Université de Montréal

A Rolling Horizon Approach for the Locomotive Routing Problem at the Canadian National Railway Company

par

Hoang Giang Pham

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Informatique et recherche opérationnelle

October 7, 2020

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

A Rolling Horizon Approach for the Locomotive Routing Problem at the Canadian National Railway Company

présenté par

Hoang Giang Pham

a été évalué par un jury composé des personnes suivantes :

Margarida Carvalho

(président-rapporteur)

Emma Frejinger

(directeur de recherche)

Jean-François Cordeau

(codirecteur)

Bernard Gendron

(membre du jury)

Résumé

Cette thèse étudie le problème du routage des locomotives qui se pose à la Compagnie des chemins de fer nationaux du Canada (CN) - le plus grand chemin de fer au Canada en termes de revenus et de taille physique de son réseau ferroviaire. Le problème vise à déterminer la séquence des activités de chaque locomotive sur un horizon de planification donné. Dans ce contexte, il faut prendre des décisions liées à l'affectation de locomotives aux trains planifiés en tenant compte des besoins d'entretien des locomotives. D'autres décisions traitant l'envoi de locomotives aux gares par mouvements à vide, les déplacements légers (sans tirer des wagons) et la location de locomotives tierces doivent également être prises en compte. Sur la base d'une formulation de programmation en nombres entiers et d'un réseau espace-temps présentés dans la littérature, nous introduisons une approche par horizon roulant pour trouver des solutions sous-optimales de ce problème dans un temps de calcul acceptable. Une formulation mathématique et un réseau espace-temps issus de la littérature sont adaptés à notre problème. Nous introduisons un nouveau type d'arcs pour le réseau et de nouvelles contraintes pour le modèle pour faire face aux problèmes qui se posent lors de la division de l'horizon de planification en plus petits morceaux. Les expériences numériques sur des instances réelles montrent les avantages et les inconvénients de notre algorithme par rapport à une approche exacte.

mots clés: Problème du Routage des Locomotives, Approche par Horizon Roulant.

Abstract

This thesis addresses the locomotive routing problem arising at the Canadian National Railway Company (CN) - the largest railway in Canada in terms of both revenue and the physical size of its rail network. The problem aims to determine the sequence of activities for each locomotive over the planning horizon. Besides assigning locomotives to scheduled trains and considering scheduled locomotive maintenance requirements, the problem also includes other decisions, such as sending locomotives to stations by deadheading, light traveling, and leasing of third-party locomotives. Based on an Integer Programming formulation and a Time-Expanded Network presented in the literature, we introduce a Rolling Horizon Approach (RHA) as a method to find near-optimal solutions of this problem in acceptable computing time. We adapt a mathematical formulation and a space-time network from the literature. We introduce a new type of arcs for the network and new constraints for the model to cope with issues arising when dividing the planning horizon into smaller ones. Computational experiments on real-life instances show the pros and cons of our algorithm when compared to an exact solution approach.

Keywords: Locomotive Routing Problem, Rolling Horizon Approach.

Contents

List of tables

List of figures

List of abbreviations

LAP	Locomotive Assignment Problem
LRP	Locomotive Routing Problem
RHA	Rolling Horizon Approach
IP	Integer Programming

Acknowledgements

Foremost, I would like to express the deepest appreciation to my supervisor Prof. Emma Frejinger and my co-supervisor Prof. Jean-François Cordeau for the continuous support of my Master's study and research, for their patience, motivation, enthusiasm, and immense knowledge. Their guidance not only helped me in all the time of studying and writing of this thesis, but also taught me how to work scientifically in my research career.

Besides my supervisor and my co-supervisor, I would like to thank Dr. Pedro Miranda, for his research that inspired me to begin doing my thesis, insightful explanations about the methodology, significant comments about all the issues, and challenging questions for me to work with.

My sincere thanks also go to Mr. Eric Larsen for helping me to access the real-life data of the Canadian National Railway Company.

Last but not the least, I would like to thank my family: my parents Mrs. Dang Thi Loc and Mr. Pham Hong Truong, for giving birth to me in the first place and supporting me spiritually throughout my life, my younger sister Ms. Pham Thu Trang, for her belief that I deserve to be her elder brother. In memory of my grandmother, I am extremely grateful for her love, prayers, and caring for both my childhood and my future.

Chapter 1

Introduction

Rail transportation is a product of the industrial era, playing a major role in economic development. A massive railway system implemented in North America contributed to it becoming one of the world's largest economies. Railroads play an essential role in transportation of people and goods over long distances not only because of their capacity to carry heavy loads but also because of their speed, safety, and relatively low environmental impact.

Canadian National Railway Company (CN) operates Canada's largest railway and is Canada's only transcontinental railway company. Its network spans Canada from the Atlantic coast in Nova Scotia to the Pacific coast in British Columbia across about 20,400 route miles (32,831 km) of track. For a Canadian Class I freight railway as CN, several thousand weekly trains are operated by using more than 2,000 locomotives, including both owned and leased ones. The operating and net income of CN was several billion Canadian dollars in 2019. Therefore, most of the planning and scheduling problems arising in railroads involve billions of dollars of resources annually.

Due to the vastness of the worldwide railway system, there are many challenging optimization problems arising. They were listed as large, relevant and complex issues in Ahuja et al. (2005a), including, e.g., *blocking problem, yard location problem, train scheduling problem, locomotive scheduling problem, maintenance planning problem, train dispatching problem,* and *crew scheduling problem.*

Among these problems, the *locomotive scheduling problem* consists in efficiently assigning different types of locomotives to the scheduled trains. This problem can be divided into two problems, namely, the *locomotive assignment problem* (LAP) and the *locomotive routing problem* (LRP). The first one is at the tactical level, where locomotives are classified into types based on their main physical characteristics, including horsepower, pulling capabilities, weight, number of axles, cost, among others. The second one arises at the operational level, where the activities of individual and uniquely identified locomotives must be specified in detail.

This thesis focuses on the LRP faced by CN and is an extension of the work of Miranda et al. (2020), who introduce a time-expanded formulation to find optimal solutions for the same problem. They propose an exact method to solve optimally 1-week instances of the LRP in an acceptable computing time (about 10-15 minutes). However, if the railway requires to consider a long-term plan such as 10-day or 2-week operational schedules, the problem becomes too hard to solve with the exact approach. Indeed the number of arcs and nodes in the space-time network and the number of variables in the integer programming model dramatically increase when expanding the planning horizon. This is a motivation for us to implement a heuristic rolling horizon method providing good quality solutions.

Inspired by the work of Miranda et al. (2020), this thesis makes the following contributions. First, we modify the space-time network, allowing us to present the LRP corresponding to each sub-instance created by dividing an instance into smaller overlapping time horizons. Maintenance deadlines of critical locomotives and the constraints related to them must be redefined to fit each sub-instance. The Rolling Horizon Approach (RHA) also requires to add a new type of arcs to the network to satisfy one of the most essential operational requirements - *train-to-train connections*, which can be broken when the real-life instances are divided into smaller fragments.

Second, we adapt and implement the IP formulation introduced in Miranda et al. (2020). The adaptations, in terms of both the IP model and the space-time network, are the study of the issues coming from constraints about critical locomotive maintenance deadlines and train-to-train connections. The conflicts, caused by those constraints, must be avoided not only by modifying the existing constraints but also by adding new constraints when switching from one planning horizon to another in the RHA.

Third, we apply the RHA to solve all sub-instances and then collect their solutions to create a final result for each real-life instance. We should note that several activities of the locomotives are not considered in the solution of each sub-instance if they start in a period of time called *overlap*. The overlap, which begins at the end of the roll period and ends at the end of the planning horizon, is a shared period of time of two consecutive planning horizons. The latest activity of each locomotive, which occurs before and finishes after the end of the roll period, will be recorded for later iterations in the RHA because it directly affects the input of the next planning horizons.

Finally, we analyze computational experiments of 7-day instances to observe all the effects of the overlap and the chosen planning horizons in the RHA. In case of longer time horizon instances, we study the quality of the solution provided by the RHA when compared with the exact IP-based method.

The rest of the thesis is organized as follows. In Chapter 2, we briefly discuss the relevant literature. Chapter 3 provides the problem description, while Chapter 4 describes in detail the space-time network and the IP model. Both of them are modified to adapt to the RHA framework. Extensive computational experiments are carried out and reported in Chapter 5. The detailed computational results are provided in Appendix A.

Chapter 2

Literature Review

We review several works that study the locomotive scheduling problem in Section 2.1. Two main categories of this problem, which are known as the locomotive assignment problem and the locomotive routing problem, are reviewed in Section 2.1.1 and Section 2.1.2 respectively. We discuss the Rolling Horizon Approach for decomposing large-scale IP models in Section 2.2.

2.1. The Locomotive Scheduling Problem

Because of the high cost of owning and operating locomotives, the *locomotive scheduling problem* plays an essential role in rail transportation. The question of making the most efficient use of the locomotives has challenged researchers for a few decades. This problem aims to satisfy pulling requirements of the scheduled trains while decreasing the cost of operating locomotives and obeying a variety of constraints, such as fleet-size constraints on different locomotive types, fueling, and maintenance constraints (Ahuja et al.; 2005b; Vaidyanathan et al.; 2008a). Because the problem of real size is large and complex, it is highly essential to employ Operations Research techniques and practical optimization tools to support locomotive scheduling decisions (Ortiz-Astorquiza et al.; 2019).

Studying the surveys of Cordeau et al. (1998) and, more recently, Piu and Speranza (2014), we understand that most models and algorithms focus on solving realistic versions of the LAP. This problem aims to determine a set of locomotive types assigned to each train, the way locomotives will deadhead or light travel in the network, and which train-to-train connections should be created. However, the result of the LAP can not be implemented directly in operation. Therefore, in the context of the locomotive scheduling problem, the LRP is an important problem raised by the railroads at the operational level. The LRP aims to decide detailed operating activities of each locomotive based on the result of the LAP. Despite being essential, publications focusing on the LRP are rather scarce.

2.1.1. The Locomotive Assignment Problem

At the tactical level, the LAP determines how to effectively assign the number of locomotives of each type to the scheduled trains while taking into account constraints on the fleet size for each locomotive type, power requirement for each train, compatibility between trains and locomotive types, balanced flow of locomotives through the railroad network. From now on, a set of locomotives assigned to each train in a given schedule is called a *consist*.

To balance the flow of locomotives through the network, the LAP considers how to move locomotives from stations where they are in surplus of power requirement to other ones with a shortage. Besides relocating locomotives by using them to pull the trains, locomotives may also be attached to the trains without actively pulling, which is called *deadheading*. They can also travel or pull others in a group of locomotives (without rail cars), which is called *light traveling* (Ahuja et al.; 2005b; Vaidyanathan et al.; 2008a). Comparing two types of re-positioning, we see that light traveling is more flexible than the deadheading, but it is more expensive due to the need of an operational crew.

Other important characteristics to incorporate in the LAP are *consist busting* and *train-to-train connections*. Consist busting is considered when a train arrives at its destination station. If the consist is busted, the corresponding locomotives become separately available at the station. However, busting is time-consuming and avoided when possible. It is not essential to re-assign consists for back-and-forth trains, or trains that have changed their ID in the given schedule but are physically the same. If two trains require the same set of locomotive types and the destination of the first train is the departure of the second train, the consist of the arriving train can be used for the departing one. In those cases, consist busting is replaced by train-to-train connection. As pointed out by Ahuja et al. (2005a), Ahuja et al. (2005b), Vaidyanathan et al. (2008a), the train-to-train connections can decrease train delays and crew time to decouple and move locomotives individually.

At the operational level, maintenance and fueling requirements of individual locomotives might reduce the locomotive availability over the planning horizon. Furthermore, in real life, planners are concerned with assigning locomotive units to trains, rather than locomotive types since the output of solving the LAP is not directly implementable. Thus, the LRP arises at the operational level to further refine and adjust the locomotive assignment plan. In this thesis, the LRP, an extension of the works of Ortiz-Astorquiza et al. (2019) and Miranda et al. (2020), determines the sequences of activities operated by each locomotive, such as pulling trains, taking part in train-to-train connections, light traveling, deadheading, and undergoing maintenance.

Among these activities, only pulling trains and deciding train-to-train connections are dependent on the consist assignment in the output of the LAP. The deadheads and light travels must be reassigned because of the impact of maintenance operations on the flow of individual locomotives over the network (Miranda et al.; 2020). The maintenance operations have a significant impact on the locomotive routing. Each locomotive must be maintained periodically. For example, locomotives have to pass through maintenance every 92 days in North America according to the law. Furthermore, the railroad companies may set deadlines to send locomotives to the shop for semi-yearly, yearly, or quadrennial major revisions and mechanical repairs. The locomotives cannot pull trains or travel on their own (i.e., light travel) if they missed their maintenance deadlines. In that case, they will be turned off and deadheaded to a shop for checkup (Bouzaiene-Ayari et al.; 2016). At each shop station, locomotives may be served right after arriving or have to wait in line for a spot since each shop has limited capacity and is frequently congested. An imbalance in power availability across the network may be created by locomotive unavailability. Therefore, railway companies prefer sending locomotives to the shops, neither too early nor too late, while maximizing their productivity.

In addition to classify problems based on planning levels as mentioned above, the LAPs and the LRPs may be distinguished by the number of pulling locomotives a train may require (Vaidyanathan et al.; 2008a). The problem can be modeled as a "single-locomotive model" if each train needs a single pulling locomotive. If some trains require more than one pulling locomotive, the problem falls into the "multiple locomotive model" category. In this thesis, the LRP is modeled by a "multiple locomotive model" because we must assign a set of locomotives (consist) to each train based on the consist provided by the result of the LAP.

Generally, "single-locomotive" problems are easier to solve than "multiple locomotive" ones. The single-locomotive model category can be divided into two types by considering how many locomotive types the model requires. Forbes et al. (1991) argue that the problem becomes similar to the single-depot bus (vehicle) scheduling problem if only one locomotive type is required. Otherwise, the problem can be considered as the multiple depot bus (vehicle) scheduling problem.

Wright (1989) implements three algorithms to solve the problem with multiple locomotive types: the first is a deterministic algorithm to find a feasible solution, the second is a local improvement method, and the third is a simulated annealing algorithm. Although the first valid solution for large scale version of multiple locomotive model is found, the author does not recommend the use of the procedure for real-life applications because the solution does not take into account the fleet size constraints.

Inspired by the work of Wright (1989), Forbes et al. (1991) introduce a model based on an integer linear program equivalent to a multi-commodity flow formulation. The model, where each commodity represents a locomotive type, obtains an exact solution for instances of 25-200 trains. It is a significant improvement over the method proposed by Wright (1989) because the model is able to take into account the fleet size constraints.

More recently, Fugenschuh et al. (2006) extend the work of Forbes et al. (1991) by adding several new aspects: cyclic train departures, time windows on start and arrival times, transfer of wagons among trains. Two linear integer programming problems with fixed and flexible start and arrival times are introduced. While the first model is solved to optimality by using CPLEX directly, a combination of a randomized parameterized greedy (PGreedy) heuristic and a special purpose reformulation to improve the formulation of the problem is implemented before using CPLEX to cope with the second one.

The most sophisticated version of the LAP (or the LRP) occurs when each train requires a consist instead of a single pulling locomotive. Florian et al. (1976) is the first to study this version of the problem. The authors consider a freight train problem arising at CN. It is formulated as a multi-commodity network flow problem. The objective is to minimize the capital investment and the maintenance costs while assigning a sufficient number of locomotives of different types to satisfy the motive power requirements of each train. A Benders decomposition method is proposed to deal with the weekly train schedule. However, their implementation can not solve the problem to optimality and can be acceptable for medium-sized problems but not for large ones.

Cordeau et al. (2000) propose an IP formulation to deal with the simultaneous assignment of locomotives and rail cars in the context of passenger transportation. It is formulated as a multi-commodity flow problem on a space-time network. In the network, each node represents an event, i.e., arrival, departure, and repositioning of a unit. Each arc performs an activity of scheduled train, such as operating, repositioning, and waiting. An exact algorithm, based on the Benders decomposition approach, is introduced to solve a set of nine instances obtained from VIA Rail Canada. Compared with three other solution methods, namely, Lagrangian relaxation, simplex-based branch-and-bound algorithm, and Dantzig–Wolfe decomposition, the authors show that the method based on Benders decomposition finds optimal solutions within a short computation time. However, this model is not sophisticated enough to be used in practice because it does not consider constraints that are important in practice. Thus, in Cordeau et al. (2001), it is modified by incorporating a much broader set of constraints and possibilities required in a commercial application.

The problem introduced in Cordeau et al. (2001) is an extended version of the one in Cordeau et al. (2000). A large-scale IP model and a heuristic approach based on column generation are introduced to solve this real-life problem. The solution, provided by applying a heuristic branch-and-bound method in which the linear relaxation lower bounds are computed by column generation, has been successfully implemented at VIA Rail. The algorithm can satisfy a long-term planning horizon and find a good quality solution in a few hours of computation.

Vaidyanathan et al. (2008a) introduce two formulations: *consist formulation*, and *hybrid formulation* for the *locomotive planning problem* arising at CSX Transportation (a Class I US Railroad). They consider an extension of the problem studied in Ahuja et al. (2005b), wherein all the real-world constraints are not incorporated to generate a fully implementable solution. In Vaidyanathan et al. (2008a), the authors not only add new constraints to the problem desired by locomotive directors, but also develop additional formulations to transition their solutions to practice. The major contribution of the paper is a new approach for routing consists instead of routing individual locomotives. The computational tests show that if good consists are created, even though the routing of consists seems to restrict the solution space, high-quality solutions are provided by the consist formulation and they are easier to implement for locomotive dispatchers. The consist formulation runs faster and is more robust than the hybrid formulation and the flow-based models in Ahuja et al. (2005b), while the objective value of the hybrid formulation is better (by 5%) than the one of the consist formulation (the flow-based models cannot converge to a feasible solution after 10 hours of running time).

Ortiz-Astorquiza et al. (2019) develop mathematical optimization models and Benders-based solution algorithms to find high-quality solution for the LAP arising at CN. Two IP formulations, namely Locomotives-Based Formulation (LBF) and Consist-and-Locomotive Flow-Based Formulation (CLF), are introduced. The results of extensive computational experiments show that the CLF model provides a good solution in reasonable running time when solved with a general-purpose solver. The performances of those formulations are also improved by two versions of an algorithm based on Benders decomposition. Those versions significantly reduce the CPU time to obtain a first solution. The solution analysis shows that the enhanced model provides a 25% reduction in the number of locomotives compared with the locomotive schedule used in actual operations. To sum up, the authors argue that the proposed model (CLF) is well-suited to provide good solutions for the LAP with significant cost and running time reduction.

2.1.2. The Locomotive Routing Problem

With the idea of solving a series of small overlapping instances extracted from the original problem, Ziarati et al. (1997) extend the work of Florian et al. (1976) and achieve a reduction of $7-7.5\%$ in both the number of locomotives in use and power consumption when studying a problem arising at CN. They define this problem as a LAP where each train requires sufficient power during its operation, and locomotives having maintenance requests must be sent to the shop within a given time limit. Due to the presence of maintenance constraints for individual locomotives, we can consider this problem as an LRP at the operational level. The authors introduce a multi-commodity network flow-based model and then decompose the model into a master problem and a set of sub-problems using the Dantzig-Wolfe decomposition technique. A branch-and-bound procedure, where the linear relaxation at each node is solved by column generation, is used to generate integer feasible solutions.

In Ziarati et al. (1999), the authors extend the work in Ziarati et al. (1997) with a cutting plane approach. First, a train whose demand is fulfilled by two locomotive types is selected. Then, a branching decision is imposed on forbidding the assignment of other locomotive types to this train. Finally, the appropriate cutting planes are added for this train to strengthen the formulation. They use a RHA in the computational experiments due to the large size of the instance, which contains thousands of trains and locomotives. By using cutting planes, the integrality gap is decreased by an average of 23%. Comparisons with CN's solutions show a reduction of 11 locomotives or, equivalently, a 1.1% saving.

To complement previous research in Vaidyanathan et al. (2008a) and Ahuja et al. (2005b), Vaidyanathan et al. (2008b) study the LRP problem faced by CSX Transportation, a major U.S. railroad company. This problem aims to determine paths of individual locomotives whose actions are limited by fueling and maintenance constraints. These constraints require each locomotive to visit a fueling station and a shop before a given number of miles of travel. The authors define the sequence of trains that the locomotive takes between fueling stops as a fuel string and the sequence of trains that the locomotive takes between servicing stops as a service string. Then, the enumerated paths are used as input parameters for an integer program that aims to decompose the locomotive assignment into flows on paths. This integer programming problem with millions of decision variables cannot be solved using a commercial solver. Therefore, they develop a fast aggregation-disaggregation based algorithm to solve this formulation within a few minutes. The computational experiments show a less than 2.2% optimality gap.

Bouzaiene-Ayari et al. (2016) introduce a suite of models, ranging from single and multi-commodity flow models to a multi-attribute model, which have been successfully implemented at Norfolk Southern. While the single and multi-commodity flow models can be solved using commercial integer programming solvers, the multi-attribute resource allocation model is solved by approximate dynamic programming (ADP). At the strategic level, all locomotives are considered at the same time in a single commodity flow model, wherein locomotives are grouped into a single type, and their flows (through the shop and foreign power) are approximated at an aggregate level. At the operational level, locomotives are grouped into four classes (high and low horsepower, high and low adhesion) as the input of the multi-commodity flow model. Small instances can be solved using CPLEX, but the dramatic growth in run times forces the authors to implement a multi-attribute resource allocation model when increasing the size of the data. By using ADP to solve the third model, they demonstrate how efficiently to implement ADP for both deterministic and stochastic models that capture locomotives and trains at a very high level of detail. Although ADP is suitable to handle high levels of details, it does not globally optimize the locomotives flow on the network over time.

Miranda et al. (2020) study the LRP arising at CN, wherein repositioning locomotives through deadheads, light travels, and leasing of third-party locomotives are considered besides maintenance operations. The authors propose an IP formulation to solve the integer multi-commodity flow problem with side constraints, based on a two-layer time-expanded network representation of the problem. In the IP model introduced in Miranda et al. (2020), regular locomotives are grouped into types, while critical locomotives are considered individually. By doing this, their model is reduced in size and can be solved optimally within reasonable computing times. As we show in Section 5.3, the method introduced in that paper can solve to optimality all instances with a planning horizon up to 10 days. However, computing times dramatically increase when expanding the planning period. For this reason, Miranda et al. (2020) inspires us to implement the RHA, with the purpose of decreasing the computing time and providing near-optimal solutions.

2.2. The RHA for Mixed Integer Programming

Mathematical programming, especially IP, has been widely used for modeling the scheduling and planning problems because of its flexibility and extensive modeling capability, as well as the existence of powerful off-the-shelf solvers. However, NP-hard problems remain challenging to solve. In this thesis, we are interested in using the mathematical formulation proposed by Miranda et al. (2020) for an optimization problem with a certain time horizon. It is a large-scale IP model and its size increases dramatically when the length of the time horizon increases. An efficient method to decompose this IP model and provide near-optimal solutions with more reasonable computing time is to solve smaller problem within a rolling horizon framework. In the following we provide a number of examples implementing the RHA to solve problems from various areas of application.

In the literature, the RHA is often used in manufacturing scheduling, as well as in other areas (see, e.g., the survey in Chand et al.; 2002). Here, the rolling horizons are implemented routinely to update or revise schedules, estimate a part of the future plan based on reliable and recent data. The basic idea is to repeatedly solve a MIP, which covers a short time horizon. An overlap between two consecutive short time horizons creates an opportunity to make a better overview in the future plan, and the result in this period does not count for the final result. The decision in the overlap will be re-optimized in order to adapt all changes in the coming manufacturing plan. When all days of the planning horizon have been considered in at least one MIP, the overall problem is completely solved.

Baker and Peterson (1979) argue the importance of using rolling schedules in production planning due to their representation of the practical means by which analysis is converted to action in dynamic problems. Even when the optimal solution is found for a long time horizon, that solution is seldom directly implemented in real-life without revision. When new information becomes available, it is more common to revise the planning and modify the previous solution to adapt to the changes in data. Baker (1977) was among the first to test the effectiveness of schedules obtained from a RHA. The author shows that the longer the planning horizon is, the better the rolling horizon performance of static models. In Baker and Peterson (1979), the authors prove that in almost cases, longer planning horizons provide monotonic improvements in performance, but tend to be more challenging to solve.

Several powerful algorithms for solving rolling horizon problems were introduced in Stauffer and Liebling (1997), Dimitriadis et al. (1997), Mercé and Fontan (2003), Araujo et al. (2007).

Since a long time horizon needs to be considered in the railway operation, it is not easy to obtain an optimal or even a near-optimal solution. To overcome this issue, the RHA is considered as an efficient way to solve the LRP in Ziarati et al. (1997) and Ziarati et al. (1999). That is, they divide the whole LRP into several stages, where each stage is only for train services in a short period. Only the plan for the near future is updated and executed at each stage. By this practice, a RHA, which helps decrease the computation time and provides a nearly optimal solution, is utilized to decompose the LRP.

Additionally, the RHA is also applied to the railway rescheduling area, wherein the railway dispatchers usually reschedule train services gradually, such as the high-speed train system. The literature on that subject is shown in the papers of Nielsen et al. (2012), Quaglietta et al. (2013), Pellegrini et al. (2014), and Zhan et al. (2016).

Nielsen et al. (2012) study real-time disruption management of railway rolling stock in the Netherlands. A generic framework is introduced to deal with disruptions of railway rolling stock schedules. They propose an online combinatorial decision problem, where a sequence of information updates represent the uncertainty of a disruption. To decompose the problem and to reduce the computation time, their RHA is described as follows: rolling stock decisions are only considered if they are within a specific time horizon from the time of rescheduling. The experimental results shows that the RHA can handle the rolling stock during a disruption with minor effects for the shunting plans. With the short computation times, the RHA is indicated as the right candidate for being used in the decision support system for rolling stock rescheduling.

Quaglietta et al. (2013) introduce a framework that couples the state-of-the-art dispatching system ROMA (Railway traffic Optimization by Means of Alternative graphs) developed by D'Ariano (2009) with the microscopic simulation model of railway networks, called EGTRAIN, in Quaglietta and Punzo (2013). A RHA is implemented to perform this integration. The authors also refer to different disturbed traffic scenarios created by sampling train entrance delays and dwell times within a typical Monte-Carlo scheme. The results of this study show that the instability of plans increases over time as stochastic disturbances propagate on the network independently from the length of the prediction horizon. Short horizons provide more solid plans in terms of reordering, but probably a lower performance in recovering delays. However, longer prediction horizons focus on reordering to reduce constant delays but make plans more unstable concerning the reordering.

Pellegrini et al. (2014) propose a mixed-integer linear programming formulation for tackling the real-time railway traffic management problem arising in two control areas in France. This formulation can model either the route-lock sectional-release interlocking system (SR) or the route-lock route-release one (RR) by applying two different alternative objective functions. Following that, they introduce SR and RR formulation, respectively. For both alternative formulations, a rolling-horizon framework is implemented to perform subsequently for scheduling and routing trains during a long time horizon. In this framework, the time interval for a single optimization advances throughout the day. Besides, previously made decisions can be either modifiable or not in order to ensure the compatibility of two decisions made in two consecutive time intervals.

Zhan et al. (2016) study the high-speed train rescheduling problem where one track of a double-track train is temporarily unavailable. Three MILP models are introduced to formulated three practical train rescheduling strategies, and then they are solved by a RHA. The results obtained by the RHA are compared with those obtained by the centralized approach. Their analysis shows that the gaps between the solutions provided by the RHA and those achieved by the centralized approach are relatively small for disruption instances with short durations. The most significant gap is around 20%, but the average gaps are smaller than 3%. For disruption instances with longer durations, the gaps are not far from the gaps mentioned above. This study proves that the disposition timetables solved by the RHA are near-optimal, and the RHA is quite efficient in obtaining a reasonable disposition timetable.

Expanding the work in Lai et al. (2008a), Lai et al. (2008b) introduce a rolling horizon model to cope with the aerodynamic efficiency of intermodal freight trains with uncertainty. A static model, formulated as a MIP, is developed to optimize the load placement on a sequence of intermodal scheduled trains. The authors also implement a dynamic model, which is a modification of the static model with exponentially decreasing weights assigned to the objective functions of future trains, to account for incomplete or uncertain information on later trains and incoming loads. A rolling horizon scheme, in which future trains are considered simultaneously with the current train, is used to deal with the challenge coming from the dynamic model. The experimental result of the dynamic model under the performance of the rolling horizon framework is analyzed based on two simulations: a terminal with a uniform arrival rate of incoming loads and a terminal with a nonuniform arrival rate of incoming loads. For the first simulation, the rolling horizon provides solutions close to the known optimum (relative optimality gap between $0.1\% - 3\%$) after 600 CPU seconds. The results of the second simulation are largely similar to uniform operations, and the rolling horizon performs an 8.6% benefit equivalent to approximately 700,000 gallons of fuel savings per year.

Samà et al. (2013) show that the RHA is also efficient in aircraft scheduling. They consider the real-time problem of scheduling aircraft in two major Italian Terminal Control Areas where airborne decisions need to be taken in given time horizons of traffic prediction. An alternative graph formulation formulates their problem, and a RHA is implemented to control busy traffic situations when a large number of aircraft are delayed. A Branch-and-Bound algorithm (BB) is compared with a First Come First Served rule presenting the dispatchers' behaviors in real-life operation. In terms of delay and travel time minimization, the RHA solved by BB provides better results than First Come First Served rule. Moreover, BB requires fewer changes of aircraft schedule than First Come First Served during consecutive look-ahead periods. Therefore, the result of BB is more stable and easier to implement in operation than First Come First Served's one. In terms of computing time, the rolling horizon configurations with both methods are fast to provide feasible solutions for one-hour instances. The authors also compare the performance of various multi-stage configurations of the RHA with the centralized approach (only one single stage). BB is implemented as a scheduler in both approaches. The solution analysis shows that the BB-based rolling horizon approach run ten times faster than the centralized approach. For both Terminal Control Areas, the RHA provides a smaller number of the maximum consecutive delay (over 40% less), while the centralized one is not able to find better solutions even if it runs ten times slower.

Chapter 3

Problem Definition

In this chapter, we describe the locomotive routing problem faced by CN, as well as real-life instances generated from CN's historical database and used in Miranda et al. (2020). In the context of this thesis, the RHA requires to divide a large instance into smaller ones. Therefore, we must implement several adaptations to cope with the lack of information when extracting sub-instances from the original ones.

3.1. Overview

Based on a schedule of trains, the LAP is the tactical level of the process that aims to determine the minimum cost assignment of locomotive to trains. The LAP satisfies several operational constraints, such as the number of locomotives of each type to assign to each train, the fleet size for each locomotive type, the pulling-power requirements for each train, and, if necessary, the operation mode (DP or conventional) of each train. The LAP is presented in detail in Ortiz-Astorquiza et al. (2019). After solving the LAP, the railroad has to determine the sequence of trains to which each locomotive is assigned. The output of the LAP is not directly implementable because the LAP does not consider, e.g., fueling constraints and servicing constraints so that the railroad needs an operational level that takes into account identified locomotives assigned to each train to satisfy them. The LRP considers these constraints.

In this thesis, the RHA is implemented to deal with the LRP. The overall time horizon is divided into smaller time horizons with different start times. The gap between two consecutive start times is called *roll period* and is denoted by r. For each time horizon, a single-stage optimization problem is solved to obtain a plan based on the historical and currently available information on the operational conditions.

Figure 3.1 shows how the RHA works. The train schedule of t days is divided into three planning horizons of the same size, where each planning horizon only contains train events in the period h. Following the time axis, two consecutive stages start at $\mathcal{T}$ and $\mathcal{T} + r$,

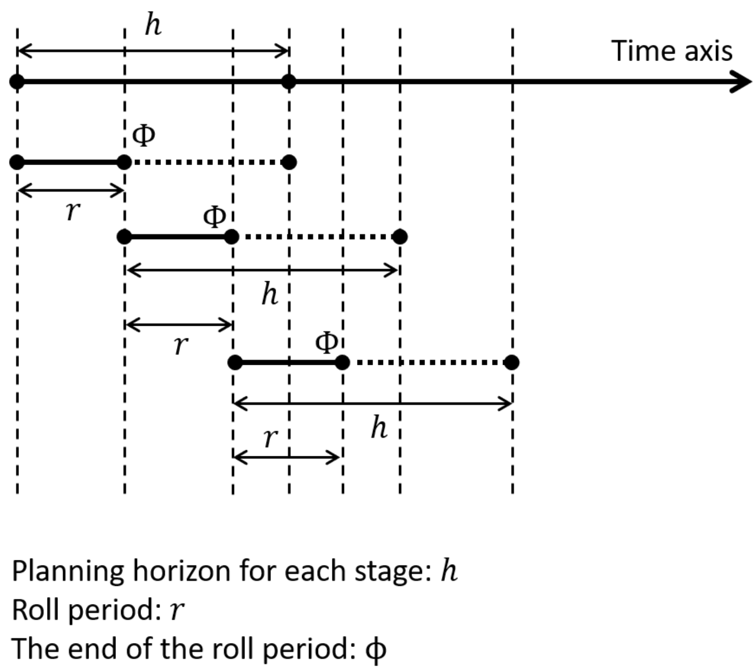

Fig. 3.1. Example of RHA illustrations and the notations

respectively, where $\mathcal{T}$ is the beginning of the earlier planning horizon. The overlap is the amount of time corresponding to the period $(h - r, h)$. Assume that ϕ is the end of a roll period, $(\phi-)$ is a time attribute less than the end of roll period, and $(\phi+)$ is a time attribute greater than the end of roll period. When the whole time horizon is divided into shorter ones, the input of each planning horizon in the RHA needs to contain all information of the corresponding part in original data, in addition to relevant information from the previous period.

3.2. Data & Constraints

In this section, we provide the details of the input given by CN for the LRP with the assumption that the LAP solutions are taken from the railroad or by using the method introduced in Ortiz-Astorquiza et al. (2019). Besides the information used in LRP, we also present how to create the input for the RHA.

Locomotives Data: The basic information of each locomotive contains: ID, type, horsepower, and weight. At the beginning of the week, the locomotives' status and location are given. If the locomotive is pulling the train, light traveling or deadheading passing the beginning of the week, its location is unknown. The locations are determined for locomotives grounded in the yards or maintained in the shops. We are also given the set of *critical locomotives* due to maintenance in the current week. For each critical locomotive, we are provided a type of maintenance request determining how long this locomotive will be maintained and the maintenance deadline.

From the second iteration of the RHA to the last one, the information about locomotives' status and location at the beginning of the planning horizon needs to be taken from the result of the previous iteration. The stations where locomotives are located and locomotives' status will be updated based on their last activities starting before the end of the roll period (see Section 4.3.4). Furthermore, the critical locomotives can be redefined by comparing their maintenance deadline with the end of the planning horizon. If critical locomotives' deadlines are greater than the end of the planning horizon, they are changed to *"temporary regular locomotives"*, which have no maintenance request. By this change, the railroad not only avoids maintaining critical locomotives too early before the deadlines but also decreases the number of leased locomotives.

Train Data: The train data contains scheduled trains within a weekly information on each train. For each train in the schedule, we are given basic information such as ID, its departure time and arrival time, its origin and destination station, tonnage, horsepower per tonnage factor. Additionally, we know the set of stations in the route of each train and the distance between each pair of stations. The output of the LAP is also given a set of train-to-train connections that must be satisfied during the week. The train-to-train connection is a combination of two trains sharing the same consist and the destination station of the first train is the origin station of the second one. Furthermore, from the result of the LAP, the type and the number of locomotives per type required to pull each train is specified as an input of the operational level.

When implementing the RHA for solving the LRP, we have to extract a part of the train schedule and train-to-train connections fitting each planning horizon. All trains departing during the planning horizon and all train-to-train connections containing both trains in this period are taken from the original schedule. The train-to-train connections including the second train out of this period will be considered later (see Section 4.3.1).

Station Data: Besides the basic information as ID, name, distance to each other station in the railroad network, each station requires a number of locomotives of each type by the end of the week in order to preserve the power demand at the beginning of the next week.

We are also given a set of stations providing maintenance service. Each station in this subset, called *shop*, can maintain an exact number, called *capacity*, of locomotives at the same time.

When implementing RHA, the capacity of the shops must be updated at the beginning of each planning horizon based on the data of locomotives. We are also given the set of stations which can connect to each other for light traveling. The locomotives cannot light travel from or to the stations, which are out of this subset.

Cost Data: Several cost parameters used at the LRP are specified by the operational cost data in real-life, such as ownership cost, fuel consumption, track maintenance, locomotive crew cost, maintaining cost of critical locomotives, and the fixed cost per light travel. The ownership cost presents the weekly spending of owning a locomotive, even if it is not used. The fuel consumption depends on the locomotive type, train fueling consumption rate, and traveling distance. The track maintenance is affected by the locomotive's weight and the distance it travels. The locomotive crew cost presents human resources spent on operating a locomotive, depending on how long and how far it is traveled by the crew. The maintenance cost is associated with the time required to maintain a critical locomotive, affected by the locomotive's type. The fixed cost per light travel is added to each light arc not only to perform how expensive when re-positioning a locomotive on that arc but also to distinguish a light travel to a deadhead.

In Miranda et al. (2020), the leasing cost is implicitly being charged on the arcs leased locomotives traverse. Note that in all arcs, the authors charge an ownership cost, which is the amount to pay for having the locomotive. In the case of a leased locomotive, the model will own it for the entire planning horizon, so they have to pay the cost of having it from time 0 (i.e., the beginning of the week) to the end of the last operation assigned to it. When a locomotive flows through some arcs of the graph, a cost that already includes a small "portion" for owning it during the current planning horizon is paid. Therefore, even if the authors set the fixed leasing cost to zero, the model tries to use the least possible number of leased locomotives to save on the ownership cost. In practice, we observe that to reduce the number of leased locomotives in the RHA, the leasing cost should be greater than zero, and its value depends on the length of the planning horizon.

Decisions: In Ortiz-Astorquiza et al. (2019), the authors not only consider determining train consists, locomotives repositioning over the network, and train-to-train connections but also take into account the operation mode of each train. The operation mode of a train is defined based on the position of locomotives pulling it. If all locomotives travel together at the head of a train, the train operates in conventional mode. If locomotives are interspersed throughout the length of a train, the train operates under distributed power mode. Additionally, the authors also consider several side-constraints and company's preferences in order to satisfy the requirement arising in practice.

Based on the result of the LAP, decisions made by the LRP ensure that the different side-constraints and company's preferences are met. Consequently, they do not need to be considered again at the operational level. For instance, if the result of the LAP decides to operate a train under distributed power, a consist is chosen considering locomotives types that are appropriately equipped to do so. Then, the LRP assigns locomotives units of the selected types to this train in order to satisfy distributed power. To sum up, the sequence of trains each locomotive should operate is determined, while considering locomotive maintenance and a balance flow of locomotives through the space-time network so as to satisfy a given train schedule at a minimum cost.

Objective: The LRP ensures that all scheduled trains and train-to-train connections must be served on time and assigned precisely consists provided by the LAP. Besides, critical locomotives should be maintained as much as possible without exceeding the capacities of shop stations. The problem also aims to minimize the number of deadheads, light travels, and leased locomotives while keeping the network balance. The objective function contains the cost of actively pulling scheduled trains, which is modeled as a function of track maintenance, fueling consumption and ownership costs; the cost of deadheads, which is a function of track maintenance and ownership costs; the cost of light travels, which is considered as a function of track maintenance, ownership, fuel consumption and crew costs; the cost of idling (ownership cost) locomotives at the stations, and the cost of moving critical locomotives through shops, which is calculated as a function of maintenance and the ownership costs.

To use the RHA to solve the LRP, we have to update activities of locomotives when switching from a small planning horizon to the next one. We also deal with all conflicts caused by maintenance requirements and train-to-train connections mentioned in Chapter 4.

Constraints: At the operational level, the LRP considers each critical locomotive and groups of regular locomotives to determine their route over the current week. Therefore, the LRP faces both constraints for individual locomotives and the ones for groups of locomotives. Besides minimizing the total of costs mentioned in the previous section, the LRP must satisfy all following constraints:

- Assigning exactly type and number of locomotives per type to operate trains on schedule.

- Preserving combinations of locomotives in train-to-train connections.

- Sending critical locomotives to the shop without exceeding shop capacities.

- Limiting the number of locomotives, both active and deadhead, attached to a train.

- Imposing a limit on the distance of each light travel.

- Guaranteeing supply of locomotives at the station at the end of each planning horizon to satisfy the next schedule.

- Obeying the light traveling rule that the locomotives only light travel between a given set of stations.

Chapter 4

Model and Solution Method

In Section 4.1, we present the two-layer space-time network provided by Miranda et al. (2020). The network, which allows us to manage and separate the flow of regular and critical locomotives, is described in Section 4.1.1 and Section 4.1.2. Although the network describes the problem over the short planning horizon, it does not take into account the train-to-train connections wherein two trains do not operate in the same period of time. A new type of arcs is introduced in the Section 4.3.1 in order to solve that issue. The IP model of Miranda et al. (2020) is introduced in Section 4.2, while Section 4.3.2 presents how we modify the IP model when adding new arcs to the network. The RHA framework is implemented in Section 4.3.3. In this section, we study some conflicts arising when dividing the whole planning horizon into smaller ones. Several constraints must be slightly modified to cope with those conflicts. Therefore, a new IP model, which is used in the RHA framework, is presented. Finally, Section 4.3.4 considers the way to update the information of locomotives when switching from a planning horizon to the next one in the RHA.

4.1. The Existing Space-time Network

We describe the two-layer space-time network introduced in Miranda et al. (2020) to describe the physical railroad over time. Let $G = \{\mathcal{N},\mathcal{A}\}$ be the graph where $\mathcal{N}$ is the set of nodes presenting the events at the stations and $\mathcal{A}$ is the set of arcs presenting activities of locomotives, such as pulling trains, grounding at the stations, going to the shops for maintenance, deadheading or light traveling. Each node is defined by type (source node, sink node, departure node, arrival node, outpost node), time, and place.

The primary purpose of the two-layer space-time network is to deal with the maintenance operation of critical locomotives. When deciding to send a locomotive to the shop, we must provide a way to maximize locomotive utilization, to meet the maintenance appointment, and to avoid exceeding the shop's capacity. Because critical locomotives might be attached to a train that takes them far away from shop locations or locomotives have to stand in line at the shop for a spot, or they have to be sent to other shops with available capacities for being served, CN allows critical locomotives to be maintained after their maintenance deadline. In this situation, the locomotives are called *overdue* and cannot be used for pulling trains or light traveling requiring locomotives' power. The overdue locomotives have to wait at the shop until spots are available for them or are sent to other shops by using deadheads. In the graph, they are sent to the top layer to do these actions to separate their flow and the flow of non-overdue locomotives.

While the bottom layer contains all arc types, such as train arcs, train-to-train arcs, deadhead arcs, light travel arcs, shop arcs, and ground arcs, the top layer does not include train arcs, train-to-train arcs, and light travel arcs. Only the critical locomotives that have not missed their deadline and the regular locomotives flow in the bottom layer. An overdue locomotive missing the maintenance appointment will be sent to the top layer right after finishing the last activity crossing the deadline. In practice, only legacy train arcs are allowed to pass the critical locomotives' maintenance deadline in the bottom layer.

4.1.1. Bottom Layer

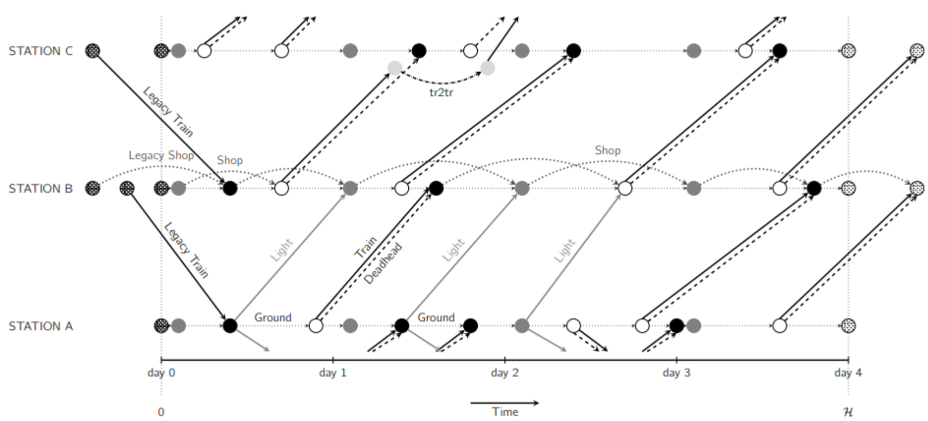

Fig. 4.1. Example of the bottom layer in a space-time network with 3 stations. Source: Miranda et al. (2020)

Figure 4.1 presents an example of the bottom layer in the space-time network with three stations. Let $\mathcal{N}^b$ and $\mathcal{A}^b$ be the set of nodes and arcs in the bottom layer. We denote by $\mathcal{N}_d^b$ (white nodes in Figure 4.1), $\mathcal{N}_a^b$ (black nodes in Figure 4.1), $\mathcal{N}_o^b$ (dark grey nodes in Figure 4.1), $\mathcal{N}_{sr}^b$ (hatched nodes in Figure 4.1), $\mathcal{N}_{si}^b$ (dotted nodes in Figure 4.1) the set of departure nodes, arrival nodes, outpost nodes, source nodes, and sink nodes in the bottom layer, respectively. A train arc in the set of train arcs $\mathcal{A}_t^b$ in the bottom layer connects a departure node and an arrival node. Each departure node contains the origin station of the train and the time attribute given by the train's departure time minus the amount of time required to build consist. Similarly, each arrival node represents the train's destination and the time attribute given by the train's arrival time plus the time required to bust consist.

At the beginning of the planning horizon, each station is presented by a source node with a time attribute set to 0. We partition the set $\mathcal{N}^b_{sr}$ into a set of initial nodes $\mathcal{N}^b_i$ acting as sources of available locomotives at the associated stations and a set of nodes $\mathcal{N}^b_{sr} \backslash \mathcal{N}^b_i$ related to legacy arcs. Similarly, each station is also represented by a sink node with a time attribute set to the end of the planning horizon. Let $\mathcal{N}^b_f$ and $\mathcal{N}^b_{si} \backslash N^b_f$ be set of final nodes acting as the sinks of available locomotives at the end of the planning horizon and the set of nodes associated with actions that cross the end of the planning horizon. In order to ensure that there exists at least one shop or light travel opportunity, outpost nodes are created at each station at the beginning of each day. From these nodes in the set $\mathcal{N}_o = \mathcal{N}^b_o \cup \mathcal{N}^t_o$, we can decide to move locomotives to the shops, or light travel to other stations, or stay grounding.

From now on, we assume that nodes at each station are sorted chronologically by their time attribute, and no pair of nodes at the same station has the same time attribute.

The set $\mathcal{A}^b$ is partitioned into small subsets: train arcs $\mathcal{A}^b_t$ (solid black arcs in Figure 4.1), train-to-train arcs $\mathcal{A}^b_{t2t}$ (dashdotted arcs in Figure 4.1), deadhead arcs $\mathcal{A}^b_{dh}$ (dashed arcs in Figure 4.1), light travel arcs $\mathcal{A}^b_l$ (solid gray arcs in Figure 4.1), ground arcs $\mathcal{A}^b_g$ (dotted black arcs in Figure 4.1), and legacy arcs $\mathcal{A}^b_{leg}$ (solid black arcs for legacy train arcs and dotted gray arcs for legacy shop arcs in Figure 4.1). As aforementioned, a train arc representing a scheduled train connects a departure node and an arrival node. Along the route of the trains, there exist some intermediate stations where the trains can pick up and release locomotives on their trips. The deadhead arcs in the set $\mathcal{A}^b_{dh}$ are created to connect the origin stations of the trains and the intermediate stations, or pairs of intermediate stations, or the intermediate stations to the destinations of the trains. The locomotives traversing deadhead arcs do not provide motive power. For RHA, if deadhead arcs cross the end of the roll period, they will be updated as a legacy train arcs (see Section 4.3.4).

Besides deadhead arcs, using light travel arcs in set $\mathcal{A}^b_l$ is also an effective way of repositioning locomotives across the railroad. As mentioned in Section 3.2, only a small subset of stations is available for locomotives to light travel among its elements. This is based on the fact that the railroad does not expect locomotives to do light traveling with a long distance. It is an expensive operation consuming track capacity and does not generate any revenue because the locomotives travel by themselves without pulling trains or being attached to railcars. The way to update the status of light travel arcs passing the end of the roll period is similar to the one used for deadhead arcs. In this thesis, we implement the idea used to generate light arcs, as mentioned in Miranda et al. (2020).

At the tactical level, one of the essential aspects to consider when solving the LAP is train-to-train connections. To deal with that, at the operational level, each train-to-train arc in set $\mathcal{A}^b_{t2t}$ is generated to connect the arrival node corresponding to the first train and the departure node corresponding to the second train. The flow of locomotives must be preserved along the path: the first train arc - train-to-train arc - the second train arc. From now on, let the *first train arc* and the *second train arc* present two trains belonging to a train-to-train connection. Each train-to-train arc is attached a label equal to the index of the second train in this connection in order to distinguish it between the set of all train-to-train arcs.

Shop arcs, $\mathcal{A}^b_{sh}$, are created based on scheduled maintenance operations for critical locomotives. The flow of a critical locomotive on a shop arc presents its visit to a shop, where a specific inspection is carried out. The frequency, length, and cost of maintenance depend on types of a maintenance request, namely, $standard$, $semi-yearly$, $yearly$, and $quadrennial$. Although the railroad aims to serve critical locomotives before their deadline, they allow serving critical locomotives after their deadlines because of the reason mentioned at the beginning of this section. The way to generate the shop arcs is presented in detail in Miranda et al. (2020).

Ground arc, $\mathcal{A}^b_g$, presents locomotives idling at the stations. These locomotives can be waiting for light travel opportunities, standing in the line for spots in shops, or be available at given stations. We should note that nodes at each station are sorted chronologically by their time attribute. Ground arcs are created between each pair of consecutive nodes, starting from the initial node until the final node is reached.

Finally, we consider the set of legacy arcs $\mathcal{A}^b_{leg}$ presenting unfinished activities starting from the previous planning horizon and finishing within the current one. This set is partitioned into $\mathcal{A}^b_{t^-}$ and $\mathcal{A}^b_{sh^-}$. The tails of all legacy arcs are always sink nodes, whose time attributes are always negative. The head node of a legacy shop arc is set as the first node at the shop station with a time attribute greater than or equal to the end of the inspection.

4.1.2. Top Layer & Connecting Layers

The top layer is built by making a copy of elements of the bottom layer except for train arcs, train-to-train arcs, and light travel arcs. Let $\mathcal{N}^t_a$, $\mathcal{N}^t_d$, $\mathcal{N}^t_o$, $\mathcal{N}^t_{sr}$, and $\mathcal{N}^t_{si}$ be sets of arrival nodes, departure nodes, outpost nodes, source nodes and sink nodes in the top layer,

Sets	Definition
$\mathcal{N}^b$	Set of nodes in the bottom layer
$\mathcal{N}^t$	Set of nodes in the top layer
$\mathcal{N}$	Set of all nodes ($\mathcal{N} = \mathcal{N}^b \cup \mathcal{N}^t$)
$\mathcal{N}_d$	Set of departure nodes ($\mathcal{N}_d = \mathcal{N}_d^b \cup \mathcal{N}_d^t$)
$\mathcal{N}_a$	Set of arrival nodes ($\mathcal{N}_a = \mathcal{N}_a^b \cup \mathcal{N}_a^t$)
$\mathcal{N}_o$	Set of outpost nodes ($\mathcal{N}_o = \mathcal{N}_o^b \cup \mathcal{N}_o^t$)
$\mathcal{N}_{sr}$	Set of source nodes ($\mathcal{N}_{sr} = \mathcal{N}_{sr}^b \cup \mathcal{N}_{sr}^t$)
$\mathcal{N}_{si}$	Set of sink nodes ($\mathcal{N}_{si} = \mathcal{N}_{si}^b \cup \mathcal{N}_{si}^t$)
$\mathcal{N}_i$	Set of initial nodes ($\mathcal{N}_i = \mathcal{N}_i^b \cup \mathcal{N}_i^t$)
$\mathcal{N}_f$	Set of final nodes ($\mathcal{N}_f = \mathcal{N}_f^b \cup \mathcal{N}_f^t$)
$\mathcal{A}^b$	Set of arcs in the bottom layer
$\mathcal{A}^t$	Set of arcs in the top layer
$\mathcal{N}$	Set of all nodes ($\mathcal{N} = \mathcal{N}^b \cup \mathcal{N}^t$)
$\mathcal{A}_t$	Set of train arcs ($\mathcal{A}_t = \mathcal{A}_t^b$)
$\mathcal{A}_{t2t}$	Set of train-to-train arcs ($\mathcal{A}_{t2t} = \mathcal{A}_{t2t}^b$)
$\mathcal{A}_{dh}$	Set of deadhead arcs ($\mathcal{A}_{dh} = \mathcal{A}_{dh}^b \cup \mathcal{A}_{dh}^t$)
$\mathcal{A}_l$	Set of light arcs ($\mathcal{A}_l = \mathcal{A}_l^b$)
$\mathcal{A}_{sh}$	Set of shop arcs ($\mathcal{A}_{sh} = \mathcal{A}_{sh}^b \cup \mathcal{A}_{sh}^t$)
$\mathcal{A}_g$	Set of ground arcs ($\mathcal{A}_g = \mathcal{A}_g^b \cup \mathcal{A}_g^t$)
$\mathcal{A}_{leg}$	Set of legacy arcs ($\mathcal{A}_{leg} = \mathcal{A}_{leg}^b \cup \mathcal{A}_{leg}^t$)
$\mathcal{A}_{t^-}$	Set of legacy train arcs ($\mathcal{A}_{t^-} = \mathcal{A}_{t^-}^b \cup \mathcal{A}_{t^-}^t$)
$\mathcal{A}_{sh^-}$	Set of legacy shop arcs ($\mathcal{A}_{sh^-} = \mathcal{A}_{sh^-}^b$)
$\mathcal{A}_{up}$	Set of upward inter-layer arcs
$\mathcal{A}_{down}$	Set of downward inter-layer arcs
$\mathcal{A}_y$	Set of inter-layer arcs ($\mathcal{A}_y = \mathcal{A}_{up} \cup \mathcal{A}_{down}$)
$\mathcal{A}$	Set of all arcs ($\mathcal{A} = \mathcal{A}^b \cup \mathcal{A}^t \cup \mathcal{A}_y$)

Table 4.1. Sets of nodes and arcs in the time-space network

respectively. Likewise, let $\mathcal{A}_{dh}^t$, $\mathcal{A}_g^t$, $\mathcal{A}_{sh}^t$, and $\mathcal{A}_{leg}^t$ be set of deadhead arcs, ground arcs, shop arcs, and legacy arcs in the top layer.

We also introduce set $\mathcal{A}_y$ as the set of the arcs connecting two layers. These arcs, called *inter-layer arcs* in Miranda et al. (2020), are used to send the overdue locomotives to the top layer and return them to the bottom layer for operation after servicing. In the RHA, when there are transitions from planning horizons to others, critical locomotives might be overdue by the end of each roll period. They would be placed on the top layer at the beginning of the upcoming planning horizon. However, it is not essential to make a copy of all legacy arcs in the bottom layer when building the top one. For example, we can remove legacy shop arcs from the top layer because the locomotives traversing legacy shop arcs can be considered as “nearly” regular ones. By doing this, we do not need to add *downward* arcs created to move locomotives from the top layer to the bottom layer after finishing their inspections.

In the case of starting and finishing the locomotive’s maintenance within the current planning horizon, we add a downward arc whose tail corresponds to the head of the associated shop arc in the top layer and the head in the bottom layer with identical type, place and time attributes. Conversely, an *upward* arc is created to connect a tail node in the bottom layer and a head node in the top layer. While the tail of the arc associates with the head node of the last event at the station, the head node is a copy of the tail with the same type, location, and time attribute but in the top layer.

4.2. The Existing Mathematical Formulation

Variables	**Definition**
r_{kl}	Number of regular locomotives of type k that traverses arc l
c_{vl}	Equals 1 if the critical locomotive v traverses arc l, 0 otherwise
u_{ki}	Number of leased locomotives of type k supplied by source i

Table 4.2. Decision variables in mathematical formulation

In this section, we provide a mathematical model built upon the space-time network in the previous section. The model introduced in Miranda et al. (2020) is formulated as an IP problem and repeatedly used in the RHA. The model solves the integer multi-commodity flow problem with side constraints to determine which regular and critical locomotives flow on the arcs of the graph. The problem faced by CN considers thousands of locomotives and trains per week. It is hence not possible to consider each locomotive as one commodity of the graph. The regular locomotives are grouped into types similarly in the LAP to decrease the number of variables in the mathematical model, while the critical locomotives are still treated individually. Because of this reduction, the problem is solved to optimality within an acceptable computing time. According to Miranda et al. (2020), it is essential to apply a flow decomposition algorithm introduced by Ahuja et al. (1993) to extract paths for individual regular locomotives while the route of each critical locomotive is presented by the network flow of the commodity associated with it.

The formulation proposed in Miranda et al. (2020) is presented as follows with all sets, parameters, and variables as summarized in the Tables 4.1-4.4.

Objective function:

$$\min \sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{A}_t}\gamma_{kl}(r_{kl}+\sum_{v\in\mathcal{V}_k^C}c_{vl})+\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{A}_{dh}}\gamma_{kl}(r_{kl}+\sum_{v\in\mathcal{V}_k^C}c_{vl})+\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{A}_l}\gamma_{kl}(r_{kl}+\sum_{v\in\mathcal{V}_k^C}c_{vl})$$
$$+\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{A}_g}\gamma_{kl}(r_{kl}+\sum_{v\in\mathcal{V}_k^C}c_{vl})+\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{A}_{t2t}}\gamma_{kl}(r_{kl}+\sum_{v\in\mathcal{V}_k^C}c_{vl})+\sum_{v\in\mathcal{V}^C}\sum_{l\in\mathcal{A}_{sh}(v)}\gamma_{vl}c_{vl}$$
$$+M\sum_{v\in\mathcal{V}^C}(1-\sum_{l\in\mathcal{A}_{sh}(v)}c_{vl})+L\sum_{k\in\mathcal{K}}\sum_{i\in N_i^b}u_{ki} \tag{4.2.1}$$

subject to:

$$r_{kl}=\eta_{kl}^R \qquad \forall k\in\mathcal{K},\forall l\in\mathcal{A}_{t^-}^b\cup\mathcal{A}_{sh^-}^b \tag{4.2.2}$$

$$c_{vl}=\eta_{vl}^C \qquad \forall v\in\mathcal{V}^C,\forall l\in\mathcal{A}_{t^-} \tag{4.2.3}$$

$$\sum_{l\in\mathcal{O}_k[i]}r_{kl}=\lambda_{ki}^R+u_{ki} \qquad \forall k\in\mathcal{K},\forall i\in\mathcal{N}_i^b \tag{4.2.4}$$

$$\sum_{l\in\mathcal{O}_v[i]}c_{vl}=\lambda_{vi}^C \qquad \forall v\in\mathcal{V}^C,\forall i\in\mathcal{N}_i \tag{4.2.5}$$

$$\sum_{i\in\mathcal{N}_{si}^b}\sum_{l\in\mathcal{I}_k[i]}r_{kl}=\sum_{i\in\mathcal{N}_{sr}^b}\sum_{l\in\mathcal{O}_k[i]}r_{kl} \qquad \forall k\in\mathcal{K} \tag{4.2.6}$$

$$\sum_{i\in\mathcal{N}_{si}}\sum_{l\in\mathcal{I}_v[i]}c_{vl}=1 \qquad \forall v\in\mathcal{V}^C \tag{4.2.7}$$

$$\sum_{l\in\mathcal{I}_k[i]}r_{kl}=\sum_{l\in\mathcal{O}_k[i]}r_{kl} \qquad \forall k\in\mathcal{K},\forall i\in\mathcal{N}_d^b\cup\mathcal{N}_a^b\cup\mathcal{N}_o^b \tag{4.2.8}$$

$$\sum_{l\in\mathcal{I}_v[i]}c_{vl}=\sum_{l\in\mathcal{O}_v[i]}c_{vl} \qquad \forall v\in\mathcal{V}^C,\forall i\in\mathcal{N}_d\cup\mathcal{N}_a\cup\mathcal{N}_o \tag{4.2.9}$$

$$\sum_{l\in\mathcal{A}_{sh}(v)}c_{vl}\le 1 \qquad \forall v\in\mathcal{V}^C \tag{4.2.10}$$

$$\sum_{l \in \mathcal{A}_{up}(v)} c_{vl} = 1 - \sum_{l \in \mathcal{A}^b_{sh}(v)} c_{vl} \qquad \forall v \in \mathcal{V}^C \tag{4.2.11}$$

$$c_{vl'} = \sum_{l=(i,j) \in \mathcal{A}^t_{sh}(v)} c_{vl} \qquad \forall v \in \mathcal{V}^C, l' = (j,h) \in \mathcal{A}_{down}(v) \tag{4.2.12}$$

$$\sum_{v \in \mathcal{V}^C} \sum_{\substack{l=(j,h) \in \mathcal{A}_{sh}(v): \\ p_j = s \\ t_i - \tau_{m_v} < t_j \leq t_i}} c_{vl} \leq \zeta_s - \sum_{\substack{l=(j,h) \in \mathcal{A}_{sh^-}: \\ p_h = s \\ t_h > t_i}} \sum_{k \in \mathcal{K}} \eta^R_{kl} \qquad \forall s \in Sh, i \in \mathcal{N}^b_{sh} : p_i = s \tag{4.2.13}$$

$$r_{kl} + \sum_{v \in \mathcal{V}^C} c_{vl} = \alpha_{kl} \qquad \forall k \in \mathcal{K}, \forall l \in \mathcal{A}_t \tag{4.2.14}$$

$$\sum_{k \in \mathcal{K}} [(r_{kl} + \sum_{v \in \mathcal{V}^C} c_{vl}) + \sum_{j \in \mathcal{E}^b_{il}} (r_{kj} + \sum_{v \in \mathcal{V}^C} c_{vj})] + \sum_{v \in \mathcal{V}^C} \sum_{j \in \mathcal{E}^t_{il}} y_{vj} \leq \omega^T \qquad l \in \mathcal{A}_t, i = 0,\ldots,\sigma_l \tag{4.2.15}$$

$$\sum_{k \in \mathcal{K}} r_{kl} + \sum_{v \in \mathcal{V}^C} c_{vl} \leq \ell \qquad \forall l \in \mathcal{A}_l \tag{4.2.16}$$

$$\sum_{\substack{i \in \mathcal{N}^b_f: \\ p_i = s}} (\sum_{l \in \mathcal{I}_k[i]} r_{kl} + \sum_{v \in \mathcal{V}^C} \sum_{l \in \mathcal{I}_v[i]} c_{vl}) + \sum_{\substack{i \in \mathcal{N}^t_f: \\ p_i = s}} \sum_{v \in \mathcal{V}^C} \sum_{l \in \mathcal{I}_v[i]} c_{vl} \geq \nu_{ks} \qquad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{4.2.17}$$

$$r_{kl} \in \mathbb{Z}^+ \qquad \forall k \in \mathcal{K}, l \in \mathcal{A}^b \backslash \mathcal{A}^b_{sh} \tag{4.2.18}$$

$$u_{ki} \in \mathbb{Z}^+ \qquad \forall k \in \mathcal{K}, i \in \mathcal{N}^b_i \tag{4.2.19}$$

$$c_{vl} \in \{0,1\} \qquad v \in \mathcal{V}^C, l \in \mathcal{A}_t \cup \mathcal{A}_{t2t} \cup \mathcal{A}_{dh} \cup \mathcal{A}_l \cup \mathcal{A}_g \cup \mathcal{A}_{sh}(v) \cup \mathcal{A}_{up}(v) \cup \mathcal{A}_{down}(v) \cup \mathcal{A}_{t^-} \tag{4.2.20}$$

The objective function (4.2.1) aims to minimize the sum of eight terms: pulling cost, deadheading cost, light traveling cost, idling cost, train-to-train connecting cost, and maintaining cost, the penalty for not serving critical locomotives at shop stations, and leasing cost. The strategy to calculate every detail of the cost γ_{kl} in each term is mentioned in Miranda et al. (2020).

Constraints (4.2.2) and (4.2.3) concern the flow of regular locomotives and critical locomotives on legacy arcs. We should note that it is not necessary to consider the top layer in constraint (4.2.2) because regular locomotives and locomotives undergoing maintenance at the beginning of the horizon only flow on the bottom layer.

At the beginning of the planning horizon, constraints (4.2.4) ensure that a total number of regular locomotives type k traversing on all outgoing arcs from each initial node must equal the supply of owned and leased locomotives type k in the yard. Similarly, constraints (4.2.5) establish the flow out of initial nodes for critical locomotives.

While constraints (4.2.6) guarantee the flow of regular locomotives from the sources to the sinks must be preserved, constraints (4.2.7) simply show that each critical locomotive must go to the sink. Constraints (4.2.8) and (4.2.9) impose the flow conversation of regular locomotives (including both owned and leased) and critical locomotives on departure nodes, arrival nodes and outpost nodes.

Constraints (4.2.10), (4.2.11), (4.2.12), and (4.2.13) are used to deal with critical locomotives' maintenance condition. Constraints (4.2.10) ensure that each critical locomotive flows on at most one shop arc. In case critical locomotives are not maintained in the shops (traversing a shop arcs) before their maintenance deadline, constraints (4.2.11) force it to follow a layer arc going to the top layer. Constraints (4.2.12) move overdue locomotives back to the bottom layer after being serviced in shops. Equalities (4.2.13) consider current number of locomotives in each shop in order to estimate the opportunity sending critical locomotives to this shop.

Constraints (4.2.14) guarantee that the consist assigned to each train is preserved during operational requirements. Equalities (4.2.15) limit the number of locomotives traversing each train arc, including both deadhead locomotives and active locomotives. Constraints (4.2.16) set the upper bound of the number of locomotives traveling themselves on light arcs while constraints (4.2.17) set the lower bound of the number of locomotives at each station by the end of the planning horizon. Finally, constraints (4.2.18), (4.2.19), (4.2.20) establish decision variables' domain.

4.3. The Rolling Horizon Approach

4.3.1. The Adaptation of the Network in the RHA

In Miranda et al. (2020), the legacy train-to-train arcs are not considered because all train-to-train connections in data contain scheduled trains within the week. When dividing the whole week or even a longer horizon into smaller planning horizons in the RHA, there

Parameters	**Definition**
γ_{kl}	Flow cost of a locomotive (regular or critical) type k on arc l
γ_{vl}	Flow cost of a critical locomotive v on arc l
M	Penalty for not taking a critical locomotive to shop
L	Leasing cost paid for each leased locomotive
λ_{ki}^{R}	Supply of the regular locomotives type k at source i
λ_{vi}^{C}	Supply of the critical locomotives v at source i
η_{kl}^{R}	Flow of regular locomotives type k on legacy arc $l \in \mathcal{A}_{t^-} \cup \mathcal{A}_{sh^-}$
η_{vl}^{C}	Flow of critical locomotive v on legacy arc $l \in \mathcal{A}_{t^-}$
α_{kl}	Number of active locomotives type k required to operate train $l \in \mathcal{A}_t$
μ_v	Maintenance type required by the critical locomotive v
δ_v	Maintenance deadline for the critical locomotive v
τ_m	Duration of maintenance type m
ζ_s	Capacity of shop s
ν_{ks}	Number of locomotives type k needed by station s at the end of the horizon
ω^T	Maximum number of locomotives per train
ℓ	Limit on number of locomotives traversing a light arcs
σ_t	Number of intermediate stops of the train t

Table 4.3. Parameters in mathematical formulation

exist some train-to-train connections starting and ending in different periods. Therefore, we introduce a set of legacy train-to-train arcs $\mathcal{A}_{t2t^-}^{b}$ to deal with these unfinished train-to-train connections. We can think of them as *partial legacy train arcs* with labels determining the remaining trains in train-to-train connections. This type of legacy train arcs includes the same origin and destination stations due to the fact that all locomotives in the consists must be grounded at the station to wait for connecting the next trains. In practice, if there exist some legacy train arcs followed by train-to-train connections, they are also attached labels similarly to legacy train-to-train arcs.

In the case of each legacy train arc, the head node is the arrival node at the destination station with time attribute equal to the train arrival time plus the time used to bust the consist (case a in Figure 4.2). For each legacy train-to-train arc, the head node is the

Sets	Definition
R	Set of regular locomotives
$\mathcal{K}$	Set of locomotive types
$\mathcal{V}_k^C$	Set of critical locomotives of type $k \in \mathcal{K}$
$\mathcal{V}^C$	Set of critical locomotives ($\mathcal{V}^C = \bigcup_{k \in \mathcal{K}} \mathcal{V}_k^C$)
$\mathcal{S}$	Set of stations
Sh	Set of shop stations ($Sh \subseteq \mathcal{S}$)
$\mathcal{N}_{sh}$	Set of nodes in the time-space network that are tails of at least one shop arc
$\mathcal{A}_{sh}^b(v)$	Set of shop arcs that can be traversed by critical locomotive v in the bottom layer
$\mathcal{A}_{sh}^t(v)$	Set of shop arcs that can be traversed by critical locomotive v in the top layer
$\mathcal{A}_{sh}(v)$	Set of shop arcs that can be traversed by critical locomotive v ($\mathcal{A}_{sh}(v) = \mathcal{A}_{sh}^b(v) \cup \mathcal{A}_{sh}^t(v)$)
$\mathcal{A}_{up}(v)$	Set of upward inter-layer arcs that can be traversed by critical locomotive v
$\mathcal{A}_{down}(v)$	Set of downward inter-layer arcs that can be traversed by critical locomotive v
$\mathcal{I}_k[i]$	Set of incoming arcs into node i that can be traversed by regular locomotive type k
$\mathcal{O}_k[i]$	Set of outgoing arcs into node i that can be traversed by regular locomotive type k
$\mathcal{I}_v[i]$	Set of incoming arcs into node i that can be traversed by critical locomotive type v
$\mathcal{O}_v[i]$	Set of outgoing arcs into node i that can be traversed by critical locomotive type v
$\mathcal{E}_{il}^b$	Set of arcs in the bottom layer that present an “ongoing” deadhead operation in train l when it departs from the ith station in its route
$\mathcal{E}_{il}^t$	Set of arcs in the top layer that present an “ongoing” deadhead operation in train l when it departs from the ith station in its route

Table 4.4. Sets in mathematical formulation

departure node of the remaining train in an unfinished train-to-train connection with the time attribute equal to the departure time of the second train.

As aforementioned, a legacy train-to-train arc is considered as a partial legacy train arc with a special label corresponding to the ID of the remaining train in an unfinished train-to-train connection to distinguish from authentic legacy train arcs. When this partial legacy train arc reaches its head node, an "unvalued" train-to-train arc with label -1 is created with the same time attributes of the tail node and head node and the cost spending for this arc equal to zero (case a in Figure 4.2). The difference between legacy train arcs in Miranda et al. (2020) and the ones in RHA is that legacy train arcs in RHA are attached special labels if they present the operation of the first trains in an unfinished train-to-train connection (case c in Figure 4.2). In this situation, train-to-train arcs are created to complete the connections, but they are also labeled -1 to distinguish from the ones created for train-to-train connections within the current planning horizon. Thus, instead of only containing the path (first train arc, train-to-train arc, second train arc) in the space-time network as in Miranda et al. (2020), our space-time network includes the path (legacy train/train-to-train arc, (unvalued) train-to-train arc, second train arc). More specifically, although $\mathcal{A}^b_{t2t-}$ is a subset of $\mathcal{A}^b_{t-}$, in practice, we still distinguish the set $\mathcal{A}^b_{t2t-}$ from the set $\mathcal{A}^b_{t-}$.

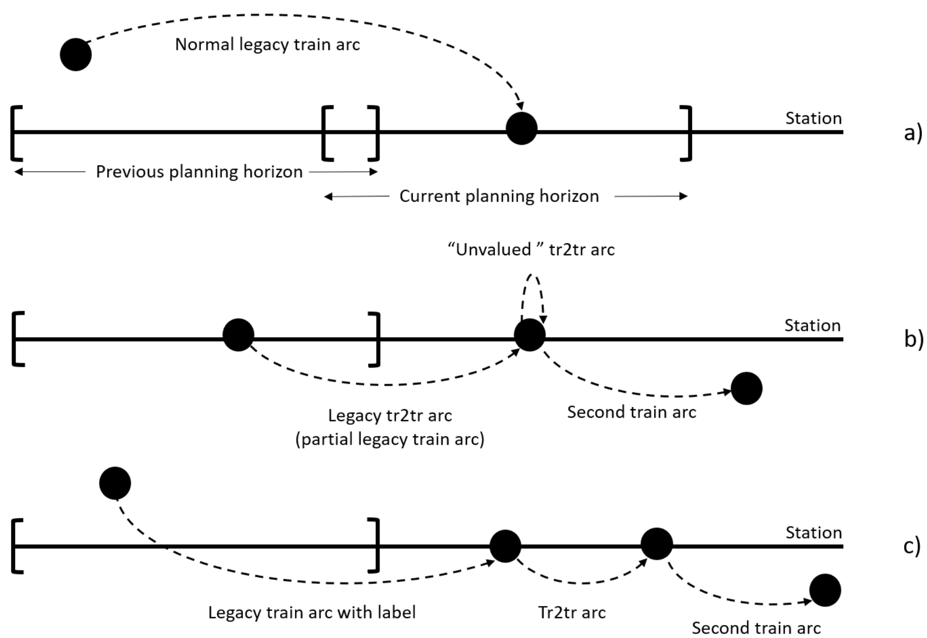

Fig. 4.2. Legacy train arcs and legacy train-to-train arcs

If a train-to-train connection contains a train out of the planning horizon, the first train arc in this connection will become a labeled legacy train arc at the beginning of the planning horizon whose end is greater than the departure time of the second train. Ortiz-Astorquiza et al. (2019) only consider train-to-train connections for a time window of six hours as the ones between two trains that are far apart in time are not necessarily. Therefore, the RHA can perform all train-to-train connections because the optimal solution of the LAP at tactical level does not contain a train-to-train connection with a large separation between the first train's arrival time and the second one's departure time.

However, in the case of the short planning horizon, a legacy train arc related to the first train might not find the second one in their train-to-train connection because the departure time of the second train is greater than the end of the considering time horizon. If the waiting time of the sharing consist does not exceed six hours, and the overlapping period is at least one day, we can assume that the legacy train arc mentioned above passes the end of the roll period. Thus, we can temporarily ignore it in the current planning horizon then add it again to the set of legacy train arcs in the next iteration of the RHA (see Figure 4.3). Additionally, case b and c in Figure 4.2 raised another issue considered at the end of Section 4.3.2.

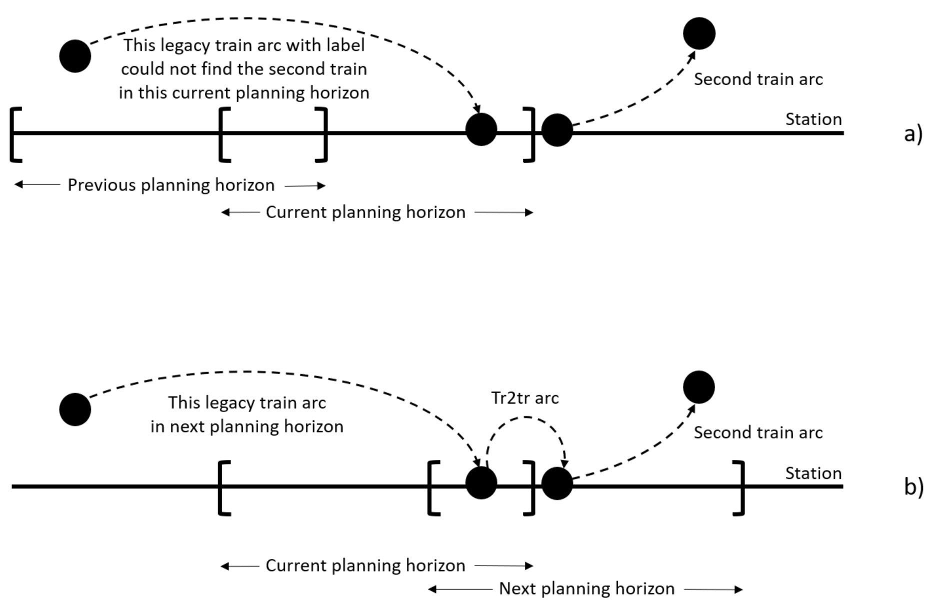

Fig. 4.3. How to deal with invalid legacy train arc with label

We should note that set $\mathcal{A}^b_{leg}$ contains all arcs whose head nodes' time attributes are smaller than the end of the considered planning horizon at each iteration of the RHA. We do not consider the legacy arcs that are unfinished within the planning horizon, and all locomotives traversing them because these locomotives cannot take part in any other events. We temporarily ignore them to decrease the size of the graph without affecting the solution.

Whenever the end time of these events is shorter than the end of a planning horizon, they will be added to the input.

We observe that the legacy train-to-train arcs and labeled legacy train arcs should be added to the bottom layer only. They can not appear on the top layer because other train arcs follow them in train-to-train connections.

4.3.2. The Adaptation of the IP Model to the RHA

We note that the IP model mentioned in Section 4.2 only considers the train-to-train connections containing two trains within the planning horizon. To apply RHA, we must create legacy train-to-train arcs to satisfy the train-to-train connections passing through the end of the roll period. Remember that we introduce set $\mathcal{A}_{t2t^-}$ as the set of legacy train-to-train arcs in Section 4.3.1. Following that, constraints (4.2.2), (4.2.3) and (4.2.20) are slightly changed to:

$$r_{kl} = \eta^R_{kl} \qquad \forall k \in \mathcal{K}, \forall l \in \mathcal{A}^b_{t^-} \cup \mathcal{A}^b_{sh^-} \cup \mathcal{A}^b_{t2t^-} \tag{4.3.1}$$

$$c_{vl} = \eta^C_{vl} \qquad \forall v \in \mathcal{V}^C, \forall l \in \mathcal{A}_{t^-} \cup \mathcal{A}_{t2t^-} \tag{4.3.2}$$

and

$$c_{vl} \in \{0,1\} \qquad v \in \mathcal{V}^C, l \in \mathcal{A}_t \cup \mathcal{A}_{t2t} \cup \mathcal{A}_{dh} \cup \mathcal{A}_l \cup \mathcal{A}_g \cup \mathcal{A}_{sh}(v) \cup \mathcal{A}_{dn}(v) \cup \mathcal{A}_{t^-} \cup \mathcal{A}_{t2t^-} \tag{4.3.3}$$

Therefore, the IP model using in RHA is reformulated as follows:

Objective function:

$$(4.2.1)$$

subject to:

$$\begin{cases} (4.3.1) \\ (4.3.2) \\ (4.2.4) - (4.2.19) \\ (4.3.3) \end{cases}$$

We should note that the creation of decision variables r_{kl} and c_{vl} for legacy train-to-train arcs or legacy train arcs belonging to train-to-train connections ($l \in \mathcal{A}_{t2t^-}$) is slightly different from the one for train-to-train arcs within the planning horizon ($l \in \mathcal{A}_{t2t}$). In the first one, we do not need to find the incoming train arcs associated to this train-to-train connection because each train-to-train arc with label -1 aforementioned in the previous section (case b) or case c) in Figure 4.2) is created for exactly one (partial) legacy train arc coming from the source node and carries known locomotives. For the second one, we have to consider the index of the train arc arriving at the tail node of the train-to-train arc. We introduce two new parameters shown in Table 4.5. This difference is presented as follows:

- In the case of train-to-train connection within the considering planning horizon:

$$c_{vt} \in \{0,1\} \qquad v \in \mathcal{V}^C, t = (i,j) \in \mathcal{A}_{t2t} : \begin{cases} \exists l \in \mathcal{A}_t \cap \mathcal{I}_v[i] \\ \exists l' \in \mathcal{A}_t \cap \mathcal{O}_v[j] \\ \Delta_{l'} = \Gamma_t \end{cases} \tag{4.3.4}$$

$$r_{kt} \in \mathbb{Z}^+ \qquad k \in \mathcal{K}, t = (i,j) \in \mathcal{A}_{t2t} : \begin{cases} \exists l \in \mathcal{A}_t \cap \mathcal{I}_k[i] \\ \exists l' \in \mathcal{A}_t \cap \mathcal{O}_k[j] \\ \Delta_{l'} = \Gamma_t \end{cases} \tag{4.3.5}$$

- In the case of unfinished train-to-train connection:

$$c_{vt} \in \{0,1\} \qquad v \in \mathcal{V}^C, t = (i,j) \in \mathcal{A}_{t2t} : \begin{cases} \exists l \in \mathcal{A}_{t2t^-} \cap \mathcal{I}_v[i] \\ \exists l' \in \mathcal{A}_t \cap \mathcal{O}_v[j] \\ \Gamma_t = -1 \\ \Gamma_l = \Delta_{l'} \end{cases} \tag{4.3.6}$$

$$r_{kt} \in \mathbb{Z}^+ \qquad k \in \mathcal{K}, t = (i,j) \in \mathcal{A}_{t2t} : \begin{cases} \exists l \in \mathcal{A}_{t2t^-} \cap \mathcal{I}_k[i] \\ \exists l' \in \mathcal{A}_t \cap \mathcal{O}_k[j] \\ \Gamma_t = -1 \\ \Gamma_l = \Delta_{l'} \end{cases} \tag{4.3.7}$$

Notation	**Definition**
$\mathcal{A}_{t2t^-}$	Set of legacy train-to-train arcs ($\mathcal{A}_{t2t^-} = \mathcal{A}^b_{t2t^-}$)
Δ_l	Index of the train related to train arc l
Γ_t	the label of arc t

Table 4.5. New attributes in the space-time network and the IP model

4.3.3. The Rolling Horizon Approach

The operational level in the real-life process of railway management has to face a long time horizon. This means that it is difficult to obtain an optimal or even a nearly optimal solution. Furthermore, railway dispatchers usually reschedule train services gradually to update and execute the plan for the near future. Due to this fact, a RHA is introduced as a fast and effective way to decompose our problem.

In Figure 3.1, we consider a h-day scheduled train at each iteration of the RHA, and the plan of the r first days provided by solving the LRP is fixed and added to the final result. After that, all the information is taken from the r latest days in the final result, such as: changing of locomotive types (from regular to critical), locomotive events (in transit: pulling trains, light traveling or deadheading; grounded in stations; in shops for maintenance) and capacity of shops, are combined with the train schedule of the next planning horizon to be used for the next iteration of the RHA.

In this section, we introduce two algorithms to deal with two types of instances. Algorithm 4.3.1, where critical locomotives are changed to "temporary regular" one, is implemented to solve 7-day instances, while Algorithm 4.3.2 is used to cope with 8-14 day instances without changing maintenance conditions of locomotives. We develop the second algorithm in order to eliminate re-optimizing time because the computing time increases dramatically when increases the length of planning horizon.

The RHA with "temporary regular locomotives"

Algorithm 4.3.1 presents how the RHA is implemented to deal with the LRP faced by CN. The input of the algorithm contains: train schedule T of the whole week with information of trains and train-to-train connections, all legacy arcs $\mathcal{A}_{leg}$, and set of all locomotives. The function extract_data() returns set $T^* \subset T$ of scheduled trains departing in $[s, s+h]$, all train-to-train connections wherein two trains are in a set T^*, a set $\mathcal{A}^*_{leg} \subset \mathcal{A}_{leg}$ of legacy arcs arriving at their destinations in the current planning horizon and all locomotives except the ones traversing legacy arcs whose arrival times are greater than the end of the day $s+h$. It also divides the set of locomotives into critical locomotives, in which their maintenance deadlines are equal or smaller than the end of the planning horizon, and regular locomotives, including "temporary regular" ones. By using this function, the flow η^R_{kl} and η^C_{kl} of regular and critical locomotives, respectively, on each legacy arc in the set $\mathcal{A}^*_{leg}$ are computed before being used in the IP model. Note that after extracting data, time attributes of trains in schedule and legacy arcs will be shifted to the left s days to adapt to the space-time network where time attributes of source nodes are set to 0.

Algorithm 4.3.1. Rolling horizon framework with "temporary" regular locomotives

---

Initial Data: $(T, \mathcal{A}_{leg}, \mathcal{V})$; **Result:** $\mathcal{S}$
Initialization: $i = 0, s = 0, \mathcal{D}^* = (T, \mathcal{A}_{leg}, \mathcal{V})$, t, h, r
while $(i < \lceil t/r \rceil)$ {
  Data $\mathcal{D}$:= extract_data $(\mathcal{D}^*, h, s)$; `// data of` h `days starting from day` s
  populate_data $(\mathcal{D})$;
  Network $\mathcal{N}$:= build $(\mathcal{D})$; `// build space-time network`
  IP Model $\mathcal{M}$:= build $(\mathcal{N})$; `// build mathematical model`
  sol := solve $(\mathcal{M})$; `// solve IP model by using CPLEX`
  $\mathcal{S}'$:= extract_solution (sol, r, s); `// solution of` r `days starting from day` s
  $bool$:= check_conflict $(\mathcal{S}')$; `// find all conflicts`
  if $(bool == true)$ {
    Data $\mathcal{D}'$:= config $(\mathcal{D})$; `// redefine types of some locos`
    populate_data $(\mathcal{D}')$;
    Network $\mathcal{N}'$:= build $(\mathcal{D}', h, s)$;
    IP Model $\mathcal{M}'$:= rebuild $(\mathcal{N}')$; `// Add new constraints to IP model`
    sol' := solve $(\mathcal{M}')$;
    $\mathcal{S}''$:= extract_solution (sol', r, s);
    $\mathcal{S}' := \mathcal{S}''$;
    $bool$:= check_conflict $(\mathcal{S}'')$;
    if $(bool == true)$ {
      Data $\mathcal{D}''$:= reconfig $(\mathcal{D}, x)$;`// use original types of locos`
      populate_data $(\mathcal{D}'')$;
      Network $\mathcal{N}''$:= build $(\mathcal{D}'', h, s)$;
      IP Model $\mathcal{M}''$:= rebuild $(\mathcal{N}'')$;
      sol'' := solve $(\mathcal{M}'')$;
      $\mathcal{S}'$:= extract_solution (sol'', r, s);
    }
  }
  $\mathcal{S}$:= add $(\mathcal{S}')$;
  $\mathcal{D}^*$:= update $(\mathcal{S}')$;
  $s := s + r$;
  $i := i + 1$;
}
return $\mathcal{S}$**;**

We should note that a critical locomotive can be temporarily considered as a regular locomotive if its maintenance deadline is out of the planning horizon. This assumption ensures that the critical locomotives can not be sent to the shops too early before its maintenance deadline. The locomotive turns back to the critical locomotive in the nearest planning horizon having the end time greater than its original deadline. Additionally, based on the "lately maintained" constraint, if a critical locomotive traverses a legacy arc passing its maintenance deadline, it is overdue for maintenance and must be sent to the top layer right after finish this activity (case a in Figure 4.4).

The function populate_data() determines current capacity ζ_s of each shop station s, supply λ_{ki}^R and λ_{ki}^C of each type k of regular and critical locomotives, respectively, at each station i at the beginning of the current planning horizon and maintenance status of critical locomotives (overdue or not) based on their maintenance deadline and arrival time of the legacy arcs they traverse.

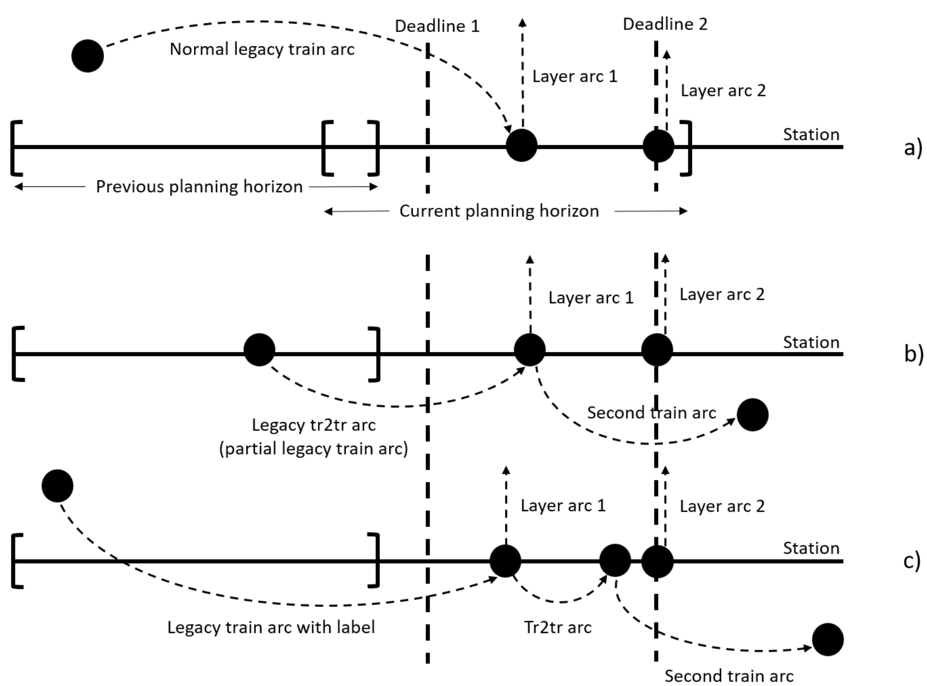

Fig. 4.4. The conflicts may occur when running the RHA step by step

However, a conflict sometimes appears when a critical locomotive changed from a "temporary regular" locomotive in the previous planning horizon to its original type in the current horizon traverses a legacy train-to-train arc or a legacy train arc connecting with another train in an unfinished train-to-train connection. In case b and case c in Figure 4.4, we consider a legacy train-to-train arc and a legacy train arc with label that both cross the maintenance deadline (Deadline 1) of critical locomotives traversing them. Because those legacy arcs cross the Deadline 1, the critical locomotives are overdue so that the layer arcs are created to move them to the top layer. Additionally, those locomotives belong to train-to-train connections, thus, the second train arcs or the train-to-train arcs are also created. It is impractical for both layer arcs and train-to-train arcs (or the second train arcs) to force the overdue locomotives to follow them at the same time. Besides, if the legacy arcs mentioned above do not cross the deadline of any critical locomotives in their consists (do not cross the Deadline 1), the second train arcs related to them can cross the Deadline 2. Our space-time network does not allow any train arc to cross the deadline of any critical locomotives belonging to its consist. Thus, this situation also raises an issue. To deal with those conflicts, the function check_conflict() is called after extracting a solution. If this function figures out conflicts from the result as the issue mentioned above, all invalid "temporary regular" locomotives are identified. We use the function config() to return their initial types then re-optimize the problem. Otherwise, the functions extract_solution() and add() are called consecutively to save all locomotives' status, their activities starting from the day s to $s + r$ and attach this piece of result to the final one.

In practice, we observe that for a short planning horizon (7-day instances), re-optimizing sub-problems sometimes occurs at the first iteration and never happens from the second to the last iteration. Thus, the RHA calls functions to re-optimize sub-problems at most once during the whole running time. In the worst case, the second **if**() condition in Algorithm 4.3.1 is accessed if there still exists invalid "temporary regular" locomotives in the solution. The function reconfig() is called to change all "temporary regular" locomotives to the critical ones with their original maintenance deadline. Then the algorithm re-optimizes the problem with this data.

Considering the IP model in Miranda et al. (2020), we realize that a given critical locomotive either are served on time (traversing a shop arc in the bottom layer), or served on the overdue state (traversing a shop arc in the top layer), or not served at all (in which case, the locomotive must flow on the top layer after its deadline). So, if the maintenance deadline is passed at the end of the planning horizon, then the locomotive will never be overdue and, therefore, it will not be moved to the top layer. This means that the locomotive must be serviced in a shop. Otherwise, the problem will be infeasible.

To overcome this problem, we introduce another function rebuild() for the IP model where the maintenance constraints are changed to:

$$\sum_{l \in \mathcal{A}_{up}(v)} c_{vl} = 1 - \sum_{l \in \mathcal{A}^b_{sh}(v)} c_{vl} \qquad v \in \mathcal{V}^C, d_v \leq EOH \tag{4.3.8}$$

with d_v = maintenance deadline of critical locomotive v and EOH = the end of the current planning horizon. By doing this, all critical locomotives, whose maintenance deadlines are greater than EOH, are not moved to the top layer because they are not overdue for maintenance and can not be served in this time horizon.

Furthermore, critical locomotives never traverse train arcs crossing their maintenance deadlines, so the conflict presented as case b and the one related to Deadline 1 in case c in Figure 4.4 never happen. The problem is infeasible if there exists a critical locomotive coming from the previous horizon by a first train arc and must traverse a second train arc crossing its maintenance deadline (Deadline 2 in case c). However, the IP model tries to serve as many critical locomotives as possible by their maintenance deadline within the considered time horizon so we can ignore this conflict. The supply of critical locomotives at each station is much less than the one of regular locomotives at the beginning of the planning horizon. Thus, it is less complicated for the model to send more critical locomotives to the shop stations and use regular ones for pulling trains instead.

To ensure that a feasible solution is always found, constraints that do not allow critical locomotives to traverse second train arcs crossing their deadlines are also added to the IP model when calling the function rebuild(). We also introduce a function $checkT2t()$ to check that a train arc passing the end of roll period belongs to a series of train-to-train connections or not. This function searches among the original data then returns ID of the last train departing in the further planning horizons in an series of train-to-train connections. Let t_arr be arrival time of this train. The constraints are presented as follows:

$$\sum_{l \in \mathcal{A}_t | checkT2t(l) \neq \varnothing, t\_arr > d_v} c_{vl} = 0 \qquad v \in \mathcal{V}^C, d_v > EOH \tag{4.3.9}$$

Therefore, the function rebuild() creates a new IP formulation:

Objective function:

(4.2.1)

subject to:

$$
\begin{cases}
(4.3.1) \\
(4.3.2) \\
(4.2.4) - (4.2.10) \\
(4.2.12) - (4.2.19) \\
(4.3.3) \\
(4.3.8) \\
(4.3.9)
\end{cases}
$$

Finally, we have to consider all locomotives' activities starting by the end of the day $s + r$ and finishing after this time because they not only exist in the current solution but also affect the next planning horizon. The detail of this task, the core of the function update(), is presented in Section 4.3.2.

The RHA without creating "temporary regular locomotives"

Algorithm 4.3.2 is introduced as a sub-process of the Algorithm 4.3.1. In this algorithm, we do not change critical locomotives into "temporary" regular when implementing the IP model mentioned above in the function rebuild(). Comparing with the Algorithm 4.3.1, this algorithm provides a shorter running time because it does not require re-optimization steps. Besides, the Algorithm 4.3.2 always finds feasible solutions, while the Algorithm 4.3.1 may return no solution or take a considerable extra computational time to re-optimize the previous sub-problems if the conflicts require to roll back more than one planning horizon. However, in the solutions of Algorithm 4.3.2, critical locomotives may be served early before their deadlines since the IP model tries to send as many of them as possible to the shops during the considered time horizon. To save time, we use Algorithm 4.3.2 to solve 8-14 day instances, while Algorithm 4.3.1 is implemented for the case of 7-day instances in order to observe the impact of changing types of locomotives on maintaining critical locomotives.

Algorithm 4.3.2. Rolling horizon framework without changing types of locomotives

Initial Data: $(T, \mathcal{A}_{leg}, \mathcal{V})$; **Result:** $\mathcal{S}$
Initialization: $i = 0, s = 0, \mathcal{D}^* = (T, \mathcal{A}_{leg}, \mathcal{V})$, t, h, r
while $(i < \lceil t/r \rceil)$ {
  Data $\mathcal{D}$:= extract_data $(\mathcal{D}^*, h, s)$;
  Data $\mathcal{D}''$:= reconfig $(\mathcal{D}, x)$;
  populate_data $(\mathcal{D}'')$;
  Network $\mathcal{N}''$:= build $(\mathcal{D}'', h, s)$;
  IP Model $\mathcal{M}''$:= rebuild $(\mathcal{N}'')$;
  sol'' := solve $(\mathcal{M}'')$;
  $\mathcal{S}'$:= extract_solution (sol'', r, s);
  $\mathcal{S}$:= add $(\mathcal{S}')$;
  $\mathcal{D}^*$:= update $(\mathcal{S}')$;
  $s := s + r$;
  $i := i + 1$;
}
return $\mathcal{S}$**;**

4.3.4. Updating Information in the RHA

We consider all activities (arcs) l starting at ϕ or before ϕ ($\phi-$) and finishing after ϕ ($\phi+$). Note that ϕ is also the beginning of the next planning horizon in the RHA. From now on, we also denote ($\phi-$) and ($\phi+$) are the current planning horizon and the next planning horizon, respectively. The stations where tail node j and head node h of arc l are placed are denoted by p_j, p_h. Parameters t_j, t_h, t_u are start time, end time of the locomotive's activity and time unit. Let $\mathcal{A}' = \mathcal{A}'_t \cup \mathcal{A}'_{t2t} \cup \mathcal{A}'_{dh} \cup \mathcal{A}'_l \cup \mathcal{A}'_g \cup \mathcal{A}'_{sh} \cup \mathcal{A}'_y \cup \mathcal{A}'_{t^-} \cup \mathcal{A}'_{sh^-} \cup \mathcal{A}'_{t2t^-}$ be the set of arcs in the next iteration. The station, where the locomotive l is grounded, is denoted by ρ_l. A parameter o_v is created to check the maintenance status of critical locomotive v. If the end time of arc l is greater than d_v, critical locomotive v is overdue for maintenance so that $o_v = true$. Otherwise, $o_v = false$. Let string id be the ID of the second train in an unfinished train-to-train connection which cannot finish before the end of roll period and whose the first train departs in the roll period. A parameter $label$ is label of the legacy train arc for the next planning horizon if the function $checkT2t()$ does not return $\varnothing$. Let $\mathcal{V}'^{C}_{k}$,

$\mathcal{V}'^C$, and R' be the set of critical locomotives type k, the set of critical locomotives, and the set of regular locomotives in the next iteration of the RHA, respectively.

In each case mentioned below, we update the status of each locomotive and the flow of each legacy arc it traverses or the supply of locomotives at the station where it is grounded. For the cases of legacy light arcs and legacy deadhead arcs, we consider them as legacy train arcs because their roles are similar for building the space-time network. We should note that all legacy arcs having arrival time out of the next planning horizon are not taken to the input of the iteration related to this horizon, but they must exist in the set $\mathcal{A}_{leg}$ for use later. If a legacy arc arrives before the end of the next planning horizon but out of roll period, it is still counted for a further estimation in the future because its flow might affect the supply of locomotives at its destination. As mentioned in Section 4.1.1, the legacy arcs must be shifted to the left s days because the time attributes of source nodes in the space-time network must be set to 0. Therefore, the start time and end time of each arc in set $\mathcal{A}'_{leg}$ equal to the start time and end time of the corresponding activity plus $s \cdot t_u$, respectively. Note that the notation $(\phi-)\{... \Rightarrow (\phi+)\{...$ presents the transformation of an arc crossing the end of the roll period into a legacy arc in the next planning horizon.

The sets, parameters in the mathematical formulation, locomotives data and stations data need to be updated at the beginning of the next planning horizon. We present the updates separately for critical, regular and leased locomotives.

Critical locomotives:

In the following we present the updates depending on the state/location of the critical locomotive. More precisely, it can be in a shop, undergo maintenance, pull a train, deadhead, light travel or be grounded. All of the updates follow a similar structure, so we describe the first cases in more detail and then focus on the differences.

If critical locomotive v of type k is in shop:

$$(\phi-) \quad \exists l : \begin{cases} c_{vl} = 1 \\ l \in \mathcal{A}_{sh}(v) \\ t_j \leq \phi < t_h \\ p_j = p_h = m \in Sh \end{cases} \qquad \Rightarrow (\phi+) \Big\{ (4.3.10\text{a}) - (4.3.10\text{i})$$

where:

$$
\begin{cases}
\mathcal{V}'^{C}_{k} = \mathcal{V}^{C}_{k} \backslash v & (4.3.10a) \\
\mathcal{V}'^{C} = \mathcal{V}^{C} \backslash v & (4.3.10b) \\
R' = R \cup v & (4.3.10c) \\
l' \in \mathcal{A}'^{b}_{sh^-} & (4.3.10d) \\
t'_j = t_j + s \cdot t_u & (4.3.10e) \\
t'_h = t_h + s \cdot t_u & (4.3.10f) \\
p'_j = p'_h = m \in Sh & (4.3.10g) \\
\eta'^{R}_{kl'} = 1 & (4.3.10h) \\
\rho_v = m & (4.3.10i)
\end{cases}
$$

If critical locomotive v of type k is maintained before ϕ:

$$
(\phi-) \quad \exists l : \begin{cases} c_{vl} = 1 \\ l \in \mathcal{A}_{sh} \\ t_h \leq \phi \\ p_j = p_h = m \in Sh \end{cases} \Rightarrow (\phi+) \begin{cases} \mathcal{V}'^{C}_{k} = \mathcal{V}^{C}_{k} \backslash v \\ \mathcal{V}'^{C} = \mathcal{V}^{C} \backslash v \\ R' = R \cup v \\ \rho_v = m \end{cases} \tag{4.3.11}
$$

Equations (4.3.10a) - (4.3.10c) ensure that for each shop arc l passing the end of the roll period, critical locomotive v traversing this arc will be changed to regular one. A legacy shop arc l' will be created as Equations (4.3.10d) based on the head node, the tail node, and time attributes of l in Equations (4.3.10e) - (4.3.10g). All critical locomotives finishing their inspections are changed into the regular ones in (4.3.11). The flow of regular locomotive type k on legacy shop arc l' will be set to 1 in Equations (4.3.10h) and the location of locomotive v will be updated as Equations (4.3.10i). The capacities of the shop stations where locomotives are located will be updated by the function populate_data() (see Section 4.3.2) at the beginning of the next planning horizon.

If critical locomotive v is pulling a train on arc l:

$$(\phi-) \quad \exists l : \begin{cases} c_{vl} = 1 \\ l \in \mathcal{A}_t \\ t_j \leq \phi < t_h \\ p_j, p_h \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} (4.3.12a) - (4.3.12j) \end{cases}$$

where:

$$\begin{cases} l' \in \mathcal{A}'^{b}_{t^-} & (4.3.12a) \\ t'_j = t_j + s \cdot t_u & (4.3.12b) \\ t'_h = t_h + s \cdot t_u & (4.3.12c) \\ p'_j = p_j & (4.3.12d) \\ p'_h = p_h & (4.3.12e) \\ \eta'^{C}_{vl'} = 1 & (4.3.12f) \\ \rho_v = \varnothing & (4.3.12g) \\ d'_v = d_v + s \cdot t_u & (4.3.12h) \\ o_v = true \qquad \text{if } d'_v \leq 0 \vee t'_h \geq d'_v > 0 & (4.3.12i) \\ label = id \qquad \text{if } checkT2t(l) \neq \varnothing & (4.3.12j) \end{cases}$$

For each train arc l passing the end of the roll period, we create a legacy train arc l' as Equations (4.3.12a). The head node, the tail node, and time attributes of l' are provided by Equations (4.3.12b) - (4.3.12e). Equations (4.3.12f) sets the flow of the legacy train arc l' to 1 because only critical locomotive v traverses this arc. The critical locomotive v traversing legacy train arc l' is not available at any station at the beginning of the next planning horizon so that ρ_v is updated as $\varnothing$ in Equations (4.3.12g). The maintenance deadline of critical locomotive v is returned to its original value in Equations (4.3.12h), and the overdue state o_v is updated in Equations (4.3.12i). Finally, if l is the first train arc in an unfinished train-to-train connection, a label will be attached to l' in Equations (4.3.12j).

If critical locomotive v is traversing deadhead arc l:

$$(\phi-) \quad \exists l : \begin{cases} c_{vl} = 1 \\ l \in \mathcal{A}_{dh} \\ t_j \leq \phi < t_h \\ p_j, p_h \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A}'^{b}_{t^-} \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p_j \\ p'_h = p_h \\ \eta'^{C}_{vl'} = 1 \\ \rho_v = \varnothing \\ d'_v = d_v + s \cdot t_u \\ o_v = true & \text{if } d'_v \leq 0 \vee t'_h \geq d'_v > 0 \end{cases} \tag{4.3.13}$$

The deadhead arcs crossing the end of the roll period are updated as legacy train arcs without Equations (4.3.12j) because the locomotives traversing these arcs cannot take part in train-to-train connections.

If critical locomotive v is traversing light travel arc l:

$$(\phi-) \quad \exists l : \begin{cases} c_{vl} = 1 \\ l \in \mathcal{A}_{l} \\ t_j \leq \phi < t_h \\ p_j, p_h \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A}'^{b}_{t^-} \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p_j \\ p'_h = p_h \\ \eta'^{C}_{vl'} = 1 \\ \rho_v = \varnothing \\ d'_v = d_v + s \cdot t_u \\ o_v = true & \text{if } d'_v \leq 0 \vee t'_h \geq d'_v > 0 \end{cases} \tag{4.3.14}$$

The light travel arcs crossing the end of the roll period are updated as legacy train arcs without Equations (4.3.12j).

If critical locomotive v is grounded at the station (l is a ground arc):

$$(\phi-) \quad \exists l : \begin{cases} c_{vl} = 1 \\ l \in \mathcal{A}_g \\ t_j \leq \phi < t_h \\ p_j = p_h = m \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \left\{ (4.3.15a) - (4.3.15d) \right.$$

where:

$$\begin{cases} \lambda'^{C}_{vm} = 1 \qquad \text{if } d'_v \geq 0 & (4.3.15a) \\ \rho_v = m & (4.3.15b) \\ d'_v = d_v + s \cdot t_u & (4.3.15c) \\ o_v = true \qquad \text{if } d'_v \leq 0 & (4.3.15d) \end{cases}$$

The supply of critical locomotive v at station m is set to 1 in Equations (4.3.15a) if v has not missed its maintenance deadline. Equations (4.3.15d) updates the overdue state of critical locomotive v. If the maintenance deadline d'_v is less than or equal to 0, it is placed at a shop station and waiting for inspection. This locomotive will be placed in the top layer at the beginning of the next planning horizon.

Aforementioned in Section 4.3.1, a legacy train-to-train arc is considered as a partial legacy train arc with a unique label corresponding to the ID of the remaining train in an unfinished train-to-train connection. The train-to-train arc crossing the end of the roll period will be changed to legacy train-to-train arc in the next planning horizon as follows:

If critical locomotive v is in train-to-train arc l:

$$(\phi-) \quad \exists l : \begin{cases} c_{vl} = 1 \\ l \in \mathcal{A}^b_{t2t} \\ t_j \leq \phi < t_h \\ p_j = p_h = m \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A}'_{t2t^-} \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p'_h = m \\ \eta'^C_{vl'} = 1 \\ \rho_v = m \\ d'_v = d_v + s \cdot t_u \\ label = id \end{cases} \tag{4.3.16}$$

Note that the overdue state is not considered as in (4.3.13) because of Constraints (4.3.9).

Regular locomotives:

The information of a regular locomotive and a critical locomotive traversing an arc, which crosses the end of the roll period, is updated similarly. However, we do not consider Equations (4.3.12i) in term of the regular locomotives because they do not require to be maintained.

For each regular locomotive i of type k traverses on train arc l:

$$(\phi-) \quad \exists l : \begin{cases} r_{kl} \geq 1 \\ l \in \mathcal{A}^b_t \\ t_j \leq \phi < t_h \\ p_j, p_h \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A}'^b_{t^-} \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p_j \\ p'_h = p_h \\ \eta'^R_{kl'} = 1 \\ \rho_i = \varnothing \\ label = id \qquad \text{if } checkT2t(l) \neq \varnothing \end{cases} \tag{4.3.17}$$

For each regular locomotive i of type k travels as a deadhead on arc l:

$$(\phi-) \quad \exists l : \begin{cases} r_{kl} \geq 1 \\ l \in \mathcal{A}^b_{dh} \\ t_j \leq \phi < t_h \\ p_j, p_h \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A}'^b_{t^-} \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p_j \\ p'_h = p_h \\ \eta'^R_{kl'} = 1 \\ \rho_i = \varnothing \end{cases} \tag{4.3.18}$$

For each regular locomotive i of type k is light travelling on arc l:

$$(\phi-) \quad \exists l : \begin{cases} r_{kl} \geq 1 \\ l \in \mathcal{A}^b_{l} \\ t_j \leq \phi < t_h \\ p_j, p_h \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A}'^b_{t^-} \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p_j \\ p'_h = p_h \\ \eta'^R_{kl'} = 1 \\ \rho_i = \varnothing \end{cases} \tag{4.3.19}$$

For each regular locomotive i of type k is in grounded arc l:

$$(\phi-) \quad \exists l : \begin{cases} r_{kl} \geq 1 \\ l \in \mathcal{A}^b_{g} \\ t_j \leq \phi < t_h \\ p_j = p_h = m \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} \lambda'^R_{km} := \lambda'^R_{km} + 1 \\ \rho_i = m \end{cases} \tag{4.3.20}$$

For each regular locomotive i of type k is in train-to-train arc l:

$$(\phi-) \quad \exists l : \begin{cases} r_{kl} \geq 1 \\ l \in \mathcal{A}^b_{t2t} \\ t_j \leq \phi < t_h \\ p_j = p_h = m \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A}'^b_{t2t-} \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p'_h = m \\ \eta'^R_{kl'} = 1 \\ \rho_i = m \\ label = id \end{cases} \tag{4.3.21}$$

In the IP model introduced in Miranda et al. (2020), the decision variables u_{ki} in Constraints (4.2.4) determine the number of leased locomotives type k supplied at source i. Thus, the model decides to lease locomotives at the beginning of the week and let them be active as much as possible in the whole week. In the RHA, variables u_{ki} are created at the beginning of each planning horizon so that many locomotives are leased after the beginning of the week, and their activities starting in roll period are saved to the final result. If we do not update the status and location of a leased locomotive for use later, it will be considered as released right after finishing the last activity passing the end of the roll period because it does not exist in the input data of the next planning horizon.

We observe that reusing the old leased locomotives can decrease the number of new ones in the future and avoid releasing them too early before the end of the whole planning horizon. Additionally, if a leased locomotive pulls the first train of an unfinished train-to-train connection, it must be used for pulling the second train in the next planning horizon. Even if the locomotives are grounded after traversing a train, deadhead, or a light arc, it should be kept because the model might add it to the supply of station for further operation. However, it is thriftless and not necessarily to reuse all leased locomotives. By considering the locomotives' sequences of activities starting before the end of the roll period, we can decide which leased locomotive is going to be eliminated from the result. In that case, all leased locomotives grounded at the station in the whole roll period will be ignored even if they depart for somewhere in the overlap as shown in Constraints (4.3.25).

Due to these reasons, we suppose that the leased locomotives are considered as "temporary owned regular" locomotives, and their paths are recorded if they pull trains, traverse light/deadhead arcs or take part in train-to-train connections. This assumption does not add or subtract any cost because the leasing cost is paid at the beginning of the current planning horizon until the end of the week.

Leased locomotives:

By denoting that $\mathcal{P}_i$ is the path of a leased locomotive departing from source i, the status and location of leased locomotives are updated the same as regular locomotives.

Assign a leased locomotive z of type k traversing train arc l:

$$(\phi-) \quad \exists l : \begin{cases} u_{ki} \geq 1 \\ l \in \mathcal{A}_t^b \\ l \in \mathcal{P}_i \\ t_j \leq \phi < t_h \\ p_j, p_h \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A}'^b_{t^-} \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p_j \\ p'_h = p_h \\ \eta'^R_{kl'} = 1 \\ \rho_z = \varnothing \\ R' = R \cup z \\ label = id & \text{if } checkT2t(l) \neq \varnothing \end{cases} \tag{4.3.22}$$

Assign a leased locomotive z of type k traversing deadhead arc l:

$$(\phi-) \quad \exists l : \begin{cases} u_{ki} \geq 1 \\ l \in \mathcal{A}_{dh}^b \\ l \in \mathcal{P}_i \\ t_j \leq \phi < t_h \\ p_j, p_h \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A}'^b_{t^-} \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p_j \\ p'_h = p_h \\ \eta'^R_{kl'} = 1 \\ \rho_z = \varnothing \\ R' = R \cup z \end{cases} \tag{4.3.23}$$

Assign a leased locomotive z of type k traversing light arc l:

$$(\phi-) \quad \exists l: \begin{cases} u_{ki} \geq 1 \\ l \in \mathcal{A}_l^b \\ l \in \mathcal{P}_i \\ t_j \leq \phi < t_h \\ p_j, p_h \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A'}_{t^-}^b \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p_j \\ p'_h = p_h \\ \eta'^R_{kl'} = 1 \\ \rho_z = \varnothing \\ R' = R \cup z \end{cases} \tag{4.3.24}$$

If leased locomotive z of type k is on a ground arc l:

$$(\phi-) \quad \exists l: \begin{cases} u_{ki} \geq 1 \\ l \in \mathcal{A}_g^b \\ l \in \mathcal{P}_i \\ \exists l_1 = (j_1, h_1) \in \mathcal{P}_i : l_1 \in \mathcal{A}_t^b \cup \mathcal{A}_{dh}^b \cup \mathcal{A}_l^b \cup \mathcal{A}_{t2t}^b \\ t_j \leq \phi < t_h \\ t_{h_1} < \phi \\ p_j = p_h = s \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} \lambda'^R_{ks} := \lambda'^R_{ks} + 1 \\ \rho_z = s \end{cases} \tag{4.3.25}$$

As presented in the left-hand side of the system of equation (4.3.25), the path of leased locomotive z must contain at least an arc l_1, which is a train arc, a deadhead arc, a light travel arc or a train-to-train arc. This condition ensures that locomotive z is not grounded at the station in the whole overlap.

Assign a leased locomotive z of type k for train-to-train connection:

$$(\phi-) \quad \exists l : \begin{cases} u_{ki} \geq 1 \\ l \in \mathcal{A}^b_{t2t} \\ l \in \mathcal{P}_i \\ t_j \leq \phi < t_h \\ p_j = p_h = m \in \mathcal{S} \end{cases} \Rightarrow (\phi+) \begin{cases} l' \in \mathcal{A'}^b_{t2t-} \\ t'_j = t_j + s \cdot t_u \\ t'_h = t_h + s \cdot t_u \\ p'_j = p_j \\ p'_h = p_h \\ \eta'^R_{kl'} = 1 \\ \rho_z = m \\ R' = R \cup z \\ label = id \end{cases} \tag{4.3.26}$$

To decrease the size of the space-time network in the next planning horizon, we combine individual unfinished train arcs in $\mathcal{A'}^b_{t-}$, including both deadhead arcs and light travel arcs, into one legacy train arc if they have the same label, departure station, destination station, and time attributes.

Similarly, we also group all arcs $l \in \mathcal{A'}^b_{t2t-}$ into one legacy train-to-train arc and add them to the set of legacy train arcs for the next iteration. Following this, the set of locomotives traversing these arcs is determined, then the flow of regular and critical locomotives on each arc is calculated.

Without loss of generality, we assume that l_y is the legacy arc traversing by regular locomotive y type k. Let $\mathcal{A'}_{t^-} = \mathcal{A'}^b_{t-} \cup \mathcal{A'}^b_{t2t-}$ be the set of considering legacy arcs, $\mathcal{A^*}_{t-}$ be the final set of legacy arcs and $\mathcal{V}_{l^*}$ be the set of locomotives traversing the legacy arc l^*. This process is presented by set of equations (4.3.27).

$\forall l_y \in \mathcal{A'}_{t^-}$:

$$L = \{x : x \neq y, \begin{cases} l_x \in \mathcal{A'}_{t^-} \\ t_{j_x} = t_{j_y} \\ t_{h_x} = t_{h_y} \\ p_{j_x} = p_{j_y} \\ p_{h_x} = p_{h_y} \\ label_x = label_y \end{cases} \} \Rightarrow \begin{cases} l^* \in \mathcal{A^*}_{t^-} & \\ t^*_j = t_{j_y} & \\ t^*_h = t_{h_y} & \\ \eta^R_{kl^*} = 1 + \sum \eta^R_{x_{kl^*}} & \text{if } x \in R, \text{ type } k \\ \eta^R_{k'l^*} = \sum \eta^R_{x_{k'l^*}} & \text{if } x \in R, \text{ type } k' \neq k \\ \eta^C_{xl^*} = 1 & \text{if } x \in \mathcal{V}^C \\ p_j = p_{j_y} & \\ p_h = p_{h_y} & \\ label = label_y & \\ (\mathcal{A'}_{t^-} \backslash l_y) \backslash \{l_x : x \in L\} & \\ \mathcal{V}_{l^*} = y \cup L & \end{cases} \tag{4.3.27}$$

In the left-hand side of the system of equations (4.3.27), for each locomotive y, we consider all locomotives $x \neq y$ traversing legacy train arcs l_x in set $\mathcal{A'}_{t^-}$. If there exists at least one locomotive x traversing a train arc l_x whose label, departure station, destination station and time attributes are similar to l_y's ones, set L including all locomotives x satisfying the conditions mentioned above is created and the right hand size of the system of equations (4.3.27) is called to add x and y to a new train arc l^*. The arc l^* is a copy of l_y and contains all locomotives x. We also increase the flow of regular locomotives type k ($\eta^R_{kl^*}$) and the flow of critical locomotives ($\eta^C_{xl^*}$) on arc l^*. Finally, we delete arcs l_y and all train arcs l_x from set $\mathcal{A'}_{t^-}$, then update set $\mathcal{V}_{l^*}$ ($y \cup L$) as set of locomotives traversing arc l^*.

Chapter 5

Computational Experiments

5.1. Instances & Parameter Setting

In this section, the effectiveness of the RHA is discussed. We test our algorithm on 12 one-week instances proposed by Miranda et al. (2020). These instances are generated from the database of CN and contain all the necessary information mentioned in Section 3.2. Each instance contains over 1900 locomotives, 3700 trains, 400 train-to-train connections, and 400 stations (see detail in Table 5.1). We are also given legacy trains from the previous week, and locomotives serviced in the shop stations at the beginning of each week.

Data	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
# trains	3811	3750	3739	3735	3598	3762	3710	3822	3786	3797	3769	3804
# stations	443	454	439	456	452	474	449	454	434	463	444	434
# locos	1933	1963	1960	1937	1930	1935	1940	1939	1916	1913	1905	1901
# tr2tr	529	528	491	522	497	485	493	523	588	513	502	531
# locos in shop	138	153	130	128	142	132	130	137	128	133	156	132
# locos in transit	332	336	361	373	356	371	350	348	371	337	355	364
# critical locos	99	79	89	91	86	92	92	99	83	102	79	108

Table 5.1. Instances overview

The RHA framework is implemented in C++ using CPLEX 12.9.0 for the resolution of the IP formulation. Both methods are run on a 3.20GHz Xeon(R) E5-2667 computer. We have tested many settings with different values of the planning horizon and the roll period, and also compared all of them to study the effects of these chosen time horizons on the final results.

While the parameters of the IP formulation in the RHA are chosen almost the same as Miranda et al. (2020), the leasing cost is set in a different way. In practice, we observe that if the leasing cost is set to zero, the RHA will focus on leasing third-party locomotives instead of using light travels and deadheads. Hence, the number of leased locomotives increases dramatically.

As described in Section 3.2, the leasing cost is ignored in Miranda et al. (2020) because of the ownership cost charged on each arc of the space-time network for having the locomotives. When the IP model decides to lease a third-party locomotive, this locomotive will be created at a source node, and then traverse several arcs until reaching a sink node. Next, the ownership cost will be calculated based on the number of arcs along its path. The objective of the IP model is to minimize the total operational cost. Therefore, the total ownership cost will be minimized, and the number of leased locomotives as well.

In the case of the RHA, the leased locomotives are created at the beginning of every roll period. We observe that the reuse of the third-party locomotives, which are leased at the previous planning horizons, is restrictive. Because we only take into account the result of the roll period, while all activities of the leased locomotives in the overlap, such as light traveling or deadheading, are ignored. The RHA does not record information for the further time horizon so that the leased locomotives may not be efficiently reused in the next planning horizon. Therefore, the IP model may decide to lease new third-party locomotives instead of reusing the older ones. Since leasing a new locomotive are always feasible, while operating light travels and deadheads are unavailable because of several conditions.

Controlling the number of leased locomotives by adding the leasing cost impacts on the number of deadheads and light travels in the solution provided by the RHA. While the number of deadheads is trivially changed, the number of light travels is significantly transformed. There exist some reasons for this phenomenon. First, the IP model always focuses on using as many deadheads as possible because of their low operational cost. Thus, the number of deadheads is much higher than the number of light travels and leased locomotives. Second, light travels are more flexible than deadheads since they do not depend on train arcs. Third, each light travel arc is charged a penalty which presents how it is more expensive than a deadhead arc. However, the limit on the maximum distance of each light travel sets a limit on light travel cost and makes them more suitable to address instead of leasing third-party locomotives with high(er) cost. Accordingly, a trade-off between the number of leased locomotives and the number of light travels and deadheads can be considered as a trade-off between leasing third-party locomotives and using light travels.

From the observation mentioned above, we test our rolling horizon framework with the leasing cost greater than 0. The value of the leasing cost depends on several aspects, such as the length of the chosen planning horizon, the length of the considering roll period, the number of iterations used in the RHA, and the penalty paid for each light travel arc.

5.2. The Effects of the Planning Horizon and the Roll Period on the Results

In the first experiment (see the results shown in Table 5.2), we compare the results of the first instance obtained by Algorithm 4.3.1 with the optimal solution proposed by Miranda et al. (2020). Notice that in the first row of Table 5.2, $h = x$,$r = y$ presents the length of the planning horizon (x days) and the length of the roll period (y days). "# deadheads", "# light travels", and "# leased locos" stand for the number of locomotives used as deadheads, light traveling, and leased from third-party companies. Each cell in the Table 5.2 presents the relative difference between the solution of Algorithm 4.3.1 and the solution provided by the exact method.

Week 1	$h = 3$,$r = 2$	$h = 4$,$r = 2$	$h = 5$,$r = 2$	$h = 4$,$r = 3$	$h = 5$,$r = 3$	$h = 5$,$r = 4$
Time	-80.64	-55.22	-24.35	-60.60	-36.20	-56.65
Total cost	18.95	13.16	9.81	23.93	20.57	11.44
Deadhead cost	-46.35	-25.60	-8.68	-27.64	-6.38	-18.38
Idle cost	25.00	19.20	16.25	45.40	40.61	18.24
Light travel cost	102.47	45.62	14.44	39.56	34.40	54.05
Total distance	-3.20	-1.62	-0.51	-1.88	-0.35	-1.13
Total deadhead distance	-45.02	-23.01	-7.39	-26.42	-4.99	-16.30
Average deadhead distance	-19.84	-8.19	-3.40	-10.11	-2.92	-4.73
Total light distance	29.95	43.81	25.19	14.14	13.58	70.12
Average light distance	-20.71	-16.50	-28.46	-29.16	-29.50	-32.70
# deadheads	-31.41	-16.15	-4.13	-18.15	-2.13	-12.14
# light travels	63.89	72.22	75.00	61.11	61.11	152.78
# leased locos	44.49	21.19	7.63	25.85	8.47	8.05

Table 5.2. The relative difference of the results of the first instance solved by the RHA with the results provided by the exact solution of the IP-based model (%)

Overall, the computing time is reduced with the RHA. It can be observed that the setting with the shortest planning horizon takes the least time to finish (save 80.64%), and the relative difference of running time decreases when the length of the planning horizon increases. The gaps between the total cost provided by using the RHA and the one in the optimal solution range between 9.81% (in the case of $h = 5$, $r = 2$ – the longest planning horizon, the shortest roll period and 3-day overlapping) and 23.93% (in the case of $h = 4$, $r = 3$ with 1-day overlapping). We do not present the costs of pulling trains and performing train-to-train connections because they are nearly equal (relative difference $< 0.1\%$) to the corresponding costs in the exact solution. The gap between those pulling costs is caused by the duplicate of insignificant ownership cost, which is calculated at the previous iteration then calculated again at the current iteration, during the time of busting or connecting locomotives when legacy trains arrival at their destinations. Hence, the RHA served all scheduled trains and did not break any train-to-train connections.

The RHA uses fewer deadheads than the optimal solution. The RHA provides a reduction of at least 4.13% and, at most 31.41% in terms of the number of deadheads. As a result, our algorithm decreases the deadhead cost, the total distance, the deadhead distance when compared with the exact solution. Although the gap of the number of light travels between two methods is high ($> 60\%$), the average values of light travel distance are decreased by at least 16.50% and at most 32.70%. Recall that one of the purposes of the deadheads and the light travels is to balance the power availability across the network. Because of using fewer deadheads and the small number of added light travels, the railroad has to lease more third-party locomotives to avoid the shortage of power requirements at stations. The number of leased locomotives increases at most 44.49% with the setting $h = 3$, $r = 2$ and at least 7.63% with setting $h = 5$, $r = 2$. The number of unserved critical locomotives is lower than the ones in the optimal solutions because some critical locomotives are on the way to the shops or waiting for spots at the shop stations at the end of the last iterations of the RHA.

To explain those analyses, we consider the impacts of changing the planning horizon's length on the result while keeping the roll period the same. Note that the deadhead arcs connecting the intermediate stations in the route of the trains are built based on the train arcs. When the length of the planning horizon increases, the number of deadheads, the total deadhead distance, the average deadhead distance, and the cost of deadheads also increase since we consider more scheduled trains in the input. Consequently, including more trains in the data creates more opportunities for assigning deadhead locomotives.

Some effects on the results follow the increase in the number of deadheads. Because of the expensive cost of light travel and third-party leasing locomotives, the IP model aims to use as many deadheads as possible to re-position locomotives among stations or sending critical locomotives to the shops. Therefore, first, the number of leased locomotives decreases since more deadheads are used to balance the power requirements at stations. Furthermore, more deadheading locomotives decreases the number of idling (unused) locomotives and the idling cost.

The comparisons of the other 11 instances are presented in the Appendix. The analysis shows that their characteristics are similar to the first one. Additionally, the setting $h = 5$, $r = 2$ provides the best solutions when solving one-week instances, although it does not obtain the best performance in running time.

5.3. The Solution Quality and Computational Time for Longer Time Horizons

We also run experiments on ten instances with Algorithm 4.3.2, gradually increasing the planning horizon from 8 to 14 days, with the setting $h = 5$, $r = 2$. A time limit of 12 hours is set in all cases. The IP model is able to optimally solve all instances with planning horizons of up to 10 days. However, with planning horizons of 11, 12, 13, 14 days, the numbers of optimal solutions provided by the exact method are 8, 3, 1, 0, respectively and there are 2, 5, 2, 1 near-optimal solution ($\leq 0.53\%$ gap). From the analyses of the results of the 7-day instances mentioned above, the planning horizon and the roll period used in the RHA will be set to 5 days and 2 days. The detailed comparison is presented in Tables 5.3 - 5.7.

Table 5.3 presents the average of the final objective function values provided by the RHA and the average values of the relaxation solutions' objective (lower bounds) obtained by the exact model when we experiment with ten instances of each planning horizon. The relative differences range between 6.09% and 17.89%.

Instances	**Relaxation (Ave.Obj)**	**RHA (Ave.Obj)**	**Relative difference (%)**
8-day	22157549.28	23508096.24	6.09
9-day	25051852.31	29127733.54	16.27
10-day	27913620.15	31288397.68	12.09
11-day	30826627.92	36340349.56	17.89
12-day	33729596.32	37918416.71	12.42
13-day	36660814.81	42700158.50	16.47
14-day	39429913.46	44913047.15	13.91

Table 5.3. Comparison of the objective values of the relaxation solutions provided by CPLEX with the objective values provided by the RHA

Tables 5.4 - 5.7 present the comparison of feasible solutions found by the two methods. In those tables, “Cost” is the average operational cost (objective value). “Distance” presents the average distance that locomotives pull the trains, light travel (“Light distance”) and travel as deadheads (“Deadhead distance”). “# deadheads”, “# light travels”, “# locos shopped” and “# leased locos” stand for the average number of locomotives used as deadheads, light traveling, served in shops and leased from third-party companies. The column “RHA”, “IP”, and “Relative difference” present the result of the RHA, the exact method, and the relative difference of the results, respectively. Note that we only summarize the detailed values of feasible solutions provided by the IP model. For instance, if the exact method solves to optimal 3/10 instances and near-optimal 5/10 instances, the average values compared with the ones in the RHA will be calculated based on 8 solutions (see the value of the row “# feasible sols” and “# sols compared”).

The following findings emerge. The computing time decreases at least 52.09% (see the case of 8-day instances in Table 5.4), while the total operation cost increases at most 17.30% (see the case of 11-day instances in Table 5.5). The comparison also shows that the RHA can provide acceptable solutions with instances containing from 8 to 14 days train schedule. The lowest relative difference in the total cost between the two methods is 5.63% in the case of 8-day instances (see in Table 5.4), and the highest one is 17.30% for 11-day instances (see in Table 5.5). We should note that the total cost performed by the RHA includes an amount of leasing cost, while the exact method ignores that cost. We set a penalty for leasing a third-party locomotive. That number is estimated based on the fixed cost per light travel and the limit on distance of each light travel to ensure that the penalty for leasing locomotives is not lower than the lowest cost paid for re-positioning locomotives by using light arcs. In the worst case where the relative difference between the number of leased locomotives used in the RHA and the one provided by the exact method is 67.43% (see the case of 14-day instances in Table 5.7), the leasing cost, which must be added in real-life operation, does not significantly affect the total cost.

As mentioned in Section 5.2, the RHA always provides a lower number of deadheads and deadhead distance than the exact method. Because of the high percentage, the less deadhead distance decreases the total traveling distance. The number of light travels and leased locomotives performed by the RHA are higher than the ones in the result of the exact method. About the number of light travel, the highest relative difference is 51.80% in the case of 12-day instances (see in Table 5.6).

We also observe that when the length of the whole planning horizon increases, the relative differences in the number of leased locomotives between two methods increases. For 14-day instances (see Table 5.7), the setting $h = 5$, $s = 2$ performs the worst relative difference in term of the number of leased locomotives. To explain that observation, we remind the way leased locomotives are created at each iteration of the RHA. At the beginning of each planning horizon, the IP model decides to lease a number of third-party locomotives at each source node based on the lack of power requirement at the corresponding station. For this reason, the RHA must lease new locomotives iteration by iteration instead of leasing locomotives only once as the exact method does.

By observing the real-life instances, we realize that when the length of the planning horizon increases, the number of critical locomotives and train-to-train connections also increase. For two instances that require the same number of iterations to solve, each iteration of the larger one can require more leased locomotives than the corresponding iteration of the smaller one. This phenomenon happens because there are more critical locomotives that can not pull trains when they are maintaining in the shops; or more scheduled train in the larger instance than in the small one.

While the exact method considers the whole planning horizon, each iteration of the RHA takes into account a smaller time horizon. Hence, the reuse of leased locomotives in the RHA is more restrictive than in the IP-based method because of the RHA's short-term overview. The RHA leases new third-party locomotives at the beginning of every roll period and it provides a greater total number of leased locomotives than the exact method does. These relative differences shown in Table 5.4 - 5.7 are greater than or equal 20.48%.

We realize that the connectivity between two space-time networks in two consecutive iterations of the RHA is not well built enough to conveniently re-position locomotives by using as many deadheads as possible. For instance, on the path of the scheduled train T, there are several intermediate stations, namely sA, sB, sC, sD, sE. Assume that the train T can pick up or release deadhead locomotives at each station mentioned above and T arrives at the station sD and sE on the overlap period. The exact method can make a decision that some locomotives can be attached to or busted from the train T at the station sD, sE. Otherwise, in the term of the RHA, the opportunity of attaching or busting locomotives at those stations are ignored because all activities happen during the overlap. Therefore, the leased locomotives are reused efficiently in the exact method by only using deadheads, while those in the RHA can be impractical in the later sub-instances because the costs of re-positioning them by using combinations of light travels and deadheads are almost higher than the cost of leasing the new ones.

	8-day instances			9-day instances		
	RHA (Ave.Value)	**IP (Ave.Value)**	**Relative difference (%)**	**RHA (Ave.Value)**	**IP (Ave.Value)**	**Relative difference (%)**
# feasible sols	10/10	10/10	-	10/10	10/10	-
Time (s)	502.41	1048.71	-52.09	721.76	3328.80	-78.32
Cost	23508096.24	22254009.42	5.63	29127733.54	25167658.90	15.73
Distance	2868269.30	2889800.80	-0.74	3249322.50	3272228.90	-0.70
Deadhead distance	187980.80	210500.90	-10.70	233370.10	257189.50	-9.26
Light distance	3418.60	2430.00	40.68	3806.70	2893.70	31.55
# deadheads	871.31	949.90	-8.27	1041.90	1133.50	-8.08
# light travels	91.41	64.60	41.49	111.81	77.90	43.52
# locos shopped	97.70	104.10	-6.15	112.70	117.00	-3.67
# leased locos	294.70	244.60	20.48	309.00	250.60	23.30

Table 5.4. Comparison of the results of instances with 8,9-day planning horizon solved by the RHA with the exact solution of the IP-based model

	10-day instances			11-day instances		
	RHA (Ave.Value)	**IP (Ave.Value)**	**Relative difference (%)**	**RHA (Ave.Value)**	**IP (Ave.Value)**	**Relative difference (%)**
# feasible sols	10/10	10/10	-	10/10	10/10	-
Time (s)	902.73	8758.80	-89.69	939.59	19916.28	-95.28
Cost	31288397.68	28052976.32	11.46	36340349.56	30983695.45	17.30
Distance	3608602.00	3648548.30	-0.95	3991923.30	4037556.10	-1.13
Deadhead distance	261534.40	298773.40	-12.14	298080.10	345446.10	-13.71
Light distance	4991.20	3452.60	41.94	5181.00	3447.80	50.27
# deadheads	1190.00	1305.10	-8.75	1344.00	1499.40	-10.36
# light travels	133.40	95.20	39.18	143.70	96.60	48.76
# locos shopped	125.00	129.20	-3.33	140.50	142.00	-1.06
# leased locos	329.60	250.60	31.13	355.70	262.80	35.35

Table 5.5. Comparison of the results of instances with 10,11-day planning horizon solved by the RHA with the exact solution of the IP-based model

	12-day instances			13-day instances		
	RHA (Ave.Value)	**IP (Ave.Value)**	**Relative difference (%)**	**RHA (Ave.Value)**	**IP (Ave.Value)**	**Relative difference (%)**
# feasible sols	10/10	8/10	-	10/10	3/10	-
# sols compared	8	8	-	3	3	-
Time (s)	893.44	34688.85	-97.42	937.61	41653.43	-97.75
Cost	37883147.11	33883815.27	11.80	41923703.13	36812046.15	13.89
Distance	4362667.25	4418672.63	-1.27	4278981.33	4395229.67	-2.65
Deadhead distance	329552.25	387988.88	-15.06	349891.67	431789.33	-18.97
Light distance	5639.00	3207.75	75.79	6518.00	3819.33	70.66
# deadheads	1507.38	1675.88	-10.05	1596.33	1821.67	-12.37
# light travels	153.13	100.88	51.80	173.00	120.67	43.37
# locos shopped	152.25	153.88	-1.06	141.00	158.33	-10.95
# leased locos	379.00	274.50	38.07	353.00	272.67	29.46

Table 5.6. Comparison of the results of instances with 12,13-day planning horizon solved by the RHA with the exact solution of the IP-based model

	14-day instances		
	RHA (Ave.Value)	**IP (Ave.Value)**	**Relative difference (%)**
# feasible sols	10/10	1/10	-
# sols compared	1	1	-
Time (s)	905.73	43299.47	-97.91
Cost	44784887.38	39195897.44	14.26
Distance	5021773.00	5126379.00	-2.04
Deadhead distance	389457.00	496925.00	-21.63
Light distance	6077.00	3215.00	89.02
# deadheads	1716.00	2028.00	-15.39
# light travels	145.00	136.00	6.62
# locos shopped	168.00	169.00	-0.59
# leased locos	437.00	261.00	67.43

Table 5.7. Comparison of the results of instances with 14-day planning horizon solved by the RHA with the exact solution of the IP-based model

Conclusion & Future Work

In this thesis, we study an essential part of locomotive scheduling problem, which is known as the locomotive routing problem (LRP). The LRP arising at CN is a large scale optimization problem with more than 3900 trains and 1900 locomotives per week. Based on the research of Miranda et al. (2020), we propose a modified space-time network wherein a new type of arc is introduced to deal with "unfinished" train-to-train connections. While all train-to-train connections contains two trains departing at the same planning horizon in Miranda et al. (2020), we need to consider new cases of train-to-train connections including two trains operating in different planning horizons. The changes in the space-time network lead to the adaptations in the IP model. New constraints are added to the mathematical formulation provided in Miranda et al. (2020) in order to cope with the conflicts of train-to-train connections and maintenance conditions. The new network and the model are applied in the RHA framework to solve the problem. Our method significantly reduces the computation time (the maximum relative difference is 97.91% in the case of 14-day instances) and provides results at the expense of a reduced solution quantity (the maximum relative difference is 17.30%) when compared with the exact solution in Miranda et al. (2020).

Future work perspectives are numerous. First, although the model provides quite good solutions, the total light traveling distance and the number of light travels are significantly larger than the ones in the optimal solution. Second, we believe that the parameters presenting the cost of third-party leasing locomotives should be controlled in a better way in order to decrease the number of leased locomotives. Another technique to deal with the issue mentioned above is that at the end of each planning horizon at each station, the minimum power requirement, which can be estimated based on historical experiments by CN, should be re-defined instead of setting equal to zero. In addition, minimizing the relative difference in number of locomotives between the solution provided by the RHA and stations' power requirements, which could be calculated based on the number of scheduled trains, also can decrease the number of leased locomotives. Finally, our RHA cannot serve all critical locomotives during the considering week. Thus, we need to figure out how to control the penalty cost corresponding to maintenance requests.

References

Ahuja, R. K., Cunha, C. B. and Sahin, G. (2005a). Network Models in Railroad Planning and Scheduling, *INFORMS Tutorials in Operations Research, 3: 54-101* .

Ahuja, R. K., Liu, J., Orlin, J. B., Sharma, D. and Shughart, L. A. (2005b). Solving Real-Life Locomotive-Scheduling Problems, *Transportation Science, 39(4): 503-517* .

Ahuja, R. K., Magnanti, T. and Orlin, J. (1993). Network Flows: Theory, Algorithms, and Applications, *Prentice Hall, New Jersey* .

Araujo, S., Arenales, M. and Clark, A. (2007). Joint Rolling-Horizon Scheduling of Materials Processing and Lot-sizing with Sequence-dependent Setups, *Journal of Heuristics, 13: 337–358* .

Baker, K. (1977). An Experimental Study of the Effectiveness of Rolling Schedules in Production Planning, *Decision Sciences, 8: 19–27* .

Baker, K. and Peterson, D. (1979). An Analytic Framework for Evaluating Rolling Schedules, *Management Science, 25: 341–351* .

Bouzaiene-Ayari, B., Cheng, C., Das, S., Fiorillo, R. and Powell, W. B. (2016). From Single Commodity to Multiattribute Models for Locomotive Optimization: A Comparison of Optimal Integer Programming and Approximate Dynamic Programming, *Transportation Science, 50(2): 366-389.* .

Chand, S., Hsu, V. N. and Sethi, S. (2002). Forecast, Solution, and Rolling Horizons in Operations Management Problems: a Classified Bibliography, *Manufacturing & Service Operations Management, 4(1): 25–43* .

Cordeau, J.-F., Desaulniers, G., Lingaya, N., Soumis, F. and Desrosiers, J. (2001). Simultaneous Locomotive and Car Assignment at VIA Rail Canada, *Transportation Research Part B: Methodological, 35: 767–787* .

Cordeau, J.-F., Soumis, F. and Desrosiers, J. (2000). A Benders Decomposition Approach for the Locomotive and Car Assignment Problem, *Transportation Science, 34: 133–149* .

Cordeau, J.-F., Toth, P., and Vigo, D. (1998). A Survey of Optimization Models for Train Routing and Scheduling, *Transportation Science, 32(4): 380-404* .

D'Ariano, A. (2009). Innovative Decision Support System for Railway Traffic Control, *IEEE Intelligent Transportation Systems Magazine, 1(4): 8-16* .

Dimitriadis, A., Shah, N. and Pantelides, C. (1997). RTN-based Rolling Horizon Algorithms for Medium Term Scheduling of Multipurpose Plants, *Computers & Chemical Engineering, 21: 1061–1066* .

Florian, M., Bushell, G., Ferland, J., Guerin, G. and Nastansky, L. (1976). The Engine Scheduling Problem in a Railway Network, *INFOR 14: 121–138* .

Forbes, M. A., Holt, J. N. and Watts, J. N. (1991). Exact Solution of Locomotive Scheduling Problems, *Journal of the Operational Research Society, 42: 825–831* .

Fugenschuh, A., Homfeld, H.and Huck, A., Martin, A. and Yuan, Z. (2006). Scheduling Locomotives and Car Transfers in Freight Transport, *Transportation Science, Volume 42: 478–491* .

Lai, Y.-C., Barkan, C. and Önal, H. (2008a). Optimizing the Aerodynamic Efficiency of Intermodal Freight Trains, *Transportation Research Part E, 44: 820–834* .

Lai, Y.-C., Ouyang, Y. and Barkan, C. (2008b). A Rolling Horizon Model to Optimize Aerodynamic Efficiency of Intermodal Freight Trains with Uncertainty, *Transportation Science, 42(4): 466–477* .

Mercé, C. and Fontan, G. (2003). MIP-based Heuristics for Capacitated Lotsizing Problems, *International Journal of Production Economics, 85: 97–111* .

Miranda, P., Frejinger, E. and Cordeau, J.-F. (2020). A Time-Space Formulation for the Locomotive Routing Problem at the Canadian National Railways, *Technical report 2020-19, CIRRELT, Montreal* .

Nielsen, L. K., Kroon, L. and Maróti, G. (2012). A Rolling Horizon Approach for Disruption Management of Railway Rolling Stock, *European Journal of Operational Research, 220(2): 496–509* .

Ortiz-Astorquiza, C., Frejinger, E. and Cordeau, J.-F. (2019). The Locomotive Assignment Problem with Distributed Power at the Canadian National Railway Company, *Technical report 2019-55, CIRRELT, Montreal* .

Pellegrini, P., Marlière, G. and Rodriguez, J. (2014). Optimal Train Routing and Scheduling for Managing Traffic Perturbations in Complex Junctions, *Transportation Research Part B: Methodological, 59: 58–80* .

Piu, F. and Speranza, M. G. (2014). The Locomotive Assignment Problem: A Survey on Optimization Models, *International Transactions in Operational Research, Volume 21(3): 327–352* .

Quaglietta, E., Corman, F. and Goverde, R. M. (2013). Impact of a Stochastic and Dynamic Setting on the Stability of Railway Dispatching Solutions, *2013 16th International IEEE Conference on Intelligent Transportation Systems-(ITSC). IEEE: 1035–1040* .

Quaglietta, E. and Punzo, V. (2013). Supporting the Design of Railway Systems by Means of a Sobol Variance-based Sensitivity Analysis, *Transportation Research Part C: Emerging Technologies, 34: 38–54* .

Samà, M., D'Ariano, A. and Pacciarelli, D. (2013). Rolling Horizon Approach for Aircraft Scheduling in the Terminal Control Area of Busy Airports, *Procedia - Social and Behavioral Sciences, 80: 531-552* .

Stauffer, L. and Liebling, T. (1997). Rolling Horizon Scheduling in a Rolling-mill, *Annals of Operations Research, 69: 323–349* .

Vaidyanathan, B., Ahuja, R. K., Liu, J. and Shughart, L. A. (2008a). Real-life locomotive planning: New Formulations and Computational Results, *Transportation Research Part B: Methodological, 42(2): 147-168* .

Vaidyanathan, B., Ahuja, R. K. and Orlin, J. B. (2008b). The Locomotive Routing Problem, *Transportation Science, 42(4): 492-507* .

Wright, M. B. (1989). Applying Stochastic Algorithms to a Locomotive Scheduling Problem, *Journal of the Operational Research Society, 40: 187–192.* .

Zhan, S., Kroon, L. G., Zhao, J. and Peng, Q. (2016). A Rolling Horizon Approach to the High Speed Train Rescheduling Problem in case of a Partial Segment Blockage, *Transportation Research Part E 95: 32–61* .

Ziarati, K., Soumis, F., Desrosiers, J., Gelinas, S., and Saintonge, A. (1997). Locomotive Assignment with Heterogeneous Consists at CN North America, *European Journal of Operational Research, 97(2): 281-292* .

Ziarati, K., Soumis, F., Desrosiers, J. and Solomon, M. M. (1999). A Branch-First, Cut-Second Approach for Locomotive Assignment, *Management Science, 45(8): 1156–1168* .

Appendix A

Detailed Experimental Results

	Week_1	Week_2	Week_3	Week_4	Week_5	Week_6
Time	557.978	445.192	561.253	407.138	593.001	556.789
Total_Cost	19537668.18	19330052.64	19275814.21	19243999.74	19516367.16	19387953.62
Pulling_Cost	9354287.142	9325657.846	9382540.755	9136311.747	9299484.715	9348296.13
Dead_Cost	656919.795	554403.134	617216.259	617868.512	519623.374	527450.564
Idle_Cost	8022426.662	8421801.951	8101345.335	8245094.988	8195859.271	8369921.965
Light_Cost	574410.745	305339.579	286659.754	400326.691	663824.684	371869.984
Shop_Cost	724869.484	510245.403	702701.48	649490.566	635292.99	573982.512
Lease_Cost	0	0	0	0	0	0
Tr2Tr_Cost	204754.355	212604.726	185350.623	194907.237	202282.128	196432.463
Total_distance	2600685	2548376	2527805	2469379	2471468	2481261
Total_pulling_distance	2413368	2397494	2346980	2299298	2326941	2334594
Pulling_distance(%)	92.797	94.079	92.847	93.112	94.152	94.089
Total_deadhead_distance	185888	150055	179907	169257	142985	145622
Deadhead_distance(%)	7.148	5.888	7.117	6.854	5.785	5.869
Average_deadhead_distance	232.651	210.456	242.462	212.902	212.459	201.414
Total_light_distance	1429	827	918	824	1542	1045
Light_distance(%)	0.055	0.032	0.036	0.033	0.062	0.042
Average_light_distance	39.694	21.763	31.655	34.333	35.045	22.717
n_deadhead_locos	799	713	742	795	673	723
n_light_locos	36	38	29	24	44	46
n_locos_due	99	79	89	91	86	92
n_locos_shopped	99	79	88	91	86	92
n_leased_locos	236	244	226	263	257	270
n_tr2tr_conn	529	528	491	522	497	485

Table A.1. Optimal solutions of first 6 weeks

	Week_7	**Week_8**	**Week_9**	**Week_10**	**Week_11**	**Week_12**
Time	317.988	585.431	482.192	587.562	420.515	526.354
Total_Cost	19223278.8	19626956.97	19196240.3	19391815.7	19032227.5	19718313.3
Pulling_Cost	9354549.272	9478566.807	9347401.419	9459984.673	9425821.928	9557495.516
Dead_Cost	514619.545	570453.031	483613.31	519207.154	617988.653	561878.858
Idle_Cost	8054711.846	8133152.216	8195967.935	8099671.592	7747625.626	8015992.197
Light_Cost	533317.187	568609.98	444703.326	517269.732	506584.338	611396.021
Shop_Cost	577910.246	664370.326	488131.276	584413.554	531596.113	756126.593
Lease_Cost	0	0	0	0	0	0
Tr2Tr_Cost	188170.698	211804.614	236423.019	211268.973	202610.804	215424.065
Total_distance	2488261	2532821	2460282	2483389	2476233	2513738
Total_pulling_distance	2346011	2376298	2326919	2345567	2304732	2352028
Pulling_distance(%)	94.283	93.82	94.579	94.45	93.074	93.567
Total_deadhead_distance	141139	155241	132226	136582	170278	159614
Deadhead_distance(%)	5.672	6.129	5.374	5.5	6.876	6.35
Average_deadhead_distance	195.755	197.759	210.551	188.129	222.877	216.867
Total_light_distance	1111	1282	1137	1240	1223	2096
Light_distance(%)	0.045	0.051	0.046	0.05	0.049	0.083
Average_light_distance	25.25	24.189	35.531	41.333	31.359	41.92
n_deadhead_locos	721	785	628	726	764	736
n_light_locos	44	53	32	30	39	50
n_locos_due	92	99	83	102	79	108
n_locos_shopped	91	99	82	101	78	107
n_leased_locos	222	258	274	272	255	313
n_tr2tr_conn	493	523	588	513	502	531

Table A.2. Optimal solutions of last 6 weeks

	Week_2	**Week_3**	**Week_4**	**Week_5**	**Week_6**	**Week_7**	**Week_8**	**Week_9**	**Week_10**	**Week_11**	**Week_12**
Time	-85.41	-84.67	-76.30	-89.99	-91.53	-85.45	-91.63	-91.82	-89.85	-84.44	-89.26
Total_Cost	20.60	20.78	20.72	18.04	18.36	18.86	20.41	18.47	20.89	19.03	20.03
Pulling_Cost	0.10	0.09	0.08	0.09	0.09	0.10	0.10	0.12	0.11	0.08	0.10
Dead_Cost	-52.85	-47.10	-51.18	-41.21	-43.39	-45.75	-49.83	-46.76	-41.44	-42.28	-50.64
Idle_Cost	26.29	25.47	25.52	23.90	21.92	23.95	25.94	24.68	22.12	26.89	25.20
Light_Cost	159.80	275.61	123.49	28.89	92.32	107.32	84.71	23.58	180.29	28.12	57.02
Shop_Cost	-4.18	-3.63	0.07	-7.58	-3.82	-1.19	0.72	-1.44	-5.41	1.59	-4.14
Tr2Tr_Cost	-0.67	-0.82	-0.51	-0.74	-0.47	-0.71	-0.36	-0.64	-0.69	-0.36	-0.57
Total_distance	-2.95	-3.38	-3.39	-2.24	-2.50	-2.57	-2.82	-2.41	-2.06	-2.95	-3.20
Total_deadhead_distance	-50.36	-47.87	-49.61	-38.60	-42.70	-45.70	-46.47	-44.72	-38.13	-43.00	-50.24
Average_deadhead_distance	-14.72	-26.32	-22.06	-9.97	-14.76	-17.23	-9.64	-19.64	-4.84	-11.30	-17.70
Total_light_distance	33.74	68.63	26.94	-18.35	6.51	42.75	61.23	-23.66	82.42	11.37	-14.79
Average_light_distance	-13.86	-7.73	-39.07	-10.19	-30.01	-15.12	37.83	-41.83	-13.13	-19.57	-32.37
n_deadhead_locos	-41.80	-29.25	-35.35	-31.80	-32.78	-34.40	-40.76	-31.21	-34.99	-35.73	-39.54
n_light_locos	55.26	82.76	108.33	-9.09	52.17	68.18	16.98	31.25	110.00	38.46	26.00
n_leased_locos	51.23	50.44	50.19	47.08	40.00	42.34	51.94	39.42	33.82	49.80	40.26

Table A.3. The relative difference of two methods with $h = 3, r = 2$ (%)

	Week_2	Week_3	Week_4	Week_5	Week_6	Week_7	Week_8	Week_9	Week_10	Week_11	Week_12
Time	-70.51	-66.64	-67.13	-63.09	-65.24	-46.04	-69.65	-75.24	-70.60	-76.61	-72.59
Total_Cost	16.55	14.74	15.38	13.82	13.49	13.40	16.03	14.28	16.83	14.32	15.34
Pulling_Cost	0.11	0.09	0.09	0.09	0.09	0.09	0.11	0.12	0.12	0.08	0.12
Dead_Cost	-28.39	-25.50	-28.92	-15.11	-24.21	-19.73	-30.35	-27.88	-22.65	-21.50	-16.26
Idle_Cost	19.10	21.20	20.25	17.79	17.71	20.15	20.67	19.47	18.16	21.55	19.03
Light_Cost	251.60	122.03	86.06	46.47	54.27	24.64	93.96	42.35	165.84	30.50	59.71
Shop_Cost	-1.78	-5.93	0.41	-3.30	-4.20	0.37	-1.58	0.44	-4.91	-2.28	-5.08
Tr2Tr_Cost	-0.66	-0.79	-0.50	-0.64	-0.45	-0.71	-0.32	-0.56	-0.62	-0.35	-0.47
Total_distance	-1.48	-1.88	-1.88	-0.83	-1.39	-0.99	-1.85	-1.45	-1.13	-1.73	-1.01
Total_deadhead_distance	-25.80	-26.63	-27.63	-14.51	-23.76	-17.47	-30.97	-26.95	-21.13	-25.28	-15.76
Average_deadhead_distance	-6.69	-16.25	-9.40	-0.28	-10.52	-0.50	-4.09	-12.62	-2.29	-9.68	-0.32
Total_light_distance	120.56	36.49	41.14	14.59	13.88	1.08	98.28	1.93	65.16	12.26	-9.06
Average_light_distance	7.45	1.49	-23.01	-13.07	-27.24	-12.79	33.02	-37.27	-28.19	-18.92	-24.22
n_deadhead_locos	-20.48	-12.40	-20.13	-14.26	-14.80	-17.06	-28.03	-16.40	-19.28	-17.28	-15.49
n_light_locos	105.26	34.48	83.33	31.82	56.52	15.91	49.06	62.50	130.00	38.46	20.00
n_leased_locos	21.72	29.65	28.52	20.62	20.74	24.77	29.84	19.34	19.49	23.53	21.41

Table A.4. The relative difference of two methods with $h = 4, r = 2$ (%)

	Week_2	Week_3	Week_4	Week_5	Week_6	Week_7	Week_8	Week_9	Week_10	Week_11	Week_12
Time	-13.08	-38.47	20.06	-34.90	-26.66	13.81	-1.75	-18.71	10.69	-9.54	-30.12
Total_Cost	12.27	10.83	12.28	10.67	10.99	10.74	12.63	13.11	13.31	11.78	10.43
Pulling_Cost	0.10	0.09	0.09	0.09	0.10	0.09	0.11	0.13	0.11	0.09	0.11
Dead_Cost	-21.08	-14.22	-13.91	-13.18	-4.22	-11.09	-13.97	-12.31	-14.35	-10.12	-8.78
Idle_Cost	18.38	19.36	17.84	15.68	15.16	17.66	17.58	17.50	16.78	18.48	17.25
Light_Cost	91.25	-7.32	67.50	34.00	51.38	23.12	83.84	99.06	119.81	42.79	-12.43
Shop_Cost	-1.02	-1.92	-1.48	-1.67	0.70	-0.49	-4.34	0.60	-5.84	-0.46	-2.92
Tr2Tr_Cost	-0.65	-0.79	-0.49	-0.64	-0.43	-0.60	-0.32	-0.53	-0.63	-0.35	-0.47
Total_distance	-1.21	-1.12	-0.83	-0.71	-0.18	-0.54	-0.78	-0.63	-0.81	-0.88	-0.67
Total_deadhead_distance	-20.87	-15.62	-12.48	-12.78	-3.48	-9.43	-13.44	-12.66	-15.29	-13.32	-10.13
Average_deadhead_distance	-7.51	-9.91	-3.63	-3.93	-1.99	-1.81	2.64	-7.97	-7.52	-4.71	-2.59
Total_light_distance	46.67	-17.43	78.28	53.37	49.86	-6.75	96.72	98.59	61.94	69.66	-31.01
Average_light_distance	-24.68	-20.18	-4.92	-7.56	2.89	-31.62	22.66	7.71	-19.03	6.72	-40.53
n_deadhead_locos	-14.45	-6.33	-9.18	-9.21	-1.52	-7.77	-15.67	-5.10	-8.40	-9.03	-7.74
n_light_locos	94.74	3.45	87.50	65.91	45.65	36.36	60.38	84.38	100.00	58.97	16.00
n_leased_locos	17.21	22.12	17.49	8.95	9.63	11.71	13.18	9.12	12.13	10.98	12.78

Table A.5. The relative difference of two methods with $h = 5, r = 2$ (%)

	Week_2	Week_3	Week_4	Week_5	Week_6	Week_7	Week_8	Week_9	Week_10	Week_11	Week_12
Time	-75.01	-69.04	-61.64	-67.87	-72.88	-36.05	-76.67	-79.57	-73.94	-83.14	-81.15
Total_Cost	26.24	26.07	25.87	24.01	25.21	23.68	26.19	24.93	25.24	25.04	26.71
Pulling_Cost	0.11	0.08	0.08	0.09	0.09	0.09	0.10	0.13	0.12	0.08	0.12
Dead_Cost	-32.89	-34.94	-28.21	-29.51	-33.42	-19.75	-30.74	-25.23	-29.93	-30.86	-29.09
Idle_Cost	44.40	48.03	44.30	46.19	44.51	45.77	45.58	44.66	43.48	49.50	46.43
Light_Cost	178.17	132.07	109.52	-5.47	53.84	12.15	93.39	37.44	89.67	6.76	67.38
Shop_Cost	-4.72	-6.49	0.87	-2.62	0.84	-4.41	-0.08	1.01	-0.27	-1.61	-0.61
Tr2Tr_Cost	-0.50	-0.50	-0.51	-0.49	-0.28	-0.34	-0.24	-0.49	-0.42	-0.31	-0.31
Total_distance	-1.90	-2.61	-1.86	-1.70	-1.89	-1.09	-1.86	-1.31	-1.63	-2.34	-1.89
Total_deadhead_distance	-32.63	-36.93	-27.46	-29.13	-32.25	-19.03	-31.02	-24.39	-29.95	-33.93	-29.71
Average_deadhead_distance	-15.44	-22.27	-6.53	-10.17	-11.58	-5.54	-8.84	-11.74	-9.19	-14.30	-5.07
Total_light_distance	54.41	37.80	67.23	-28.27	2.68	-15.12	85.57	10.99	35.40	-8.18	-10.07
Average_light_distance	-23.80	-9.18	-40.98	-23.03	-34.40	-23.78	24.49	-27.51	-45.84	-12.66	-31.87
n_deadhead_locos	-20.34	-18.87	-22.39	-21.10	-23.37	-14.29	-24.33	-14.33	-22.87	-22.91	-25.95
n_light_locos	102.63	51.72	183.33	-6.82	56.52	11.36	49.06	53.13	150.00	5.13	32.00
n_leased_locos	22.54	35.40	20.91	29.57	26.30	25.68	25.58	17.88	17.65	29.41	24.92

Table A.6. The relative difference of two methods with $h = 4, r = 3$ (%)

	Week_2	Week_3	Week_4	Week_5	Week_6	Week_7	Week_8	Week_9	Week_10	Week_11	Week_12
Time	-45.92	-55.40	4.49	-6.57	-36.51	-19.02	-51.30	-52.50	-41.66	-50.16	-44.52
Total_Cost	22.36	22.70	22.42	20.46	21.84	20.78	22.71	22.14	21.89	21.65	21.07
Pulling_Cost	0.10	0.09	0.09	0.09	0.09	0.08	0.11	0.13	0.11	0.08	0.11
Dead_Cost	-21.18	-20.16	-14.42	-10.72	-13.23	-14.41	-14.35	-14.06	-13.75	-14.20	-13.74
Idle_Cost	42.76	43.75	41.77	39.90	40.71	41.96	42.04	43.40	40.20	43.22	43.56
Light_Cost	54.12	114.58	66.73	20.61	49.50	28.03	76.03	10.73	74.34	41.15	-15.83
Shop_Cost	1.61	0.33	0.76	0.09	1.37	-5.06	0.49	-1.89	-0.88	0.86	-0.73
Tr2Tr_Cost	-0.50	-0.50	-0.48	-0.48	-0.28	-0.34	-0.24	-0.39	-0.38	-0.24	-0.33
Total_distance	-1.20	-1.54	-0.93	-0.56	-0.80	-0.77	-0.79	-0.77	-0.77	-1.18	-0.95
Total_deadhead_distance	-20.57	-21.76	-13.99	-9.97	-13.82	-13.51	-13.78	-14.60	-14.44	-17.53	-14.62
Average_deadhead_distance	-9.38	-12.96	-4.10	-1.00	-6.59	-3.92	-0.18	-8.64	-7.29	-9.34	-2.42
Total_light_distance	43.17	25.49	80.46	22.11	33.97	-9.72	99.06	29.64	40.08	57.56	-32.16
Average_light_distance	-7.79	-19.13	-24.02	-13.34	-10.68	-36.95	33.55	9.17	-39.09	-0.89	-38.32
n_deadhead_locos	-12.34	-10.11	-10.31	-9.06	-7.75	-9.99	-13.63	-6.53	-7.71	-9.03	-12.50
n_light_locos	55.26	55.17	137.50	40.91	50.00	43.18	49.06	18.75	130.00	58.97	10.00
n_leased_locos	17.21	18.58	12.55	7.39	12.22	9.91	13.57	12.77	7.35	9.02	15.65

Table A.7. The relative difference of two methods with $h = 5, r = 3$ (%)

	Week_2	Week_3	Week_4	Week_5	Week_6	Week_7	Week_8	Week_9	Week_10	Week_11	Week_12
Time	-70.40	-67.92	70.52	-62.01	-40.40	-10.86	-47.09	-64.06	-38.94	-55.18	-49.83
Total_Cost	12.61	15.16	13.94	12.02	12.89	12.10	13.86	12.80	13.47	13.05	11.86
Pulling_Cost	0.10	0.09	0.09	0.09	0.09	0.09	0.11	0.13	0.10	0.09	0.10
Dead_Cost	-26.27	-28.36	-32.29	-21.20	-22.65	-21.85	-20.87	-21.23	-18.66	-24.42	-19.34
Idle_Cost	20.15	22.81	21.10	18.15	19.15	19.78	20.24	20.71	17.64	21.61	20.26
Light_Cost	72.57	196.12	101.62	48.45	75.81	49.82	87.84	34.44	117.30	51.96	0.04
Shop_Cost	-0.43	0.75	0.74	-1.35	-1.16	-1.06	-2.45	-3.66	-2.57	1.24	-2.32
Tr2Tr_Cost	-0.24	-0.38	-0.33	-0.34	-0.08	-0.43	-0.20	-0.20	-0.25	-0.21	-0.20
Total_distance	-1.33	-2.09	-2.21	-1.19	-1.28	-1.18	-1.14	-1.15	-1.02	-1.72	-1.27
Total_deadhead_distance	-23.04	-29.72	-32.69	-21.23	-22.42	-21.02	-19.33	-22.09	-19.16	-25.72	-19.68
Average_deadhead_distance	-6.68	-16.30	-18.55	-6.99	-10.40	-4.62	-0.12	-12.00	-5.94	-10.92	-4.96
Total_light_distance	72.67	78.10	105.46	57.39	83.25	16.83	94.62	77.22	76.37	99.02	-19.27
Average_light_distance	-0.58	9.89	-21.73	-18.52	8.07	-34.09	17.21	11.20	-33.02	14.14	-39.76
n_deadhead_locos	-17.53	-16.04	-17.36	-15.30	-13.42	-17.20	-19.24	-11.46	-14.05	-16.62	-15.49
n_light_locos	73.68	62.07	162.50	93.18	69.57	77.27	66.04	59.38	163.33	74.36	34.00
n_leased_locos	17.21	29.65	19.39	9.73	17.78	13.06	17.83	13.50	8.46	16.47	15.34

Table A.8. The relative difference of two methods with $h = 5, r = 4$ (%)

	Week_1+2	Week_3+4	Week_5+6	Week_7+8	Week_9+10	Week_11+12
Time	167.033	144.221	209.808	243.497	224.843	227.08
Total_Cost	44503069.65	46545837.17	45299952.72	45165284.69	45746449.71	45926787.71
Pulling_Cost	18698951.15	18535221.37	18664540.52	18849659.49	18831123.94	19000463.32
Dead_Cost	747542.049	757900.987	672928.966	655205.41	634956.912	733301.973
Idle_Cost	19337330.19	19937835.88	19813151.38	19325021.42	19574346.13	18883324.42
Light_Cost	1376007.996	2701968.467	1389033.388	1833871.542	2106531.475	2507560.526
Shop_Cost	1128705.528	1255844.664	1197214.026	1215599.588	1086440.962	1254038.973
Lease_Cost	2780400	2961000	3141600	2872800	3049200	3112200
Tr2Tr_Cost	434132.738	396065.797	421484.44	413127.236	463850.292	435898.493
Total_distance	5038188	4872328	4863893	4912435	4870268	4868376
Total_pulling_distance	4810862	4646278	4661535	4722309	4672486	4656760
Pulling_distance(%)	95.488	95.361	95.840	96.130	95.939	95.653
Total_deadhead_distance	224343	221441	199797	187167	193807	207359
Deadhead_distance(%)	4.453	4.545	4.108	3.810	3.979	4.259
Average_deadhead_distance	196.276	202.599	190.102	169.382	182.492	199.960
Total_light_distance	2983	4609	2561	2959	3975	4257
Light_distance(%)	0.059	0.095	0.053	0.060	0.082	0.087
Average_light_distance	33.517	60.645	33.260	30.505	41.406	46.780
n_deadhead_locos	1143	1093	1051	1105	1062	1037
n_light_locos	89	76	77	97	96	91
n_locos_due	170	173	176	193	188	188
n_locos_shopped	162	168	169	186	180	182
n_leased_locos	662	705	748	684	726	741
n_tr2tr_conn	1104	1053	1033	1051	1142	1079

Table A.9. Results of 2-week instances with $h = 3, r = 2$