each car should be assigned to exactly one outbound train.

In the remainder of the paper, we will use hump yards as an application context to illustrate the general idea, but the discussions provided in this paper are also applicable in a flat yard setting.

We now take a closer look at how the classification process is performed at hump yards.

3.1 Physical operations

Humping is a means to perform the classification operation in hump yards. Cars in such yards are not humped individually, but in groups, called *cut*s. A block and a cut refer to different grouping schemes, where the former implies a group formed solely for travelling purposes, and the latter corresponds to a group assembled to facilitate the humping process. Once the train enters the receiving area, the cars are assembled into cuts and each cut is positioned on one of the receiving tracks (although each receiving track may hold more than one cut). The number of cars in a single cut may vary, and in relatively big hump yards, this number is usually between 50 and 100.

Each cut has a *cut-off time* that specifies the latest point in time that the cut (hence the cars within) may be humped in order to make the connection to the outbound train. This is usually determined by allowing a pre-specified amount of slack time (denoted by T_s) before the associated outbound trains leave the yard. T_s is usually not less than four hours, which includes the required time for processing the cars over the hump, forming the blocks on the classification tracks, and moving the blocks to the departure tracks to form the train. Hence, a car in the queue must be humped at least T_s time units before its outbound train departs.

Trains are generally operated according to schedules and this implies running regular and cyclically-scheduled services with fixed composition. More specifically, a train with a specific capacity (number of cars, length, tonnage) and a number and definition of blocks (i.e., origin, destination, length) is operated based on a given frequency. This frequency is often once a day, but it can be lower, e.g. two or three times a week. In the latter case, the long headway between consecutive departures may cause problems for missed connections. Indeed, in the event that a car is humped after its cut-off time, the connection is missed and the car is reassigned to the next departure of the same train.

Factors contributing to the missed connections are generally delays in other yards or changes in network conditions, which cause fluctuations in the arrival times of inbound trains. The occurrence of such events imply an increase in a car's dwell time when it misses its connection, proportional to the frequency of its outbound train.

Even in the absence of connection-missing events, some cars may unnecessarily wait in the yard longer than they should. This is due to the way that cuts are formed. More specifically, each cut includes a number of cars with different outbound trains, and the cut-off time of the cut is determined with respect to the car that has the earliest departure of its outbound train amongst all included in the cut.

3.1.1 Reassignments based on cuts and cut orders

Most often, the classification process in rail yards is carried out with respect to a given ordering of the cuts, and the engine pulls the cuts out of the receiving tracks up to the hump based on this order. While this ordering does not imply physically positioning the cuts as a "waiting line" in the yard (as such a line would block the track leading to the hump and prevent the engine from running efficiently), it should nevertheless be treated as a conceptual queue (what we call here the *cut queue*). To illustrate this process, we depict an example classification process in Figure 2.

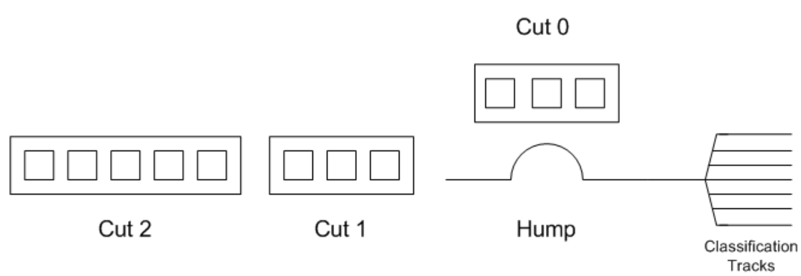


Figure 2: Illustration of the humping process and the cut queue

Figure 2 shows one cut that is being humped (named `Cut 0`) and two other cuts (named `Cut 1` and `Cut 2`) that are waiting in the cut queue to be humped in the given order. In this case, the problem we tackle in the paper is restricted therefore by the additional constraints imposed by this ordering. In other words, the order of the reassignments depends upon the ordering of the cuts which can not be changed.

Alternatively, there may be situations where cut orders are not as strict and one may dynamically change the ordering if this proves to be advantageous. In such cases, the reassignment problem involves additional decisions pertaining to the changes in the ordering of the cuts. Although this flexibility may result in higher improvements with respect to the total dwell time, one must be careful to respect the cut-off times if and when modifying the order of the cuts. This issue is especially relevant for the loaded cars in the cut which have priority over empty cars.

3.1.2 Reassignments based on individual units

In the freight railroad industry, the classification process is almost always carried out with respect to cuts. However, there may be cases where there are no formations of such groups. This phenomena can be observed especially in container terminals. These terminals are used to transfer the containerized cargo from one mode of transportation to the other, and to store the containers in the meantime. When loading and unloading, each container is treated as an individual unit. The reassignments are therefore not restricted by any grouping schemes and can be performed in a much more flexible manner as opposed to rail. However, there is still an issue of an ordering, since containers are usually stacked on top of each other. In this case, in order to retrive and process any container, the containers placed on top of it (if any) should be relocated.

3.2 The focus of this paper

In this paper, since we are inspired by an actual application at a major rail yard in North America, we consider the case where cuts are scheduled to be humped in a given order. This paper will therefore investigate the opportunities of time improvements under such a restriction. Although this order can not be changed in practice, we will nevertheless also perform an analysis of how the results might change if the ordering of the cuts could be changed. Details on how this may be accomplished will be provided in Section 5.

4 Related literature

To the best of our knowledge, no prior work nor a solution algorithm exists in the literature for the problem addressed in this paper. However, there has been considerable amount of research on railroad network operations and relatively less on empty cars. In this section, we start by briefly reviewing some introductory papers for the former case, followed by a somewhat more detailed review for the latter.

For an introduction to railroad operations, we refer to the reader to the introductory papers by Assad (1980a,b), and Crainic (1987). A detailed description of yard operations can also be found in Petersen (1977a). More recent surveys include those by Cordeau et al. (1998) and Crainic (2003), in which reviews of optimization models for train routing and scheduling can be found. Crainic and Kim (2007) provide in-depth discussions of related topics.

Models exist in the literature addressing the rail freight service network design problem. One of the first models in this context is presented by Crainic et al. (1984), who consider the problem from a tactical planning perspective and take into account the interactions between blocking, makeup and train and traffic routing policies. The authors present a comprehensive nonlinear mixed integer multicommodity flow model, which is solved through a heuristic algorithm that is based on a decomposition scheme. Tested on data obtained from a Canadian rail network, the authors obtain improvements through the proposed algorithm, in terms of total car delays, train miles and train costs. A generalization of this model for the multicommodity, multimode freight transportation problem is described in Crainic and Rousseau (1986). In the same paper, an algorithm based on decomposition and column generation principles is offered for the solution of this problem.

Keaton (1992) describes an integer linear programming formulation for the problem of designing railroad operating plans incorporating decisions pertaining to optimal train connections, frequencies, and blocking and routing plans for freight cars. The objective is to minimize the costs associated to trains, cars and yard classification. The author describes an algorithm based on Lagrangian relaxation and a dual adjustment procedure to solve the problem.

Early studies on the railroad blocking problem include that of Bodin et al. (1980) and Assad (1983). Later on, the problem has been studied by Newton et al. (1998) who propose a network design based model and offer a column-generation based branch and bound algorithm for its solution. Another study on this problem is due to Barnhart et al. (2000), where the authors make use of a network design model with maximum degree and flow constraints on the nodes. A solution method based on Lagrangian relaxation coupled with subgradient optimization is shown to yield results on realistic data sets with significantly reduced operating costs. In a very recent paper, Ahuja et al. (2006) propose a Very Large Scale Neighborhood Search algorithm to obtain near-optimal solutions to the problem within one or two hours of computation time. Using this algorithm, the authors show that good quality solutions can be obtained on several data sets obtained from several U.S. railroads.

The block-to-train assignment problem has been studied by several authors, e.g. Nozick and Morlok (1997) and Kwon et al. (1998). In a very recent study by Jha et al. (2006), the authors provide arc and path-based formulations for the problem, and develop exact and heuristic algorithms based on the latter. Tested on a real data set, the heuristic algorithm is able to calculate solutions that are within 1%-2% of the optimal solution in a matter of few seconds.

Work has been conducted on studying and analyzing the operations at rail yards, and amongst these we mention the rather earlier papers by Petersen (1977a,b). In these papers, the author presents models for the yard operations using queueing theory, which he uses to derive probability distributions of various yard operations and describes how service rates of these operations can be related to the physical characteristics of the yard and the traffic handled. The models proposed by this author can therefore be used to estimate yard performance under different traffic conditions.

In a more recent study, Bostel and Dejax (1998) address the problem of optimizing the allocation of containers to trains in a rapid transhipment shunting yard. In these yards, trains succeed one another upon arrival and containers are transferred from one train to the other. The problem, in general, is to determine the initial loading place of the containers on the arriving trains and their reloading place on the departing trains so as to minimize the transfers in the yard. The authors study several variants of the problem, namely, with or without imposed initial loading and with limited or unlimited storage capacity. They present four models and describe several solution algorithms.

4.1 Empty car management

Prior research on empty car management solely focuses on the distribution of empty cars in the rail network, but does not investigate the issues in processing these cars in a rail yard. We provide here a short review of some studies regarding the former case.

One of the earliest studies on empty car management in railroads is due to White and Bomberault (1969), who consider the problem of distributing empty cars throughout the whole network by taking into account the forecasted demand. The overall goal is to reposition the empty cars to the locations where they will be available to serve the demand, while, at the same time, to minimize the costs of repositioning the cars. Such operations are traditionally performed by dispatchers, who are informed by the railroad agents of the requirements and the rate of use of these cars. The authors argue that fluctuations on the rate of use may arise quite frequently, which may result in an unbalanced system. Having this as their motivation, the authors develop a network flow-based algorithm to solve the empty car distribution problem.

A review of empty vehicle flow problems and models can be found in Dejax and Crainic (1987). This paper also offers a taxonomy of empty flow problems which distinguish between strategy oriented, long or medium-term *policy models* and shorter-term *operational models*. The authors conclude with two types of modelling approaches that may be appropriate for allocation and distribution of empty vehicle problems that are stated as follows: (i) deterministic, linear, transportation and transhipment formulations on dynamic (space-time) networks, and (ii) stochastic modelling where travel times or supply/demand may be treated as being probabilistic.

Powell and Carvalho (1998) consider the problem of managing a fleet of railroad flatcars over a network in a real-time setting. The authors introduce a model that is used to assign trailers and containers onto flatcars. An introduction to fleet management with a focus on rail car distribution can be found in Powell and Topaloglu (2002).

Holmberg et al. (1998) propose an optimization model for improving the process of distributing empty cars in a Swedish railroad company, which is based on a time-expanded multicommodity network flow problem with integer requirements. This model explicitly considers transportation capacities of trains, as well as arrival and departure times.

In a recent study, Joborn et al. (2004) analyze the cost structure for repositioning empty cars in a scheduled railroad system so as to take into account the economy-of-scale effect of grouping cars at the yards. The authors offer an optimization model that fundamentally serves operational planning purposes, although it could also be utilized at the tactical and strategic planning levels. The model, having the structure of a capacitated multicommodity network flow problem with fixed cost, is solved through a tabu search metaheuristic.

5 Proposed Methodology

A most straightforward way to address the issues considered in this paper is to change the make-up policies of the trains by solving from scratch the railroad blocking and block-to-train assignment problems whenever delays occur in the network. However, these decisions are of the tactical level, and may require performing significant changes in the original plans, hence the operations. Such changes also may have undesirable consequences with respect to meeting the demands at other yards and disrupting the timeliness of the network in general. On the contrary, the approach we propose attempts to improve the efficiency of the operations by not significantly changing the original plan as issued by the railroad planners, but rather perform simple and effective modifications at the operational level. In fact, the modifications we propose are local as opposed to global, do not alter the blocking plans, and ensure that demand for empty cars are still met after the reassignments.

5.1 General idea

To reduce the time a cars sits idle at the yard, the procedure we propose dynamically changes the itineraries of the empty cars. Each empty car has associated a tag that specifies its destination. Changes in the itineraries can not be performed in a random manner, and should be based on the existing set of demands. Given the set of empty cars and their outbound trains, one may then perform a reassignment operation such that the total number and type of empty cars assigned to each outbound train in the initial plan remain the same after the reassignment. This suggests, for a pair of empty cars, an operation that we call *swapping*, where one of the cars is assigned to the outbound train of the other, and vice versa. We should mention here that swapping is not a physical process in the yard, but rather just a matter of changing the itineraries and tags of the empty cars. Hence, it can be carried out in an effective manner.

Empty cars come in different types, such as box cars, flat cars, gondolas, etc. A swapping operation can therefore only be performed in between two cars of the same type.

A natural question that arises here is how to identify and make use of such swapping opportunities. As humping is a continuous process at the yard, the approach we propose here is based on a *sliding time window* concept. A *time window* is defined with respect to an occurrence of a humping event. In any given time window, all cars that are placed in the cut queue and that have not yet been humped may be swapped. However, all cars that have already been humped and are located on the classification tracks can not be subject to swapping as this would require a physical movement of the car from the classification tracks to the receiving tracks. Therefore, in each time window, cars can be swapped only if they have not been classified.

Once the reassignments have been made and the humping event is performed, the time window slides as to include the new cut that enters the queue, and swapping is performed once more. This operation continues iteratively.

Formal description of the swapping procedure along with notation is given below.

5.2 Notation and formal description of the procedure

The reassignment operations for empty cars can be modelled by a constrained assignment problem as follows. Let $\mathcal{P}$ denote the set of cars that are waiting in the cut queue (which will hereafter be referred to as the car pool), $\mathcal{R}$ denote the set of outbound trains, and $\mathcal{S}$ denote the set of empty car types. We will use $t(i)$ to refer to the type of any car $i \in \mathcal{P}$, hence $t(i) \in \mathcal{S}$. We will denote by C_j^k the number of empty cars of type $k \in \mathcal{S}$ originally assigned to train $j \in \mathcal{R}$, and by c_{ij} the dwell time of car $i \in \mathcal{P}$ when assigned to train $j \in \mathcal{R}$ as initially planned.

Let x_{ij} be a binary decision variable that is equal to 1 if car $i \in \mathcal{P}$ is assigned to train $j \in \mathcal{R}$, and 0 otherwise. The formulation of the problem can then be given as

$$\text{Minimize} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{R}} c_{ij} x_{ij} \tag{5.1}$$

subject to

$$\sum_{j \in \mathcal{R}} x_{ij} = 1 \qquad \forall i \in \mathcal{P} \tag{5.2}$$

$$\sum_{i \in \mathcal{P}: t(i) = k} x_{ij} = C_j^k \qquad \forall j \in \mathcal{R}, \forall k \in \mathcal{S} \tag{5.3}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in \mathcal{P}, j \in \mathcal{R}. \tag{5.4}$$

In this formulation, constraints (5.2) indicate that each car should be assigned to only one outbound train. Constraints (5.3) are, in a sense, capacity restrictions, which imply that after the reassignment each train has the same number and type of empty cars assigned to it as in the initial timetable. Solving this model for each predefined time window, one may thus obtain a reassignment scheme that would result in a minimized total dwell for that time window.

We now show that, taking advantage of the fact imposed by constraint (5.3), one can devise an efficient algorithm to solve the model defined by (5.1)-(5.4) that runs in polynomial time. At this point, we make a distinction between the cars in the first cut of the cut queue (the cut that will be humped next) and those in the remaining cuts. Then, for any given time window, it will suffice to consider the possible swaps of the cars in the former set with those in the latter set. One need not consider the latter set for swapping opportunities as these may be swapped yet another time in subsequent time windows.

A single example given in Figure 3 should serve to illustrate the idea. Assume that within a given time window there is a cut queue that is composed of two cuts (shown inbetween the dotted lines in Figure 3a) with different types of empty cars denoted by A-D. Cut 0 is being humped in Figure 3a. Since Cut 1 is to be humped next in the queue, the goal is to see if the cars in this cut may be swapped with others in the remaining cuts (in our example, Cut 2). Assume that we are interested in trying to decrease the dwell time of the type A car in Cut 1 (A_1) with the itinerary given in Table 1. The only possible swap that can be performed here is with the car of the same type in Cut 2 (A_2), having the itinerary also given in Table 1. A swap operation performed on these two cars (Figure 3b) yields the new itineraries given in Table 2. As one may see, the dwell time of A_1 has decreased from 19 hours to 14 hours, which is a savings of approximately 26%. Although the total dwell of these two cars has not changed, we are not interested in the "temporary" change in the dwell time of A_2 at this point, as this may change yet again in subsequent iterations. However, as Cut 1 will be humped in the next time window (Figure 3c), all the reassignments for the cars in this cut are considered as "final" since no further changes in the itineraries are possible for a humped car.

Table 1: Car itineraries for the example before the swap operation

	Arrival	Departure	Dwell
A_1	10/01/2006 01:00	10/01/2006 20:00	19h
A_2	10/01/2006 03:00	10/01/2006 15:00	12h

Table 2: Car itineraries for the example after the swap operation

	Arrival	Departure	Dwell
A_1	10/01/2006 01:00	10/01/2006 15:00	14h
A_2	10/01/2006 03:00	10/01/2006 20:00	17h

We formally define a *swap* between a pair of cars (i, j), $i, j \in \mathcal{P}$ as the operation of assigning car

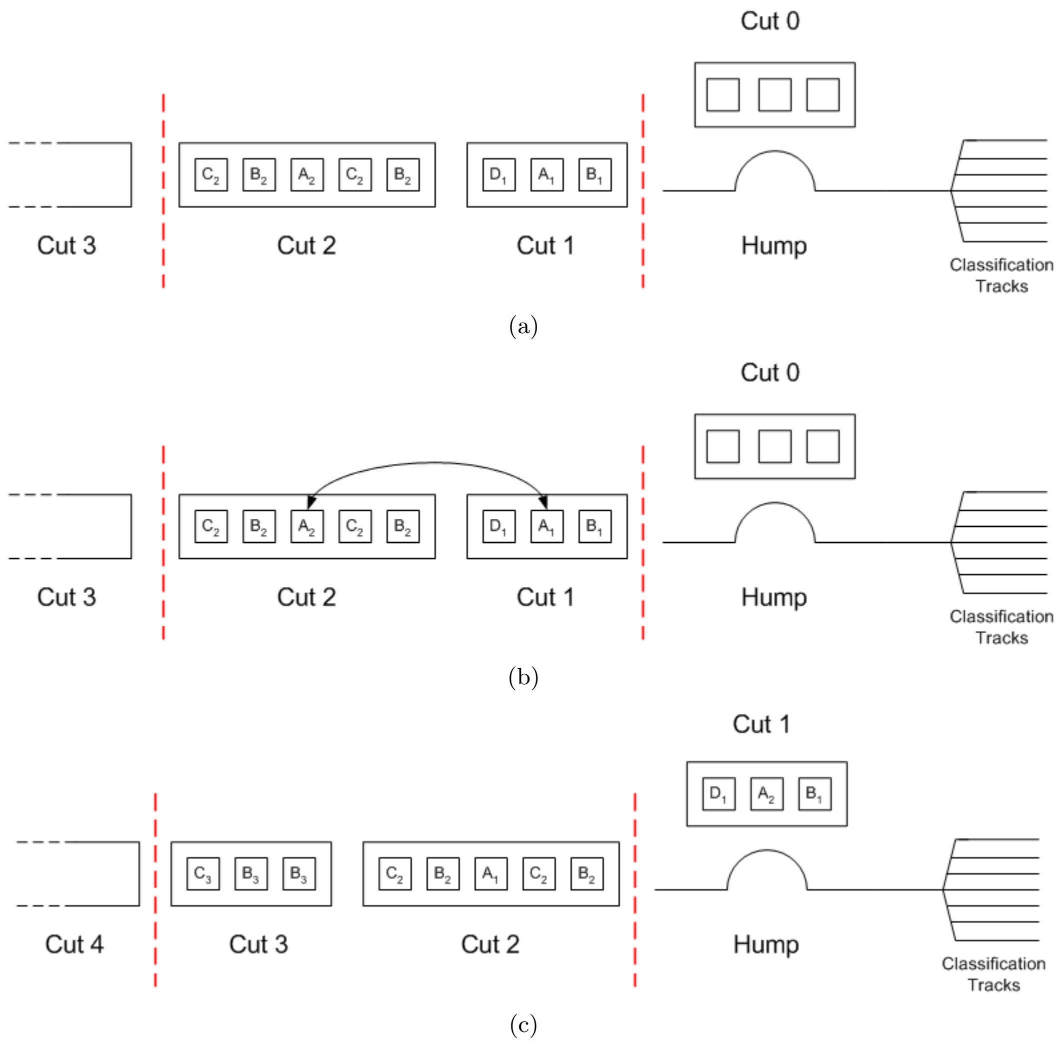


Figure 3: Illustration of the the sliding time window procedure with (a) Cut 0 being humped (b) Swapping operation takes place inbetween Cuts 1 and 2 (c) Cut 1 being humped

i to the outbound train of car j, and vice versa, and denote this assignment operation by s_{ij}. We also denote by r_{ij} the reduction in dwell time as a result of an assignment operation s_{ij}. The r_{ij} parameter can be determined as follows. Let t_i^0 denote the initial dwell time of car i as given by the original plan, which can be calculated as

$$t_i^0 = d_i - a_i, \tag{5.5}$$

where d_i denotes the scheduled departure and a_i denotes the actual arrival time of car i. Under any assignment operation s_{ij}, the revised dwell times for car i, denoted by t_i^j, can be calculated as

$$t_i^j = d_j - a_i. \tag{5.6}$$

The total reduction in the dwelling time for car i can then be written as the following expression

$$r_{ij} = t_i^0 - t_i^j. \tag{5.7}$$

Clearly, $r_{ii} = 0$ for any $i \in \mathcal{P}$. An important fact worth mentioning here is as follows. Since the cut-off time for any car i implies some T_s units of slack time (that is necessary for further operations before it leaves the yard and is usually fixed regardless of the car), we disregard any assignment s_{ij} that results in a revised dwell time which is less than T_s. Hence, graph $\mathcal{G}$ does not include any (i, j) with r_{ij} such that $t_i^j < T_s$.

The assignment procedure works as follows. At any given time, the yard has a number of cuts of which the composition and the order in the cut queue is given in the original timetable. Let the cuts be ordered in the queue as $\mathcal{P} := \{C_t, C_{t+1}, \ldots, C_K\}$, where C_t is the first cut in the queue (we will also refer to this cut as the *next cut to be humped* for each time window). We would like to see, if there exists a better assignment of cars $i \in C_t$ to the positions of those in $j \in \mathcal{P}$, such that the total dwell time for the empty cars in set C_t is reduced. To perform this, we model the assignment operation on a weighted bipartite graph $\mathcal{G} = (V_h \bigcup V_p, \mathcal{A})$, where $V_h = \{i | i \in C_t\}$ is the set of cars that will be humped next and $V_p = \{j | j \in \mathcal{P}\}$ is the set of all cars in the cut pool. The arc set is defined as $\mathcal{A} := \{(i,j) | i \in V_h, j \in V_p\}$. The weights r_{ij} for each $(i,j) \in A$ denote the savings in dwell time of car i by assigning it to the position of car j. The resulting graph is depicted in Figure 4. In the case of no reassignment, the car would be assigned to itself (i.e., s_{ii} with $r_{ii} = 0$).

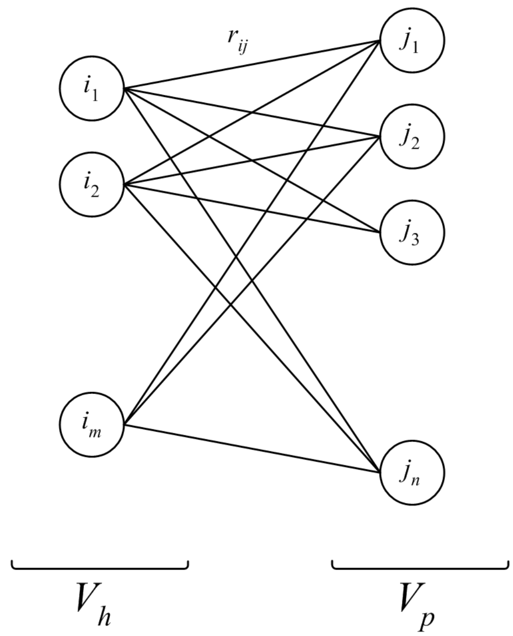


Figure 4: Modeling the swapping problem as a maximum weighted matching problem on a bipartite graph

The procedure can now be stated as finding the maximum weighted matching problem on graph $\mathcal{G}$, which is equivalent to the well-known assignment problem. The problem, defined on this graph, can then be solved in $\mathcal{O}(|C_0|^2|\mathcal{A}|)$ time using the Hungarian algorithm (Kuhn, 1955).

Having constructed the graph, we now show that there are some opportunities to reduce the size of the graph by using the following reduction rules. Let x^*_{ij} denote the value of the decision variable related to arc (i,j) in the optimal solution. Then,

1. Any arc $(i,j) \in \mathcal{A}$ with $r_{ij} < 0$ may be eliminated from the graph as it is easy to see this arc will never be selected in the optimal solution. Note that this operation would only reduce the size of the number of arcs in the graph.

2. For any node $i \in V_h$, if $r_{ij} < 0$ for all $j \in V_p$, then $x^*_{ii} = 1$ (i.e., the initial assignment will not

change). Hence, for this node, $V_h \leftarrow V_h \setminus \{i\}$ and $\mathcal{A} \leftarrow \mathcal{A} \setminus (i,j)$ for all $j \in V_p$. Notice that this would reduce the size of the assignment problem.

At any given time, the reduction in the dwell times are calculated only over cars that are humped, as dwell times for those cars that are not humped may change yet another time in the subsequent re-assignment phases. The procedure runs by solving the problem on the bipartite graph for each time window, performing the humping operation (updating the car pool) and continuing in such an iterative manner. A formal description of the procedure is given in Algorithm 1.

Algorithm 1 The `Dynamic Reassignment` Procedure

1: Initialize with a set of empty cars $\mathcal{P} = \{C_0, C_1, C_2, \ldots, C_K\}$.
2: Let $t := 0$, $T_0 := 0$, and $T_r := 0$.
3: **while** $t \leq T$ **do**
4:     Let $V_h = \{i | i \in C_t\}$, $V_p = \{j | j \in \mathcal{P}\}$.
5:     **for all** $(i,j) \in A$ **do**
6:         $r_{ij} = t_i^0 - t_i^f$
7:     **end for**
8:     Let $\mathbf{x}^* \in \{0,1\}^{|C_0||\mathcal{P}|}$ be the solution of the assignment problem on graph $G = (V_h \bigcup V_p, \mathcal{A})$ with $\mathcal{A} := \{(i,j) | i \in V_h, j \in V_p\}$.
9:     **for all** $(i,j) \in \mathcal{A}$ **do**
10:         **if** $x_{ij}^* = 1$ **then**
11:             $d_i' = d_j$, $d_j' = d_i$, $t_i^0 = d_i - a_i$ and $t_i^j = d_i' - a_i$.
12:             $T_0 \leftarrow T_0 + t_i^0$ and $T_r \leftarrow T_r + t_i^j$.
13:         **end if**
14:     **end for**
15:     $t \leftarrow t + 1$. Update car pool as $\mathcal{P} \leftarrow \mathcal{P} \setminus \{C_0\} \bigcup \{C_{K+t}\}$.
16:     Go to Step 4.
17: **end while**

As an output, the `Dynamic Reassignment` procedure returns the total initial (T_0) and reduced (T_r) dwell times of the cars that are humped in the system for the given time horizon (e.g. day). One can then calculate the reduction (in percent) in the dwell times for these cars as $100(T_0 - T_r)/T_0$.

As the proposed procedure runs on a continuous time horizon and each car can be subject to a numerous number of reassignments, we can not give a theoretical bound on the overall savings that can be achieved. We can, however, derive the following properties that would shed light on the savings that can be obtained for a given time window. Before doing so, we denote headway, i.e. the time between consecutive departures of a train, by H_s. As previously mentioned, H_s is generally 24 hours, but may be up to 72 hours depending on the service frequency of a train.

Proposition 1 *The savings that can be achieved in a given time window is at least the value of the maximum matching on the bipartite graph* $\mathcal{G} = (V_h \bigcup V_p, \mathcal{A})$ *as defined above.*

Proof Follows from the construction of the bipartite graph. □

In the next proposition, we show that the swapping procedure can be especially effective when a delay in an inbound train results in a car missing its connecting outbound train. As mentioned above, such cars are automatically assigned to the next departure of the same train, thereby resulting in an additional H_s hours of dwell.

Proposition 2 *Given a time window at point* T *in time characterized by sets* V_h *and* V_p *as defined above, let* $\mathcal{F} = \{i \in V_h | d_i - h_i \geq T_s\}$*, where* h_i *denotes the time that car* i *will be classified. Also define, for each* $i \in \mathcal{F}$, $\mathcal{R}_i = \{j \in V_p | d_j - h_j < T_s \text{ and } d_j - h_i \geq T_s \text{ and } d_i - h_j \geq T_s\}$*. Then, the value of the savings that can be achieved in this time window is at least* $H_s\delta$*, where* δ *is the number of* $i \in \mathcal{F}$ *such that* $\mathcal{R}_i \neq \emptyset$*.*

Proof The proposition is based on the fact that, any car $i \in V_h$ (i.e., that will be classified next in the queue) with less than T_s hours to its connection train, will miss that train and will be incurred an additional dwell time of H_s hours until the departure of the next train in the next day. Now consider a pair (i, j) with $i \in \mathcal{F}, j \in \mathcal{R}_i$. The dwell time of car i will be $d_i - a_i$ since by assumption it has sufficient slack time to connect to its outbound train ($d_i - h_i \geq T_s$). This is, however, not the case for car j since the slack time $d_j - h_j$ is less than four hours. The dwell time for car j is then calculated as $d_j - a_j + H_s$. Consider now an assignment s_{ij}, after which car i incurs a revised dwell time of $d_j - a_i$ (by the assumption $d_i - h_j \geq T_s$) and car j incurs a dwell time of $d_i - a_j$ (by the assumption $d_j - h_i \geq T_s$). For a single such pair, the time savings is $(d_i - a_i + d_j - a_j + H_s) - (d_j - a_i + d_i - a_j) = H_s$ hours and will be increased at multiples of H_s for every pair exhibiting this property. □

We illustrate Proposition 2 in the example given in Table 3, showing the itineraries of two cars (C_1 and C_2) in a given time window, where car C_2 is to be classified next in the queue. We assume that both departure trains have a headway of $H_s = 24$ hours.

Table 3: Example illustrating Proposition 2

	Arrival	Hump time	Departure	Dwell
C_1	29/09/2006 03:00	29/09/2006 10:30	30/09/2006 01:00	22h
C_2	29/09/2006 04:00	29/09/2006 12:00	29/09/2006 15:00 (30/09/2006 15:00)	11h (+ 24h)

Table 4: Example illustrating Proposition 2

	Arrival	Hump time	Departure	Dwell
C_1	29/09/2006 03:00	29/09/2006 10:30	29/09/2006 15:00	12h
C_2	29/09/2006 04:00	29/09/2006 12:00	30/09/2006 01:00	21h

In this example, the dwell time for car C_1 is 22 hours and that of car C_2 is 35 hours, including the additional $H_s = 24$ hours resulting from an insufficient slack time of three hours between it hump time (29/09/2006 12:00) and the scheduled departure time (which will be updated by the system to be 30/09/2006 15:00). The total dwell time for both cars is therefore 57 hours. Now consider the revised schedules after a swap operation $s_{C_1C_2}$ given in Table 4. Since this operation results in sufficient slack time for cars C_1 and C_2 (4.5 and 13 hours, respectively), the revised schedule can be carried out as planned. With this operation, the total dwell time for both cars is reduced to 33 hours, which corresponds to a reduction of approximately 42%.

Looking at Table 3, the reader may wonder how (and if) it is possible to have two cars C_1 and C_2 with $a_{C_1} < a_{C_2}$, even though C_2 is scheduled to leave earlier than C_1. Such schedules can be constructed due to the fact that cars are not humped individually, but in cuts and the cut-off (and hump) time of each car is determined not individually, but by the cut-off (and hump) time of the cut it belongs to.

6 Application of the Procedure and Computational Results

In this section, we demonstrate the application of the proposed procedure on a real-life data set obtained from CN, one of the largest railroad companies in North America. In particular, we are motivated by the desire to improve the efficiency of one of CN's rail yards, called MacMillan. MacMillan yard is a major hump yard in CN's rail network, in which incoming cars are classified, formed into blocks according to their destinations, made into trains and sent to their next destination. This yard operates on a 24 hour clock and handles over a million cars (loaded and empties) per year. About half of the cars processed in the yard are empty cars, of which 50% are owned by CN.

The data set we use to test the proposed procedure includes over 1.5 million records associated with statistics of 19,069 different cars at the MacMillan yard within the period of Sept 30, 2006 through Oct 6, 2006. The cars included in the set are composed of those that were processed at the yard in the given time interval, as well as those who were on their way to this yard (called *en-route* cars). The data set includes information on the estimated time of arrival, estimated hump time, cut off time and scheduled departure time of each car (either in the yard or on an inbound train), in addition to their type and their sequence number in the cut. This data set also specifies, given a specific point in time, the order of the cuts that are ready to be humped. We have identified, amongst the whole set, 4466 empty cars that satisfy the interchangeability criteria (i.e., swappable cars). These empty cars constitute approximately 21% of the whole set of cars. In total, these cars form 115 groups based on their types.

6.1 Results of the swapping procedure on the test data

The procedure has been coded in C under Cygwin and run using the real-life data set on a 1.73GHz Pentium PC with 512MB RAM. We give the full results in Table 5, and summarize the results obtained on the real-life data set in Table 6.

The results given in Table 6 show that it is possible to achieve an average time savings of 4.98% with

Table 5: Detailed results of computational experiments on the data set

id	T_0	T_r	imp	N_h	id	T_0	T_r	imp	N_h
1	17.15	17.15	0.00	1	59	584.35	512.35	12.32	28
2	290.07	290.07	0.00	12	60	168.23	168.23	0.00	8
3	80.38	80.38	0.00	5	61	185.37	185.37	0.00	10
4	133.35	133.35	0.00	5	62	44.52	44.52	0.00	2
5	200.58	200.58	0.00	8	63	45.83	45.83	0.00	2
6	7816.53	6897.03	11.76	316	64	61.10	61.10	0.00	4
7	45.68	45.68	0.00	3	65	419.72	414.72	1.19	21
8	2183.40	2007.40	8.06	80	66	14.38	14.38	0.00	1
9	403.58	403.58	0.00	17	67	75.87	75.87	0.00	4
10	42.45	42.45	0.00	3	68	128.80	104.80	18.63	5
11	294.90	294.90	0.00	11	69	42.70	42.70	0.00	1
12	129.03	129.03	0.00	8	70	30.82	30.82	0.00	2
13	35.03	35.03	0.00	1	71	108.28	108.28	0.00	4
14	1187.97	1163.97	2.02	50	72	119.65	119.65	0.00	6
15	471.70	471.70	0.00	19	73	155.08	155.08	0.00	7
16	22.85	22.85	0.00	1	74	55.83	55.83	0.00	3
17	699.18	675.18	3.43	30	75	14.28	14.28	0.00	1
18	250.47	250.47	0.00	12	76	72.57	72.57	0.00	5
19	5494.35	5280.40	3.89	195	77	84.00	84.00	0.00	4
20	500.92	500.92	0.00	21	78	116.85	116.85	0.00	6
21	315.17	291.17	7.62	14	79	71.08	71.08	0.00	3
22	2590.97	2412.97	6.87	116	80	2025.00	1992.00	1.63	91
23	1642.28	1533.28	6.64	73	81	96.73	96.73	0.00	3
24	133.25	133.25	0.00	6	82	18.08	18.08	0.00	1
25	421.02	421.02	0.00	21	83	28.72	28.72	0.00	1
26	26.75	26.75	0.00	1	84	293.20	293.20	0.00	15
27	903.97	903.97	0.00	43	85	95.68	95.68	0.00	4
28	55.97	55.97	0.00	2	86	792.97	792.97	0.00	37
29	542.72	542.72	0.00	21	87	56.38	56.38	0.00	3
30	310.20	310.20	0.00	12	88	22.35	22.35	0.00	1
31	23.82	23.82	0.00	1	89	3865.65	3671.65	5.02	170
32	3083.67	2833.17	8.12	125	90	549.42	549.42	0.00	26
33	2776.97	2544.97	8.35	109	91	485.65	485.65	0.00	21
34	422.02	422.02	0.00	19	92	65.95	65.95	0.00	3
35	742.18	742.18	0.00	37	93	25.47	25.47	0.00	1
36	725.10	725.10	0.00	34	94	489.17	489.17	0.00	30
37	53.98	53.98	0.00	3	95	127.85	127.85	0.00	6
38	59.78	59.78	0.00	3	96	19.22	19.22	0.00	1
39	24.03	24.03	0.00	1	97	503.02	503.02	0.00	25
40	95.75	95.75	0.00	4	98	351.57	351.57	0.00	17
41	1905.28	1881.28	1.26	77	99	661.75	661.75	0.00	41
42	4166.87	3878.87	6.91	200	100	13.25	13.25	0.00	1
43	1495.32	1427.32	4.55	71	101	34.83	34.83	0.00	1
44	197.13	197.13	0.00	8	102	646.22	646.22	0.00	32
45	1546.70	1318.20	14.77	66	103	57.92	57.92	0.00	2
46	70.28	70.28	0.00	3	104	193.30	193.30	0.00	6
47	171.17	171.17	0.00	7	105	22.53	22.53	0.00	1
48	507.82	459.82	9.45	25	106	29.12	29.12	0.00	1
49	438.90	414.90	5.47	18	107	22.72	22.72	0.00	1
50	143.78	143.78	0.00	6	108	14.90	14.90	0.00	1
51	112.80	112.80	0.00	5	109	56.10	56.10	0.00	3
52	390.67	390.67	0.00	19	110	127.33	127.33	0.00	4
53	19.87	19.87	0.00	1	111	31.83	31.83	0.00	1
54	2919.35	2813.85	3.61	136	112	248.83	248.83	0.00	11
55	79.27	79.27	0.00	4	113	56.98	56.98	0.00	3
56	390.60	390.60	0.00	15	114	11.42	11.42	0.00	1
57	299.92	299.92	0.00	14	115	541.67	541.67	0.00	26
58	202.90	202.90	0.00	8					

id: Group number, T_0: initial total dwell time of the cars classified in the group, T_r: reduced dwell time of the cars classified in the group, imp: percent reduction in dwelling time (calculated as $(T_0 - T_r)/T_0$), N_h: Total number of cars classified in the group

Table 6: Summary of the computational results on the test data

Total initial dwell	65604.22 hrs
Total reduced dwell	62339.27 hrs
Improvement	4.98 %
Total number of humped cars	2846
Computation time	17.01 CPU seconds

the procedure, which reduces the average dwell time of a car from 23.05 hours to 21.90 hours. At a first glance, this savings may not seem to be as high as expected. However, one must consider that all the reassignments are performed by respecting a given schedule. In other words, all the changes are done without modifying the initial timetable with respect to demands and outbound train configurations. Given such a tight schedule which has quite a restricted space for improvement, we believe that the reduction in time which has been achieved is noteworthy.

Looking in detail to the results obtained by the procedure in Table 5, we were able to make some additional observations on the parameters that contribute to achieving time savings. Our investigation revealed that the number of cars in each group does not have a noteworthy impact on the savings. This is contrary to intuition as one expects that the larger the groups are, the larger the room for opportunities for swapping operations, hence the savings. However, a significant, albeit less obvious, factor turned out to be the behavior of the composition and the size of the car pool throughout a given time interval. More specifically, the increase in the number of different cars in the car pool per time window (with respect to each group) is observed to increase the number of possible reassignments, and therefore the savings in time.

We illustrate this situation by an example taken from the data set. This example consists of two different groups, named G1 and G2, which include 26 and 18 different cars, respectively. The example data belongs to the first day of the data set, in which a total of 10 cars were humped for both groups. We show, in Figure 5, the change in the size of the car pool at random points during the 24-hour time horizon.

Running the procedure on groups G1 and G2 yields time savings of 3.68% for the former and none for the latter. Although both groups have exactly the same number of cars humped within the

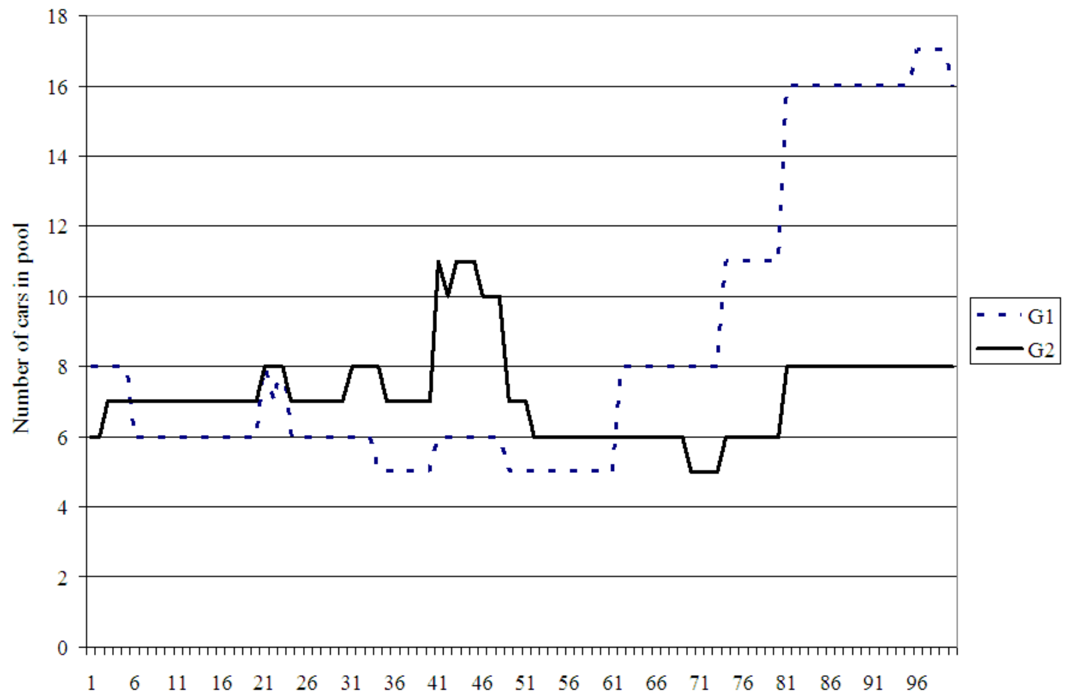


Figure 5: Comparison of two different groups of cars in terms of the car pool size

same time interval, we see from Figure 5 that G1 exhibits a significantly high number of cars in the car pool towards the end of the time interval. The fact that positive time savings were obtained for group G1 can be explained by this phenomena. Group G2, on the other hand, has a relatively lower and stable number of cars in the car pool throughout, which does not seem to be high enough to permit achieving similar savings in terms of time.

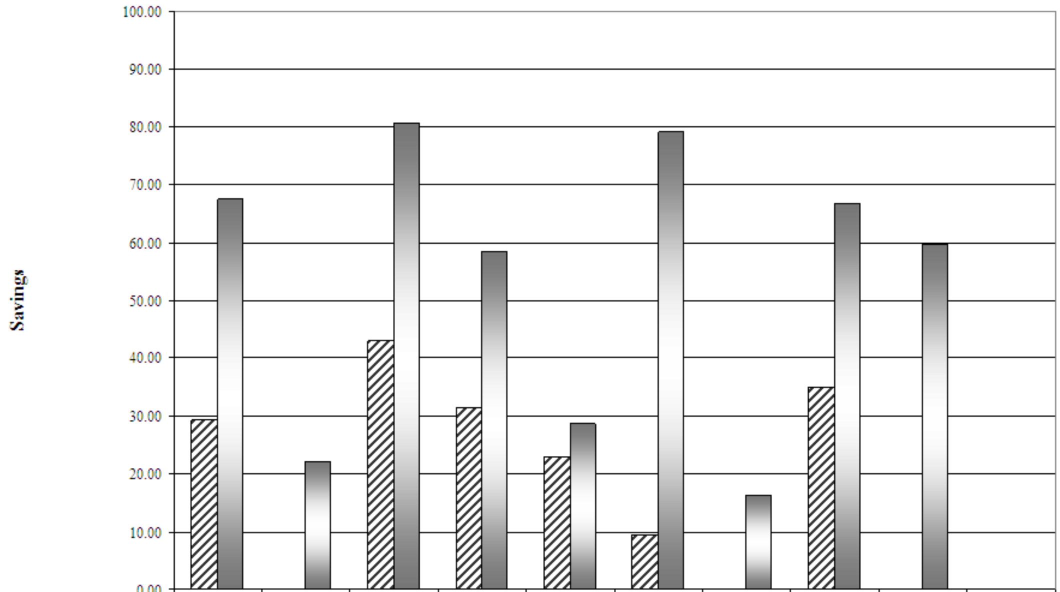
6.2 Effect of changing the humping order on time savings

Up to now, our efforts have concentrated on improvement of the dwell times of the empty cars given a specific sequencing of the cars. This sequence, known as the humping order, is determined by the system a priori. In this section, our goal is to investigate the effect of running the proposed procedure on the reduction of the dwelling times when the humping order is not fixed. In other words, we wish to evaluate the effect on the time savings of jointly finding the best order to hump empty car(s) and performing the associated swapping operations. Note that, in all generality, such a task would require a full-scale simulation of the railyard operations and the humping events, which lies beyond the scope of this paper. We would nevertheless like to give the reader an idea of the potential benefits that could be obtained were we allowed to change, at least partially, the humping order. For this purpose, the following experiments have been conducted.

We have selected within the one-week data set ten random humping events. For each event, a simulation has been performed where a set of cars humped as specified in the original data set (denoted by h_0) has been replaced by other sets of possible cars that could have been humped instead. Given a cut that is composed of h_c cars, this simulation would require testing $\binom{h_c}{h_0} - 1$ alternatives. As our aim is to obtain only an idea of what the impact of such moves could be, we have focused on humping events with $h_0 = 1$, i.e., when only a single car in a cut is humped (and the results of our experiments confirm that savings could be substantial even in this case). The procedure has then been run for each possible alternative.

The results of these experiments are shown in Figure 6 for each of the ten randomly selected humping events. For each event, the series on the left show the savings obtained by the procedure with the humping sequence fixed as given in the data set. The series on the right show the best possible savings that could be obtained by changing the humping order. The specific values that have been obtained as a result of these experiments are also included in the figure.

As the figure shows, changing the humping order almost always has a positive (and mostly significant) effect on the time savings that could be achieved. Out of the ten cases, we have only one case where no savings were obtained for either of the two situations. In some cases (events 2, 7, and 9), although there were no savings were obtained initially, the proposed modifications result in positive and substantial savings. Note that these figures are not directly comparable with those presented in Table 4, as the former set presents the savings for a given instant in time, whereas the latter is calculated over a 24 hour clock. Given the fact that these results are obtained only with cuts where a single car is humped, we suspect that the potential savings could be even higher when multiple changes in the humping sequence are performed.



	1	2	3	4	5	6	7	8	9	10
Original Savings	29.33	0.00	42.90	31.33	22.97	9.59	0.00	35.01	0.00	0.00
New Savings	67.59	22.12	80.60	58.45	28.67	79.03	16.38	66.82	59.71	0.00

Events

Figure 6: Effect of changing the hump cut order on time savings

6.3 Effect of including inbound “en-route” trains on time savings

The analysis and the results shown in the previous sections were based solely on the trains that are located in the yard and ready to be classified. However, there is yet more room for improvement by extending the scope of the analysis with the inclusion of en-route cars. En-route cars are those who are traveling on a train that is within a certain range of the yard based on their estimated time of arrival, i.e., cars which are on their way to the yard. Although these cars are not yet in the yard, their cut-off and and humping times are already planned and this information would allow one to consider these for the swapping operations and therefore for the time reduction opportunities.

In order to see what the effect of the procedure is, we gradually increase the scope of the swapping procedure, in units of hours. The increments are selected so as to consider the en-route cars that will be at the yard within T hours, where parameter T ranges from 2 to 12 in increments of two. For each T, we then run the swapping procedure to identify the resulting savings for the whole week. Figure 7 shows the results of this experiment, where the improvement values correspond to the average savings in time.

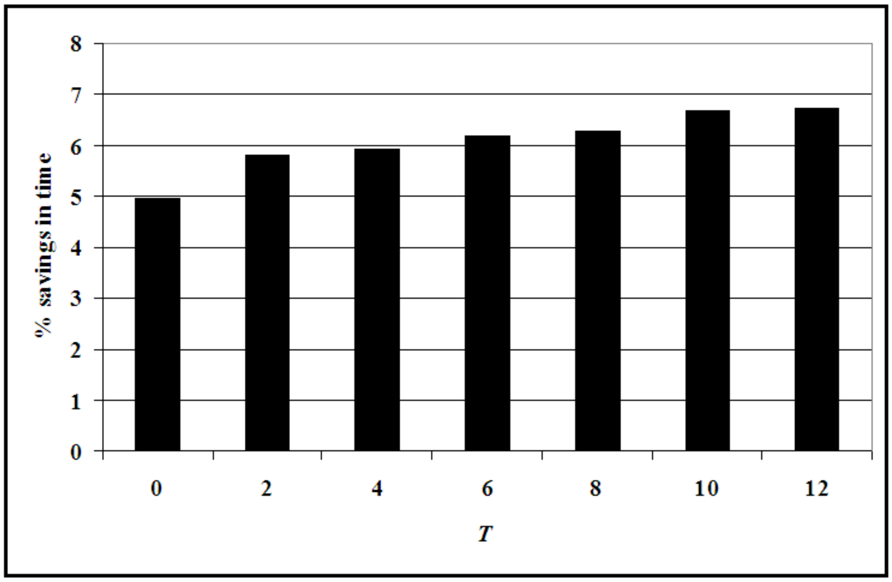


Figure 7: Summary of the statistics of the computational results obtained by inclusion of en-route cars

As can be seen in Figure 7, inclusion of en-route cars has a positive effect on the time savings, even for low values of T. This suggests that one can increase T to obtain even better results. However, we must note that swapping analysis for en-route cars are based on their *estimated* time of arrival only, as opposed to the *actual* time of arrival for the cars that are at the yard. This estimated arrival time can change infrequently based on the delays that occur in the network. Therefore, as we increase the scope of the analysis in including en-route trains, we also reduce the reliability of the predicted connections. This is coupled with the fact that because the procedure only evaluates swaps against the next cut to be humped, the further out a car is, the less likely it is to be a valid swap. Our discussions with CN has brought us to the conclusion that using a $T = 6$ hour limit shows a good balance between dwell-reduction benefits and risks due to accuracy of the estimated time of arrivals.

7 Conclusions

Railroads are showing great interest to improve the efficiency and timeliness of their operations, in an attempt to increase the quality and profits of their services. New trends in logistics, such as intermodal transportation, have now put even more pressure on railroads in competing with other modes of transportation, such as trucking and maritime shipping. The improvement of the efficiency of the rail network as a whole depends on the performance of the rail yards, as they are the sources of major time-consuming operations in the system. This implies that railroad should focus on processing cars in yards as rapidly as possible.

In this paper, we have contributed towards this goal by identifying opportunities in accelerating the flow of cars in rail yards by focusing on empty cars. We offered a simple, fast and yet efficient procedure that performs such reassignments on a sliding time window basis. We showed that noteworthy time savings in dwell time of empty cars can be achieved through the proposed procedure, when tested on a real-life data obtained from one of the major railroads in North America.

Although the ideas and the analysis presented in this paper were motivated by railroad applications, the proposed procedure is not limited to this case and can readily be adapted to other modes of transportation involving empty cars or containers. It is our belief that an interesting application of this method would be in the context of intermodal transportation. Intermodal transportation involves movement of (mostly containerized) freight using at least two modes of transport (e.g. rail, air, shipping) and the transfer of containers from one mode to the other takes place at intermodal terminals (see Crainic and Kim (2007) for a description of the operations in intermodal terminals). Containers are temporarily stored in these terminals until it is time for them to be transferred to the next leg of their journey.

Containers in intermodal terminals are either loaded or empty. Based on our experience with rail yards and empty cars, we anticipate that repositioning or reassigning empty containers can also yield beneficial results. The proposed procedure described in this paper can be used for this purpose in the most straightforward manner. In fact, since containers come in much less number of groups than rail cars, the swapping operations will not be as restricted as in the case of rail freight, hence we expect that even more significant savings can be achieved in this case. This is certainly an interesting issue we believe that should be looked into and hence recommend it as a further research topic.

Acknowledgement

This research has been financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Chairs and Faculty Support programs. The authors gratefully acknowledge the support by CN, a sponsoring partner of the NSERC Industrial Research Chair in Logistics Management held by Theodor G. Crainic. We also wish to express our special thanks to Mack Barker, Assistant Vice-President, Network Optimization at CN, for the original idea on empty car redistribution within a yard.

References

R.K. Ahuja, K.C. Jha, and J. Liu. Solving real-life railroad blocking problems. Research Paper, Department of Industrial & Systems Engineering, University of Florida, Gainesville, FL, 2006.

A.A. Assad. Models for rail transportation. *Transportation Research A*, 14A:205–220, 1980a.

A.A. Assad. Models of rail network: toward a routing/make-up model. *Transportation Research B*, 14B:101–114, 1980b.

A.A. Assad. Analysis of rail classification policies. *INFOR*, 21:293–314, 1983.

C. Barnhart, H. Jin, and P.H. Vance. Railroad blocking: A network design application. *Operations Research*, 48:603–614, 2000.

T. Bektaş and T.G. Crainic. A brief overview of intermodal transportation. In D. Taylor, editor, *Logistics Engineering Handbook*. CRC Press, 2007. In press.

L.D. Bodin, B.L. Golden, A.D. Schuster, and W. Romig. A model for the blocking of trains. *Transportation Research*, 14B:115–120, 1980.

N. Bostel and P. Dejax. Models and algorithms for container allocation problems on trains in a rapid transhipment shunting yard. *Transportation Science*, 32:370–379, 1998.

J.-F. Cordeau, P. Toth, and D. Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32:380–404, 1998.

T. G. Crainic and J.-M. Rousseau. Multicommodity, multimode freight transportation: a general modeling and algorithmic framework for the service network design problem. *Transportation Research B*, 20B:225–242, 1986.

T.G. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122:272–288, 2000.

T.G. Crainic. Long-haul freight transportation. In R.W. Hall, editor, *Handbook of Transportation Science*, pages 451–516. Kluwer Academic Publishers, New York, 2nd edition, 2003.

T.G. Crainic. Rail tactical planning: issues, models and tools. Technical Report CRT-520, Center for research on transportation, 1987.

T.G. Crainic and K.H. Kim. Intermodal transportation. In C. Barnhart and G. Laporte, editors, *Transportation*, Handbooks in Operations Research and Management Science, chapter 8, pages 467–537. North-Holland, Amsterdam, 2007.

T.G. Crainic, J.-A. Ferland, and J.-M. Rousseau. A tactical planning model for rail freight transportation. *Transportation Science*, 18:165–184, 1984.

P.J. Dejax and T.G. Crainic. A review of empty flows and fleet management models in freight transportation. *Transportation Science*, 21:227–247, 1987.

ECMT. Trends in the transport sector 1970-2003. Technical report, European Conference of Ministers of Transport, France, 2005. Available online at http://www.cemt.org/pub/pubpdf/Depl2003E.pdf.

K. Holmberg, M. Joborn, and J.T. Lundgren. Improved empty freight car distribution. *Transportation Science*, 32:163–173, 1998.

K.C. Jha, R.K. Ahuja, and G. Şahin. New approaches for solving the block-to-train assignment problem. Research Paper, Department of Industrial & Systems Engineering, University of Florida, Gainesville, FL, 2006.

M. Joborn, T.G. Crainic, M. Gendreau, K. Holmberg, and J.T. Lundgren. Economies of scale in empty freight car distribution in scheduled railways. *Transportation Science*, 38:121–134, 2004.

M. H. Keaton. Designing railroad operating plans: A dual adjustment method for implementing Lagrangian relaxation. *Transportation Science*, 26:263–279, 1992.

H.W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

O.K. Kwon, C.D. Martland, and J.M. Sussman. Routing and scheduling temporal and heterogeneous freight car traffic on rail networks. *Transportation Research*, 34E:101–115, 1998.

H. Newton, C. Barnhart, and P.H. Vance. Constructing railroad blocking plans to minimize handling costs. *Transportation Science*, 32:330–345, 1998.

L.K. Nozick and E.K. Morlok. A model for medium-term operations planning in an intermodal rail-truck service. *Transportation Research*, 31A:91–107, 1997.

E.R. Petersen. Railyard modelling - Part I: Prediction of putthrough time. *Transportation Science*, 11:37–49, 1977a.

E.R. Petersen. Railyard modelling - Part II: The effect of yard facilities on congestion. *Transportation Science*, 11:50–59, 1977b.

W.B. Powell and T.A. Carvalho. Real-time optimization of containers and flatcars for intermodal operations. *Transportation Science*, 32:127–141, 1998.

W.B. Powell and H. Topaloglu. Fleet Management. In S. Wallace and W. Ziemba, editors, *Applications of Stochastic Programming*, SIAM Series on Optimization, pages 185–216. SIAM Math Programming Society, Philadelphia, PA, 2002.

W.W. White and A.M. Bomberault. A network algorithm for empty freight car allocation. *IBM Systems Journal*, 8:147–169, 1969.