



HAL
open science

Benders decomposition for the real-time Railway Traffic Management Problem

Kaba Keita

► **To cite this version:**

Kaba Keita. Benders decomposition for the real-time Railway Traffic Management Problem. Data Structures and Algorithms [cs.DS]. Ecole Centrale Lille, 2017. English. NNT : . tel-01667801v1

HAL Id: tel-01667801

<https://hal.science/tel-01667801v1>

Submitted on 19 Dec 2017 (v1), last revised 28 Mar 2018 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre:

3	3	9
---	---	---

CENTRALE LILLE

THÈSE

présentée en vue d'obtenir le grade de

DOCTEUR

en

Spécialité : Automatique, Génie Informatique, Traitement du Signal et des Images

par

Kaba KEITA

DOCTORAT DÉLIVRÉ PAR CENTRALE LILLE

Titre de la thèse :

**Décomposition de Benders pour la gestion
opérationnelle du trafic ferroviaire**

**Benders decomposition for the real-time Railway Traffic
Management Problem**

Soutenance prévue le 04 décembre 2017 devant le jury d'examen :

Présidente	Mme. Marie-Christine COSTA	Professeur, ENSTA - Paris-Tech
Rapporteur	M. Roberto WOLFER CALVO	Professeur, Université de Paris 13 - LIPN - AOC
Rapporteur	M. Xavier DELORME	Professeur, Mines - Saint-Etienne - LIMOS
Directeur de thèse	M. Joaquin RODRIGUEZ	DR, IFSTTAR - Villeneuve d'Ascq
Encadrante	Mme. Paola PELLEGRINI	CR, IFSTTAR - Villeneuve d'Ascq

Thèse préparée dans l'IFSTTAR
COSYS – ESTAS

École Doctorale Sciences pour l'ingénieur ED 072

Contents

List of Figures	v
List of Tables	vii
1 Introduction	1
1.1 Context and motivation	1
1.2 Thesis objective and contribution	2
1.3 Outline	3
2 Railway system	5
2.1 Introduction	5
2.2 Decision making process	5
2.2.1 Strategic level	6
2.2.2 Tactical level	7
2.2.3 Operational level	9
2.3 Railway Infrastructure	10
2.4 Infrastructure representation	12
2.4.1 Macroscopic representation	12
2.4.2 Microscopic representation	14
2.5 real-time Railway Traffic Management Problem: rtRTMP	16
2.5.1 Objective function	17
2.5.2 Constraints	17
2.5.3 Fixed speed model vs variable speed one	18
2.6 Conclusions	19
3 Literature review on the rtRTMP	21
3.1 Introduction	21
3.2 Monolithic approaches	21
3.2.1 ILP & MILP formulations	22

Contents

3.2.2	AG formulations	24
3.2.3	Meta-heuristics	26
3.3	Decomposition approaches	26
3.3.1	Spatial decomposition	26
3.3.2	Temporal decomposition	28
3.3.3	Mathematical decomposition	28
3.3.3.1	Lagrangian relaxation	29
3.3.3.2	Dantzig-Wolfe decomposition	30
3.3.3.3	Benders decomposition	32
3.4	Conclusions	35
4	RECIFE-MILP	36
4.1	Introduction	36
4.2	Assumptions	36
4.3	Formulation	38
4.3.1	Notation	38
4.3.2	Variables	40
4.3.3	Objective and constraints	41
4.4	Algorithm	42
4.5	RECIFE-MILP applications	44
4.6	Conclusions	45
5	Standard Benders Reformulation for RECIFE-MILP	46
5.1	Introduction	46
5.2	Standard Benders Reformulation: SBR	47
5.2.1	Sub-Problem (SP)	47
5.2.2	Dual of the Sub-Problem (DSP)	48
5.2.3	Restricted Master Problem (RMP)	49
5.2.4	Unrestricted Master Problem (UMP)	50
5.3	Benders Algorithm: BA	50
5.4	Computational experiments	51
5.4.1	Pierrefitte–Gonesse junction	52
5.4.2	Rouen-Rive-Droite control area	52
5.4.3	Results	53
5.5	Conclusions	56

6	Reduced Benders reformulation for RECIFE-MILP	58
6.1	Introduction	58
6.2	Reduced Benders Reformulation: RBR	59
6.2.1	Reduced Sub-Problem (Red-SP)	60
6.2.2	Reduced Dual of the Sub-Problem (Red-DSP)	61
6.3	Computational experiments	62
6.4	Conclusions	65
7	Inequalities for the unrestricted master problem	66
7.1	Introduction	66
7.2	Inequalities for the unrestricted master problem	67
7.2.1	Family $F1$	67
7.2.2	Family $F2$	69
7.2.3	Family $F3$	69
7.2.4	Family $F4$	70
7.2.5	Family $F5$	72
7.2.6	Family $F6$	77
7.3	Algorithm configuration	81
7.4	Computational analysis	82
7.5	Conclusions	82
8	Three-step Benders Algorithm for RECIFE-MILP	84
8.1	Introduction	84
8.2	Three-step Benders Algorithm: 3BA	84
8.2.1	New Unrestricted Master Problem (NUMP)	85
8.2.2	Mixed Integer Linear Sub-Problem (MILSP)	86
8.2.3	Linear Sub-Problem (LSP)	87
8.2.4	Dual of the Linear Sub-Problem (DLSP)	88
8.3	Computational experiments	90
8.4	Conclusions	98
9	Conclusions and future works	100
9.1	Conclusions	100
9.2	Future works	102
Bibliography		105
		111

List of Figures

- 2.1 Decision levels in the railway service planning process (strategic, tactical and operational). Problems are ranked firstly by the type of entity responsible of their solution ([Infrastructure Manager \(IM\)](#) and [Railway Undertaking \(RU\)](#)), and secondly, by their temporal occurrence. 7
- 2.2 Track composed of rails, ties and ballast. 10
- 2.3 A switch on the left, a crossing on the right. 11
- 2.4 Example of railway infrastructure including track-circuits, signals and block sections. 12
- 2.5 A macroscopic representation of the regional railway network of Nord-pas-de-Calais, France (source: [Arenas Pimentel \(2016\)](#)). 13
- 2.6 Microscopic representation of the railway infrastructure in which the track is divided into a set of [track-circuits](#) grouped into [block section](#) whose access is controlled by a [signal](#). 14
- 2.7 Elements of the blocking time (source: [Arenas Pimentel \(2016\)](#)). 15
- 2.8 Evolution of train A speed profile over distance and time when it meets a restrictive signal aspect in case of application of a fixed speed model. 19
- 2.9 Evolution of train A speed profile over distance and time when it meets a restrictive signal aspect in case of application of a variable speed model. . . . 19

- 4.1 One train longer than the shortest track-circuit. 37
- 4.2 The shortest track-circuit is longer than all trains. 37
- 4.3 Set of routes R_t , track-circuits TC_t available for train t and set of track-circuits TC^{r1} composing route $r1$ 39
- 4.4 Main variables and data concerning the utilization of tc belonging to route r of train t 40
- 4.5 Graphical representation of track segments \hat{TC}_{t,t',tc_7} , $\hat{TC}_{t,t',tc_{11}}$ in which the order between trains cannot change. 44

List of Figures

5.1	Pierrefitte–Gonesse junction.	52
5.2	Rouen-Rive-Droite control area.	53
7.1	Situation in which Inequalities (7.1) and (7.2) apply.	68
7.2	Situation in which Inequalities (7.5) apply.	70
7.3	Situation in which Inequalities (7.11) and (7.12) apply.	72
7.4	Situation in which Inequalities (7.13) apply.	73
7.5	Situation in which Inequalities (7.14) apply.	74
7.6	Situation in which Inequalities (7.15) apply.	74
7.7	Situation in which Inequalities (7.16) apply.	75
7.8	Situation in which Inequalities (7.17) apply.	76
7.9	Situation in which Inequalities (7.18) apply.	76
7.10	Situation in which Inequalities (7.19) apply.	77
7.11	Situation in which Inequalities (7.20) apply.	78
7.12	Situation in which Inequalities (7.21) apply.	79
7.13	Situation in which Inequalities (7.22) apply.	79
7.14	Situation in which Inequalities (7.23) apply.	80
7.15	Situation in which Inequalities (7.24) apply.	80
8.1	Standard Benders Algorithm in which the RECIFE-MILP formulation is decomposed into two problems.	85
8.2	Three-step Benders Algorithm in which the RECIFE-MILP formulation is decomposed into three problems.	85
8.3	Comparison of the average delay achieved by RECIFE-MILP and Three-step Benders Algorithm (3BA) on the two control areas depending the maximum number of routes per train.	93
8.4	Comparison of the number of instances solved to the optimum by RECIFE-MILP and 3BA on the two control areas depending the maximum number of routes per train.	93
8.5	Comparison of the average delay achieved by RECIFE-MILP and 3BA on the two control areas depending the number of x -variables.	94
8.6	Comparison of the number of instances solved to the optimum by RECIFE-MILP and 3BA on the two control areas depending the number of x -variables.	95
8.7	Comparison of the average delay achieved by RECIFE-MILP and 3BA on the two control areas depending the number of y -variables.	96
8.8	Comparison of the number of instances solved to the optimum by RECIFE-MILP and 3BA on the two control areas depending the number of y -variables.	96

List of Tables

5.1	Results of Benders Algorithm (BA) - Standard Benders Reformulation (SBR) and RECIFE-MILP on 30 Gonesse instances without initialization (time limit 180 seconds).	54
5.2	Results of BA-SBR and RECIFE-MILP on 30 Rouen instances without initialization (time limit 180 seconds).	54
5.3	Results of BA-SBR and RECIFE-MILP on 30 Gonesse instances with initialization (time limit 180 seconds).	55
5.4	Results of BA-SBR and RECIFE-MILP on 30 Rouen instances with initialization (time limit 180 seconds).	55
6.1	Results of BA-Reduced Benders Reformulation (RBR) and RECIFE-MILP on 30 Gonesse instances without initialization (time limit 180 seconds).	63
6.2	Results of BA-RBR and RECIFE-MILP on 30 Rouen instances without initialization (time limit 180 seconds).	64
6.3	Results of BA-RBR and RECIFE-MILP on 30 Gonesse instances with initialization (time limit 180 seconds).	64
6.4	Results of BA-RBR and RECIFE-MILP on 30 Rouen instances with initialization (time limit 180 seconds).	65
7.1	Performance of the RBR variants vs BA-RBR over the two control areas (time limit 180 seconds).	82
7.2	Performance of BA-RBR-F3 vs RECIFE-MILP over the two control areas (time limit 180 seconds).	83
8.1	Performance of 3BA vs RECIFE-MILP and BA-RBR-F3 over the two control areas (time limit 180 seconds).	90
8.2	Comparison of 3BA and RECIFE-MILP performance on the Gonesse instances when varying the maximum number of routes per train (time limit 180 seconds).	91

List of Tables

- 8.3 Comparison of [3BA](#) and [RECIFE-MILP](#) performance on the Rouen instances when varying the maximum number of routes per train (time limit 180 seconds). [92](#)

Chapter 1

Introduction

1.1 Context and motivation

To address the challenges set to try to achieve millennium development goals to deal with issues as overcrowding, natural resource depletion, and climate change, it is necessary to introduce new approaches to production, consumption and transport, capable of supporting the harmonious development of economy, society and the environment.

Rail transport, considered as an effective means of mass transport and as an environmentally-friendly mode, can play a major role in setting up a new approach to transport that meets the novel ecological requirements. However, rail transport remains less exploited than it may, being road and air often preferred options. To promote the rail, European white paper on transport recommends a 50% shift in middle distance passenger and longer distance freight journeys from road to the rail mode or the inland water transport by 2050 to achieve a 60% reduction in CO₂ emissions and comparable reduction in oil dependency (Kallas, 2011). The resulting growth in rail transport brings new challenges for this sector, concerning for example increase of efficiency of operations and improvement of quality of service provided to customers.

Even without considering the expected grow of traffic volume, today in many European countries and during peak hours the infrastructure capacity is completely exploited for ensuring the trains circulations. Many trains travel within short time through critical points. In this situation, if a disturbance occurs, traffic may be perturbed and, as a result, [conflicts](#) may emerge. In a conflict, multiple trains traveling at the planned speed would claim the same [track segment](#) concurrently. Hence, some trains must be stopped or decelerated for ensuring safety, and delays propagate. In locations such as junctions, which are areas where multiple lines cross, the emergence of conflicts is very frequent and effectively dealing with them may be particularly difficult. The [real-time Railway Traffic Management Problem \(rtRTMP\)](#) is the problem of modifying trains route and schedule to limit the delay propagation. Currently,

1 Introduction

this problem is tackled by dispatchers. They do it manually, so the result of their choices is generally suboptimal. The use of an efficient algorithm to help dispatchers decision making is crucial to ensure an effective traffic management when disturbances occur. The design of such an efficient algorithm has been the object of a remarkable branch of research, to which this thesis belongs.

1.2 Thesis objective and contribution

The objective of this thesis is to propose a decomposition algorithm, particularly a **Benders Decomposition (BD)**, to deal with the **rtRTMP**. Note that relevant contributions can be found in the operations research literature which propose algorithms, including decomposition ones, to address the **rtRTMP**. However, many of them cope with the problem considering cases in which the size of instances tackled are relatively small. Moreover, several approaches neglect the train re-routing possibility. These and other strong assumptions often limit the actual applicability of the proposed algorithms.

The contribution of this thesis goes in the direction of increasing the applicability of an existing algorithm in the practice. Namely, we consider the **RECIFE-MILP** algorithm. **RECIFE-MILP** is a **Mixed Integer Linear Programming (MILP)** based heuristic which has been developed by [Pellegrini et al. \(2015\)](#) as part of the decision support tool named **REcherche sur la Capacité des Infrastructures Ferroviaires - Research on the Railway Infrastructures Capacity (RECIFE)** introduced by [Rodriguez \(2007\)](#). The **RECIFE-MILP** algorithm allows solving some **rtRTMP** instances to optimality considering all details in the infrastructure. Indeed, **RECIFE-MILP** allows modeling specific circulation constraints that can accurately emulate the railway traffic in real circumstances. This is relevant for this thesis as shown in our experiments where we tackle real case studies of the French railway network. However, it has been shown ([Pellegrini et al., 2015](#)) that the performance of the **RECIFE-MILP** algorithm may strongly worsen when tackling large instances in the short time allowed by the real-time nature of the problem. For the **MILP** formulation at the basis of this algorithm, we propose **BD** approaches to improve the performance of the **RECIFE-MILP** algorithm when tackling large instances. After observing that the standard **BD** does not allow the effective solution of **rtRTMP** instances, we study three possible approaches to improve the performance. Specifically, we first propose a modification of the problem reformulation which is typical of **BD**, obtaining what we call reduced **BD**. Then, we introduce some inequalities to the Benders master problem. Finally, we split the solution process in three steps rather than two as in the standard **BD**. As we show in a thorough experimental analysis, the combination of the first and last approaches outperforms

the original [RECIFE-MILP](#) algorithm when tackling large instances with some important features.

1.3 Outline

The rest of this thesis is organized as follows.

In Chapter 2 we introduce main concepts and procedures characterizing the railway system. More specifically, we start by defining the decision process for traffic planning. Then, we describe the railway infrastructure elements and the two main approaches used to represent them: macroscopic and microscopic. Finally, we give extended descriptions of the [rtRTMP](#), as it is the problem tackled in this thesis.

In Chapter 3 we present an overview of the state of the art for the [rtRTMP](#). We classify and discuss contributions on the [rtRTMP](#) according to two different approaches: the monolithic approach and the decomposition one. The former includes the contributions dealing with the problem as a unique element. The latter groups the contributions in which the problem is split in sub-problems, either in its definition or in the solution technique used to solve it. The decomposition approaches often aim to the solution of large instances.

In Chapter 4 we present [RECIFE-MILP](#) for the [rtRTMP](#). The aim is to present [RECIFE-MILP](#) model assumptions, mathematical formulation and algorithm. We also present some [RECIFE-MILP](#) applications which have appeared in the literature. These applications motivate the interest we devote to [RECIFE-MILP](#), which is among the state of the art algorithms for the [rtRTMP](#) but can be incapable of dealing with really large instances. This chapter is a basis for the next chapters in which we propose a [BD](#) approach for the [RECIFE-MILP](#) formulation¹.

In Chapter 5 we present the [SBR](#) of the [RECIFE-MILP](#) formulation. It consists in splitting the [MILP](#) formulation presented in Chapter 4 into the [Unrestricted Master Problem \(UMP\)](#) and the [Sub-Problem \(SP\)](#). The [UMP](#) is the real-time train routing and scheduling problem. It contains the re-routing and rescheduling variables (binary variables) of the overall problem, and one dummy variable representing the contribution of continuous variables to the [UMP](#) objective function. The [SP](#) contains the continuous variables which represent train passing and stopping times, and deduces the delay suffered by each train. The decomposed problem is solved with the [BA](#) in which we add progressively Benders [optimality cuts \(OCs\)](#) and [feasibility cuts \(FCs\)](#) to the [UMP](#), obtaining them through the iterative solution of the [Dual of the Sub-Problem \(DSP\)](#). After presenting the reformulation, we perform computational experiments on two case studies representing traffic in two control areas in France: the Pierrefitte–Gonesse junction and

¹We will name the [RECIFE-MILP](#) formulation to designate the [MILP](#) formulation solved in [RECIFE-MILP](#) in the rest of this thesis.

1 Introduction

a line section around the Rouen–Rive–Droite station. The results show that the **SBR** is not appropriate for the **rtRTMP**.

In Chapter 6, to improve the performance of the **SBR** presented in Chapter 5, we propose the **RBR** for the **RECIFE-MILP** formulation. Unlike the classical decomposition approach (i.e., the **SBR**), the **RBR** exploits the nature of the **RECIFE-MILP** formulation to propose a more suitable reformulation. In our experiments on the same two case studies used in Chapter 5, the results show that the **RBR** is better than the **SBR**. Despite the improvement achieved, the performance of **RBR** is still significantly worse than the one of **RECIFE-MILP**.

In Chapter 7 we propose some inequalities to boost the performance of **RBR** presented in Chapter 6. The proposed inequalities are added to the **UMP**, to strengthen the consistency of routing and scheduling decisions. After an algorithm configuration phase in which we identify the most convenient inequalities to add to the **UMP**, we tackle instances representing traffic at the two control areas tackled in Chapters 5 and 6. The results show that, although adding inequalities to the **UMP** improves the **RBR** performance, the improvement of the proposed algorithm compared to **RECIFE-MILP** remains modest.

Finally, in Chapter 8 we propose the **3BA**. It is an algorithm based on the decomposition of the **RECIFE-MILP** formulation into three problems: the **New Unrestricted Master Problem (NUMP)**, the **Mixed Integer Linear Sub-Problem (MILSP)** and the **Linear Sub-Problem (LSP)**. The **NUMP** contains only the binary re-routing variables. Given the re-routing decisions made in the **NUMP**, we build the **MILSP** containing the binary rescheduling variables and the continuous re-timing variables. We obtain the optimal scheduling decisions by solving the **MILSP**, and we construct the **LSP** with the re-routing decisions made in the **NUMP** and the corresponding optimal rescheduling decisions made in the **MILSP**, to define the cuts to be added to the **NUMP**. This reformulation allows the generation of Benders cuts without the big-M parameter and the avoidance of the exploration of rescheduling infeasible or sub-optimal solutions. By doing so, we obtain an algorithm capable of reaching significantly better performance than **RECIFE-MILP** for the Rouen instances, where the number of alternative routes available for trains is very high.

In Chapter 9 we summarize the main results of this thesis and present some hints for future works.

Chapter 2

Railway system

2.1 Introduction

In this chapter we introduce the main concepts and terminology concerning the railway system that are relevant to understand the rest of this thesis. In Section 2.2 we describe the railway decision making process. Then, we present in Section 2.3 some elements of the railway infrastructure that are to be taken into account when dealing with the [rtRTMP](#), which is the problem we tackle in the next chapters. The different modeling approaches for the representation of these elements for the [rtRTMP](#) are depicted in Section 2.4. An introduction to the [rtRTMP](#) is presented in Section 2.5, and Section 2.6 concludes the chapter.

2.2 Decision making process

The railway system is very complex. It is composed of many components such as the infrastructures, rolling stocks, crews, operating protocols, etc. All these elements interact in a coordinated and regulated manner to allow the safe circulation of trains for transportation of passengers or freight. Furthermore, the railway system consists of very expensive assets. To ensure the best use of these valuable assets and to allow their efficient exploitation, an attentive planning of the railway operations is necessary. To this end, the management of the railway system is generally subdivided into three main decision levels, which are strategic, tactical and operational. In each decision level, two main types of entities are involved: The [IM](#) and the [RU](#). An [IM](#) is a body or firm responsible for establishing, managing and maintaining a railway infrastructure. A [RU](#) is a public or private entity that provides services for the transport of goods and/or passengers by rail.

2 Railway system

In the following, we will detail each of the three decision levels by shortly describing the problems tackled in each of them. Figure 2.1 shows these problems. They are ranked firstly by the type of entity responsible for their solution (**IM** and **RU**), and secondly, by their temporal occurrence. Although the nominal flow of problems, represented by the directed arrows in Figure 2.1, is from top to bottom, problems at the same height may be performed simultaneously. However, in practice, the flow of problems solving is not entirely sequential. For example, the solution of a problem at a lower level can influence those at an upper level. Hence a feedback loop allows the coordination between decision levels, to converge toward a feasible or better solution.

2.2.1 Strategic level

Decisions made at the strategic level include infrastructure development, facility upgrade, etc. Indeed, this level is based on traffic demand forecasting and concerns mainly the acquisition and renovation of resources to cover the target demand. The decision making process at the strategic level goes from 15 to 5 years prior to the execution of the decisions themselves. These decisions concern two main problems:

- 1. Network design:** This problem consists in elaborating plans for the construction or the modification of the infrastructure. It is typically managed by the **IM** in cooperation with the public authorities. However, **RUs** may also participate in the process by providing studies of customer habits, demand and projections. The modification and the construction of new infrastructure is motivated by changes of the travel requirements, increased demand of capacity or willingness of implementation of new technologies or standards. For more details on the network design problem, we refer the reader to [Magnanti and Wong \(1984\)](#); [Hooghiemstra et al. \(1999\)](#) and [García-Archilla et al. \(2013\)](#).
- 2. Line planning:** Once the infrastructure is designed, lines have to be defined and associated with individual frequencies. A train line is usually defined as an itinerary between two designated stations in the network, generally major ones, that is to be recurrently traversed by a train. Note that the itinerary may also include the stop pattern at a set of intermediate stations. A line must also include the specification of the train type to be used, i.e., type of engine, number of wagons, and its frequency, in case of regular periodic services. The line planning problem consists in selecting a set of lines and their frequencies subject to certain constraints and pursuing given objectives. For example, the line plan tries to meet the passenger travel demand and respect existing network capacities and properties. Line planning is mainly carried out by the **RUs**, but it might also require some collaboration with the **IM** and to some extent with national or regional transport authorities. Relevant

2.2 Decision making process

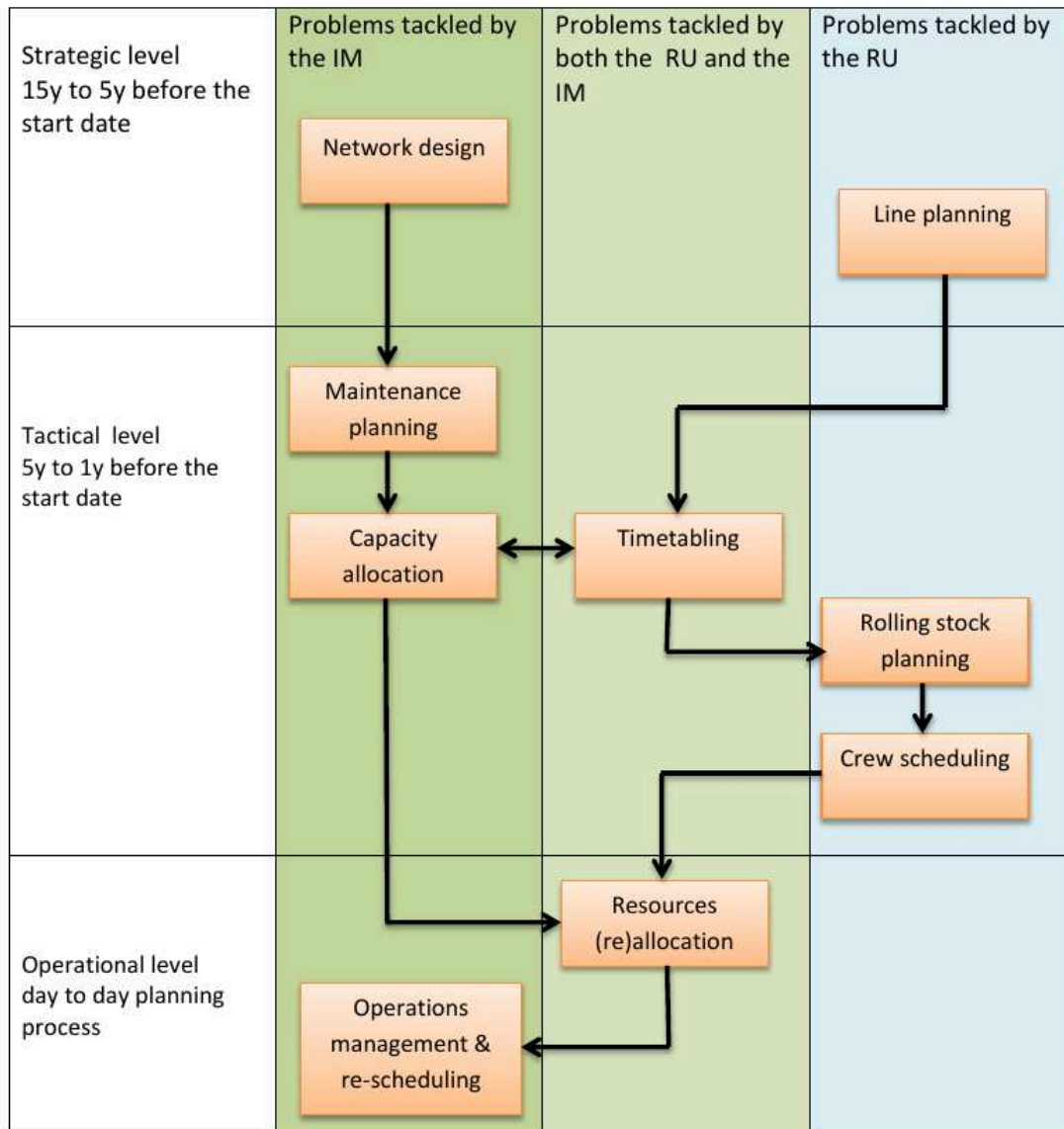


Figure 2.1 Decision levels in the railway service planning process (strategic, tactical and operational). Problems are ranked firstly by the type of entity responsible of their solution (**IM** and **RU**), and secondly, by their temporal occurrence.

work on the line planning problem can be found, for example, in [Bussieck et al. \(1997\)](#); [Claessens et al. \(1998\)](#); [Goossens et al. \(2006\)](#); [Schöbel \(2012\)](#) and [Fu et al. \(2015\)](#).

2.2.2 Tactical level

At the tactical level, decisions concern mainly five problems, which are maintenance planning, timetabling, capacity allocation, rolling stock planning and crew scheduling. These decisions

2 Railway system

are made between 5 and 1 year prior to their execution. The problems tackled at this level are defined in the following:

- 1. Maintenance planning:** It is also known as the maintenance scheduling problem. It defines an infrastructure maintenance plan that meets the maintenance needs of the network. The **IM** is responsible for establishing the maintenance plan, and this is typically done before the definition of the timetable. For more information on the maintenance planning problem see [Higgins \(1998\)](#); [Caprara et al. \(2006\)](#) and [Forsgren et al. \(2013\)](#).
- 2. Timetabling:** This problem consists in establishing a timetable which determines trains circulation in the network. Moreover, it dictates train arrival time at stations, [dwell times](#) at platforms and departure times from stations, connections between services, etc. This problem is widely studied in the literature where it is also known as the train scheduling problem. The **IM** is in charge of producing the timetable after receiving [RUs](#) train path requests. Relevant literature addressing the timetabling problem can be found (e.g., [Cordeau et al. \(1998\)](#); [Lusby et al. \(2011\)](#) and [Cacchiani and Toth \(2012\)](#)).
- 3. Stations capacity allocation:** The objective of this problem is to produce a capacity allocation plan compatible with all trains scheduled in the timetable. This plan is the assignment of [routes](#) (for a formal definition of a route see Section 2.3) and platforms in stations to each train while respecting their schedules. The **IM** is responsible for the management of the capacity allocation problem. However, in case of impossible allocation, negotiations are carried out with the [RUs](#). The capacity allocation problem has been named in several different ways in the literature such as track allocation problem, train routing problem, train path allocation problem and, in some cases, train platforming problem. Some relevant contributions addressing the capacity allocation are: [Caprara et al. \(2011\)](#) and [Schöbel \(2012\)](#).
- 4. Rolling stock planning:** This problem consists in finding a minimum cost rolling stock assignment to the trains scheduled in the timetable. The scheduling of empty rides and shunting movements is also included in the rolling stock planning. The aim of the rolling stock planning typically is to minimize the number of vehicles, considered representative of the total cost, necessary to meet the requirements of the timetable and respecting the periodic rolling stock maintenance needs. The main actors responsible for the generation of the rolling stock plan are the [RUs](#). Some relevant academic contributions on this problem are: [Abbink et al. \(2004\)](#); [Maròti \(2006\)](#); [Steinzen et al. \(2010\)](#) and [Cacchiani et al. \(2012\)](#).

5. Crew scheduling: This problem consists in the generation of crew duties to operate trains at minimal cost, while respecting all work regulations and operational requirements. The **RUs** are in charge of producing their crew schedule. We refer interesting readers to [Caprara et al. \(1997\)](#); [Abbink et al. \(2005\)](#); [Sahin and Yüceoglu \(2011\)](#) and [Jütte and Thonemann \(2012\)](#) for more details on the crew scheduling problem.

2.2.3 Operational level

The operational level refers to the daily and real-time planning and re-planning work. In this level, the problems tackled concern the day of operation or few days before it. Two main problems are tackled in this level: Resource (re)allocation and operations management and rescheduling.

- 1. Resources (re)allocation:** In case of new or unanticipated resource requirements, resources must be (re)allocated. Conditions that provoke this need for (re)allocation can be for instance: new freight trains due to an exceptional convoy, unexpected corrective maintenance due to a device malfunction, changes on the crew rotations due to strikes, etc. Although these events are unforeseen in the tactical phase, some time is available to deal with them, in general up to a few days. When the resource required to be (re)allocated is infrastructure capacity, e.g., due to unforeseen maintenance activities, new trains requests, etc., the **IM** is responsible for the (re)allocation. Instead, when the resources concerned are rolling stock or crews the **RUs** are in charge. For more details on the resources (re)allocation, we refer the reader to [Huisman et al. \(2005\)](#) and [Chu and Oetting \(2013\)](#).
- 2. Operations management and rescheduling:** The aim of this problem is manage and coordinate train movements in the railway network during operations. Two types of problems can be identified. On the one hand, there are the problems dealing with large disruptions, as the interruption of a line for several hours. They deal with the definition of a new timetable including train cancellations and short-turning ([Zhan et al., 2015](#)), as well as with rolling stock re-allocation ([Nielsen et al., 2012](#)) and crew rescheduling ([Veelenturf et al., 2012](#)). What differentiates these problems with respect to the ones which we mentioned in the resources (re)allocation paragraph is that here no anticipation at all is possible. On the other hand, there are the problems dealing with small perturbations, which occur on a daily basis. The **rtRTMP**, also known as the train rescheduling problem, is the most important problem tackled in this case. The **rtRTMP** aims to reduce delay propagation through rescheduling and local re-routing. This problem is faced by **IMs**. We will present an extended description of the **rtRTMP** in Section 2.5. Then, we will propose a detailed review of the literature on this problem in Chapter 3.

2.3 Railway Infrastructure

As mentioned at the beginning of Section 2.2, a very important component of the railway system is the infrastructure. The infrastructure is, in turn, composed of several elements, the first of which corresponds to tracks. Tracks consist of two parallel steel rails, anchored perpendicularly to components called ties (sleepers) of timber, concrete, steel, or plastic to maintain a consistent distance (or gauge) between the rails (see Figure 2.2). The track guides the conical, flanged wheels of the trains, keeping the cars on the tracks themselves without active steering and therefore allowing trains to be much longer than road vehicles. The rails and ties are usually placed on a foundation made of compressed earth on top of which is placed a bed of ballast. Its role is to distribute the load from the ties and to prevent the track from buckling as the ground settles over time, under the weight of the vehicles passing above. The ballast also serves as a means of drainage.

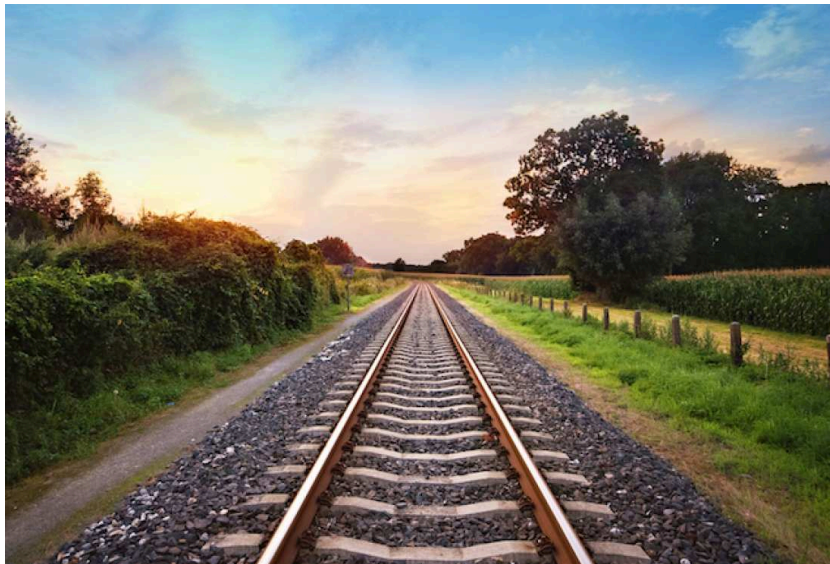


Figure 2.2 Track composed of rails, ties and ballast.

Tracks are also composed by other devices necessary for trains guide: Switches (or points) and crossings (see Figure 2.3). The former are mechanical installations enabling trains to be guided from one track to another. The latter are used to divert trains from one track to another: a special arrangement is made so that the inside flange of the wheel can move on the diverted track. Switches and crossings are generally situated at junctions. A junction is a location where two or more rails converge or diverge.

Moreover, tracks are divided into **track-circuits**. A track-circuit is a detection device on which the presence of a train is automatically detected. A sequence of track-circuits whose access is controlled by a **signal** is called **block section**. A signal is a semaphore that gives



Figure 2.3 A switch on the left, a crossing on the right.

the train driver some information about the utilization state of the following block sections. Often, signals have three colors, i.e., red, yellow and green, which are also called aspects. In a three-aspect signaling system, the red aspect indicates that the driver must stop since the next block section is currently being utilized by another train. The yellow aspect indicates that the driver can proceed but with caution because the block section after the next one is currently being utilized by another train. Commonly, a yellow aspect implies a considerable reduction of the train speed. The green aspect indicates that the next two block sections are clear so the driver can proceed at regular speed. When the signaling system has more than three aspects, the additional ones are restrictive aspects which indicate to the driver additional steps for the deceleration process to stop the train before the unavailable section. Typically, the higher the number of aspects is, the shorter the block sections can be, since several of them are available to pass from the planned speed to the complete stop of the train. With this system it is possible to ensure that only one train uses a block section at any time and all trains have enough space to brake to respect the safety distance.

The coordination between [signals](#) and [block sections](#) utilization is done through the [interlocking](#) system. In the practice, there are two main interlocking systems: route-lock route-release and route-lock sectional-release. In both, if a block section contains several track-circuits, all of them are reserved at the same time. This mechanism is called route-lock. As previously introduced, the circulation of a train in a block section is detected by the track-circuits composing it. When the train finishes the occupation of a track-circuit, it is released and then can be reserved by another train. The procedure of releasing independently the track-circuits composing a block section is known as sectional-release. On the contrary, in the case of the route-release, the release of all [track-circuits](#) in the [block section](#) is synchronized, i.e., it occurs at the same time, when the last track-circuit is released. The sectional-release is specifically useful in locations where track-circuits are shared by multiple block sections, as

2 Railway system

it helps reducing the separation between trains and, thus, maximizing the exploitation of the capacity, as at stations.

Figure 2.4 shows an example of infrastructure including **track-circuits**, **signals** and **block sections**. Let two trains, *A* and *B*, pass through this infrastructure, *A* using block sections $s1 - s2$, $s2 - s5$ and $s5 - s6$, and *B* using block-sections $s3 - s4$, $s4 - s7$ and $s7 - s8$. The two intermediate block sections share track-circuits $tc3$ and $tc4$. Suppose *A* passes before *B*. With the route-lock sectional-release **interlocking** system, *A* releases the last common track-circuit $tc4$ as soon as its tail exits it, plus the **release time**. Instead, with the route-lock route-release system, $tc4$ is released after *A* exits $tc5$, plus the release time. Indeed, train *B* closely follows *A*, it will encounter a red aspect at signal $s4$. This red aspect will change to green earlier with the former interlocking system, which allows a better exploitation of the infrastructure.

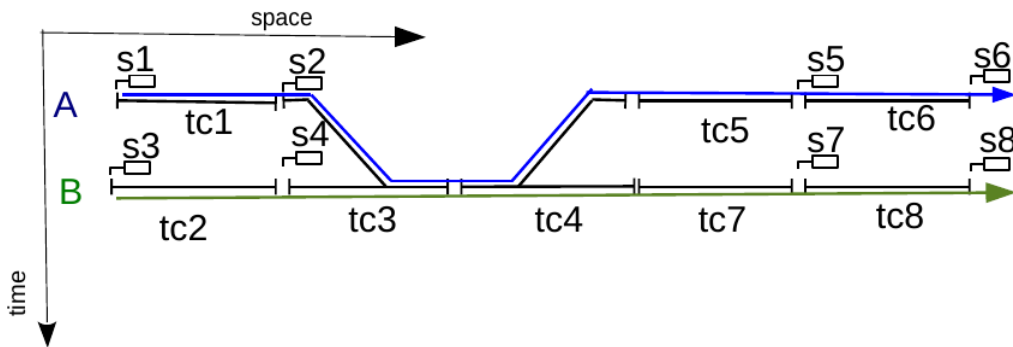


Figure 2.4 Example of railway infrastructure including track-circuits, signals and block sections.

Stations are located along tracks. They are particular pieces of infrastructure used to perform services such as loading/unloading of passengers and/or freight, exchange of on-board staff, etc. They generally include a set of facilities next to tracks and platforms.

2.4 Infrastructure representation

The railway infrastructure elements described in Section 2.3 must be modeled to formalize the problems mentioned in Section 2.2, and in particular the **rtRTMP** which is detailed in Section 2.5. In the literature, two main approaches are used to represent the railway infrastructure: macroscopic and microscopic, which we describe in the following.

2.4.1 Macroscopic representation

In the macroscopic approach, the infrastructure is seen as a graph. Here, nodes are very often used to represent either stations or junctions. The connections between these nodes are assured

2.4 Infrastructure representation

by tracks that are aggregated and considered as links. In other words, all elements between two nodes (track-circuits and block sections) are considered as a whole in a link. The purpose of the macroscopic representation of the infrastructure is to provide a global view of the network. Figure 2.5 shows a macroscopic representation of the region Nord-pas-de-Calais network in France. We can see that only the main stations are represented, together with the links that connect them.

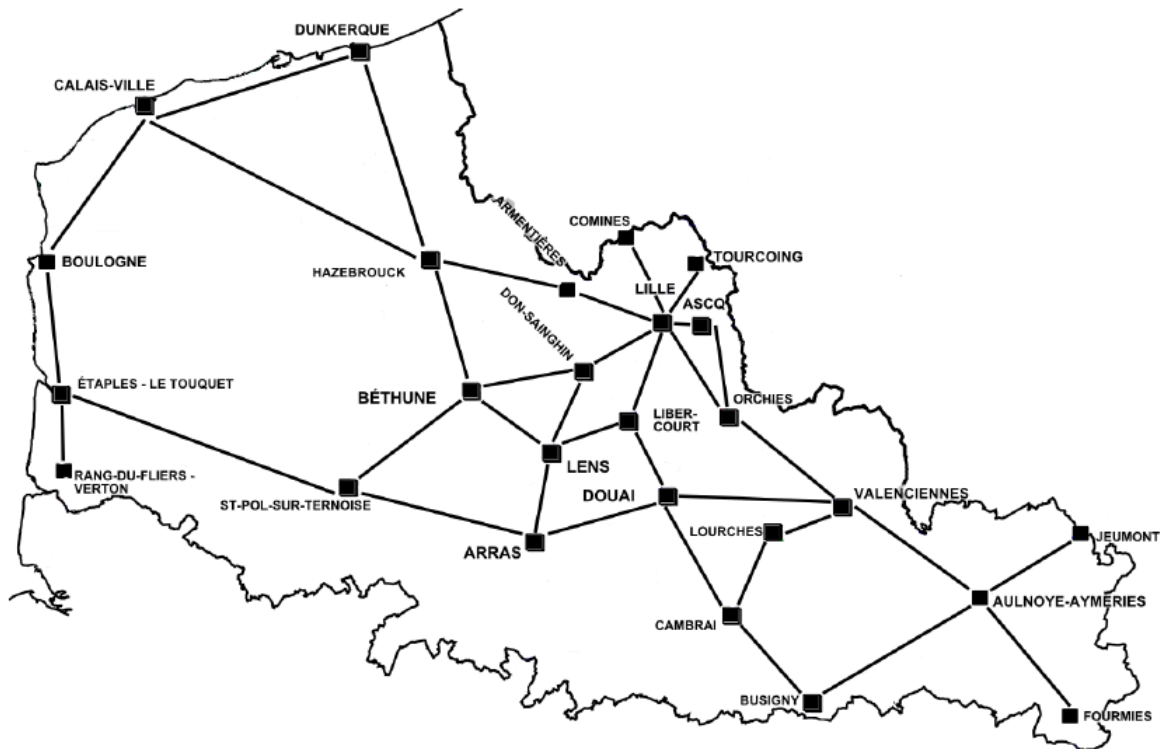


Figure 2.5 A macroscopic representation of the regional railway network of Nord-pas-de-Calais, France (source: Arenas Pimentel (2016)).

Remark that due to the rough representation of the infrastructure obtained in the macroscopic approach, that do not allow considering all details on the railway infrastructure, this level of abstraction is used generally to deal with the problems in which detailed information on the infrastructure and train dynamics are not needed. Indeed, the problems very often tackled using a macroscopic representation are treated in the strategic and the tactical decision level (see Sections 2.2.1 and 2.2.2). A typical problem that is faced with the macroscopic representation of the infrastructure is, for example, the network design problem.

Concerning train dynamics, in the macroscopic approach, no detail is taken into account: only the total travel time of trains traveling between pairs of nodes is considered.

2 Railway system

For the capacity of nodes, this is usually simply quantified as a theoretic maximum number of train that can be handled.

The concept of headway is used to ensure safety in the railway system when the macroscopic representation of the infrastructure is applied. It consists in imposing a temporal or physical separation between the utilization by two trains of a link rather using precise separations depending on the specific signaling system. To calculate the headway several factors are taken into account as the characteristics of both the tracks and the rolling stock (maximum speed, weight and brake capacities, etc).

2.4.2 Microscopic representation

In the microscopic approach, all details of the infrastructure (see Figure 2.6) are considered. The purpose of a microscopic representation is to provide a detailed description of the train movements, in the form of complete routes and passing times in critical locations. To this end, precise information about track topology and trains' dynamic properties have to be taken into consideration.

Concerning the track structure, in the microscopic approach, tracks are divided into track-circuits. Note that the precise description of stations and junctions are also considered in the microscopic approach (platforms, switches, crossings, etc.).

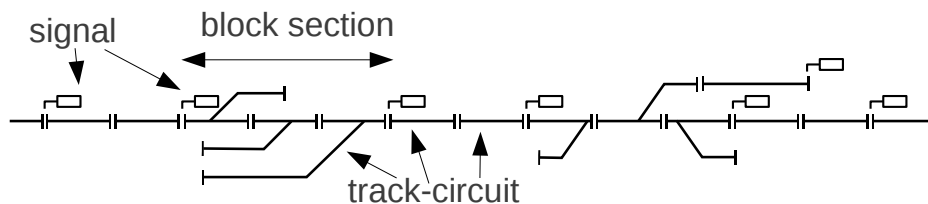


Figure 2.6 Microscopic representation of the railway infrastructure in which the track is divided into a set of track-circuits grouped into block section whose access is controlled by a signal.

Concerning the trains' dynamic properties, complete information on the blocking time of a track-circuit is given. The blocking time is defined as the time interval that a block section is exclusively reserved for a train movement (Pachl, 2009). Note that the blocking time is applicable only with the microscopic representation of the railway infrastructure. The utilization of the blocking time theory gives a precise modeling of the interlocking system. The blocking time determines the minimum time or space that separate two succeeding trains. Indeed, in a 3-aspect signaling system, the blocking time implies that a train is authorized to cross a block section if two subsequent ones are not reserved by another train. This simulates the fact that a train can enter a block section only with a green signal. The following intervals compose the blocking time of a track-circuit:

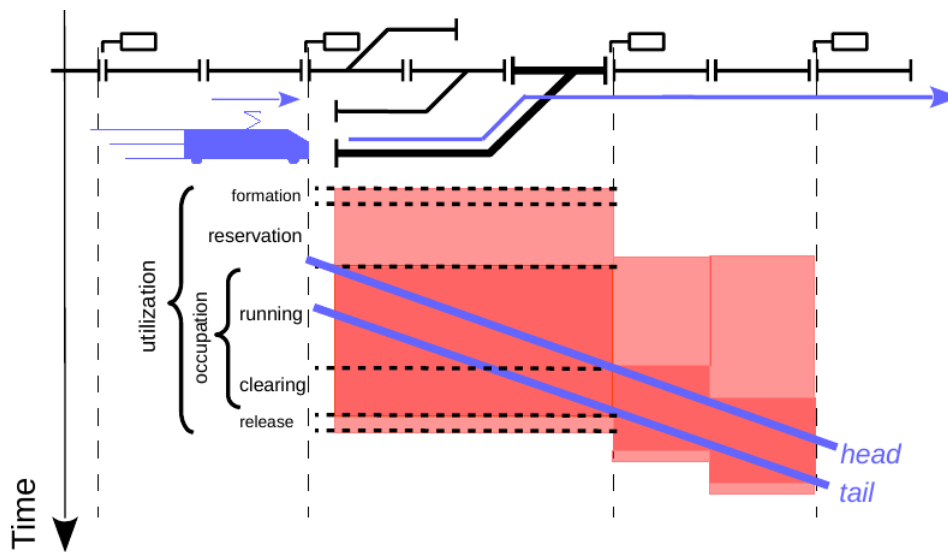


Figure 2.7 Elements of the blocking time (source: Arenas Pimentel (2016)).

- the time to set the block sections (signal and moving elements) to which the track-circuit belongs;
- the **signal** visibility time, i.e., the time needed by the driver to see the aspect of the signal opening the block section and react to it;
- the approach time, i.e., the time necessary for the travel across the previous block-section (necessary for ensuring that the signal opening the block section has green aspect);
- the **running time** along the track-circuit itself and all the ones preceding the one considered in the block section used (time necessary for the head of the train to traverse a track-circuit);
- the **clearing time**, i.e., the time needed for the tail of the train to exit the track-circuit once the head entered the following;
- the **release time** to unlock the system.

Figure 2.7 illustrates the blocking time for a train on three **track-circuits** composing two **block sections**. In this figure, the **formation time** corresponds to the time to set the block section plus the visibility time. The **reservation time** corresponds to the approach time. We can remark that the blocking time is significantly longer than the time used to physically traverse the track-circuit.

For visually representing the train travel along an infrastructure, we can draw into a space-time diagram the blocking times for all block sections crossed by train along its route. By

2 Railway system

doing so, we obtain the blocking time stairway (colored boxes in Figure 2.7). Indeed, the use of the blocking time stairway allows the calculation of the minimum headway between two trains using one or more common track-circuits: this headway is the time separation which ensures that the blocking times of no track-circuit overlap.

In the microscopic representation of the railway infrastructure, the capacity of stations is determined depending the specific characteristics of the tracks and platforms that compose them.

Remark that the microscopic representation of the infrastructure allows obtaining a detailed quantification of track sections capacity. Hence, this type of representation is particularly useful when coping with problems in which precise information is needed. Indeed, the problems very often tackled using a microscopic representation are treated in the tactical and operational decision level (see Section 2.2.2 and 2.2.3). In particular, a problem that is appropriately faced with the microscopic representation of the infrastructure is, for example, the [rtRTMP](#).

2.5 real-time Railway Traffic Management Problem: [rtRTMP](#)

As the contributions of this thesis are related to the [rtRTMP](#), we provide in this section a detailed description of this problem. The [rtRTMP](#) consists in modifying trains route and schedule to limit delay propagation. In the practice, this problem is tackled by [IM](#) dispatchers in control areas, which are portions of the infrastructure. They do it manually, so the result of their choices is generally suboptimal.

In principle, the timetables defined in the tactical phase (see Section 2.2) can be smoothly operated: if there are no disturbances nor disruptions, then the timetable can be executed exactly as planned. However, in the real-time operations disturbances and disruptions are inevitable. Disturbances are relatively small perturbations that influence the railway traffic causing trains' *primary* delay. They consist for example in a longer than planned [running time](#) or [dwell time](#) from one station to another. These delays often propagate generating the so-called *secondary* or *knock-on* delays. Disruptions are large perturbations leading to major modifications of the timetable. As mentioned when describing operations management and rescheduling problems in Section 2.2.3, the [rtRTMP](#) concerns the management of traffic in case of disturbances. In the next sections, we detail the [rtRTMP](#) starting with the identification of the objective pursued. Then we describe the constraints that must be satisfied, and finally we describe the two models used in the literature to describe trains speed variation dynamics in case of unplanned brakings and accelerations.

2.5.1 Objective function

The objective functions used in literature when tackling the [rtRTMP](#) are typically functions of trains' delay or arrival time at destination.

Trains' delay is the non-negative difference between the actual arrival times at stations and the scheduled ones. Different functions of delay are minimized in the [rtRTMP](#) literature: total delay, maximum delay, maximum consecutive delay, cumulative consecutive delay, sum of delays with penalties when exceeding a threshold, etc..

Focusing on arrival times, some authors use also the minimization of the total completion time, that is the arrival time of the last train at its destination.

Other objective functions considering different performance indicators are: the minimization of the trains' travel time, which is a measure of the time spent by all trains in the network, the schedule deviation, which penalizes both trains' advance and delay, and the maximization of trains' punctuality, which refers to the number of trains that do not arrive at their final destination on time, or suffering a delay smaller than a given threshold.

In some cases weights are associated to different trains in the objective function to allow taking into account different priorities.

Moreover, customer perspective is sometimes considered when solving the [rtRTMP](#). For example, passengers' inconvenience to be minimized can be measured as the additional waiting time at platforms and number of transfers.

2.5.2 Constraints

Constraints of the [rtRTMP](#) are imposed to respect operational requirements:

Time concerning constraints: These constraints impose the respect of operational requirements related to the time. For instance, at a station a train cannot depart before its scheduled departure time. Another requirement handled by these constraints is the coherent physical [occupation time](#) of [track-circuits](#) along a train [route](#): a train cannot start occupying a track-circuit along its route if it has not spent in the preceding one at least the corresponding [running time](#). Time constraints also allow modeling the complete blocking time.

Connection constraints: In railway traffic, train services are sometimes in connection: passengers or crew members are planned to transfer from a train to another at a specific station. In this case we must impose constraints that manage requirements such as the guarantee of a minimum time separation between the arrival of the feeder train and the departure of the connecting train.

2 Railway system

Rolling-stock re-utilization constraints: If train services are to be operated with the same rolling-stock, their arrival and departure must be coherent, in terms of both time and space. Hence, a minimum time interval must pass between arrival and departure, and the concerned trains must arrive at and depart from the same track-circuit.

Delay management constraints: If the difference between the estimated train arrival time and the scheduled arrival time at the exit from the infrastructure is positive there is a delay. Constraints are imposed to capture this delay.

Capacity constraints: The track-circuits utilization by two trains must not overlap.

2.5.3 Fixed speed model vs variable speed one

In the [rtRTMP](#), two main models for train movements are generally used in the literature to capture the train speed variation dynamics due to unplanned brakings and accelerations: the variable and the fixed speed ones. In a fixed speed model trains brake and re-accelerate instantaneously in case of [conflict](#), while in a variable speed one they follow more or less realistic braking and acceleration curves. As an example, Figure 2.8 shows the evolution of train A speed profile in case of a fixed speed model when encountering a yellow [signal](#). Here, train A is hindered by the presence of train B in the third [block section](#) of its [route](#). The left figure shows this evolution in a distance-speed diagram. According to the planned speed profile, train A runs at 120km/h for the first 3.9km , then it accelerates to 160km/h until $\text{km } 10$, when it needs to decelerate to pass a switch at 60km/h at distance 11.8km . After passing the switch it can accelerate again to 160km/h . In the fixed speed model, due to the conflict, train A stops in front of the yellow [signal](#): its speed passes instantaneously from 120 to 0 km/h . When the signal becomes green, i.e., when train B releases its first block section, train A regains its planned speed of 120km/h instantaneously. The evolution of train A speed profile over time is depicted in the right figure. In this figure, train A stops instantaneously at the time 150s and re-starts at the time 380s after 130s of stopping time. The realistic speed evolution of train A in the same conflict situation is shown in Figure 2.9 representing the variable speed model. Here, instead of stopping at distance 3.5km or at time 150s , train A decelerates after passing the yellow signal, to be able to stop at distance 4.8km . Its speed follows a coherent braking curve and the train reaches its stop at time 250s . After train B releases its first block section, train A accelerates, at time 380s , re-gaining its planned speed profile at time 500s and at distance 9km .

The trains' entrance times in the different track-circuits in case of [conflict](#) are hence imprecise when a fixed speed model is applied. However, this imprecision is not an issue for the practical implementation of the traffic management decisions since what is actually imposed in

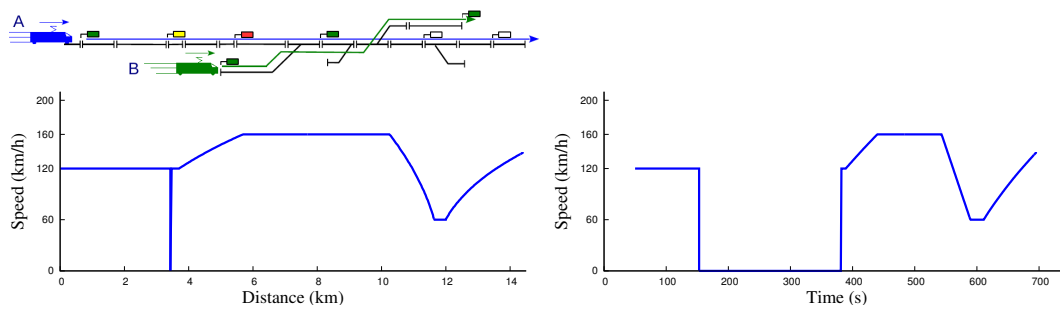


Figure 2.8 Evolution of train A speed profile over distance and time when it meets a restrictive signal aspect in case of application of a fixed speed model.

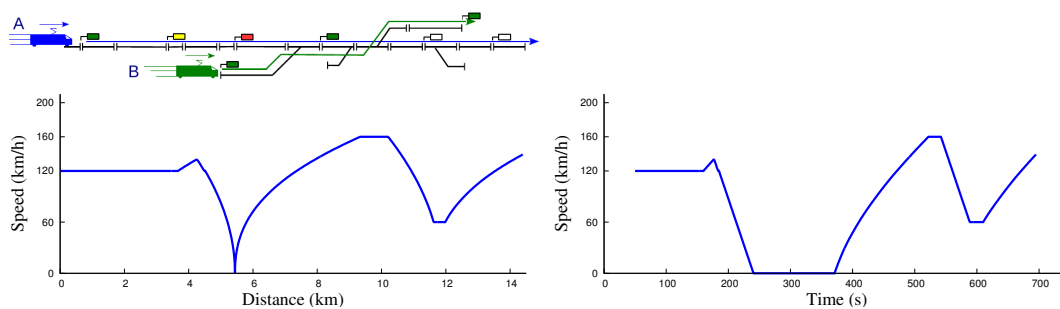


Figure 2.9 Evolution of train A speed profile over distance and time when it meets a restrictive signal aspect in case of application of a variable speed model.

reality is the set of precedences between trains, rather than the precise time at which trains are supposed to pass.

As we illustrate in Chapter 3, most existing algorithms for the `rtRTMP` apply a fixed speed model for limiting the difficulty of the problem. The impact of the application of a fixed speed model on the quality of the solution returned by `rtRTMP` algorithms has not been thoroughly studied in the literature. To the best of our knowledge, the only attempt to do so is the paper by [Sobieraj et al. \(2011\)](#). Here, the authors measure the difference in terms of delay propagation when the same train precedences are assessed with a fixed and a variable speed model. In some different traffic situations, the authors show that although the delay propagation indeed differs, the two models give coherent indications in terms of solution quality ranking.

2.6 Conclusions

In this chapter we introduced the main concepts and terminology concerning the railway system that are relevant to understand the rest of this thesis.

The description of the decision making process helps to understand in which part of this complex process are situated the main contribution of this thesis, i.e., the `rtRTMP`. Moreover,

2 Railway system

the introduction of the [rtRTMP](#), coupled with the discussion of the different approaches to represent the railway infrastructure, serves as a basis for Chapter 5 - 8 in which we present a decomposition method to tackle this problem using a microscopic representation.

Chapter 3

Literature review on the rtRTMP

3.1 Introduction

This chapter presents an overview of models and algorithms for the [rtRTMP](#) introduced in Section 2. This problem consists in modifying trains' route and schedule to limit the delay propagation in the railway network. The [rtRTMP](#) is recognized in the literature to be an NP-hard problem ([Kroon et al., 1997](#); [Mascis and Pacciarelli, 2002](#)). Instances of realistic size are often difficult to solve to optimality even using the best existing solvers. Many algorithms exist in the literature to tackle this problem, aiming at better and better performance on larger and larger instances.

The aim of this chapter is to classify and discuss various algorithms for the [rtRTMP](#) grouped according to the approach they consider: monolithic versus decomposition approach. The former includes the contributions facing the problem as a unique element. The latter groups the contributions in which the problem is split in sub-problems, either in its definition or in the solution technique used to solve it. The decomposition approaches often aim to the solution of large instances.

The rest of the chapter is organized as follows. In Section 3.2, we present contributions on the [rtRTMP](#) that use the monolithic approach. In Section 3.3, we describe decomposition approaches used in the literature to address the [rtRTMP](#), and in Section 3.4 we draw some conclusions.

3.2 Monolithic approaches

In this section, we classify and discuss the most relevant contributions on the [rtRTMP](#) that propose algorithms to tackle the problem considering it as a unique element, i.e., the monolithic

3 Literature review on the rtRTMP

approaches. We group the algorithms into three main categories according to the solution techniques. The most important algorithms for the **rtRTMP** are based on three solution techniques (for recent surveys, see [Lusby et al. \(2011\)](#); [Cacchiani et al. \(2014\)](#); [Corman and Meng \(2015\)](#); [Fang et al. \(2015\)](#)). The first technique is based on integer programming including **Integer Linear Programming (ILP)** and **MILP** formulations. The second technique uses an **Alternative Graph (AG)** formulation. The third one applies meta-heuristics.

3.2.1 ILP & MILP formulations

In an **ILP** formulation, all decision variables are binaries and/or integers. For instance, in such a formulation the order of two trains, assignment of resources, etc., can be represented by binary variables, while trains' arrival, departure times and delay can be represented by integer variables. Unlike the **ILP** formulation, in a **MILP** some decision variables are continuous. For instance, in a **MILP** for the **rtRTMP**, trains' arrival, departure times and delay are often represented by continuous variables. The binary decision variables are similar to those in the **ILP** formulation. In both, the **rtRTMP** objective function and constraints are linear expressions.

Relevant contributions modeling the **rtRTMP** as an **ILP** are the ones by [Caimi et al. \(2011, 2012\)](#) and [Dollevoet et al. \(2014\)](#). [Caimi et al. \(2011\)](#) propose a multi-commodity flow formulation for the train routing and scheduling problem where a set of different train movements (paths) represent the different commodities. A solution to the problem is a collection of the train paths, where a train path is defined as a feasible sequence of track sections and associated blocking times. In the formulation, the authors maximize the number of scheduled train movements which can be performed in a given time horizon under the following constraints: each train must choose no more than one path and two trains' paths cannot use a same resource at a same time. The case study tackled in the paper represents traffic in a station control area, in Switzerland. In a subsequent work, [Caimi et al. \(2012\)](#) propose a discrete-time formulation for rescheduling trains in complex central railway station areas. The computational analysis is done in the Berne station control area, in Switzerland. In the two formulations, the microscopic representation of the railway infrastructure described in Section 2.3 is used. More recently, [Dollevoet et al. \(2014\)](#) combine a macroscopic representation of the railway infrastructure (see Section 2.3) and a microscopic one to tackle the train scheduling problem when considering connections. In particular, the main decision which is to be taken concerns the set of connections to preserve, possibly delaying a departing train to wait for the arrival of another one. This decision is made considering a macroscopic model of a large network. A heuristic algorithm is then used to solve the microscopic problem for each station, to ensure the respect of the decisions on the connections to preserve, and hence on the trains' departure times. However, the re-routing

possibility is not considered. The instances used in the analysis are random perturbations of the real timetable on a portion of the Dutch network.

The **rtRTMP** is also modeled by several authors using **MILP** formulations. This technique is used by most of the publications on the **rtRTMP** (see [Fang et al. \(2015\)](#)). Among them, a **MILP** formulation based on a graph formulation of the problem is used by [Dessouky et al. \(2006\)](#). A branch-and-bound algorithm is used by the authors to solve the problem. The macroscopic representation of the infrastructure is used. [Törnquist and Persson \(2007\)](#) propose a **MILP** formulation based on a microscopic representation of the infrastructure to tackle the train rescheduling problem in case of N-tracked railway network. They propose some strategies, which simplify and restrict the solution space, to allow solving the problem in a reasonable computation time. Different objectives and different delay characteristics are used to test their influence in the solution procedure performance. Instances based a part of the Swedish network are used in the computational experiments. [Acuna-Agost et al. \(2011a\)](#) present an extension of the formulation by [Törnquist and Persson \(2007\)](#) to tackle the railway rescheduling problem. A **MILP**-based local search method is proposed by the authors to solve the problem on instances representing traffic in two different networks. The first one is located in France and the second in Chile. In a subsequent paper, [Acuna-Agost et al. \(2011b\)](#) propose a new solution approach to improve the computational results called SAPI, that performs a statistical analysis of possible propagation of the incidents before solving the problem. In particular, SAPI studies the probability that an event is affected by a given set of incidents. Using this information, it is possible to identify the set of events that are not affected by the incidents.

More recently, [Törnquist Krasemann \(2015\)](#) proposes a **MILP** formulation to address the train rescheduling problem. The formulation is adapted to the special restrictions of the congested, single-tracked Iron Ore line located in northern Sweden. The paper investigates potential configuration challenges in the development of a rescheduling support for train traffic dispatchers controlling this line. The author discusses how the characteristics of different situations shall influence the problem formulation and the resulting rescheduling solutions.

Finally, [Pellegrini et al. \(2014\)](#) propose a **MILP** formulation to tackle the **rtRTMP** when the microscopic representation of the infrastructure is done at the **track-circuit** level (see Section 2.4). In the paper, in addition to proposing the novel formulation, the authors show that the consideration of such a fine granularity of the infrastructure representation may allow the detection of better solutions than the use of a rough granularity limited to the **block section** level (Section 2.4). The computational experiments are based on the triangle of Gagny control area and on the Lille-Flandres one in France. In a subsequent paper, [Pellegrini et al. \(2015\)](#) extend the model of [Pellegrini et al. \(2014\)](#) by including the consideration of intermediate stops in the trains' route and further peculiarities of the movements, and propose a heuristic

3 Literature review on the rtRTMP

algorithm named [RECIFE-MILP](#) based on the truncated solution of the [MILP](#) formulation and on the application of some preprocessing procedures. [RECIFE-MILP](#) has been used in several applications since its proposal, and it plays a very important role in this thesis. Hence, we devote to its detailed explanation a whole chapter in the following (Chapter 4).

3.2.2 AG formulations

An [AG](#) formulation generalizes the disjunctive graph one and can be used to model no-store job shop scheduling problems ([Mascis and Pacciarelli, 2002](#)). As the definition of the [rtRTMP](#) can be linked to that of a no-store job shop scheduling problem ([Mascis et al., 2004](#); [D’Ariano et al., 2007a](#)), some authors have used an [AG](#) for the [rtRTMP](#). To do so, in the [rtRTMP](#), a [block section](#) corresponds to a no-store machine and a train corresponds to a job. No-store (or blocking) constraints model the absence of buffers between machines (or block sections). The [AG](#) is composed by three elements, which are a set of nodes, a set of fixed arcs, and a set of pairs of alternative arcs. A node represents the passing of a train through a particular block section (an operation). A fixed arc models the sequence of operations to be executed by a train. A pair of alternative arcs models the train rescheduling decision when a potential [conflict](#) arises. In an [AG](#), a solution is a complete selection, where an arc for each alternative pair is selected. It is feasible if the connected graph has no positive length cycles. The longest path in an [AG](#) is equivalent to the maximum propagated delay of the corresponding train sequencing. The objective of the [rtRTMP](#) modeled as an [AG](#) is to minimize the length of the longest path satisfying the capacity constraints. Moreover, the nature of the formulation used does not allow to perform re-routing. As underlined in the following, when re-routing is to be considered, the [AG](#) formulation needs to be integrated in a meta-heuristic algorithm exploring separately different re-routing possibilities.

The [AG](#) formulation for the [rtRTMP](#) are used to tackle several variants of the problem. [D’Ariano et al. \(2007a\)](#) propose a truncated branch-and-bound algorithm to solve it. In the computational analysis, the effectiveness of the truncated branch-and-bound is evaluated with respect to three other heuristics: First Come First Served (FCFS), First Leave First Served (FLFS) and a greedy heuristic. The experiments are carried out on a Dutch control area. In a following paper, [D’Ariano et al. \(2007b\)](#) extend the previous studies based on a fixed speed model to a variable speed one (see Section 2.5). The latter takes into account the realistic consequences of braking and subsequent acceleration imposed due to [conflicts](#) in the network. The variable speed model is solved in an iterative rescheduling procedure. At each step the [rtRTMP](#) is solved and if the solution includes conflicts, the first conflict in time is considered and the train speed profile of the braking train is re-computed. With this new speed profile, the blocking time stairway of the train is modified and the [rtRTMP](#) is solved once more starting at

the time of the treated [conflict](#). The procedure terminates when the end of the time horizon is reached or there are no new conflicts in the [rtRTMP](#) solution. The computational experiments are based on a Dutch control area. Although this procedure has the merit of dealing with a variable speed model, it makes the solution of the [rtRTMP](#) with no re-routing much longer. For this reason, it is not considered in the next papers. [D’Ariano et al. \(2008\)](#) present a real-time traffic management system, called [Railway traffic Optimization by Means of Alternative graphs \(ROMA\)](#), in which train re-scheduling and re-routing problems are tackled. A branch-and-bound algorithm is used for train rescheduling, as in the previous papers on the [AG](#) formulation, while a local search algorithm is developed for re-routing optimization purposes. An extensive computational study is carried out, based on a Dutch control area. The [rtRTMP](#) including a re-routing possibilities is also tackled by [Corman et al. \(2010\)](#). A tabu search algorithm is proposed to solve the problem. In the same control area as the previous paper, the authors address two variants of the [rtRTMP](#). The first one implements the green wave policy, which consists in letting trains wait at the stations to avoid speed profile modifications in open corridors ([Corman et al., 2009](#)). The second considers a bi-objective version of the [rtRTMP](#) ([Corman et al., 2012a](#)), where the minimization concerns both the maximum train delays and a measure of the missed connections. The proposed algorithm provides a set of feasible non-dominated schedules to support the dispatcher decision process. More recently, [Samà et al. \(2015\)](#) propose a Data Envelopment Analysis (DEA) framework for assessing a multi-objective version of the [rtRTMP](#). In the experimental analysis, no re-routing is allowed, to speed up the solution of the traffic management problem. However, the framework proposed does not impose any constraint in this sense, and in principle it may be used also when alternative routes are considered. [Corman and Quaglietta \(2015\)](#) study the performance of [ROMA](#) in a closed-loop framework. In a closed-loop, the optimization algorithm, [ROMA](#) in this case, exchanges information with the field (or with a simulator) on a regular basis. Specifically, the field sends information on the traffic state to the algorithm. The algorithm, based on the most recent information available, optimizes traffic management decisions and transmits them to the field. Further modules, as a human machine interface to allow dispatchers deciding whether to implement some specific decisions, may also be included in the loop.

Finally, in [Kecman et al. \(2013\)](#) the [AG](#) formulation is not used to tackle a specific variant of the [rtRTMP](#), but for assessing the impact of a macroscopic versus a microscopic representation of the infrastructure. The computational experiments based on a part of the Dutch network show that the more realistic the representation is, then when passing from macroscopic to microscopic, the more the computational time necessary to find good quality solutions increases and the more the model is realistic.

3.2.3 Meta-heuristics

Some authors propose meta-heuristics to tackle different variants of the [rtRTMP](#). [T.K.Ho and T.H.Yeung \(2001\)](#) propose three meta-heuristic algorithms (genetic algorithm, simulated annealing and tabu search) for routing trains through a railway junction. Then, [Fan et al. \(2012\)](#) compare various meta-heuristics on common scenarios. In the experimental analysis, a timetable generated specifically for the paper represents perturbed traffic in a portion of the English network. The authors find that simple disturbances (i.e., one train delayed) can be managed efficiently using straightforward approaches, such as first-come-first-served (FCFS). For more complex scenarios, advanced methods as colony optimization and genetic algorithms are found to be more appropriate. More recently, [Samà et al. \(2016\)](#) introduce the real-time Train Routing Selection Problem (rtTRSP). The rtTRSP is a sub-problem of the [rtRTMP](#). It consists in selecting an appropriate set of routing possibilities for each train, to be considered in the solution of the [rtRTMP](#). By doing so, the size of the instance of the latter problem is reduced, in principle without excluding the best quality solutions. The problem is modeled as an [ILP](#) and efficiently solved with an ant colony optimization algorithm in a short computational time. In a subsequent paper, [Samà et al. \(2017\)](#) investigate when the best moment to solve the Train Routing Selection Problem (TRSP) is, comparing the tactical and operational application. The authors claim that solving the TRSP at the operational level is the best choice when the traffic disturbance is large or very different from the perturbations which are typically observed on the part of the network considered.

Although the common techniques used to tackle the [rtRTMP](#) are based to the above three ones, other techniques as Constraint Programming (CP) is also used to deal with the [rtRTMP](#). In particular, [Rodriguez \(2007\)](#) uses CP for the [rtRTMP](#). Computational experiments on a French control area show that the CP algorithm can find satisfactory solutions within 3 min of computational time.

3.3 Decomposition approaches

In this section, we present decomposition approaches used in the literature to deal with the [rtRTMP](#). These approaches can be divided into three categories (spatial, temporal and mathematical decompositions), which are described in the following.

3.3.1 Spatial decomposition

The spatial decomposition is interesting when dealing with the [rtRTMP](#) in large control area. This kind of decomposition splits a control area into smaller dispatching zones. Each dispatch-

ing zone is tackled separately. This approach somehow mimics the procedure implemented by **IM** operators to manage traffic in large control areas.

The advantage of the spatial decomposition is the reduction of the problem difficulty by reducing the size of instances. However, a decision on how the dispatching zones shall be coordinated must be made, and this is often done accepting the sub-optimality of the overall solution.

The solution of the **rtRTMP** using a spatial decomposition is not often considered in the literature. To the best of our knowledge, only [Corman et al. \(2010, 2012b, 2014\)](#) present an approach to solve the problem of coordinating the traffic management decisions of multiple dispatchers. In [Corman et al. \(2010\)](#), the problem is formulated as a bi-level program with the objective of minimizing delay propagation. At the lower level, the system manages traffic in each dispatching zone independently, without any knowledge of traffic elsewhere. At the higher level, a coordinator module is responsible for the traffic management over the overall control area, with a global vision of traffic. The dispatchers' and the coordinator's problems are formulated with **AGs**, as explained in Section 3.2.2. The coordinator problem consists of imposing border constraints to dispatchers, i.e., constraints imposed on border block sections. These border constraints ensure that locally feasible schedules are globally feasible. The coordinator minimizes the maximum consecutive delay over all trains in the whole control area. In the model, there are two types of border constraints at the limit between dispatching zones. These constraints must be satisfied by the local solutions of all the concerned dispatching zones. The first type of constraints concerns time windows indicating <earliest, latest> entrance/exit times of a train into/out of a border block section: trains must enter and exit neighbor dispatching zones at coherent times. The second type of constraints concerns sequencing between two trains passing a border block section: the order of the trains entering and exiting two neighbor dispatching zones must be the same. A specific **AG** is adopted to model coordination constraints, while detailed graphs as the ones used in the papers described in Section 3.2.2 are used for the dispatching zones. Each dispatching zone is managed with the objective of minimizing the maximum consecutive delays of trains within it. A branch-and-bound algorithm is used to solve the problem. Computational experiments are made on a large part of the railway network in the South-East of the Netherlands. The papers by [Corman et al. \(2012b, 2014\)](#) follow the just mentioned work, with the difference that in [Corman et al. \(2010\)](#) only two dispatching control areas are taken into account, whereas in [Corman et al. \(2012b\)](#) the spatial decomposition is applied to multiple control areas. In [Corman et al. \(2014\)](#), the assessment of monolithic and spatial decomposition approaches for the **rtRTMP** is performed. The results suggest that when dealing with the **rtRTMP** in a large dispatching zone, the spatial decomposition approach outperforms the monolithic one in terms of solution quality while

3 Literature review on the rtRTMP

keeping computation time at an acceptable level. The re-routing possibility is not considered by the authors.

3.3.2 Temporal decomposition

The temporal decomposition is interesting to cut a long time horizon into tractable intervals. The solution of the [rtRTMP](#) over a long time horizon is obtained by solving progressively the problems over the shorter intervals.

As the spatial decomposition, the advantage of this approach is the reduction of the problem difficulty by reducing the size of instances. However, it may give a global sub-optimal solution.

To the best of our knowledge, the only temporal decomposition applied to the [rtRTMP](#) is the paper of [D’Ariano and Pranzo \(2009\)](#). The authors tackle the [rtRTMP](#) over several hours. They compare the solution obtained through the temporal decomposition with the so-called global approach, which corresponds to a monolithic one following the classification of this thesis. Computational experiments are made on instances representing traffic on the control area between Utrecht and Den Bosch in the Netherlands. The instances are solved using two algorithms: a truncated branch-and-bound algorithm and the first comes first served algorithm. According the authors, the computational analysis suggests the following observations. (1) In terms of delay minimization, the global approach gives better results compared to the application of the temporal decomposition. (2) When the time horizon is enlarged, the computational time for finding good solutions increases considerably for the global approach. This increase depends on the scheduling algorithm and on the instance size. The authors claim that when dealing with real time applications, the global approach is only suitable for time horizons of up to three hours. Furthermore the authors suggest that the reasonable time horizon for the [rtRTMP](#) is one hour. The re-routing possibility is not considered by the authors.

3.3.3 Mathematical decomposition

The mathematical decomposition exploits the structure of the [rtRTMP](#) formulation instead of decomposing it over time or space. This type of decomposition is widely used in integer programming to tackle large combinatorial optimization problems. There are mainly three mathematical decomposition approaches used in the [rtRTMP](#) literature: Lagrangian relaxation, Dantzig-Wolfe decomposition and [BD](#). They consist in decomposing a problem into sub-problems. These sub-problems are coordinated by a master problem in the solution process. A decomposition yields a new formulation (i.e., reformulation) of the original problem. If the algorithm used to solve the reformulation converges, it returns an optimal solution for the original problem. Unlike the two previous decomposition approaches used to tackle the

rtRTMP which return a global sub-optimal solution, a mathematical decomposition returns a global optimum if sufficient computational time is available. In the following, we present a short and easy description of the main principles of each type of mathematical decomposition. It is aimed to be an introduction of the main idea at its basis, and not a precise explanation of its functioning. For such an explanation, we refer the interested reader to the large body of dedicated literature existing.

3.3.3.1 Lagrangian relaxation

The Lagrangian relaxation separates the constraints of a problem into two groups, namely the ‘easy’ and the ‘hard’ ones. The hard constraints are removed and transferred into the objective function. For example, consider the following integer linear problem.

$$Z = \min \quad cx \tag{3.1}$$

$$Ax \geq b, \tag{3.2}$$

$$Dx \geq e, \tag{3.3}$$

$$x \in \mathbb{N}^n. \tag{3.4}$$

where $c \in \mathbb{R}^n$, $e, b \in \mathbb{R}^m$, $A, D \in \mathbb{R}^{nm}$.

If Constraints 3.2 make Z difficult, the Lagrangian relaxation can be used to relax these constraints: they are brought into the objective function with an associated vector μ called Lagrange multiplier. We refer to the resulting problem

$$Z_{RL}(\mu) = \min \quad cx + \mu(b - Ax) \tag{3.5}$$

$$Dx \geq e \tag{3.6}$$

$$x \in \mathbb{N}^n; \mu \in \mathbb{R}_+^m \tag{3.7}$$

as the Lagrangian relaxation or the Lagrangian sub-problem, and we refer to the function $Z_{RL}(\mu)$ as the Lagrangian function. As mentioned, we penalize in the objective function the violation of Constraints 3.2 through the Lagrangian multiplier μ .

To solve Z , the following optimization problem

$$Z_{(\mu)}^* = \max_{\mu} \quad Z_{RL}(\mu) \tag{3.8}$$

3 Literature review on the rtRTMP

is used, which we refer to as the Lagrangian dual problem associated with the original optimization problem. This dual problem is often solved with a sub-gradient method. This method consists in updating iteratively the value of variables of $Z_{RL}^*(\mu)$. Each iteration k consists in two main steps:

1. Given μ^k (i.e., Lagrangian multiplier of iteration k), compute $Z_{RL}(\mu^k)$ to obtain a solution x^k and the sub-gradient of Z_{RL} at point μ^k . We name this sub-gradient g^k and it corresponds to $(b - Ax^k)$,
2. Compute μ^{k+1} using μ^k and the step size $P^k = \frac{UB - Z_{RL}(\mu^k)}{\|b - Ax^k\|^2}$ of the iteration k , i.e., $\mu^{k+1} = \mu^k + P^k * g^k$. Where UB is an upper bound on the optimal solution we want to compute.

To the best of our knowledge, only [Meng and Zhou \(2014\)](#) propose a Lagrangian relaxation to cope with the **rtRTMP**. The authors propose a network flow formulation to model the problem which tackles the simultaneous train re-routing and rescheduling. The objective function of the model minimizes the total delay of all involved trains. The network flow formulation contains several big-M constraints reformulated by introducing new variables called cumulative flow variables. The cumulative flow variables are time-indexed. They represent both temporal and spatial resource consumption by trains. In the experimental analysis a discretization step of one minute is considered to keep the number of variables tractable. The Lagrangian relaxation technique is used to reformulate and solve the problem. The block section capacity constraints are brought in the objective function. By doing so, the original problem is separated into a path-finding sub-problem for each train. A Lagrangian relaxation algorithm based on the sub-gradient method and embedding a computationally efficient time-dependent shortest path algorithm is used to solve these path-finding sub-problems. The performance of this algorithm is tested in a fictitious network. Based on the obtained results, the authors claim that the algorithm can find feasible solutions with small gaps (up to 10%) for all test cases within about 1.3 min.

3.3.3.2 Dantzig-Wolfe decomposition

The Dantzig-Wolfe decomposition ([Dantzig and Wolfe, 1960](#)) is widely used in the linear programming field. Its application is particularly suitable if the problem to be tackled presents a specific structure. Specifically, two types of constraints are distinguished: the coupling constraints and the independent constraints. The coupling constraints contain all variables, while the independent ones contain a subset of variables.

As an example of application of the Dantzig-Wolfe decomposition, consider the same integer problem used in Section [3.3.3.1](#) and suppose that Constraints [\(3.2\)](#) are the coupling

3.3 Decomposition approaches

constraints and Constraints (3.3) are the independent ones. The Dantzig-Wolfe decomposition consists in separating the problem Z into two problems. The first problem contains the coupling constraints ($Ax \geq b$), whereas the second one contains the independent constraints ($Dx \geq e$). The coordination between the two problems is done by Dantzig-Wolfe Master Problem (MP_{DW}) that corresponds to:

$$MP_{DW} = \min \quad cx \quad (3.9)$$

$$Ax \geq b \quad (3.10)$$

$$x \in X \quad (3.11)$$

where $X = \{x \in N^n : Dx \geq e\}$ represents the polyhedron formed by the independent constraints. Another way to express X is as the convex hull combination of extreme points (J) and extreme rays (R), i.e.,

$$X = \{x \in N^n : x = \sum_{j \in J} \lambda_j x_j + \sum_{r \in R} \lambda_r x_r \mid \sum_{j \in J} \lambda_j = 1, \forall j \in J \quad \lambda_j \geq 0, \forall r \in R \quad \lambda_r \geq 0\} \quad (3.12)$$

Replacing X by its expression in (3.12), we obtain the formulation of the MP_{DW} equivalent to the original problem Z .

$$MP_{DW} = \min \quad c \sum_{j \in J} \lambda_j x_j + c \sum_{r \in R} \lambda_r x_r \quad (3.13)$$

$$A \sum_{j \in J} \lambda_j x_j + A \sum_{r \in R} \lambda_r x_r \geq b \quad (3.14)$$

$$\sum_{j \in J} \lambda_j = 1 \quad (3.15)$$

$$\lambda_j \geq 0 \quad \forall j \in J \quad (3.16)$$

$$\lambda_r \geq 0 \quad \forall r \in R. \quad (3.17)$$

Note that, in the Dantzig-Wolfe decomposition only the coupling constraints are relaxed. So, the approach may provide a better bound than the linear relaxation.

The Dantzig-Wolfe decomposition is solved using the column generation algorithm sketched in the following. Specifically, first, a restricted version of MP_{DW} , commonly referred to as the Restricted Master Problem (RMP_{DW}) using a subset of extreme points J' and extreme rays R'

3 Literature review on the rtRTMP

$$RMP_{DW} = \min \quad c \sum_{j \in J'} \lambda_j x_j + c \sum_{r \in R'} \lambda_r x_r \quad (3.18)$$

$$A \sum_{j \in J'} \lambda_j x_j + A \sum_{r \in R'} \lambda_r x_r \geq b \quad (3.19)$$

$$\sum_{j \in J'} \lambda_j = 1 \quad (3.20)$$

$$\lambda_j \geq 0 \quad \forall j \in J' \quad (3.21)$$

$$\lambda_r \geq 0 \quad \forall r \in R'. \quad (3.22)$$

is solved to obtain optimal values for the dual variables u and v associated with Constraints (3.19) and (3.20) respectively. Second, a sub-problem called Pricing Problem (PP) is used to generate a new column to be added to the RMP_{DW} that has negative reduced cost with respect to the dual solutions u and v of the RMP_{DW} . The PP is formulated as follows.

$$PP = \min_{j \in J, r \in R} (c - uA)x_j + (c - uA)x_r - v \quad (3.23)$$

The algorithm iterates these two steps and it stops when all variables have positive reduced cost, i.e., $PP \geq 0$ or the computational time limit is elapsed.

To the best of our knowledge, the only application of a column generation algorithm to tackle the rtRTMP is the one by [Lusby et al. \(2012\)](#). The authors use a set packing formulation, which is an integer linear program with a resource based constraint system. Two instances on a German complex junction from an actual midweek timetable operated by Deutsch Bahn with respectively one hour and two hour time horizons are tackled. The solution of the column generation algorithm and the solution of the plain Cplex are compared. The authors show that the column generation algorithm obtains the optimal solution of the two instances within 60s and outperforms the plain Cplex result. The authors do not consider re-routing possibilities.

3.3.3.3 Benders decomposition

The **BD** ([Benders, 1962](#)) is used to tackle **MILP** formulations, which contain both integer (or binary) and continuous variables. The principle of this decomposition consists in separating the integer (or binary) part (**UMP**) from the continuous one (**SP**) of the **MILP**. The decomposed problem is solved adding progressively to the **UMP** Benders cuts obtained from the **SP**.

As an example, consider the following **MILP**

$$Z : \min cx + fy \quad (3.24)$$

$$Ax + By \leq b \quad (3.25)$$

$$x \in \mathbb{N}, y \in \mathbb{R}. \quad (3.26)$$

where $f, c \in \mathbb{R}$, $A, B \in \mathbb{R}^m$ and $b \in \mathbb{R}^m$.

Suppose that the values of the integer variables x are given (\bar{x}). The above problem is decomposed in the following Benders **SP** with dual **DSP**:

$$\mathbf{SP} : \min c\bar{x} + fy \quad (3.27)$$

$$By \leq b - A\bar{x} \quad (3.28)$$

$$y \in \mathbb{R}. \quad (3.29)$$

$$\mathbf{DSP} : \max u(b - A\bar{x}) + c\bar{x} \quad (3.30)$$

$$uB \geq f \quad (3.31)$$

$$u \in \mathbb{R}. \quad (3.32)$$

where u are dual variables associated to Constraints (3.28).

Based on the above **DSP**, the Benders **Master Problem (MP)**, commonly referred as the **Restricted Master Problem (RMP)** is:

$$\mathbf{RMP} : \min cx + z \quad (3.33)$$

$$u(b - Ax) + cx \leq z \quad \forall u \in P \quad (3.34)$$

$$u(b - Ax) + cx \leq 0 \quad \forall u \in R \quad (3.35)$$

$$x \in \mathbb{N} \quad (3.36)$$

$$z \in \mathbb{R}. \quad (3.37)$$

where P and R are, respectively, the subsets of extreme points and extreme rays of polyhedron defined by Constraints (3.31) and (3.32). The additional dummy variable z represents the contribution of the continuous variables to the **RMP** objective function. The **RMP** has only one dummy variable z and the integer variables x of the original problem Z . Note that the values of u are known because the right hand side of (3.34) and (3.35) is either an extreme point P or an extreme ray R calculated in the **DSP**. Constraints (3.34) and (3.35) represent respectively the classic Benders **OC** and **FC**.

To solve the **BD**, one starts seeking a feasible solution to the **RMP** by fixing sets P and R to empty. The problem with $P = R = 0$, i.e., without Constraints (3.34) and (3.35), is called the

3 Literature review on the rtRTMP

UMP, which is formulated as follows.

$$UMP : \min cx + z \quad (3.38)$$

$$x \in \mathbb{N} \quad (3.39)$$

$$z \in \mathbb{R}. \quad (3.40)$$

The BD is solved using a row generation algorithm, commonly named as a BA. In each iteration of the algorithm, the UMP and the DSP are solved and a Benders cut, OC or FC, is added to the UMP depending the solution of the DSP. The algorithm stops when either the optimum of the overall problem is found (the objective function value is the same for the UMP and the DSP) or the computational time limit is elapsed.

Some heuristic algorithms based on the BD dealing with the rtRTMP exist. First of all, in the paper by Lamorgese and Mannino (2015) the rtRTMP is decomposed into two sub-problems. They are called Line Dispatching (LD) and Station Dispatching (SD) problem respectively. The LD problem deals with finding a minimum cost schedule such that all pairs of trains only meet in a station. Once this problem is solved then in the SD problem a platform is assigned to each train at each station separately. The authors consider this approach as equivalent to the classical BD method where the MP corresponds to the LD problem and the SP corresponds to the SD one. However, note that although the authors consider this approach as a BD, it is slightly different as both the LD and the SD are MILP problems. The LD problem, which is modeled macroscopically, provides train tentative arrival and departure times in the stations of the railway line. The SD problem modeled microscopically tries to assign to each train a platform at each station so that the given tentative arrival and departure times are satisfied. Cuts are progressively added to the LD problem based on the station capacity violations identified in the SD one. The experimental analysis made on Italian and Norwegian single and double line tracks show the algorithm improvement with respect to a previously proposed heuristic algorithm. However, to keep the solution of the SD problem quick, only a rough approximation of the microscopic representation of each station is considered. In a subsequent paper, Lamorgese et al. (2016) propose a variation of the former algorithm by making it more macroscopic. Specifically, in Lamorgese et al. (2016), instead of being associated to the precise locations in which trains can meet or pass each other, the binary variables are associated with entire sub-regions. Moreover, some algorithmic enhancements are added for solving the LD problem. On the same Italian and Norwegian instances, the authors report that the computational results with the new approach improve the ones presented in Lamorgese and Mannino (2015). Finally, in a recent seminar, Lamorgese and Mannino (2016) present a BD approach to tackle the train's rescheduling problem formulated with an AG model. In their approach, the authors replace the

Benders feasibility and optimality cuts with strong cuts obtained by strengthening and lifting the standards ones. In all these papers, the authors do not consider re-routing possibilities when dealing with the [rtRTMP](#).

3.4 Conclusions

In this chapter, we presented an overview of the state of the art for the [rtRTMP](#). We classified and discussed contributions on the [rtRTMP](#) according to two different approaches: the monolithic approach and the decomposition one. The former includes the contributions facing the problem as a unique element. The latter groups the contributions in which the problem is split in sub-problems, either in its definition or in the solution technique used to solve it. The decomposition approaches often aim to the solution of large instances.

Although many algorithms have been proposed to tackle the [rtRTMP](#) in the literature, most of them deal with the problem using the monolithic approaches. In several studies, the size of instances tackled are relatively small. Even if some authors use the decomposition approaches to deal with the [rtRTMP](#), they very often neglect the train re-routing possibility. The only decomposition approach that considers this possibility is the one proposed by [Meng and Zhou \(2014\)](#). However, the authors tackle a rather small artificial case study and are obliged to consider a very rough time discretization to keep the solution times reasonable. Moreover, the microscopic representation of the infrastructure at the [track-circuit](#) level is never considered in the cited literature that use the decomposition approach.

In this thesis, we consider the [RECIFE-MILP](#) algorithm, which allows solving some [rtRTMP](#) instances to optimality considering all details in the infrastructure. However, it has been shown ([Pellegrini et al., 2015](#)) that its performance may strongly worsen when tackling very large instances in the short time allowed by the real-time nature of the problem. For the [MILP](#) formulation at the basis of this algorithm, we propose [BD](#) approaches to improve the performance of [RECIFE-MILP](#) when tackling large instances.

Chapter 4

RECIFE-MILP

4.1 Introduction

In this chapter, we present [RECIFE-MILP](#) for the [rtRTMP](#) proposed by [Pellegrini et al. \(2015\)](#). [RECIFE-MILP](#) has been developed as part of the decision support platform called [RECIFE](#) introduced by [Rodriguez et al. \(2007\)](#). More specifically, [RECIFE-MILP](#) is a heuristic algorithm based on the truncated solution of a [MILP](#) formulation. This formulation is based on the microscopic representation of the infrastructure described in [Section 2.4](#) and it is then capable of considering all the characteristics of the infrastructure in deep details.

This chapter is organized as follows. [Section 4.2](#) details the [RECIFE-MILP](#) formulation assumptions. [Section 4.3](#) presents the mathematical formulation. [Section 4.4](#) describes the [RECIFE-MILP](#) algorithm. Some applications of [RECIFE-MILP](#) which appeared in the literature are presented in [Section 4.5](#). [Section 4.6](#) concludes the chapter.

4.2 Assumptions

The following assumptions are made to model the [rtRTMP](#) in [RECIFE-MILP](#).

The first assumption considered in [RECIFE-MILP](#) concerns the **blocking time theory utilization** presented in [Section 2.4](#). As previously explained, a blocking time is defined as the time interval that a [block section](#) is exclusively occupied by a train movement. It implies that a train is authorized to cross a [block section](#) if the two subsequent block sections are not reserved by another train. Moreover, in a 3-aspect (green, yellow and red) signaling system defined in [Section 2.3](#), a train is allowed to use a [block section](#) only if the [signal](#) is green. However, in real-life railway traffic situation, this condition does not need to be respected: a train can enter a block section opened by a yellow signal adjusting its speed to be able to stop in front of the

forthcoming signal, if red. The **blocking time theory utilization** assumption is made in most existing **rtRTMP** decision support tools as for example **ROMA**, a real-time traffic management system introduced by **D'Ariano et al. (2008)**.

The second assumption consists in the use of the so-called **fixed speed model** to describe train movements presented in Section 2.5. As previously mentioned, in a fixed speed model trains brake and re-accelerate instantaneously in case of **conflict**. Hence, the trains' entrance times in the different block sections in case of conflict are imprecise when a fixed speed model is applied. However, this imprecision is not an issue for the practical implementation of the traffic management decisions since what is actually imposed in reality is the set of precedences between trains, rather than the precise time at which trains are supposed to pass.

In addition to the above assumptions, in the variant of **RECIFE-MILP** considered in this thesis, we make the following ones to deal with slightly easier formulation. Remark that these assumptions are often made in the literature and do not strongly penalize the realism of the model.

First, we consider that **the shortest track-circuit is longer than all trains**. In real cases, a train can be longer than a **track-circuit** as exemplified in Figure 4.1. In this case, the duration of a track-circuit **utilization** can depend on a train standing still on another track-circuit. For example, the utilization of tc_1 lasts as long as train t stops in tc_2 . Instead, With this assumption exemplified in Figure 4.2, the duration of a track-circuit utilization is not influenced by a train standing still on another track-circuit. In the example, then, even if the blue train is stopped in tc_2 , we allow the green train to enter tc_1 .

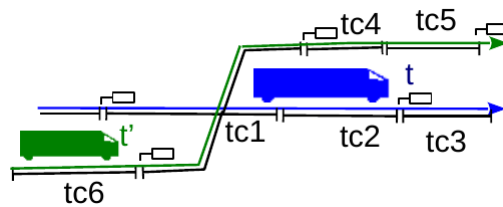


Figure 4.1 One train longer than the shortest track-circuit.

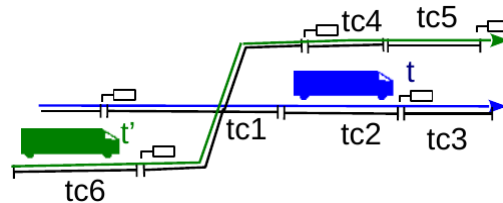


Figure 4.2 The shortest track-circuit is longer than all trains.

Second, we suppose the **dwell time**, that corresponds to the time for which a train stays at a station platform to allow passengers to disembark, transfer and board the train, **as fixed**. With the **fixed dwell time** assumption, we consider that the **dwell time** cannot be adjusted during the traffic optimization phase. In reality, the dwell time can be reduced or increased according to the dispatching strategies to suitably manage the delay propagation.

Finally, we do not consider any **Rolling-stock re-utilization**. Indeed, two trains t and t' can use a same rolling-stock. Hence, a minimum time must separate their arrival and departure, which must take place on the same track-circuit. Moreover, if train t and t' use the same rolling-stock and t' results from t , the **track-circuit** tc where the turnaround, join, or split takes place must be utilized for the whole time between t 's arrival and t' 's departure.

4.3 Formulation

In this section, we present the **RECIFE-MILP** formulation. This is one of the most accurate formulations for the **rtRTMP**, in which the infrastructure is modeled at the level of **track-circuit** which allows a representation of the route-lock sectional-release **interlocking** system (see Section 2.3) actually used in the practice.

In the **RECIFE-MILP** formulation, in addition to the actual track-circuits, two dummy ones are considered, tc_0 and tc_∞ . They represent the entry and the exit locations of the infrastructure, respectively: all routes start from tc_0 and end at tc_∞ . Each **track-circuit** has a **running time** and a **clearing time** which depend on the type of train traversing them and on the **route** traveled. Indeed, trains with different acceleration capabilities, for example, may need different running times to traverse a track-circuit. Moreover, several routes can share a **track-circuit** and the speed authorized on this track-circuit may depend on the route along which it is used. For the dummy track-circuits, the **running time** and **clearing time** are null for all train types and along all routes.

4.3.1 Notation

In the **RECIFE-MILP** formulation, the following notation is used for the input data:

$T \equiv$ set of trains;

$w_t \equiv$ weight associated to train t 's delay;

$ty_t \equiv$ type corresponding to train t (train characteristics);

$init_t, exit_t \equiv$ earliest time at which train t can be operated and earliest time at which it can reach its destination given $init_t$ and the **route** assigned in the timetable;

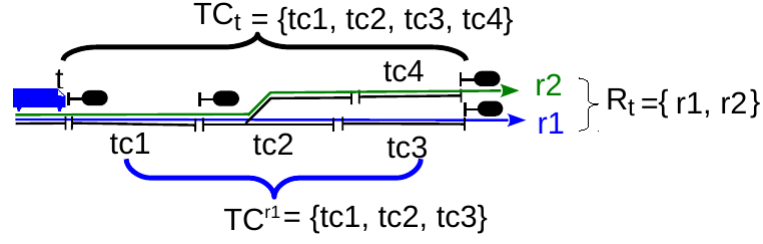


Figure 4.3 Set of routes R_t , track-circuits TC_t available for train t and set of track-circuits TC^{r1} composing route $r1$.

$R_t, TC_t \equiv$ set of routes and track-circuits available for train t , a graphical representation is showed in Figure 4.3;

$TC^r \equiv$ set of track-circuits composing route r , see the example of route $r1$ used by train t in Figure 4.3;

$TC(tc, tc', r) \equiv$ set of track-circuits between tc and tc' along r ;

$p_{r,tc}, s_{r,tc} \equiv$ track-circuits preceding and following tc along r ;

$rt_{ty,r,tc}, ct_{ty,r,tc} \equiv$ running time and clearing time of tc along r for a train of type ty ;

$bs_{r,tc} \equiv$ block section including track-circuit tc along route r ;

$for_{bs}, rel_{bs} \equiv$ formation time and release time for block section bs ;

$\hat{TC}_{t,t',tc} \equiv$ set of track-circuits tc' which may be used by both t and t' such that if t precedes (\prec) t' on tc , then necessarily $t \prec t'$ on tc' , and so on (e.g., if the track-circuits follow each other on a straight track segment). In other words, it is the set of track-circuits composing the track segment which starts with tc . $\hat{TC}_{t,t',tc} = \{tc\}$ if $tc \in TC_t \cap TC_{t'}$ and no implied precedence relation links tc to other track-circuits. $\hat{TC}_{t,t',tc} = \emptyset$ if $\exists tc' \in TC_t \cap TC_{t'}$ such that $tc \in \hat{TC}_{t,t',tc'}$, i.e., that is tc belongs to a track segment which starts with a different tc' . Hence, each track-circuit belongs to one and only one set $\hat{TC}_{t,t',tc}$;

$\hat{TC}_{t,t'} \equiv \cup_{tc \in TC_t \cap TC_{t'}} \hat{TC}_{t,t',tc}$, i.e., $\hat{TC}_{t,t'}$ is the set of all the track segments for the two trains;

$M \equiv$ large constant.

We will use this notation in the rest of this thesis.

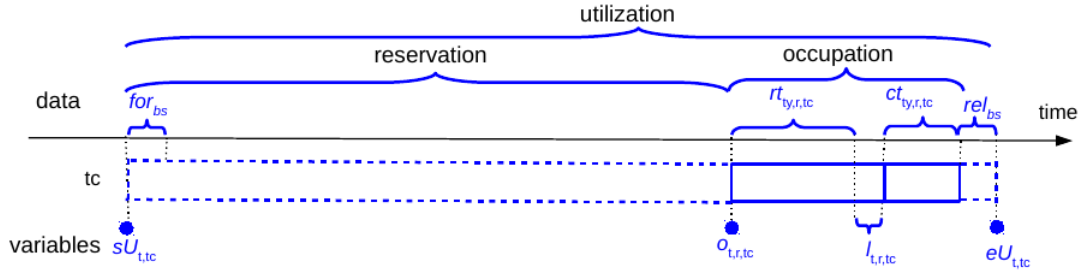


Figure 4.4 Main variables and data concerning the utilization of tc belonging to route r of train t .

4.3.2 Variables

The formulation contains non-negative **continuous variables**:

for all triplets of $t \in T$, $r \in R_t$ and $tc \in TC^r$:

$o_{t,r,tc}$: time at which t starts the **occupation time** of tc along r ;

$l_{t,r,tc}$: longer stay of t 's head on tc along r , due to **dwell time** and scheduling decisions (delay);

for all pairs of $t \in T$ and $tc \in TC_t$:

$sU_{t,tc}$: time at which t starts tc **utilization**;

$eU_{t,tc}$: time at which t ends tc utilization;

The above **continuous variables** are represented in Figure 4.4, together with the main input data (**clearing time**, **running time**, **release time** and **formation time**) concerning block sections and track-circuits.

for all $t \in T$:

D_t : delay suffered by train t when exiting the infrastructure.

In addition it includes **binary variables**:

for all pairs of $t \in T$ and $r \in R_t$:

$$x_{t,r} = \begin{cases} 1 & \text{if } t \text{ uses } r, \\ 0 & \text{otherwise,} \end{cases}$$

for all triplets of $t, t' \in T$ such that the index of t is smaller than the index of t' , and $tc \in TC_t \cap TC_{t'}$:

$$y_{t,t',tc} = \begin{cases} 1 & \text{if } t \text{ utilizes } tc \text{ before } t' (t \prec t'), \\ 0 & \text{otherwise } (t \succ t'). \end{cases}$$

We impose the index of t smaller than the index of t' to avoid creating two re-scheduling variables $y_{t,t',tc}$ and $y_{t',t,tc}$ for track-circuit tc . One variable is enough in this case since $y_{t',t,tc}$ is a complement to one of $y_{t,t',tc}$. By doing so, we reduce the number of y -variables in the formulation.

4.3.3 Objective and constraints

The [RECIFE-MILP](#) formulation is as follows:

$$\min \sum_{t \in T} w_t D_t. \quad (4.1)$$

$$o_{t,r,tc} \geq \text{init}_t x_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r : p_{r,tc} = tc_0, \quad (4.2)$$

$$o_{t,r,tc} \leq M x_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r, \quad (4.3)$$

$$o_{t,r,tc} = o_{t,r,p_{r,tc}} + l_{t,r,p_{r,tc}} + r_{t,r,ty,p_{r,tc}} x_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r : tc \neq tc_0, \quad (4.4)$$

$$\sum_{r \in R_t} x_{t,r} = 1 \quad \forall t \in T, \quad (4.5)$$

$$D_t \geq \sum_{r \in R_t} o_{t,r,tc_\infty} - \text{exit}_t \quad \forall t \in T, \quad (4.6)$$

$$sU_{t,tc} \leq \sum_{\substack{r \in R_t: \\ tc \in TC^r}} \left(o_{t,r,ref_{r,tc}} - \text{for}_{bs_{r,tc}} x_{t,r} \right) \quad \forall t \in T, tc \in TC_t, \quad (4.7)$$

$$eU_{t,tc} \geq \sum_{\substack{r \in R_t: \\ tc \in TC^r}} o_{t,r,ref_{r,tc}} + \text{rel}_{bs_{r,tc}} x_{t,r} + \text{ct}_{r,ty,tc} x_{t,r} + \sum_{tc' \in TC(\text{ref}_{r,tc}, tc, r)} r_{t,r,ty,tc'} x_{t,r} + l_{t,r,tc'} \quad \forall t \in T, tc \in TC_t, \quad (4.8)$$

$$eU_{t,tc} - M(1 - y_{t,t',tc}) \leq sU_{t',tc} \quad \forall t, t' \in T, t < t', tc \in TC_t \cap TC_{t'}, \quad (4.9)$$

$$eU_{t',tc} - My_{t,t',tc} \leq sU_{t,tc} \quad \forall t, t' \in T, t < t', tc \in TC_t \cap TC_{t'}. \quad (4.10)$$

In the formulation, the objective (4.1) is the minimization of the total weighted delays suffered by trains at their exit from the infrastructure. The weighting in the objective function allows taking into account different train priorities. These priorities may be linked to the type of circulation (e.g., freight or passenger) or other aspects as the number of passengers traveling on each train. Constraints (4.2) state that a train t cannot be operated earlier than $init_t$. Constraints (4.3) indicate that the start time of [track-circuit occupation time](#) along a [route](#) is zero if the route itself is not used. Constraints (4.4) impose that a train starts occupying track-circuit tc along a route after spending in the preceding track-circuit its longer stay and its [running time](#), if the route is used. Constraints (4.5) state that a train must use exactly one route. Constraints (4.6) indicate that the value of delay D_t at least equals the difference between the actual and the scheduled arrival times at the exit of the infrastructure. Constraints (4.7) impose that a train's [utilization](#) of a track-circuit tc starts as soon as the train starts occupying the track-circuit $ref_{r,tc}$ along one of the routes including tc , minus the [formation time](#). Constraints (4.8) indicate that the utilization of a track-circuit lasts till the train utilizes it along any route, plus the [formation time](#) and the [release time](#). Constraints (4.9) and (4.10) impose that the track-circuit utilization by two trains must not overlap.

4.4 Algorithm

[RECIFE-MILP](#) is a heuristic algorithm which mainly consists in tackling a [rtRTMP](#) instance by solving a [MILP](#) formulation for a limited computational time through a [MILP](#) solver. If the optimum is proven, the run is interrupted, otherwise, the best solution found after the elapse of this time is returned.

The [RECIFE-MILP](#) algorithm uses three boosting methods to improve the solution performance. In particular, their aim is, first, to increase the speed of detection of high quality feasible solutions and, second, to decrease the computation time needed for proving optimality.

In the following, we present these three boosting methods used to improve the [RECIFE-MILP](#) algorithm performance:

Initialization: providing an initial solution. In [RECIFE-MILP](#), a two-optimization-step cycle is implemented. In the first step, the [MILP](#) solver optimizes the train scheduling without modifying the routes with respect to the default ones (i.e., the routes defined in the initial

timetable). In the second step, it optimizes also in terms of train routing, using as initial solution the best solution found in the first step. Disregarding the optimality proof, the first step is terminated after 30 seconds provided that one feasible solution has been found. Otherwise, the first step continues until the first feasible solution is detected; in all experiments run so far, the search with no re-routing (first optimization step) always found at least a feasible solution within very few seconds.

Decreasing the big- M value: The solution obtained in the first optimization step is exploited to decrease the value of M , the large constant used in the formulation. For ensuring the coherence of Constraints (4.11) and (4.12), M needs to be at least equal to the latest end of a concerned **track-circuit utilization**. Let S^{*1} be the solution returned by the first optimization step and wD^{*1} its total weighted delay. In the second optimization step, all solutions improving over S^{*1} will have an associated total weighted delay not greater than wD^{*1} . Thus, the latest **utilization** of a **track-circuit** tc by a train t will be at most equal to the sum of wD^{*1}/w_t and the maximum of the earliest possible exit time from tc along each available route. Formally, let $\bar{u}_{t,tc}$ be the latest time at which t may end the utilization of track-circuit tc during the second optimization step. In any solution improving over S^{*1} it holds that

$$\bar{u}_{t,tc} \leq \frac{wD^{*1}}{w_t} + \max_{r \in R_t} \left\{ \text{init}_t + \sum_{\substack{tc' \in TC(tc_0, tc, r) \\ tc \in TC^r}} rt_{r, tyt, tc'} + ct_{r, tyt, tc} + rel_{bsr, tc} \right\};$$

M can be set equal to the maximum of these quantities across the trains involved in each constraint.

Infrastructure topology exploitation: The **RECIFE-MILP** algorithm exploits the characteristics of the infrastructure topology to reduce as much as possible the number of binary y -variables, i.e., the variables appearing in the disjunctive Constraints (4.9) and (4.10) that define the precedence relation between couples of trains utilizing the same track-circuit. To this aim, it exploits the fact that the topology of an infrastructure frequently imposes that the precedence relation between couples of trains must be identical on different track-circuits. Consider the situation exemplified in Figure 4.5 in which a simple infrastructure is traversed by two trains t and t' , using routes $r1$ and $r2$, respectively. The topology of this infrastructure and the routes used by trains t and t' define two **track segments** $\hat{TC}_{t, t', tc_7} = \{tc_7, tc_8\}$ and $\hat{TC}_{t, t', tc_{11}} = \{tc_{11}, tc_{12}\}$. Indeed, in every feasible solution where t uses $r1$ and t' uses $r2$, if t precedes t' (respectively, t' precedes t) on tc_7 or tc_{11} then t (respectively, t') must necessarily traverse tc_8 or tc_{12} before t' (respectively, t). Hence, in a preprocessing phase, for the couple of trains t, t' and on the basis of the physical network topology, first the set of shared track-circuits $TC_t \cap TC_{t'}$ is partitioned into subsets defined as follows: two track-circuits belong to the same

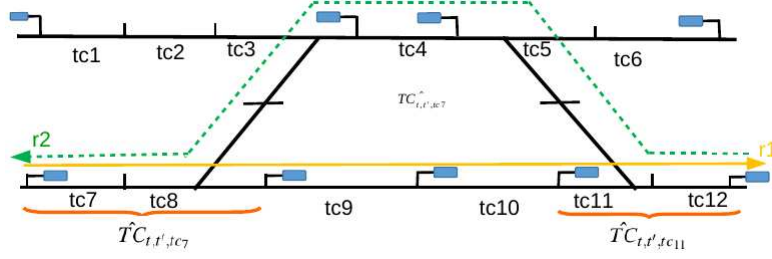


Figure 4.5 Graphical representation of track segments $\hat{T}C_{t,t',tc7}$, $\hat{T}C_{t,t',tc11}$ in which the order between trains cannot change.

subset $\hat{T}C_{t,t',\hat{f}c}$ if the same precedence relation between t and t' must hold on both track-circuits (e.g, in Figure 4.5, the precedence relation between t and t' cannot change on track-circuits $tc7$ and $tc8$, so they are grouped into same track segment $\hat{T}C_{t,t',tc7}$). Then, for each non empty $\hat{T}C_{t,t',\hat{f}c}$, a single $y_{t,t',\hat{f}c}$ variable is created in Constraints (4.9) and (4.10): $y_{t,t',tc7}$ is used for $tc7$ and $tc8$, $y_{t,t',tc11}$ is used for $tc11$ and $tc12$

As a consequence, Constraints (4.9) and (4.10) become Constraints (4.11) and (4.12), respectively:

$$eU_{t,tc} - M(1 - y_{t,t',\hat{f}c}) \leq sU_{t',tc} \quad \forall t, t' \in T, t < t', tc, \hat{f}c \in TC_t \cap TC_{t'} : tc \in \hat{T}C_{t,t',\hat{f}c}, \quad (4.11)$$

$$eU_{t',tc} - My_{t,t',\hat{f}c} \leq sU_{t,tc} \quad \forall t, t' \in T, t < t', tc, \hat{f}c \in TC_t \cap TC_{t'} : tc \in \hat{T}C_{t,t',\hat{f}c}. \quad (4.12)$$

For a more detailed discussion of the RECIFE-MILP formulation and algorithm we refer the interested reader to Pellegrini et al. (2014, 2015).

4.5 RECIFE-MILP applications

Since its proposal, RECIFE-MILP has been validated on several applications coming from Sweden, the UK, the Netherlands (Quaglietta et al., 2016; Ghaemi et al., 2017) and France (Pellegrini et al., 2016; Samà et al., 2016). In the following, we summarize some applications in which RECIFE-MILP is used. In Quaglietta et al. (2016), the authors propose a modular framework which may be used to practically deploy real-time traffic management algorithms. They use both RECIFE-MILP and ROMA (D'Ariano et al., 2008) in closed-loop with a microscopic traffic simulator. The two algorithms are applied to tackle instances representing traffic in three very different infrastructures: a portion of the East Coast Main Line in the

UK, a part of the Dutch network and a section of the Iron Ore line in Sweden and Norway. The perturbations considered are based on real timetables and specified by the concerned rail infrastructure managers. [Ghaemi et al. \(2017\)](#) present a **MILP** model for optimally short-turning trains in case of complete blockage and no train can use part of the track for several hours. Here, the authors adopt a variant of the **RECIFE-MILP** formulation which allows for short-turning in case of complete blockage. In [Pellegrini et al. \(2016\)](#), the authors compare the traffic management decisions of **RECIFE-MILP** to the ones of different strategies, according to several KPI's. Specifically, the authors compare the decisions made by **RECIFE-MILP** with those made by the dispatchers. This comparison is made on scenarios suggested by SNCF Réseau, French infrastructure manager on portions of the French network. The authors show that on all case studies tackled, **RECIFE-MILP** clearly outperformed the other strategies assessed. [Samà et al. \(2016\)](#) introduce the real-time Train Routing Selection Problem (rtTRSP) which consists in selecting a best subset of routes for each train to compute the **rtRTMP**. In the paper, the authors analyze the performance of **RECIFE-MILP** when using various routing subsets. The authors claim that, giving a subset of alternative routes determined when solving the rtTRSP, **RECIFE-MILP** achieves better performance than ACO meta-heuristic.

Despite the good performance achieved in all these applications, it is indeed possible to define instances for which **RECIFE-MILP** finds it difficult to return a high quality solution in real-time. The difficulty is very often linked to the size of the formulation describing the instances, which may easily include several tens of thousands of binary variables.

4.6 Conclusions

In this chapter, we presented **RECIFE-MILP** for the **rtRTMP**. The aim was to present **RECIFE-MILP** model assumptions, mathematical formulation and algorithm. We also presented some **RECIFE-MILP** applications which have appeared in the literature. These applications motivate the interest we devote to **RECIFE-MILP**, which is among the state of the art algorithms for the **rtRTMP** but can be incapable of dealing with really large instances. This chapter is a basis for the next chapters in which we propose a **BD** approach for the **RECIFE-MILP** formulation.

Chapter 5

Standard Benders Reformulation for RECIFE-MILP

5.1 Introduction

In Section 4.3, we presented the [RECIFE-MILP](#) formulation, which contains both binary and continuous variables. The binary variables determine the re-routing and rescheduling decisions, whereas the continuous ones set the re-timing decisions during the [rtRTMP](#) optimization.

As stated in Chapter 1, in this thesis we propose a [BD](#) to improve the performance of [RECIFE-MILP](#) when tackling large instances. In our [BD](#), we separate the [rtRTMP](#) into two problems: the [UMP](#) and [SP](#) following the principles described in Section 3.3. The [UMP](#) is the real-time train routing and scheduling problem: it contains the re-routing and rescheduling variables (binary variables) and one dummy variable representing the contribution of continuous variables to the [UMP](#) objective function. The schedule here is intended as the order in which trains cross common tracks, and does not include time information. The [SP](#) contains the continuous variables which determine the [track-circuit occupation time](#) and [reservation times](#), and deduces the delay suffered by each train. The decomposed problem is solved by progressively adding to the [UMP](#) Benders cuts obtained from the [SP](#).

This chapter is organized as follows. Section 5.2 presents the [SBR](#) for [RECIFE-MILP](#). Section 5.3 details the [BA](#) which we apply to solve the [SBR](#). Computational experiments are reported in Section 5.4, and Section 5.5 concludes the chapter.

5.2 Standard Benders Reformulation: SBR

In this section, we present the **SBR** for the **RECIFE-MILP** formulation. In the **SBR**, the **SP** contains all constraints and continuous variables related to a given routing and scheduling. These decisions are determined by the **UMP** and include the information on the routes assigned to the trains $(\bar{x}_{t,r})$ and the scheduling decisions $(\bar{y}_{t,t',tc})$.

5.2.1 Sub-Problem (SP)

By fixing routing and scheduling variables in the **RECIFE-MILP** formulation (let them be $\bar{x}_{t,r}$ and $\bar{y}_{t,t',tc}$) we get the **SP**. This **SP** contains the continuous variables which determine the **track-circuits occupation time** and **reservation times**, and deduces the delay suffered by train. Its formulation is as follows.

$$\min \sum_{t \in T} w_t D_t. \quad (5.1)$$

$$o_{t,r,tc} \geq \text{init}_t \bar{x}_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r : p_{r,tc} = tc_0, \quad (5.2)$$

$$o_{t,r,tc} \leq M \bar{x}_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r, \quad (5.3)$$

$$-o_{t,r,tc} + o_{t,r,p_{r,tc}} + l_{t,r,p_{r,tc}} = -rt_{r,ty,p_{r,tc}} \bar{x}_{t,r} \quad \forall t \in T, r \in R_t, tc \in TC^r : tc \neq tc_0, \quad (5.4)$$

$$D_t - \sum_{r \in R_t} o_{t,r,tc_\infty} \geq -\text{exit}_t \quad \forall t \in T, \quad (5.5)$$

$$sU_{t,tc} - \sum_{\substack{r \in R_t: \\ tc \in TC^r}} o_{t,r,ref_{r,tc}} \leq - \sum_{\substack{r \in R_t: \\ tc \in TC^r}} \text{for}_{bs_{r,tc}} \bar{x}_{t,r} \quad \forall t \in T, tc \in TC_t, \quad (5.6)$$

$$eU_{t,tc} - \sum_{\substack{r \in R_t: \\ tc \in TC^r}} o_{t,r,ref_{r,tc}} - \sum_{tc' \in TC(\text{ref}_{r,tc}, tc, r)} l_{t,r,tc'} \geq \sum_{\substack{r \in R_t: \\ tc \in TC^r}} \text{rel}_{bs_{r,tc}} \bar{x}_{t,r} + ct_{r,ty,tc} \bar{x}_{t,r} + \sum_{tc' \in TC(\text{ref}_{r,tc}, tc, r)} rt_{r,ty,tc'} \bar{x}_{t,r} \quad \forall t \in T, tc \in TC_t, \quad (5.7)$$

5 Standard Benders Reformulation for RECIFE-MILP

$$eU_{t,tc} - sU_{t',tc} \leq M - M\bar{y}_{t,t',\hat{c}} \quad \forall t, t' \in T, t < t', tc, \hat{c} \in TC_t \cap TC_{t'} : tc \in \hat{TC}_{t,t',\hat{c}}, \quad (5.8)$$

$$eU_{t',tc} - sU_{t,tc} \leq M\bar{y}_{t,t',\hat{c}} \quad \forall t, t' \in T, t < t', tc, \hat{c} \in TC_t \cap TC_{t'} : tc \in \hat{TC}_{t,t',\hat{c}}. \quad (5.9)$$

Let $\alpha_{t,r,tc}$, $\beta_{t,r,tc}$, $\lambda_{t,r,tc}$, θ_t , $\phi_{t,tc}$, $\omega_{t,tc}$, $\eta_{t,t',tc}$, $\psi_{t,t',tc}$, be the dual variables associated with Constraints (5.2)-(5.9) respectively. As such, $\alpha_{t,r,tc} \geq 0$, $\beta_{t,r,tc} \leq 0$, $\lambda_{t,r,tc} \in \mathbb{R}$, $\theta_t \geq 0$, $\phi_{t,tc} \leq 0$, $\omega_{t,tc} \geq 0$, $\eta_{t,t',tc} \leq 0$ and $\psi_{t,t',tc} \leq 0$. The DSP is written as follows.

5.2.2 Dual of the Sub-Problem (DSP)

$$\begin{aligned} & \max \sum_{t \in T} \sum_{r \in R_t} \sum_{tc \in TC^r} \bar{x}_{t,r} (\text{init}_t \alpha_{t,r,tc} + M\beta_{t,r,tc} - rt_{r,ty,p_{r,tc}} \lambda_{t,r,tc}) + \\ & + \sum_{t \in T} \sum_{r \in R_t} \sum_{tc \in TC_t} \bar{x}_{t,r} [(rel_{bs_{r,tc}} + ct_{r,ty,tc} + \sum_{tc' \in TC(ref_{r,tc},tc,r)} rt_{r,ty,tc'}) \omega_{t,tc} - for_{bs_{r,tc}} \phi_{t,tc}] - \\ & - \sum_{t \in T} exit_t \theta_t + \sum_{t,t' \in T} \sum_{\substack{\hat{c}, tc \in TC_t \cap TC_{t'} \\ tc \in \hat{TC}_{t,t',\hat{c}}}} M[(1 - \bar{y}_{t,t',\hat{c}}) \eta_{t,t',tc} + \bar{y}_{t,t',\hat{c}} \psi_{t,t',tc}]. \quad (5.10) \end{aligned}$$

$$\alpha_{t,r,tc} + \beta_{t,r,tc} + \lambda_{t,r,tc} - \phi_{t,tc} - \omega_{t,tc} \leq 0 \quad \forall t \in T, r \in R_t, tc \in TC^r : p_{r,tc} = tc_0, \quad (5.11)$$

$$\lambda_{t,r,tc} - \lambda_{t,r,p_{r,tc}} + \beta_{t,r,tc} - \phi_{t,tc} - \omega_{t,tc} \leq 0 \quad \forall t \in T, r \in R_t, tc \in TC^r, \quad (5.12)$$

$$\lambda_{t,r,tc} - \omega_{t,tc} \leq 0 \quad \forall t \in T, r \in R_t, tc \in TC^r, \quad (5.13)$$

$$\alpha_{t,r,tc_\infty} + \beta_{t,r,tc_\infty} - \lambda_{t,r,tc_\infty} - \theta_t \leq 0 \quad \forall t \in T, r \in R_t, \quad (5.14)$$

$$\theta_t \leq 1 \quad \forall t \in T, \quad (5.15)$$

$$\phi_{t,tc} - \eta_{t,t',tc} - \psi_{t,t',tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC_t \cap TC_{t'}, \quad (5.16)$$

$$\omega_{t,tc} + \eta_{t,t',tc} + \psi_{t,t',tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC_t \cap TC_{t'}. \quad (5.17)$$

Given the [DSP](#), we can generate the cuts that are necessary to reformulate the [RECIFE-MILP](#) formulation as the [MP](#), commonly referred as the [RMP](#).

5.2.3 Restricted Master Problem (RMP)

$$\min z \quad (5.18)$$

$$\begin{aligned} & z - \left(\sum_{t \in T} \sum_{r \in R_t} \sum_{tc \in TC^r} \bar{x}_{t,r} (\text{init}_t \alpha_{t,r,tc} + M \beta_{t,r,tc} - rt_{r,ty,pr,tc} \lambda_{t,r,tc}) + \right. \\ & + \sum_{t \in T} \sum_{r \in R_t} \sum_{tc \in TC_t} \bar{x}_{t,r} [(rel_{bs_{r,tc}} + ct_{r,ty,tc} + \sum_{tc' \in TC(ref_{r,tc},tc,r)} rt_{r,ty,tc'}) \omega_{t,tc} - for_{bs_{r,tc}} \phi_{t,tc}] - \\ & \left. - \sum_{t \in T} exit_t \theta_t + \sum_{t,t' \in T} \sum_{\hat{tc} \in TC_t \cap TC_{t'}: tc \in \hat{TC}_{t,t',\hat{tc}}} M[(1 - \bar{y}_{t,t',\hat{tc}}) \eta_{t,t',tc} + \bar{y}_{t,t',\hat{tc}} \psi_{t,t',tc}] \right) \geq 0 \\ & ((\alpha, \beta, \lambda, \theta, \phi, \omega, \eta, \psi) \in P_S), \quad (5.19) \end{aligned}$$

$$\begin{aligned} & \sum_{t \in T} \sum_{r \in R_t} \sum_{tc \in TC^r} \bar{x}_{t,r} (\text{init}_t \alpha_{t,r,tc} + M \beta_{t,r,tc} - rt_{r,ty,pr,tc} \lambda_{t,r,tc}) + \\ & + \sum_{t \in T} \sum_{r \in R_t} \sum_{tc \in TC_t} \bar{x}_{t,r} [(rel_{bs_{r,tc}} + ct_{r,ty,tc} + \sum_{tc' \in TC(ref_{r,tc},tc,r)} rt_{r,ty,tc'}) \omega_{t,tc} - for_{bs_{r,tc}} \phi_{t,tc}] - \\ & - \sum_{t \in T} exit_t \theta_t + \sum_{t,t' \in T} \sum_{\hat{tc} \in TC_t \cap TC_{t'}: tc \in \hat{TC}_{t,t',\hat{tc}}} M[(1 - \bar{y}_{t,t',\hat{tc}}) \eta_{t,t',tc} + \bar{y}_{t,t',\hat{tc}} \psi_{t,t',tc}] \leq 0 \\ & ((\alpha, \beta, \lambda, \theta, \phi, \omega, \eta, \psi) \in R_S), \quad (5.20) \end{aligned}$$

$$\sum_{r \in R_t} x_{t,r} = 1 \quad \forall t \in T, \quad (5.21)$$

$$x_{t,r} \in \{0, 1\} \quad \forall t \in T, r \in R_t, \quad (5.22)$$

$$y_{t,t',\hat{tc}} \in \{0, 1\} \quad \forall t, t' \in T, t < t', \hat{tc} \in \hat{TC}_{t,t'}, \quad (5.23)$$

5 Standard Benders Reformulation for RECIFE-MILP

$$z \geq 0. \quad (5.24)$$

The additional dummy variable z represents the contribution of the **RECIFE-MILP** continuous variables to the **RMP** objective function, while sets P_S and R_S contain the extreme points and extreme rays, respectively, of the polyhedron S representing the feasible solution space of the **DSP**. The **RMP** has only one dummy variable z and the binary variables $x_{t,r}$ and $y_{t,t',\hat{c}}$ of the original problem. Note that the values of $\alpha, \beta, \lambda, \theta, \phi, \omega, \eta, \psi$ are known because they are either the extreme points P_S or the extreme rays R_S calculated in the **DSP**. Constraints (5.19) and (5.20) represent respectively the classic Benders **OC** and **FC**. Constraints (5.21) represent the routing constraints equivalent to (4.5).

To solve the **BD**, we start seeking a feasible solution to the **RMP** by fixing sets P_S and R_S to empty. The problem with $P_S = R_S = \emptyset$, i.e., without Constraints (5.19) and (5.20), is called the **UMP**, which is formulated as follows.

5.2.4 Unrestricted Master Problem (UMP)

$$\min z \quad (5.25)$$

$$\sum_{r \in R_t} x_{t,r} = 1 \quad \forall t \in T, \quad (5.26)$$

$$x_{t,r} \in \{0, 1\} \quad \forall t \in T, r \in R_t, \quad (5.27)$$

$$y_{t,t',\hat{c}} \in \{0, 1\} \quad \forall t, t' \in T, t < t', \hat{c} \in \hat{T}C_{t,t'}, \quad (5.28)$$

$$z \geq 0. \quad (5.29)$$

In **BA**, Benders cuts (5.19) and (5.20) obtained in the **DSP** are added to the **UMP** iteratively until the optimal solution is found or the computational time limit is reached.

5.3 Benders Algorithm: BA

We solve our **SBR** of the **RECIFE-MILP** formulation with a **BA**.

In our **BA**, the **UMP** and the **DSP** are iteratively solved at the optimum, see Algorithm 1. At each iteration, we add to the **UMP** either an **OC** or a **FC** depending the solution of the **DSP** until the optimum is found (the objective function value is the same for the **UMP** and the **DSP**) or the computational time limit is elapsed.

Algorithm 1: Benders algorithm (BA).

Data: Problem instance

Result: Solution of the overall problem

Initialization:

set $UB = +\infty$, $LB = -\infty$, $BestSolution = \emptyset$;

while $LB \neq UB$ and Computational time available **do**

Solve **UMP** to optimality and find \bar{z} , $\bar{x}_{t,r}$, $\bar{y}_{t,t',\hat{c}}$;

Set $LB = \bar{z}$;

Solve **DSP** with $\bar{x}_{t,r}$ and $\bar{y}_{t,t',\hat{c}}$ and find $\bar{o}_{t,r,tc}$, $\bar{l}_{t,r,tc}$, $\bar{s}U_{t,tc}$, $\bar{e}U_{t,tc}$ and \bar{D}_t ;

Add corresponding cut (**OC** or **FC**) to the **UMP**;

if **DSP** bounded and $\bar{z} \leq UB$ **then**

Set $UB = \bar{z}$ record $BestSolution = \{\bar{x}_{t,r}, \bar{y}_{t,t',\hat{c}}, \bar{o}_{t,r,tc}, \bar{l}_{t,r,tc}, \bar{s}U_{t,tc}, \bar{e}U_{t,tc}, \bar{D}_t\}$;

Stop.

Return $BestSolution$;

Note that the **UMP** cannot be infeasible due to its formulation. As a result, the infeasibility case of the **UMP** is not considered in the algorithm.

Similarly to what is done in **RECIFE-MILP**, we initialize the search with a feasible solution. In particular, we solve the **RECIFE-MILP** formulation by fixing the train routing decisions to the default routes defined in the timetable (fixed routes). Then, we initialize the **SBR** with this fixed route optimal solution or with the best solution found after 30 seconds of computation. In our experiments, this threshold was never reached.

5.4 Computational experiments

In this section, we assess the performance of the **SBR**. The reformulation is solved using the **BA** through Cplex (from now on **BA-SBR**). Specifically, the implementation is done using IBM ILOG CPLEX Concert Technology for C++ (IBM ILOG CPLEX version 12.7 (**IBM**, 2017)). We consider two case studies representing traffic in two control areas in France: the Pierrefitte–Gonesse junction (Gonesse) and a line section around the Rouen–Rive–Droite station (Rouen). These case studies are the ones used in all the experimental analyze presented

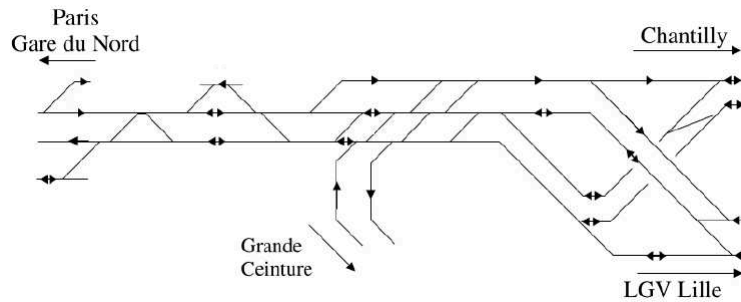


Figure 5.1 Pierrefitte–Gonesse junction.

in this thesis. We set the computational time available for the optimization to three minutes as often done in the literature (e.g., [Pellegrini et al. \(2015\)](#); [Samà et al. \(2015\)](#)).

The weight associated to delay in the objective function is set to 1 for all trains, since we have no information on different train priorities. The route formation and [release times](#) are 15 and 5 seconds for all [block sections](#), respectively.

For each control area, we consider a one-day timetable, and we create 30 scenarios in which 20% of trains, randomly selected, suffer a random delay between 5 and 15 minutes at their entrance in the control area. We generate one [rtRTMP](#) instance from each of these 30 scenarios by considering all the trains entering the control area within a one-hour horizon. We set the time horizon from 6:00 am to 07:00 am. This time horizon corresponds to the morning peak hour. We use the so obtained 60 instances for testing the [BA-SBR](#) performance.

Before presenting the results, we describe each control area in the next subsections.

5.4.1 Pierrefitte–Gonesse junction

Gonesse is a critical control area with intense mixed traffic. Figure 5.1 schematically depicts its infrastructure. It includes 89 track-circuits, grouped into 174 block sections and 39 routes. A week-day timetable includes 340 trains: 120 high-speed and 129 conventional passenger trains, and 91 freight trains. The one-hour instances we tackle include between 14 and 17 trains (mean 15). Each train can use between 5 and 13 routes (mean 8), which translates into a [RECIFE-MILP](#) formulation with about 121 000 continuous variables, 1 800 binary variables and 35 000 constraints for an instance with 15 trains.

5.4.2 Rouen-Rive-Droite control area

The control area including Rouen-Rive-Droite comprises six stations, with two to six platforms, and one junction. The presence of multiple stations with several possible platform assignments implies the availability of a very large number of alternative routes. The existence of these

5.4 Computational experiments

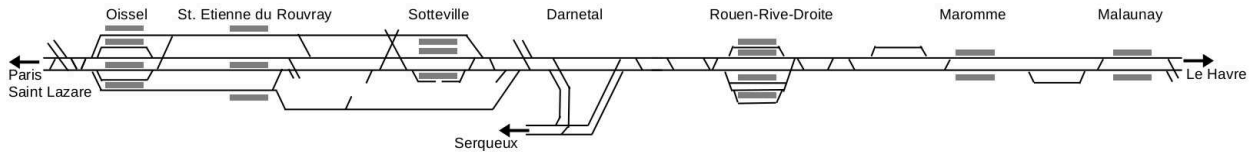


Figure 5.2 Rouen-Rive-Droite control area.

routes is the main source of complexity of traffic management in this control area, together with the presence of mix traffic. The control area is depicted in Figure 5.2. The 190 track-circuits compose 189 block sections and 11 347 routes. The one-day timetable considered includes 186 trains: 2 high-speed and 107 conventional passenger trains, 33 freight trains and 44 local movements. The obtained one-hour instances include between 10 and 13 trains (mean 11). Each train can use between 1 and 384 routes (mean 68), which translates into a **RECIFE-MILP** formulation with about 949 800 continuous variables, 1 030 binary variables and 224 900 constraints, for an instance including 11 trains.

5.4.3 Results

In this section, we present the results of our computational experiments on the two control areas. The aim of these computational experiments is to evaluate the **BA-SBR** performance. As mentioned in Section 5.3, **BA-SBR** starts with an initial solution found in the first optimization step, which is equal to the one of **RECIFE-MILP** (see Section 4.4). However, for sake of completeness, we first compare the results of **BA-SBR** without the initialization with the ones of **RECIFE-MILP**. Then, we compare them when using the initialization.

Tables 5.1-5.4 report the results obtained over the 30 instances described in Section 5.4 for each control area, Gonesse and Rouen, after 180 seconds of computation. In each table, column 1 indicates the instance tackled. Although 30 instances were solved for each control area, for sake of readability we only show the results for the 10 first instances and the average over all the instances in the last row. Columns 2 and 3 report the number of trains and the total number of routes in each instance, respectively. Columns 4-7 report the objective function value, the number of generated **OC** and **FC**, and the number of iteration of **BA-SBR** within the computational time available, respectively. Columns 8-9 report the objective function value and the optimality gap of **RECIFE-MILP** within the same time limit of 180 seconds, respectively.

When no initial solution is supplied, the results show that the performance of **BA-SBR** is very bad compared to **RECIFE-MILP**: **RECIFE-MILP** finds the optimal solution for all the Gonesse instances and at least a feasible solution for all the Rouen ones, while **BA-SBR** never reaches feasibility. A remarkable number of **FC** are added to the **UMP** throughout the runs, but they do not appear to be capable of driving the solution toward feasibility quickly enough. The

5 Standard Benders Reformulation for RECIFE-MILP

Table 5.1 Results of **BA-SBR** and **RECIFE-MILP** on 30 Gonesse instances without initialization (time limit 180 seconds).

Inst	# T	# R	BA-SBR				RECIFE-MILP	
			Obj	# OC	# FC	# it	Obj	Gap (%)
I1	16	124	x	0	377	377	30	0
I2	15	118	x	0	402	402	220	0
I3	14	106	x	0	443	443	0	0
I4	15	118	x	0	398	398	236	0
I5	16	124	x	0	377	377	0	0
I6	16	124	x	0	378	378	204	0
I7	16	124	x	0	384	384	1	0
I8	16	124	x	0	378	378	82	0
I9	16	124	x	0	377	377	347	0
I10	16	124	x	0	377	377	15	0
L...
Average	15	119	x	0	392	392	120	0

x = no feasible solution found.

Table 5.2 Results of **BA-SBR** and **RECIFE-MILP** on 30 Rouen instances without initialization (time limit 180 seconds).

Inst	# T	# R	BA-SBR				RECIFE-MILP	
			Obj	# OC	# FC	# it	Obj	Gap (%)
I1	12	856	x	0	27	27	1318	100
I2	12	856	x	0	29	29	771	100
I3	12	856	x	0	28	28	328	100
I4	11	664	x	0	37	37	23	100
I5	11	664	x	0	37	37	328	100
I6	10	661	x	0	37	37	148	100
I7	11	664	x	0	36	36	148	100
I8	12	856	x	0	28	28	328	100
I9	10	661	x	0	32	32	256	51
I10	12	856	x	0	27	27	100	100
L...
Average	11	808	x	0	30	30	419	92

x = no feasible solution found.

5.4 Computational experiments

Table 5.3 Results of **BA-SBR** and **RECIFE-MILP** on 30 Gonesse instances with initialization (time limit 180 seconds).

Inst	# T	# R	BA-SBR				RECIFE-MILP	
			Obj	# OC	# FC	# it	Obj	Gap (%)
I1	16	124	70	1	373	374	30	0
I2	15	118	257	1	394	395	220	0
I3	14	106	136	1	437	438	0	0
I4	15	118	261	1	395	396	236	0
I5	16	124	105	1	370	371	0	0
I6	16	124	234	1	376	377	204	0
I7	16	124	73	1	380	381	1	0
I8	16	124	159	1	372	373	82	0
I9	16	124	762	1	372	373	347	0
I10	16	124	27	1	372	373	15	0
I...
Average	15	119	188	1	348	349	120	0

Table 5.4 Results of **BA-SBR** and **RECIFE-MILP** on 30 Rouen instances with initialization (time limit 180 seconds).

Inst	# T	# R	BA-SBR				RECIFE-MILP	
			Obj	# OC	# FC	# it	Obj	Gap (%)
I1	12	856	1318	1	26	27	1318	100
I2	12	856	771	1	27	28	771	100
I3	12	856	328	1	26	27	328	100
I4	11	664	251	1	35	36	23	100
I5	11	664	328	1	34	35	328	100
I6	10	661	251	1	34	35	148	100
I7	11	664	251	1	35	36	148	100
I8	12	856	328	1	27	28	328	100
I9	10	661	454	1	30	31	256	51
I10	12	856	100	1	26	27	100	100
I...
Average	11	808	477	1	27	28	419	92

5 Standard Benders Reformulation for RECIFE-MILP

conclusion that can be drawn is that **BA-SBR** with no initialization is inappropriate for solving the **rtRTMP**.

With the initialization, the search process of **BA-SBR** is not actually ameliorated (see the results in Tables 5.3 and 5.4). Indeed, **BA-SBR** fails to improve the initial integer feasible solution provided. However, compared to the performance without the initialization, the results are slightly better as at least one feasible solution is found for each instance, i.e., the one corresponding to the one used for the initialization.

5.5 Conclusions

In this chapter, we presented the **SBR** of the **RECIFE-MILP** formulation. It consists in splitting the **MILP** formulation presented in Section 4.3 into the **UMP** and the **SP**. The **UMP** is the real-time train routing and scheduling problem. It contains the re-routing and rescheduling variables (binary variables) of the overall problem, and one dummy variable representing the contribution of continuous variables to the **UMP** objective function. The schedule here is intended as the order in which trains cross common tracks, and does not include time information. The **SP** contains the continuous variables which determine the **track-circuit** occupation and reservation times, and deduces the delay suffered by each train. The decomposed problem is solved with the **BA** in which we add progressively Benders optimality and feasibility cuts to the **UMP** obtained through the iterative solution of the **DSP**.

We performed some computational experiments on two case studies representing traffic in two control areas in France: the Pierrefitte–Gonesse junction and a line section around the Rouen–Rive–Droite station. The results show that the **SBR** is not appropriate for the **rtRTMP**. To improve the performance of the **BA-SBR** algorithm, we initialized the search with a feasible solution. In particular, we performed a first optimization-phase in which we solved the **RECIFE-MILP** formulation by fixing the train routing decisions to the default routes defined in the timetable (fixed routes). Then, we initialized the solution of the **SBR** with the so-obtained solution. However, even with the initialization, the results are not promising as the **BA** failed to improve the initial integer feasible solution within the computational time limit imposed (180 seconds).

Note that, although the results are clearly negative, **BD** has been proven to be good for a number of applications. For example, it was successfully used in several applications such as the engine scheduling problem (Florian et al., 1976), the aircraft routing problem (Richardson, 1976), the vehicle routing problem (Fisher and Jaicumeur, 1978), and the locomotive and car assignment problem (Cordeau et al., 1975). However, the **BD** is also known to be inappropriate for some applications like the network design problem (Wong, 1978). For the **rtRTMP**, the bad

performance of the **SBR** is also confirmed by [Lamorgese and Mannino \(2016\)](#) when tackling the train's rescheduling problem formulated with an **AG** model. To improve their algorithm performance, the authors replace the standard Benders feasibility and optimality cuts with strong cuts obtained by strengthening and lifting the standards ones. Several other methods have been proposed in the literature to obtain well performing **BD** ([Wong, 1978](#); [Magnanti and Wong, 1981](#); [Geoffrion, 2005](#); [Papadakos, 2008](#)). In the next chapters we propose and test some variants of the **SBR** and the **BA** to try to improve the **BA** performance.

Chapter 6

Reduced Benders reformulation for RECIFE-MILP

6.1 Introduction

In Chapter 5, we presented a **SBR** for the **RECIFE-MILP** formulation. This reformulation is a classical application of the **BD** approach. Specifically, we separated the original problem into two problems: the **UMP** and **SP**. We kept the binary variables related to the re-routing and rescheduling decisions in the **UMP** and we delegated the re-timing decisions in the **SP**.

Recall that in the **SBR**, the **SP** contains all constraints and continues variables related to the predefined routing and scheduling. These decisions are determined by the **UMP** and include the information on the routes assigned to the trains $(\bar{x}_{t,r})$ and the scheduling decisions $(\bar{y}_{t,t',fc})$. The **SBR** proved not to be suitable for solving the **rtRTMP**.

As a step forward, in this chapter we exploit the structure of the **RECIFE-MILP** formulation to propose a reduced reformulation which we call **RBR**. The idea behind this reformulation is the removal from the **SP** assessment of unnecessary variables and constraints. Indeed, when a train does not use a route, all variables and constraints related to the timing decisions along this route are actually useless, since they can be fixed *a priori* without the need of optimization. For example, the occupation start variables will be equal to zero. Similarly, when several routes are available for a pair of trains t and t' , typically not all **track-circuits** in $TC_t \cap TC_{t'}$ will be used by both trains given a specific set of route choices. For the unused track-circuits, a scheduling decision will be made in the **UMP**, but it will actually be useless in the practice: if only t , for example, uses a track-circuit, then defining a precedence relation there is not really necessary. In the **RBR**, we replace the complete **SP** of **SBR** with a reduced one, only containing useful variables and constraints given the route choices. The potential merit of this idea is linked to

the observation that some **UMP** solutions which are infeasible for the standard **SP** due to some inappropriately set unnecessary variables will actually be feasible for the reduced sub-problem. When this happens, an **OC** will be included in the **UMP** rather than a (typically weaker) **FC**. The possible drawback of the **RBR** is that the cuts are in principle weaker than in the **SBR**, since they exclude all the temporarily unnecessary binary variables. In this chapter, after the detailed description of the **RBR**, we assess whether its merit overcomes its drawback in practice.

The rest of this chapter is organized as follows. Section 6.2 describes the **RBR** for the **RECIFE-MILP** formulation. Section 6.3 presents the computational experiments, and Section 6.4 concludes the chapter.

6.2 Reduced Benders Reformulation: RBR

In this section, we present the **RBR** in which the information on the routes not used by trains in the **UMP** solution in input of the **SP** are neglected.

Indeed as mentioned in the introduction of this chapter, by studying the **rtRTMP** structure, we realize that many constraints in the **SP** are negligible if one already knows a route is not used by a train: For example Constraints (5.3) in the **SP** become unnecessary because they are trivially satisfied or their role is simply to set some variables equal to zero. By simply not generating these null variables we can actually avoid writing the constraints without actually changing the solution. Hence, many variables (as $\beta_{t,r,tc}$) and many constraints (as some Constraints (5.11) and (5.12)) in the **DSP** are also unnecessary. For the same reason, we can discard all constraints in the **SP** associated with the unused routes ($\bar{x}_{t,r} = 0$).

Moreover, we observe that, in the **SP**, the disjunctive Constraints (5.8) and (5.9) are imposed for all **track-circuits** which may possibly be used by two trains t and t' . However, if the routes chosen for the two trains imply that a possibly common track-circuit is used by either only one or none of them, the scheduling decision implied by the corresponding y -variable is actually negligible from a practical point of view. **RECIFE-MILP** will fix this variable, but this information will not be used for the practical implementation of the solution since, in reality, no precedence needs to be set unless both trains are passing in the same location. Hence, the value of these practically negligible y -variable can be disregarded in the **SP**, together with the corresponding constraints.

Recall that, for a pair of trains t, t' and shared track-circuit $\hat{t}c$, we define the disjunctive variable $y_{t,t',\hat{t}c}$ that takes value 1 if t utilizes $\hat{t}c$ before t' , 0 otherwise. Remark that, when only one train uses $\hat{t}c$, only one assignment of $y_{t,t',\hat{t}c}$ is feasible: if t is using it, then the variable must be equal to 0 since the **utilization** variables of t' must be set to 0; if t' is using it, then the variable must be set to 1. This implication is not explicit in the **UMP**: it can generate solutions

6 Reduced Benders reformulation for RECIFE-MILP

which result infeasible for the SP due to the negligible y -variables. By removing from the SP Constraints (5.8) and (5.9) which correspond to the track-circuits not actually used by each pair of trains, the generation of many Benders feasibility cuts can be avoided. The result of the constraint and variable removals, from the SP and hence from the DSP, is the RBR, whose detailed formulation is described in the rest of this section.

6.2.1 Reduced Sub-Problem (Red-SP)

Let, $\bar{r} \in R_t$ be the route used by train t and $\bar{r}' \in R_{t'}$ the route used by train t' , where $t, t' \in T$ in the UMP solution to be evaluated. Remark that the set of track-circuits available for train t (TC_t) can be reduced to $TC^{\bar{r}}$ (set of track-circuits composing route \bar{r}) and the set of track-circuits available for train t' ($TC_{t'}$) can also be reduced to $TC^{\bar{r}'}$. The Reduced Sub-Problem (Red-SP) can then be formulated as follows.

$$\min \sum_{t \in T} w_t D_t. \quad (6.1)$$

$$o_{t,\bar{r},tc} \geq \text{init}_t \quad \forall t \in T, tc \in TC^{\bar{r}} : p_{\bar{r},tc} = tc_0 \quad (6.2)$$

$$-o_{t,\bar{r},tc} + o_{t,\bar{r},p_{r,tc}} + l_{t,\bar{r},p_{\bar{r},tc}} = -rt_{\bar{r},ty,p_{\bar{r},tc}} \quad \forall t \in T, tc \in TC^{\bar{r}} : tc \neq tc_0 \quad (6.3)$$

$$D_t - o_{t,\bar{r},tc_\infty} \geq -\text{exit}_t \quad \forall t \in T \quad (6.4)$$

$$sU_{t,tc} - o_{t,\bar{r},ref_{\bar{r},tc}} \leq -\text{for}_{bs_{\bar{r},tc}} \quad \forall t \in T, tc \in TC^{\bar{r}} \quad (6.5)$$

$$\begin{aligned} eU_{t,tc} - o_{t,\bar{r},ref_{\bar{r},tc}} - \sum_{tc' \in TC(ref_{\bar{r},tc},tc,\bar{r})} l_{t,\bar{r},tc'} \geq \text{rel}_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \\ + \sum_{tc' \in TC(ref_{\bar{r},tc},tc,\bar{r})} rt_{\bar{r},ty,tc'} \quad \forall t \in T, tc \in TC^{\bar{r}} \end{aligned} \quad (6.6)$$

$$eU_{t,tc} - sU_{t',tc} \leq M - M\bar{y}_{t,t',\hat{f}c} \quad \forall t, t' \in T, t < t', tc, \hat{f}c \in TC^{\bar{r}} \cap TC^{\bar{r}'} : tc \in \hat{TC}_{t,t',\hat{f}c} \quad (6.7)$$

$$eU_{t',tc} - sU_{t,tc} \leq M\bar{y}_{t,t',\hat{t}c} \quad \forall t, t' \in T, t < t', tc, \hat{t}c \in TC^{\bar{r}} \cap TC^{\bar{r}'} : tc \in \hat{TC}_{t,t',\hat{t}c}. \quad (6.8)$$

Let $\alpha_{t,\bar{r},tc}$, $\lambda_{t,\bar{r},tc}$, θ_t , $\phi_{t,tc}$, $\omega_{t,tc}$, $\eta_{t,t',tc}$, $\psi_{t,t',tc}$, be the dual variables associated with Constraints (6.2)-(6.8) respectively. As such, $\alpha_{t,\bar{r},tc} \geq 0$, $\lambda_{t,\bar{r},tc} \in \mathbb{R}$, $\theta_t \geq 0$, $\phi_{t,tc} \leq 0$, $\omega_{t,tc} \geq 0$, $\eta_{t,t',tc} \leq 0$ and $\psi_{t,t',tc} \leq 0$. The **Reduced Dual of the Sub-Problem (Red-DSP)** is written as follows. It is the dual of the **Red-SP**.

6.2.2 Reduced Dual of the Sub-Problem (Red-DSP)

$$\begin{aligned} & \max \sum_{t \in T} \sum_{tc \in TC^{\bar{r}}} \text{init}_t \alpha_{t,\bar{r},tc} - rt_{\bar{r},ty,p_{\bar{r},tc}} \lambda_{t,\bar{r},tc} + \\ & + \sum_{t \in T} \sum_{tc \in TC^{\bar{r}}} (\text{rel}_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \sum_{t' \in TC(\text{ref}_{\bar{r},tc},tc,\bar{r})} rt_{\bar{r},ty,t'}) \omega_{t,tc} - \text{for}_{bs_{\bar{r},tc}} \phi_{t,tc} - \\ & - \sum_{t \in T} \text{exit}_t \theta_t + \sum_{t,t' \in T} \sum_{\substack{\hat{t}c, tc \in TC^{\bar{r}} \cap TC^{\bar{r}'} \\ tc \in \hat{TC}_{t,t',\hat{t}c}}} M[(1 - \bar{y}_{t,t',\hat{t}c}) \eta_{t,t',tc} + \bar{y}_{t,t',\hat{t}c} \psi_{t,t',tc}]. \end{aligned} \quad (6.9)$$

$$\alpha_{t,\bar{r},tc} + \lambda_{t,\bar{r},tc} - \phi_{t,tc} - \omega_{t,tc} \leq 0 \quad \forall t \in T, tc \in TC^{\bar{r}} : p_{\bar{r},tc} = tc_0 \quad (6.10)$$

$$\lambda_{t,\bar{r},tc} - \lambda_{t,\bar{r},p_{\bar{r},tc}} - \phi_{t,tc} - \omega_{t,tc} \leq 0 \quad \forall t \in T, tc \in TC^{\bar{r}} \quad (6.11)$$

$$\lambda_{t,\bar{r},tc} - \omega_{t,tc} \leq 0 \quad \forall t \in T, tc \in TC^{\bar{r}} \quad (6.12)$$

$$\alpha_{t,\bar{r},tc_\infty} - \lambda_{t,\bar{r},tc_\infty} - \theta_t \leq 0 \quad \forall t \in T \quad (6.13)$$

$$\theta_t \leq 1 \quad \forall t \in T \quad (6.14)$$

$$\phi_{t,tc} - \eta_{t,t',tc} - \psi_{t,t',tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}'} \quad (6.15)$$

$$\omega_{t,tc} + \eta_{t,t',tc} + \psi_{t,t',tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}'}. \quad (6.16)$$

6 Reduced Benders reformulation for RECIFE-MILP

Finally, we need also to change the formulation of the **RMP** to take into account the changes in the **Red-DSP**. In particular, Constraints (5.19) and (5.20) become Constraints (6.17) and (6.18). Here, the cuts include only the variables which we considered in the **Red-DSP**.

$$\begin{aligned}
& z - \left(\sum_{t \in T} \sum_{tc \in TC^{\bar{r}}} x_{t,\bar{r}} (\text{init}_t \alpha_{t,\bar{r},tc} - r_{\bar{r},ty,p_{\bar{r},tc}} \lambda_{t,\bar{r},tc}) + \right. \\
& + \sum_{t \in T} \sum_{tc \in TC^{\bar{r}}} x_{t,\bar{r}} [(rel_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \sum_{t' \in T} r_{\bar{r},ty,tc'}) \omega_{t,tc} - for_{bs_{\bar{r},tc}} \phi_{t,tc}] - \\
& \left. - \sum_{t \in T} \text{exit}_t \theta_t + \sum_{t,t' \in T} \sum_{\substack{\hat{t}c,tc \in TC^{\bar{r}} \cap TC^{t'} \\ tc \in \hat{TC}_{t,t',\hat{t}c}}} M[(1 - y_{t,t',\hat{t}c}) \eta_{t,t',tc} + y_{t,t',\hat{t}c} \psi_{t,t',tc}] \right) \geq 0 \\
& ((\alpha, \lambda, \theta, \phi, \omega, \eta, \psi) \in P_S) \quad (6.17)
\end{aligned}$$

$$\begin{aligned}
& \sum_{t \in T} \sum_{tc \in TC^{\bar{r}}} x_{t,\bar{r}} (\text{init}_t \alpha_{t,\bar{r},tc} - r_{\bar{r},ty,p_{\bar{r},tc}} \lambda_{t,\bar{r},tc}) \\
& + \sum_{t \in T} \sum_{tc \in TC^{\bar{r}}} x_{t,\bar{r}} [(rel_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \sum_{t' \in T} r_{\bar{r},ty,tc'}) \omega_{t,tc} - for_{bs_{\bar{r},tc}} \phi_{t,tc}] - \\
& \left. - \sum_{t \in T} \text{exit}_t \theta_t + \sum_{t,t' \in T} \sum_{\substack{\hat{t}c,tc \in TC^{\bar{r}} \cap TC^{t'} \\ tc \in \hat{TC}_{t,t',\hat{t}c}}} M[(1 - y_{t,t',\hat{t}c}) \eta_{t,t',tc} + y_{t,t',\hat{t}c} \psi_{t,t',tc}] \leq 0 \\
& ((\alpha, \lambda, \theta, \phi, \omega, \eta, \psi) \in R_S) \quad (6.18)
\end{aligned}$$

6.3 Computational experiments

In this section, we assess the quality of the **RBR**. To do so, we solve the reformulations with the **BA** presented in Section 5.3. The experiments are performed on the two French control areas described in Section 5.4 and an identical experimental setup is used.

For each control area, we use the 30 instances utilized in Section 5.4 to assess the **RBR** quality with the **BA** (from now on **BA-RBR**).

As done in Section 5.3, we first assess **BA-RBR** without the initialization for sake of completeness. Then we add the initialization and we repeat the assessment.

Tables 6.1-6.4 report the results with the same format used in Tables 5.1-5.4. Specifically, Column 1 indicates the instance tackled. Columns 2 and 3 report the number of trains and the total number of routes in each instance, respectively. Columns 4-7 report the objective function value, the number of generated **OC** and **FC**, and the number of iteration of

6.3 Computational experiments

Table 6.1 Results of **BA-RBR** and **RECIFE-MILP** on 30 Gonesse instances without initialization (time limit 180 seconds).

Inst	# T	# R	BA-RBR				RECIFE-MILP	
			Obj	# OC	# FC	# it	Obj	Gap (%)
I1	16	124	14551	48	108	156	30	0
I2	15	118	14946	56	114	170	220	0
I3	14	106	9693	57	136	193	0	0
I4	15	118	10957	32	144	176	236	0
I5	16	124	14136	48	107	155	0	0
I6	16	124	16487	51	97	148	204	0
I7	16	124	29564	48	103	151	1	0
I8	16	124	17182	45	104	149	82	0
I9	16	124	16281	44	100	144	347	0
I10	16	124	17028	47	106	153	15	0
I...
Average	15	119	16721	49	113	162	122	0

BA-RBR, respectively. Columns 8-9 report the objective function value and the optimality gap of **RECIFE-MILP** within the same time limit of 180 seconds, respectively.

Without the initialization, although the results achieved by **BA-RBR** are still worse from the ones of **RECIFE-MILP** (see Tables 6.1 and 6.2), the performance of **BA-RBR** is better than the one of **BA-SBR** (see Section 5.4): the **BA** finds several integer feasible solutions when used to solve the **RBR**.

With the initialization, as we observed for the **SBR** in Section 5.4, (see the results in Tables 6.3 and 6.4), **BA-RBR** fails to improve the initial integer feasible solution provided. However, compared to the performance without the initialization, we consider the search process more appropriate since the algorithm finds several integer feasible solutions (see the number of **OC** generated) although the first incumbent is not improved.

6 Reduced Benders reformulation for RECIFE-MILP

Table 6.2 Results of [BA-RBR](#) and [RECIFE-MILP](#) on 30 Rouen instances without initialization (time limit 180 seconds).

Inst	# T	# R	BA-RBR				RECIFE-MILP	
			Obj	# OC	# FC	# it	Obj	Gap (%)
I1	12	856	21865	224	62	286	1318	100
I2	12	856	22887	222	64	286	771	100
I3	12	856	23648	223	63	286	328	100
I4	11	664	15233	225	185	410	23	100
I5	11	664	10550	228	185	413	328	100
I6	10	661	16798	216	219	435	148	100
I7	11	664	16218	221	188	409	148	100
I8	12	856	23648	223	63	286	328	100
I9	10	661	2696	273	160	433	256	51
I10	12	856	23547	223	62	285	100	100
I...
Average	11	808	20466	237	85	322	419	92

Table 6.3 Results of [BA-RBR](#) and [RECIFE-MILP](#) on 30 Gonesse instances with initialization (time limit 180 seconds).

Inst	# T	# R	BA-RBR				RECIFE-MILP	
			Obj	# OC	# FC	# it	Obj	Gap (%)
I1	16	124	70	48	90	138	30	0
I2	15	118	257	59	112	171	220	0
I3	14	106	136	66	118	184	0	0
I4	15	118	261	23	151	174	236	0
I5	16	124	105	41	109	150	0	0
I6	16	124	234	50	88	138	204	0
I7	16	124	73	51	111	162	1	0
I8	16	124	159	52	100	152	82	0
I9	16	124	762	48	97	145	347	0
I10	16	124	27	53	83	136	15	0
I...
Average	15	119	188	44	98	142	120	0

Table 6.4 Results of **BA-RBR** and **RECIFE-MILP** on 30 Rouen instances with initialization (time limit 180 seconds).

Inst	# T	# R	BA-RBR				RECIFE-MILP	
			Obj	# OC	# FC	# it	Obj	Gap (%)
I1	12	856	1318	216	71	287	1318	100
I2	12	856	771	250	35	285	771	100
I3	12	856	328	223	63	286	328	100
I4	11	664	251	216	191	407	23	100
I5	11	664	328	204	206	410	328	100
I6	10	661	251	227	217	444	148	100
I7	11	664	251	228	185	413	148	100
I8	12	856	328	223	63	286	328	100
I9	10	661	454	268	163	431	256	51
I10	12	856	100	208	80	288	100	100
I...
Average	11	808	477	231	78	309	419	92

6.4 Conclusions

In this chapter, we proposed the **RBR** for the **RECIFE-MILP** formulation. Unlike the classical decomposition approach applied in Chapter 5 (i.e., the **SBR**), the **RBR** exploits the nature of the **RECIFE-MILP** formulation to propose a more suitable reformulation.

To assess this reformulation, we performed computational experiments on two case studies representing traffic in two French control areas. The results show that the **RBR** is better than the **SBR**. In our understanding, the main reason behind the difference in performance is the fact that with the **RBR** many feasible solutions are found to the overall problem, while with the **SBR** the **FCs** generated in the **DSP** do not drive toward feasibility quickly enough. Indeed, feasibility here disregards the value set for a number of actually unnecessary binary variables, once route choices are made. In the rest of this thesis, we will use **BA-RBR** for the experiments. Despite the improvement achieved, the performance of **BA-RBR** is still significantly worse than the one of **RECIFE-MILP**. In the next chapter we propose a further approach for trying to improve this performance.

Chapter 7

Inequalities for the unrestricted master problem

7.1 Introduction

In Chapter 5, we presented the **SBR** for **RECIFE-MILP**. This reformulation is a classical application of **BD** to the **RECIFE-MILP** formulation. Indeed, the **SBR** presents a performance issue when it is used to tackle an instance of the **rtRTMP**. As a result, often the computational time limit elapses without finding a feasible solution. To overcome the **SBR** inefficiency, in Chapter 6, we proposed the **RBR**. This reformulation exploits the nature of the **rtRTMP**. Specifically, we discard in the **SP (DSP)** some constraints and variables which are unnecessary given a set of re-routing decisions made in the **UMP**.

Despite the improvement achieved with the **RBR**, the performance remains poor with respect to the **RECIFE-MILP** algorithm presented in Chapter 4. To try to improve it, we propose, in this chapter, some inequalities to be added to the **UMP**. Indeed, the **UMP**, which determines the value of the binary variables (i.e., the re-routing and rescheduling decisions), has no constraints to force their consistency. In other words, there is no link between the re-routing decisions and the rescheduling ones in the **UMP**, or between different rescheduling decisions. The inequalities which we present impose this consistency in several circumstances.

This chapter is organized as follows. In Section 7.2, we propose some inequalities for the **UMP**. These inequalities are grouped into six families. To select which ones are suitable to be added to the **UMP**, an algorithm configuration phase is presented in Section 7.3. Then, to assess the impact of this addition, computational experiments are reported in Section 7.4. The conclusions of the chapter are reported in Section 7.5.

7.2 Inequalities for the unrestricted master problem

We propose here some inequalities to be added to the **UMP** presented in Section 5.2.4 to avoid the exploration of many **Red-DSP** unbounded solutions during the search. These inequalities would be redundant in **RECIFE-MILP** because they are implicitly imposed through constraints including continuous variables. However, they may be useful in the **UMP** because they establish a link between the re-routing decisions and the rescheduling ones, and among different rescheduling decisions. In particular, they force the consistency of these decisions.

The inequalities proposed in this thesis avoid some inconsistent re-routing and rescheduling decisions between two or three trains.

In the following, we explain the proposed inequalities using the infrastructure of the Gonesse control area, introduced in Section 5.4. These inequalities are found by studying the causes of the **Red-DSP** unboundedness with respect to some **UMP** solutions. We recall that in the **BAs**, we determine in the **UMP** the re-routing and rescheduling decisions then we solve the **Red-DSP** given these decisions. If the **Red-DSP** is bounded, it returns a Benders **OC** for the **UMP**; if unbounded, normally the **Red-DSP** returns a Benders **FC**. Hence, by studying the unboundedness cases and formulating the inequalities to cut the concerned **UMP** solutions, we limit the number of **FC** added.

In the explanation of these inequalities, we will use the concepts of:

Common track-circuit: a **track-circuit** is common to a set of trains if it belongs to at least one route available for each of them.

Shared track-circuit: a track-circuit is shared by a set of trains if it belongs to the specific route chosen by each of them.

Reservation zone: a reservation zone is a set of track-circuits reserved concurrently by a train due to the route chosen and the **interlocking** system.

In the following, we present the inequalities proposed and their explanation. They are grouped into six families ($F1$, $F2$, $F3$, $F4$, $F5$ and $F6$).

7.2.1 Family $F1$

The following Inequalities (7.1)-(7.4) compose family $F1$:

$$x_{t,r} + x_{t',r'} + y_{t,t',tc_1} - y_{t,t',tc_2} \leq 2 \quad \forall t, t' \in T, t < t', r \in R_t, r' \in R_{t'}, tc_1, tc_2 \in TC^r \cap TC^{r'},$$

$$(s_{r,tc_1} = tc_2, s_{r',tc_1} = tc_2) \vee (s_{r,tc_1} = tc_2, p_{r',tc_1} = tc_2).$$

(7.1)

7 Inequalities for the unrestricted master problem

$$\begin{aligned}
 -x_{t,r} - x_{t',r'} + y_{t,t',tc_1} - y_{t,t',tc_2} \geq -2 \quad \forall t,t' \in T, t < t', r \in R_t, r' \in R_{t'}, tc_1, tc_2 \in TC^r \cap TC^{r'}, \\
 (s_{r,tc_1} = tc_2, s_{r',tc_1} = tc_2) \vee (s_{r,tc_1} = tc_2, p_{r',tc_1} = tc_2).
 \end{aligned} \tag{7.2}$$

Inequalities (7.1) and (7.2) state that if tc_1 and tc_2 are adjacent shared track-circuits in the routes used by trains t and t' then the value of the rescheduling variables in tc_1 and tc_2 , y_{t,t',tc_1} and y_{t,t',tc_2} , must be equal. Figure 7.1 shows an example of network topology where these inequalities operate.

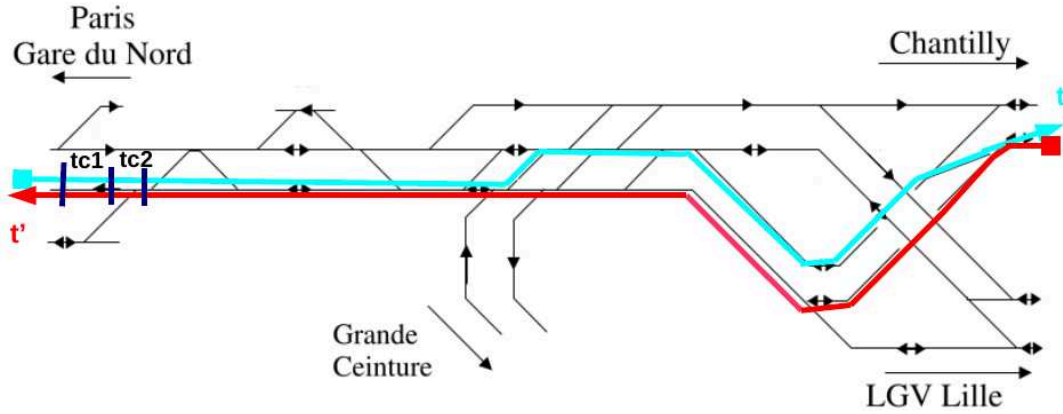


Figure 7.1 Situation in which Inequalities (7.1) and (7.2) apply.

$$\begin{aligned}
 x_{t,r} + x_{t',r'} + y_{t,t',tc_1} - y_{t,t',tc_2} \leq 2 \\
 \forall t,t' \in T, t < t', r \in R_t, r' \in R_{t'}, tc_1, tc_2 \in TC^r \cap TC^{r'}, \\
 (tc_1 \in TC(ref_{r,tc_2}, tc_2, r)) \vee (tc_2 \in TC(ref_{r,tc_1}, tc_1, r)), \\
 (tc_1 \in TC(ref_{r',tc_2}, tc_2, r')) \vee (tc_2 \in TC(ref_{r',tc_1}, tc_1, r')).
 \end{aligned} \tag{7.3}$$

$$\begin{aligned}
 -x_{t,r} - x_{t',r'} + y_{t,t',tc_1} - y_{t,t',tc_2} \geq -2 \\
 \forall t,t' \in T, t < t', r \in R_t, r' \in R_{t'}, tc_1, tc_2 \in TC^r \cap TC^{r'}, \\
 (tc_1 \in TC(ref_{r,tc_2}, tc_2, r)) \vee (tc_2 \in TC(ref_{r,tc_1}, tc_1, r)), \\
 (tc_1 \in TC(ref_{r',tc_2}, tc_2, r')) \vee (tc_2 \in TC(ref_{r',tc_1}, tc_1, r')).
 \end{aligned} \tag{7.4}$$

7.2 Inequalities for the unrestricted master problem

Inequalities (7.3) and (7.4) extend the requirement imposed by (7.1) and (7.2) to reservation zones. In words, (7.3) and (7.4) are formulated as Inequalities (7.1) and (7.2) but they are imposed if the two **track-circuits** tc_1 and tc_2 belong to a same reservation zone. Note that Inequalities (7.3) and (7.4) are not imposed if tc_1 and tc_2 are adjacent track-circuits, to avoid the redundancy with (7.1) and (7.2).

7.2.2 Family $F2$

Inequalities (7.5) constitute family $F2$:

$$x_{t,r} + x_{t',r'} + y_{t,t',tc_2} - y_{t,t',tc_1} \leq 2 \quad \forall t, t' \in T, t < t', r \in R_t, r' \in R_{t'}, \\ tc_1, tc_2 \in TC^r \cap TC^{r'}, \quad tc_2 \in TC(tc_1, tc_\infty, r), \quad tc_2 \in TC(tc_0, tc_1, r'). \quad (7.5)$$

Inequalities (7.5) impose that the value of the rescheduling variable y_{t,t',tc_2} must be smaller than or equal to the value of the rescheduling variable y_{t,t',tc_1} if the following conditions are respected:

- **track-circuits** tc_1 and tc_2 are shared by trains t and t' ,
- track-circuit tc_2 follows tc_1 in the route of the train t ,
- track-circuit tc_2 precedes tc_1 in the route of t' .

Figure 7.2 exemplifies a case in which these conditions are met. Here we can see that either the precedence relation between the two trains on tc_1 and tc_2 is the same, or t passes first through tc_1 , t' passes first through tc_2 , and they cross in the portion of track which is not common to the two routes. It is not possible that t passes first on tc_2 and second on tc_1 , and this is cut by Inequalities (7.5).

7.2.3 Family $F3$

Family $F3$ includes Inequalities (7.6)–(7.8):

$$y_{t,t',tc} - \sum_{\substack{r \in R_t: \\ tc \in TC^r}} x_{t,r} - \sum_{\substack{r \in R_{t'}: \\ tc \in TC^r}} x_{t',r} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC_t \cap TC_{t'}. \quad (7.6)$$

Inequalities (7.6) force the value of the rescheduling variables $y_{t,t',tc}$ to be 0 unless at least one of trains t and t' uses tc . These inequalities eliminate the symmetry in the branch-and-bound

7 Inequalities for the unrestricted master problem

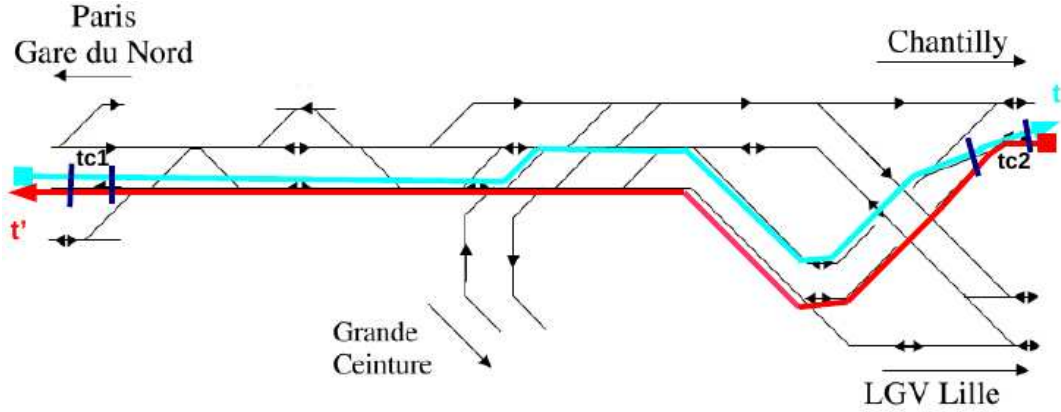


Figure 7.2 Situation in which Inequalities (7.5) apply.

tree. In the original model, both values 0 and 1 for this variable are feasible if the [track-circuit](#) is not used by either train, and this has no impact on the objective function.

$$y_{t,t',tc} + \sum_{\substack{r \in R_t: \\ tc \in TC^r}} x_{t,r} - \sum_{\substack{r \in R_{t'}: \\ tc \in TC^r}} x_{t',r} \geq 0 \quad \forall t, t' \in T, t < t', tc \in TC_t \cap TC_{t'}. \quad (7.7)$$

$$y_{t,t',tc} + \sum_{\substack{r \in R_t: \\ tc \in TC^r}} x_{t,r} - \sum_{\substack{r \in R_{t'}: \\ tc \in TC^r}} x_{t',r} \leq 1 \quad \forall t, t' \in T, t < t', tc \in TC_t \cap TC_{t'}. \quad (7.8)$$

Inequalities (7.7) and (7.8) state that if both t and t' may use the common track-circuit tc but only t' chooses a route which actually does, then the rescheduling variables $y_{t,t',tc}$ must be set to 1; on the contrary if only t chooses a route including tc , then $y_{t,t',tc}$ must be set to 0.

7.2.4 Family F4

Family F4 contains the following Inequalities (7.9)–(7.12):

$$y_{t,t',tc} + y_{t',t'',tc} - y_{t,t'',tc} \leq 1 \quad \forall t, t', t'' \in T, t < t' < t'', tc \in TC_t \cap TC_{t'} \cap TC_{t''}. \quad (7.9)$$

$$y_{t,t',tc} + y_{t',t'',tc} - y_{t,t'',tc} \geq 0 \quad \forall t, t', t'' \in T, t < t' < t'', tc \in TC_t \cap TC_{t'} \cap TC_{t''}. \quad (7.10)$$

7.2 Inequalities for the unrestricted master problem

Inequalities (7.9) and (7.10) impose the transitivity between the triplets of y -variables concerning each common **track-circuit** for three trains, t , t' and t'' . Specifically, Inequalities (7.9) impose that if train t precedes t' ($t \prec t'$) and t' precedes t'' ($t' \prec t''$) then t precedes t'' ($t \prec t''$). Inequalities (7.10) impose the opposite relation.

$$\begin{aligned}
 & y_{t,t',tc_1} + y_{t',t'',tc_2} - y_{t,t'',tc_3} + \sum_{\substack{r \in R_t: \\ tc_1, tc_3 \in TC^r}} x_{t,r} + \sum_{\substack{r \in R_{t'}: \\ tc_1, tc_2 \in TC^r}} x_{t',r} + \sum_{\substack{r \in R_{t''}: \\ tc_2, tc_3 \in TC^r}} x_{t'',r} \leq 4 \\
 & \forall t, t', t'' \in T, t < t' < t'', tc_1 \in TC_t \cap TC_{t'}, tc_2 \in TC_{t'} \cap TC_{t''}, tc_3 \in TC_t \cap TC_{t''}, \\
 & \exists r \in R_t : tc_1, tc_3 \in TC^r, (tc_1 \in TC(ref_{r,tc_3}, tc_3, r)) \vee (tc_3 \in TC(ref_{r,tc_1}, tc_1, r)), \\
 & \exists r \in R_{t'} : tc_1, tc_2 \in TC^r, (tc_1 \in TC(ref_{r,tc_2}, tc_2, r)) \vee (tc_2 \in TC(ref_{r,tc_1}, tc_1, r)), \\
 & \exists r \in R_{t''} : tc_2, tc_3 \in TC^r, (tc_2 \in TC(ref_{r,tc_3}, tc_3, r)) \vee (tc_3 \in TC(ref_{r,tc_2}, tc_2, r)). \quad (7.11)
 \end{aligned}$$

$$\begin{aligned}
 & y_{t,t',tc_1} + y_{t',t'',tc_2} - y_{t,t'',tc_3} - \sum_{\substack{r \in R_t: \\ tc_1, tc_3 \in TC^r}} x_{t,r} - \sum_{\substack{r \in R_{t'}: \\ tc_1, tc_2 \in TC^r}} x_{t',r} - \sum_{\substack{r \in R_{t''}: \\ tc_2, tc_3 \in TC^r}} x_{t'',r} \geq -3 \\
 & \forall t, t', t'' \in T, t < t' < t'', tc_1 \in TC_t \cap TC_{t'}, tc_2 \in TC_{t'} \cap TC_{t''}, tc_3 \in TC_t \cap TC_{t''}, \\
 & \exists r \in R_t : tc_1, tc_3 \in TC^r, (tc_1 \in TC(ref_{r,tc_3}, tc_3, r)) \vee (tc_3 \in TC(ref_{r,tc_1}, tc_1, r)), \\
 & \exists r \in R_{t'} : tc_1, tc_2 \in TC^r, (tc_1 \in TC(ref_{r,tc_2}, tc_2, r)) \vee (tc_2 \in TC(ref_{r,tc_1}, tc_1, r)), \\
 & \exists r \in R_{t''} : tc_2, tc_3 \in TC^r, (tc_2 \in TC(ref_{r,tc_3}, tc_3, r)) \vee (tc_3 \in TC(ref_{r,tc_2}, tc_2, r)). \quad (7.12)
 \end{aligned}$$

Inequalities (7.11) and (7.12) extend the transitivity requirements of (7.9) and (7.10) to reservation zones. Let tc_1 be the shared **track-circuit** between train t , t' and tc_2 the shared track-circuit between train t , t'' . Moreover, let tc_3 be the shared track-circuit between t' , t'' . Inequalities (7.11) impose that if train t precedes t'' ($t \prec t''$) in tc_2 and t' precedes t ($t' \prec t$) in tc_1 then t' precedes t'' ($t' \prec t''$) in tc_3 . Inequalities (7.12) impose the opposite relation. In Figure 7.3 we show an example in which tc_2 and tc_1 belong to the same reservation zone for train t (they are both in block section $s3 - s4$), tc_3 and tc_1 belong to the same reservation zone for t' (they are both in block section $s2 - s4$), and tc_2 and tc_3 belong to the same reservation zone for t'' (they are both in block section $s1 - s2$). In this situation, if train t passes before t'' in tc_2 and t' before t in tc_1 then t'' cannot precede t' in tc_3 . Indeed this violates Inequalities (7.11) and (7.12).

7 Inequalities for the unrestricted master problem

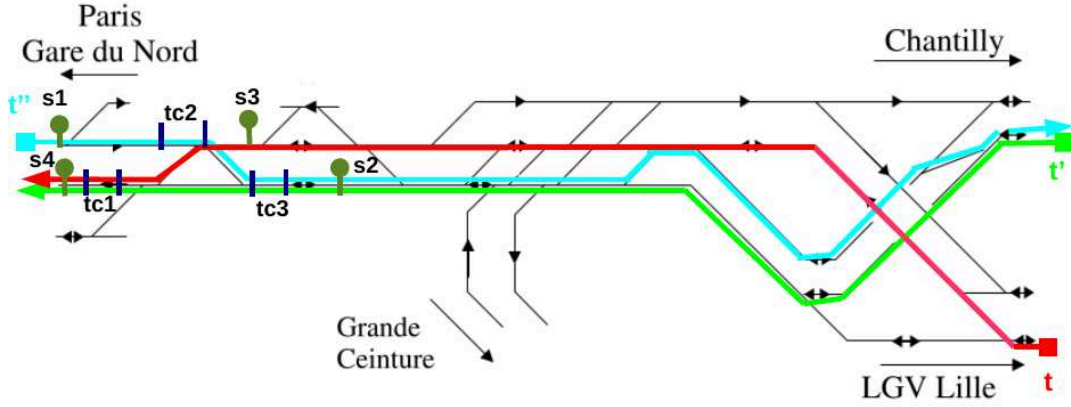


Figure 7.3 Situation in which Inequalities (7.11) and (7.12) apply.

7.2.5 Family $F5$

Family $F5$ includes Inequalities (7.13) - (7.17):

$$\begin{aligned}
 & y_{t,t'',tc_2} - y_{t',t'',tc_3} - y_{t,t',tc_1} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 3 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^r \cap TC^{r'} \cap TC^{r''}, tc_2 \in TC^r \cap TC^{r''}, tc_3 \in TC^{r'} \cap TC^{r''}, \\
 & tc_1 \in TC(tc_0, tc_2, r), tc_3 \in TC(tc_0, tc_1, r'), \\
 & tc_1 \in TC(tc_0, tc_2, r''), tc_3 \in TC(tc_1, tc_\infty, r''), \\
 & (tc_2 \in TC(ref_{r'',tc_3}, tc_3, r'')) \vee (tc_3 \in TC(ref_{r'',tc_2}, tc_2, r'')). \tag{7.13}
 \end{aligned}$$

Inequalities (7.13) consider three trains t , t' and t'' using three **track-circuits** tc_1 , tc_2 and tc_3 . They state that if t'' passes before t' on tc_3 ($y_{t',t'',tc_3} = 0$) and t' passes before t on tc_1 ($y_{t,t',tc_1} = 0$) then t' must pass before t on tc_2 ($y_{t,t',tc_2} = 0$). These inequalities are imposed under the following conditions:

- track-circuit tc_1 is shared by trains t , t' and t'' ,
- track-circuit tc_2 is shared by t and t'' which travel in the same direction,
- track-circuit tc_3 is shared by t' and t'' which travel in opposite directions,
- tc_1 and tc_3 precede tc_2 in the route of t'' and tc_2 and tc_3 are in the same reservation zone,
- tc_1 precedes tc_2 and tc_3 precedes tc_1 in the route of t and t' , respectively.

An example in which Inequalities (7.13) apply is shown in Figure 7.4.

7.2 Inequalities for the unrestricted master problem

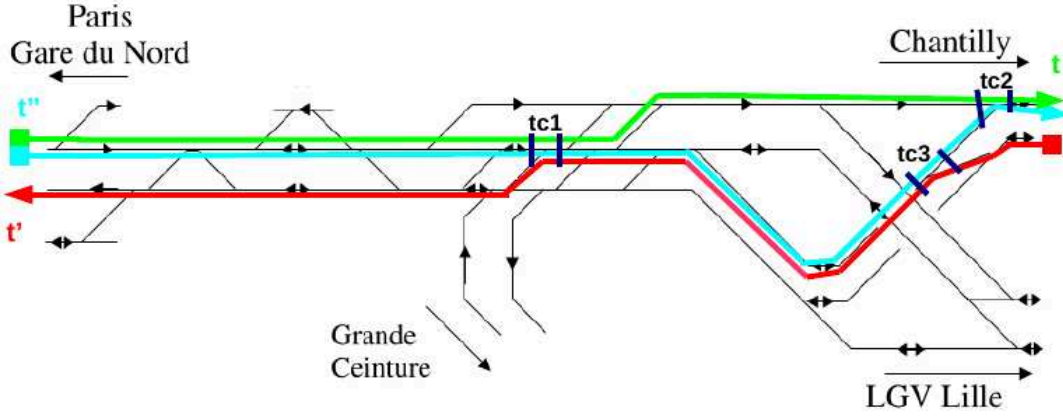


Figure 7.4 Situation in which Inequalities (7.13) apply.

Recall that we define y -variables only for pairs of trains t, t' such that the index of t is smaller than the one of t' . Hence, if the trains traveling in the same direction on tc_2 change, for example rather than being t and t'' as in Figure 7.4 they are t and t' , or the trains having tc_3 in their route change, then the conditions to cut off the infeasible assignment change as well. The five sets of Inequalities (7.14) - (7.18) similar to (7.13) must be imposed. When t and t' traverse tc_2 traveling in the same direction and t' has tc_3 in its route (Figure 7.5), Inequalities (7.14) impose the coherence of the precedences. When the two trains are t' and t , and t has tc_3 in its route (Figure 7.6), this is imposed by Inequalities (7.15). In the case where t' and t'' traverse tc_2 traveling in the same direction and t' has tc_3 in its route (Figure 7.7), Inequalities (7.16) impose the coherence of the precedences. For the case where t'' and t' traverse tc_2 traveling in the same direction and t'' has tc_3 in its route (Figure 7.8), Inequalities (7.17) play. Finally, if t'' and t traverse tc_2 traveling in the same direction and t has tc_3 in its route (Figure 7.9) then we set Inequalities (7.18).

$$\begin{aligned}
 & y_{t,t',tc_2} + y_{t',t'',tc_3} - y_{t,t'',tc_1} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 4 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^r \cap TC^{r'} \cap TC^{r''}, tc_2 \in TC^r \cap TC^{r'}, tc_3 \in TC^{r'} \cap TC^{r''}, \\
 & tc_1 \in TC(tc_0, tc_2, r), tc_3 \in TC(tc_0, tc_1, r''), \\
 & tc_1 \in TC(tc_0, tc_2, r'), tc_3 \in TC(tc_1, tc_\infty, r'),, \\
 & (tc_2 \in TC(ref_{r',tc_3}, tc_3, r')) \vee (tc_3 \in TC(ref_{r',tc_2}, tc_2, r')). \tag{7.14}
 \end{aligned}$$

7 Inequalities for the unrestricted master problem

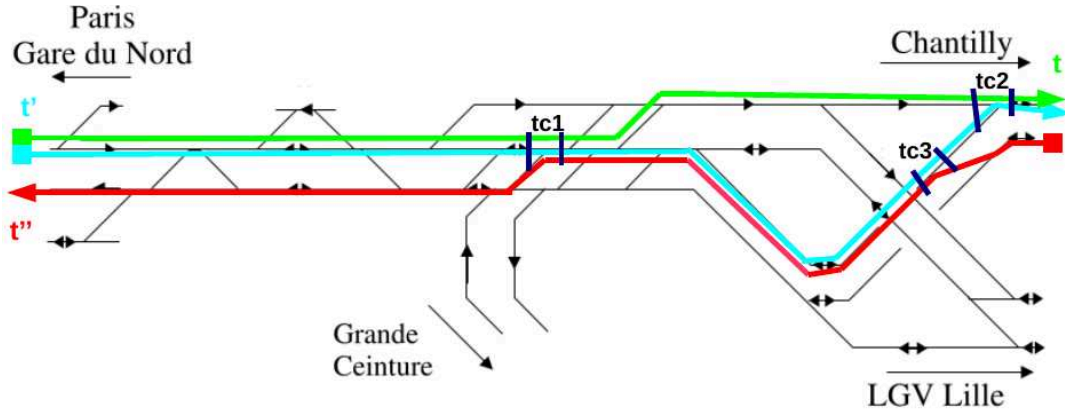


Figure 7.5 Situation in which Inequalities (7.14) apply.

$$\begin{aligned}
 & y_{t,t'',tc_3} - y_{t,t',tc_2} - y_{t',t'',tc_1} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 3 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^r \cap TC^{r'} \cap TC^{r''}, tc_2 \in TC^r \cap TC^{r'}, tc_3 \in TC^r \cap TC^{r''}, \\
 & tc_1 \in TC(tc_0, tc_2, r), tc_3 \in TC(tc_1, tc_\infty, r), \\
 & tc_1 \in TC(tc_0, tc_2, r'), tc_3 \in TC(tc_0, tc_1, r''), \\
 & (tc_2 \in TC(ref_{r,tc_3}, tc_3, r)) \vee (tc_3 \in TC(ref_{r,tc_2}, tc_2, r)). \tag{7.15}
 \end{aligned}$$

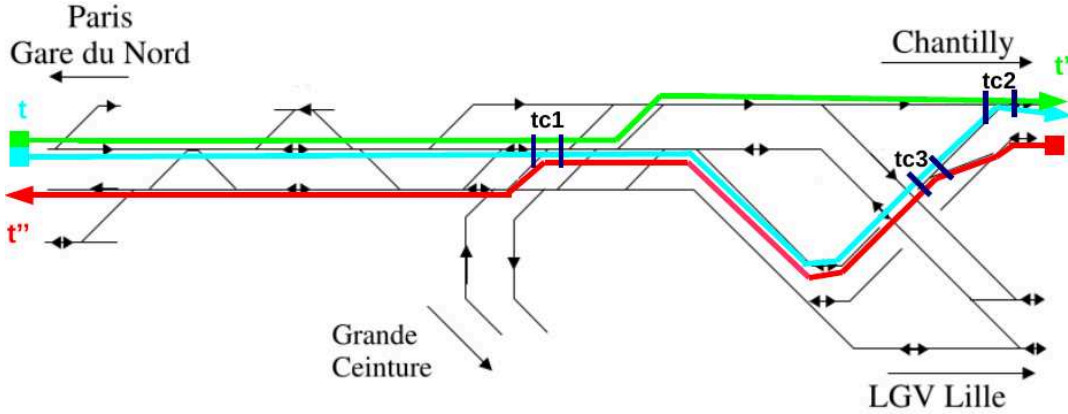


Figure 7.6 Situation in which Inequalities (7.15) apply.

7.2 Inequalities for the unrestricted master problem

$$\begin{aligned}
 & y_{t,t'',tc_1} - y_{t,t',tc_3} - y_{t',t'',tc_2} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 3 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^r \cap TC^{r'} \cap TC^{r''}, tc_2 \in TC^{r'} \cap TC^{r''}, tc_3 \in TC^r \cap TC^{r'}, \\
 & tc_1 \in TC(tc_0, tc_2, r'), tc_3 \in TC(tc_1, tc_\infty, r'), \\
 & tc_3 \in TC(tc_0, tc_1, r), tc_1 \in TC(tc_0, tc_2, r''), \\
 & (tc_2 \in TC(ref_{r',tc_3}, tc_3, r')) \vee (tc_3 \in TC(ref_{r',tc_2}, tc_2, r')). \tag{7.16}
 \end{aligned}$$

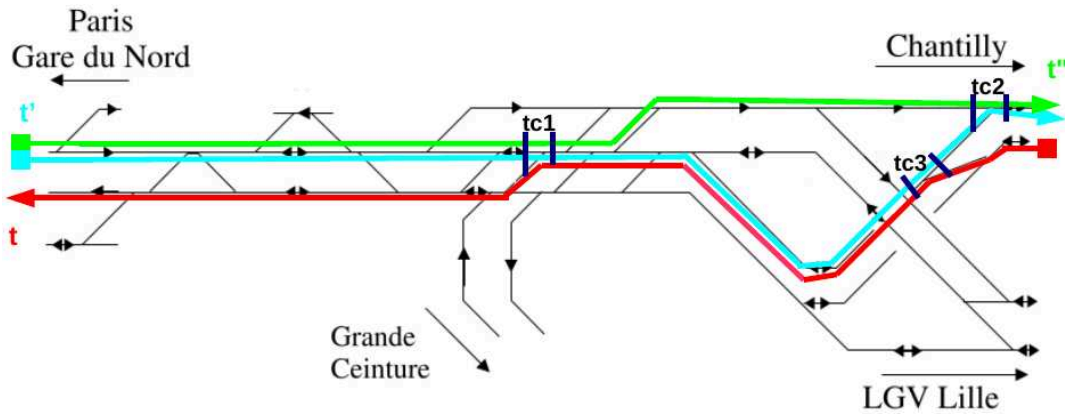


Figure 7.7 Situation in which Inequalities (7.16) apply.

$$\begin{aligned}
 & y_{t,t',tc_1} - y_{t,t'',tc_3} - y_{t',t'',tc_2} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 3 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^r \cap TC^{r'} \cap TC^{r''}, tc_2 \in TC^{r'} \cap TC^{r''}, tc_3 \in TC^r \cap TC^{r''}, \\
 & tc_1 \in TC(tc_0, tc_2, r''), tc_3 \in TC(tc_1, tc_\infty, r''), \\
 & tc_3 \in TC(tc_0, tc_1, r) tc_1 \in TC(tc_0, tc_2, r'), \\
 & (tc_2 \in TC(ref_{r'',tc_3}, tc_3, r'')) \vee (tc_3 \in TC(ref_{r'',tc_2}, tc_2, r'')). \tag{7.17}
 \end{aligned}$$

7 Inequalities for the unrestricted master problem

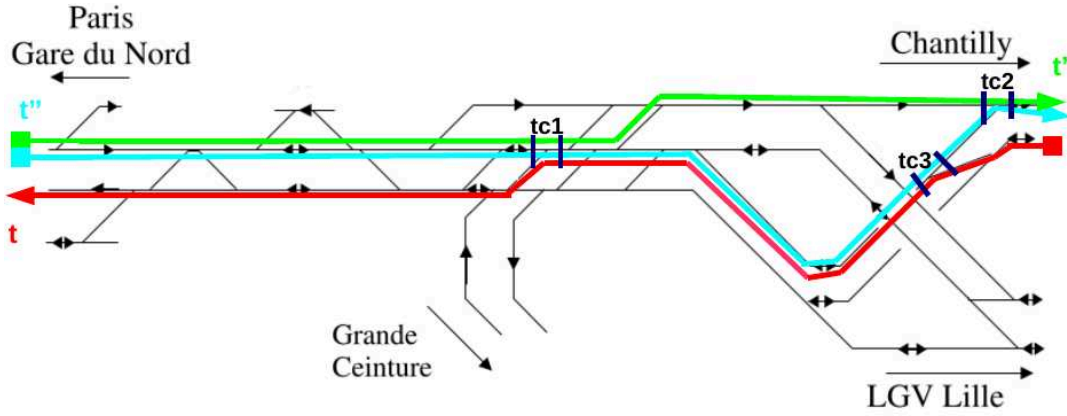


Figure 7.8 Situation in which Inequalities (7.17) apply.

$$\begin{aligned}
 & y_{t',t'',tc_1} + y_{t,t',tc_3} - y_{t,t'',tc_2} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 4 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^r \cap TC^{r'} \cap TC^{r''}, tc_2 \in TC^r \cap TC^{r''}, tc_3 \in TC^r \cap TC^{r'}, \\
 & tc_1 \in TC(tc_0, tc_2, r), tc_3 \in TC(tc_1, tc_\infty, r), \\
 & tc_1 \in TC(tc_0, tc_2, r''), tc_3 \in TC(tc_0, tc_1, r'), \\
 & (tc_2 \in TC(ref_{r,tc_3}, tc_3, r)) \vee (tc_3 \in TC(ref_{r,tc_2}, tc_2, r)). \tag{7.18}
 \end{aligned}$$

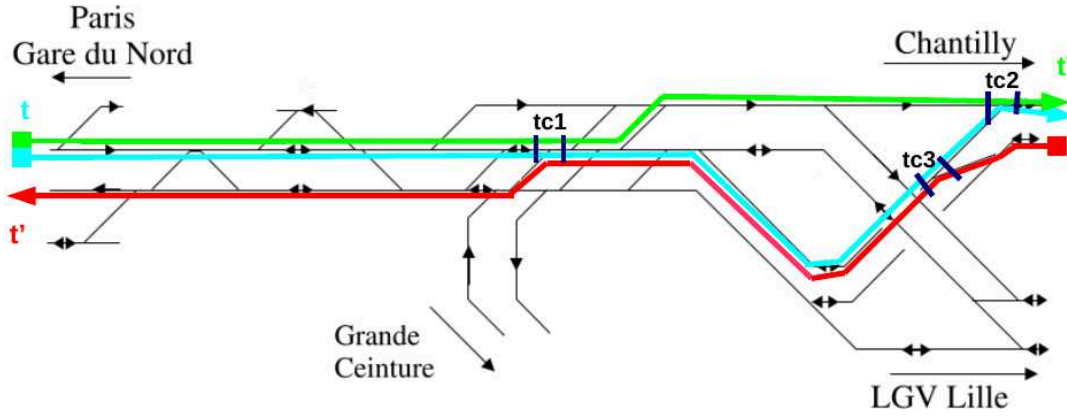


Figure 7.9 Situation in which Inequalities (7.18) apply.

7.2.6 Family F6

Family F6 contains Inequalities (7.19) - (7.24):

$$\begin{aligned}
 & y_{t,t'',tc_2} - y_{t,t',tc_1} - y_{t',t'',tc_3} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 3 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^r \cap TC^{r'}, tc_2 \in TC^r \cap TC^{r''}, tc_3 \in TC^{r'} \cap TC^{r''}, \\
 & tc_1 \in TC(tc_2, tc_\infty, r), (tc_1 \in TC(ref_{r,tc_2}, tc_2, r)) \vee (tc_2 \in TC(ref_{r,tc_1}, tc_1, r)), \\
 & tc_1 \in TC(tc_3, tc_\infty, r'), tc_2 \in TC(tc_0, tc_3, r''). \quad (7.19)
 \end{aligned}$$

Given three trains t , t' and t'' using three track-circuits tc_1 , tc_2 and tc_3 , Inequalities (7.19) force that if t passes before t'' on tc_2 and t'' before t' on tc_3 then t' must precede t on tc_1 . They must be imposed under the following conditions:

- tc_1 is shared by trains t and t' which travel in the same direction,
- tc_2 is shared by t and t'' which travel in opposite direction,
- tc_3 is shared by t' and t'' which travel in opposite direction,
- in the route of t , tc_1 follows tc_2 and they belong to the same reservation zone,
- in the route of t' and t'' , tc_1 follows tc_3 and tc_2 precedes tc_3 , respectively.

Figure 7.10 shows a case in which Inequalities (7.19) apply. For the same reason discussed for

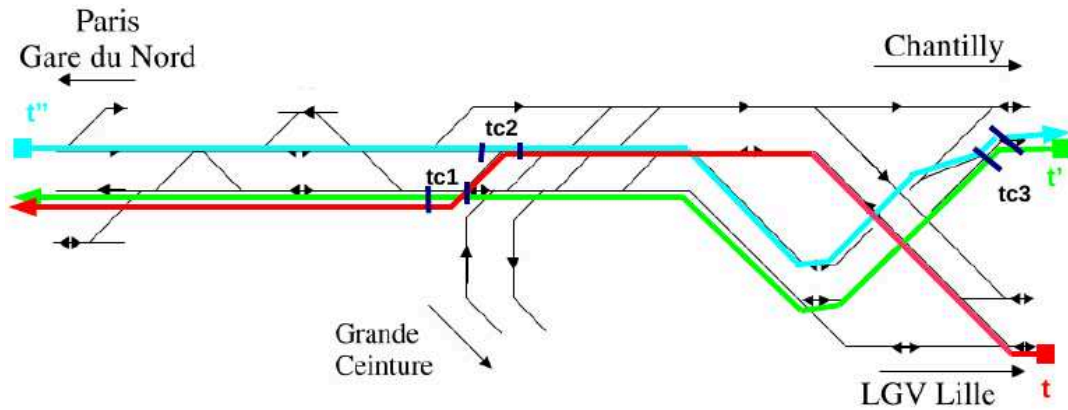


Figure 7.10 Situation in which Inequalities (7.19) apply.

Inequalities (7.13), the following five sets of Inequalities (7.20) - (7.24) similar to (7.19) are

7 Inequalities for the unrestricted master problem

imposed if the trains traveling in the same direction on tc_1 change.

$$\begin{aligned}
 & y_{t',t'',tc_2} + y_{t',t',tc_1} - y_{t',t'',tc_3} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 4 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^r \cap TC^{r'}, tc_2 \in TC^{r'} \cap TC^{r''}, tc_3 \in TC^r \cap TC^{r''}, \\
 & tc_1 \in TC(tc_3, tc_\infty, r), tc_2 \in TC(tc_0, tc_3, r''), \\
 & tc_1 \in TC(tc_2, tc_\infty, r'), (tc_1 \in TC(ref_{r',tc_2}, tc_2, r')) \vee (tc_2 \in TC(ref_{r',tc_1}, tc_1, r')). \quad (7.20)
 \end{aligned}$$

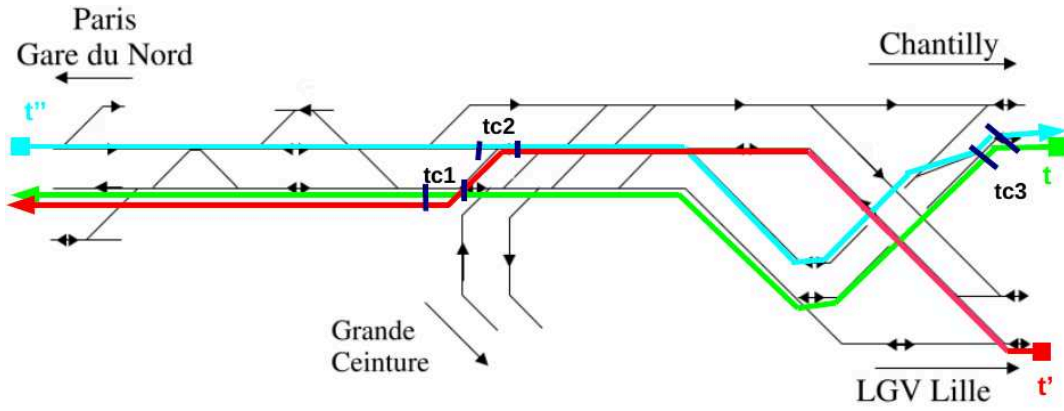


Figure 7.11 Situation in which Inequalities (7.20) apply.

$$\begin{aligned}
 & y_{t',t'',tc_3} - y_{t',t'',tc_1} - y_{t',t',tc_2} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 3 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^{r'} \cap TC^{r''}, tc_2 \in TC^r \cap TC^{r'}, tc_3 \in TC^r \cap TC^{r''}, \\
 & tc_1 \in TC(tc_2, tc_\infty, r'), (tc_1 \in TC(ref_{r',tc_2}, tc_2, r')) \vee (tc_2 \in TC(ref_{r',tc_1}, tc_1, r')), \\
 & tc_1 \in TC(tc_3, tc_\infty, r''), tc_2 \in TC(tc_0, tc_3, r). \quad (7.21)
 \end{aligned}$$

$$\begin{aligned}
 & y_{t',t'',tc_1} + y_{t',t',tc_3} - y_{t',t'',tc_2} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 4 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^{r'} \cap TC^{r''}, tc_2 \in TC^r \cap TC^{r''}, tc_3 \in TC^r \cap TC^{r'}, \\
 & tc_2 \in TC(tc_0, tc_3, r), tc_1 \in TC(tc_3, tc_\infty, r'), \\
 & tc_1 \in TC(tc_2, tc_\infty, r''), (tc_1 \in TC(ref_{r'',tc_2}, tc_2, r'')) \vee (tc_2 \in TC(ref_{r'',tc_1}, tc_1, r'')). \quad (7.22)
 \end{aligned}$$

7.2 Inequalities for the unrestricted master problem

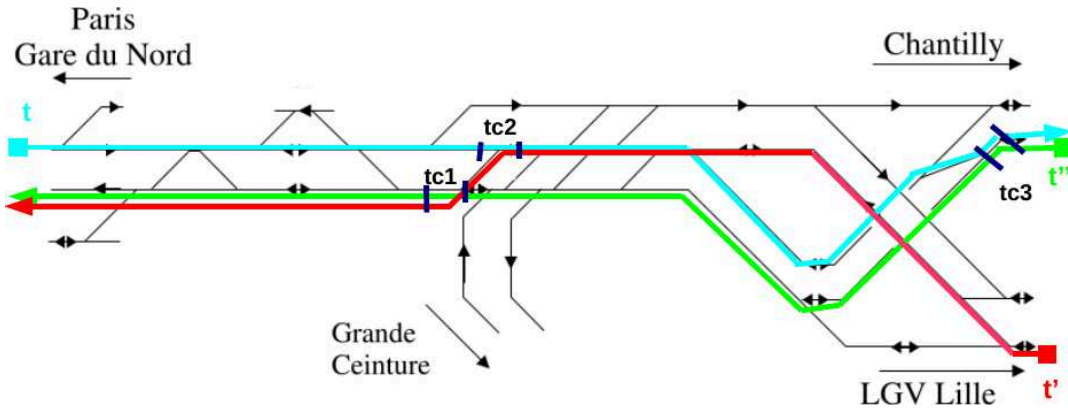


Figure 7.12 Situation in which Inequalities (7.21) apply.

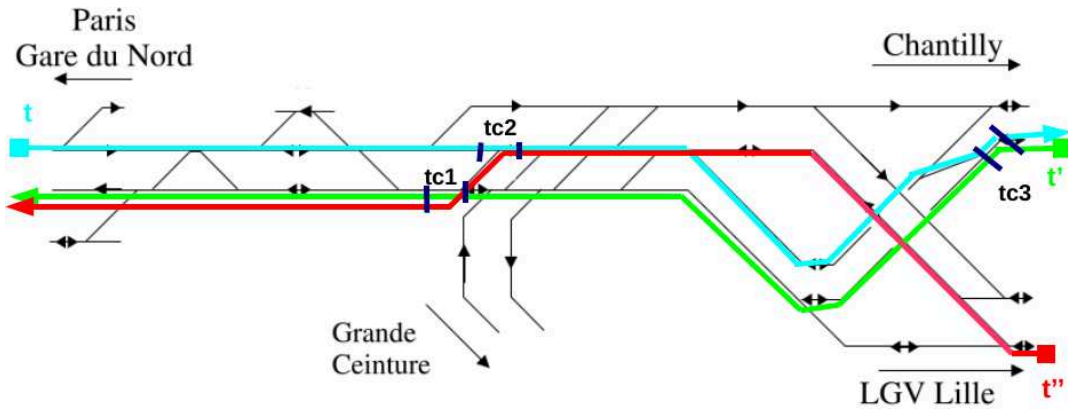


Figure 7.13 Situation in which Inequalities (7.22) apply.

$$\begin{aligned}
 & y_{t,t',tc_2} + y_{t',t'',tc_3} - y_{t,t'',tc_1} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 4 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^r \cap TC^{r''}, tc_2 \in TC^r \cap TC^{r'}, tc_3 \in TC^{r'} \cap TC^{r''}, \\
 & tc_1 \in TC(tc_2, tc_\infty, r), (tc_1 \in TC(ref_{r,tc_2}, tc_2, r)) \vee (tc_2 \in TC(ref_{r,tc_1}, tc_1, r)), \\
 & tc_1 \in TC(tc_3, tc_\infty, r''), tc_2 \in TC(tc_0, tc_3, r'). \quad (7.23)
 \end{aligned}$$

7 Inequalities for the unrestricted master problem

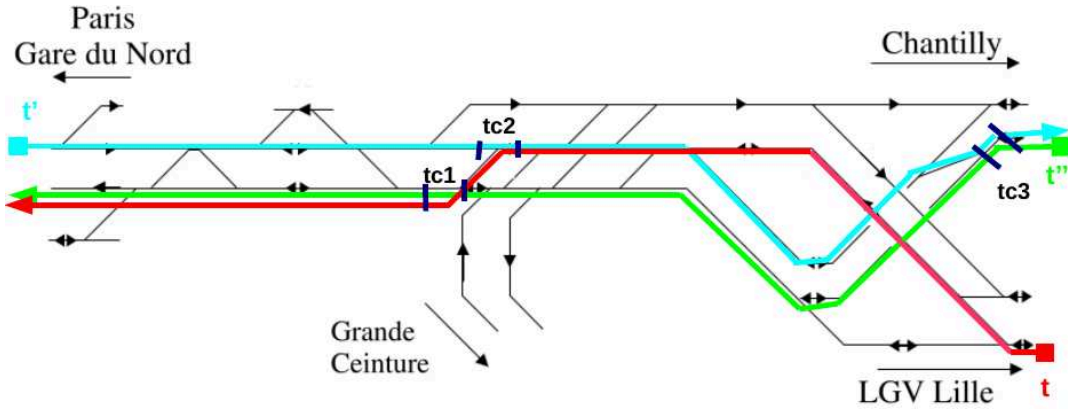


Figure 7.14 Situation in which Inequalities (7.23) apply.

$$\begin{aligned}
 & y_{t,t'',tc_1} - y_{t,t',tc_3} - y_{t',t'',tc_2} + x_{t,r} + x_{t',r'} + x_{t'',r''} \leq 3 \\
 & \forall t, t', t'' \in T, t < t' < t'', r \in R_t, r' \in R_{t'}, r'' \in R_{t''}, \\
 & tc_1 \in TC^r \cap TC^{r''}, tc_2 \in TC^{r'} \cap TC^{r''}, tc_3 \in TC^r \cap TC^{r'}, \\
 & tc_1 \in TC(tc_3, tc_\infty, r), tc_2 \in TC(tc_0, tc_3, r'), \\
 & (tc_1 \in TC(tc_2, tc_\infty, r''), tc_2, r'') \vee (tc_2 \in TC(ref_{r'',tc_1}, tc_1, r'')). \quad (7.24)
 \end{aligned}$$

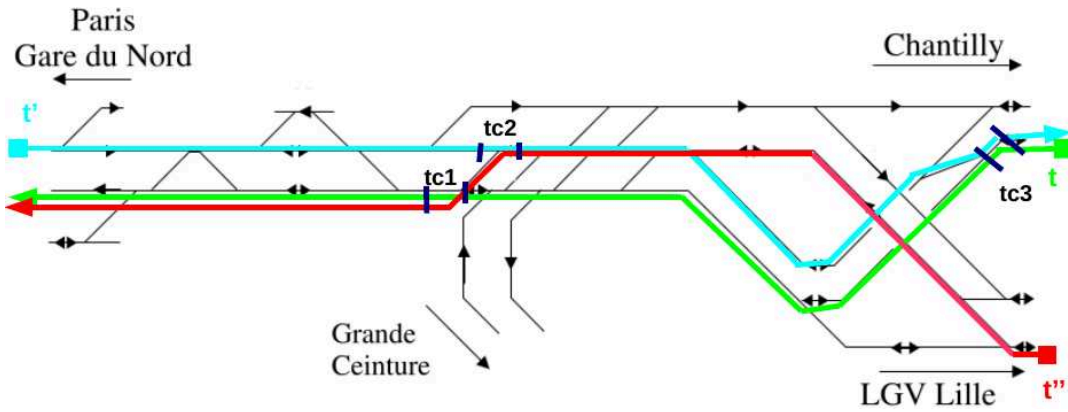


Figure 7.15 Situation in which Inequalities (7.24) apply.

7.3 Algorithm configuration

In this section, we aim at selecting the best variant of the **RBR**, which may include one or more families of inequalities proposed in Section 7.2. To do so, we consider the two case studies representing traffic in two control areas described in Section 5.4, with identical experimental setup.

For each control area, we use the 30 instances utilized in Section 5.4 to select the best variant of the **RBR** and we name them *tuning instances*. Then, to compare the best variant of the **RBR** to **RECIFE-MILP** we generate 30 novel instances, named *test instances* which will then be used in Section 7.4. We consider two separate sets of instances for the tuning and the testing for avoiding over-tuning (Birattari, 2009).

The six variants of the **RBR** which we obtain by adding a family of inequalities to the **UMP** are the following:

- **BA-RBR-F1**: **BA-RBR** with the inequalities of family $F1$,
- **BA-RBR-F2**: **BA-RBR** with the inequalities of family $F2$,
- **BA-RBR-F3**: **BA-RBR** with the inequalities of family $F3$,
- **BA-RBR-F4**: **BA-RBR** with the inequalities of family $F4$,
- **BA-RBR-F5**: **BA-RBR** with the inequalities of family $F5$,
- **BA-RBR-F6**: **BA-RBR** with the inequalities of family $F6$.

Table 7.1 present the comparison between the variants of the **RBR** and the standard one presented in Chapter 6 on the *tuning instances* considered. The first column indicates the name of the variant. The two following sets of four columns contain, for the two control areas, the number of instances solved to the optimum within the available computational time, the percentage improvement with respect to the **BA-RBR** objective function, the average delay in seconds and the average optimality gap achieved by each variant, respectively. The results show that, for the Gonesse *tuning instances*, no variants of the **RBR** improves the performance of **BA-RBR**. However, for the Rouen *tuning instances*, **BA-RBR-F3** improves the performance of **BA-RBR**. **BA-RBR-F3** improves the objective function value of **BA-RBR** by 9 % in average and solves four instances to the optimum instead of only one. We think that the failure of the other families is due to a high solution time overhead brought to the **UMP**, which is not balanced by a significant improvement of the solutions produced. As only one family of inequalities achieves improvement over Rouen, we do not investigate the possibility of adding multiple sets of inequalities simultaneously.

7 Inequalities for the unrestricted master problem

Table 7.1 Performance of the [RBR](#) variants vs [BA-RBR](#) over the two control areas (time limit 180 seconds).

Variants	Gonesse				Rouen			
	#Opt	Imp (%)	Delay (s)	Gap (%)	#Opt	Imp (%)	Delay (s)	Gap (%)
BA-RBR	7	–	188	36	1	–	465	96
BA-RBR-F1	7	0	188	36	1	0	465	96
BA-RBR-F2	7	0	188	36	1	0	465	96
BA-RBR-F3	7	0	188	36	4	9	437	76
BA-RBR-F4	7	0	188	36	1	0	465	96
BA-RBR-F5	7	0	188	36	1	0	465	96
BA-RBR-F6	7	0	188	36	1	0	465	96

7.4 Computational analysis

In this section, we compare the performance of the [BA-RBR-F3](#) to [RECIFE-MILP](#) on the two control areas of Gonesse and Rouen. We use the 30 *test instances* for each control area.

In Table 7.2, we report the results achieved, setting the computational time available to three minutes. The first column indicates the name of the algorithm. The last two sets of three columns contain the number of instances solved to the optimum within the available computational time, the average delay in seconds and the average optimality gap achieved by each algorithm, respectively. The table shows that, despite the addition of the inequalities proposed, [BA-RBR-F3](#) remains inefficient with respect to [RECIFE-MILP](#): [RECIFE-MILP](#) always achieves the best performance in terms of the average delay. Moreover, for the Gonesse instances, [BA-RBR-F3](#) is also outperformed in terms of number of optimal solutions found. This does not happen for the Rouen instances, instead, where [BA-RBR-F3](#) finds three optimal solutions instead of the one found by [RECIFE-MILP](#). However, we do not consider these results sufficient to declare [BA-RBR-F3](#) promising. An opportunity which we did not consider in this thesis, but may be worth investigating in the future, is the wise addition of some inequalities, based for example on some measure of criticality of the decisions influenced. This may help reducing the time overhead while still suitably driving the solution process.

7.5 Conclusions

In this chapter, we proposed some inequalities to boost the performance of [BA-RBR](#). The proposed inequalities are added to the [UMP](#), to strengthen the consistency of routing and

Table 7.2 Performance of **BA-RBR-F3** vs **RECIFE-MILP** over the two control areas (time limit 180 seconds).

Algos	Gonesse			Rouen		
	#Opt	Delay (s)	Gap (%)	#Opt	Delay (s)	Gap (%)
BA-RBR-F3	7	188	36	3	406	83
RECIFE-MILP	36	122	0	1	405	95

scheduling decisions. We tackled instances representing traffic at the two control areas tackled in Chapters 5 and 6. The results show that the performance of the proposed algorithm compared to **RECIFE-MILP** remains poor.

We conjecture that one of the reasons behind the bad performance of all the variants tested of **BD** algorithm which we tested up to now may be the presence of the big-M parameter in the cuts generated. The bad performance of the **BD** due to this presence is also confirmed by some authors. For example, [Binato et al. \(2001\)](#) face this issue when dealing with the power transmission network design problem. The authors consider a linear (0-1) disjunctive model which contains a big-M parameter. To cope with the performance issue of their **BA** due to the presence of the big-M parameter in Benders cuts, the authors specialize their Benders approach by using small values for this parameter. [Lamorgese and Mannino \(2016\)](#) face similar performance issues when using the **BD** to tackle the **rtRTMP** modeled as a **MILP** containing a big-M parameter, based on an **AG** (see Section 3.3). To improve their algorithm performance, the authors replace the standard Benders cuts that contain the big-M parameter with stronger ones.

In the next chapter, following this conjecture, we remove the big-M parameter from the cuts generated by proposing a novel variant of **BA**.

Chapter 8

Three-step Benders Algorithm for RECIFE-MILP

8.1 Introduction

In Chapter 7, we concluded that the performance of **BA-RBR-F3** is not satisfactory when compared to **RECIFE-MILP**: **RECIFE-MILP** always achieves the best performance in terms of average delay. One possible reason of the poor performance of our algorithm is the presence of the big-M parameter in Benders cuts.

In this chapter, we propose a **3BA**. **3BA** generates Benders cuts without the big-M parameter.

The rest of this chapter is organized as follows. Section 8.2 presents **3BA**. Section 8.3 presents the computational experiments, and Section 8.4 concludes the chapter.

8.2 Three-step Benders Algorithm: 3BA

In this section, we propose **3BA**. Unlike the reformulations proposed in Chapters 5 and 6 where the **RECIFE-MILP** formulation is decomposed into two problems, **3BA** separates the **RECIFE-MILP** formulation into three problems: the **NUMP**, the **MILSP** and the **LSP**. Figures 8.1 and 8.2 show the procedures behind the classic **BA** used in Chapters 5, 6 and 7, and **3BA**. In **3BA**, the **NUMP** contains only the binary re-routing variables $x_{i,r}$. Given the re-routing decisions made in the **NUMP**, we build the **MILSP** containing the binary rescheduling variables $y_{i,t',\hat{c}}$ and the continuous re-timing variables and we solve the **MILSP** to obtain the optimal rescheduling decisions. Then, we construct the **Dual of the Linear Sub-Problem (DLSP)** with the re-routing decisions made in the **NUMP** and the corresponding optimal rescheduling decisions made in the **MILSP** to define the cuts to be added. These cuts will include only x -variables, as they are to

8.2 Three-step Benders Algorithm: 3BA

be added to the **NUMP**. Remark that **3BA** is promising only as the solution of the rescheduling problem with no re-routing, **MILSP**, is quick. According to our experience, and as proven in the computational analysis, it is in general the case in the **rtRTMP**. The details on **3BA** are described in the rest of this section.

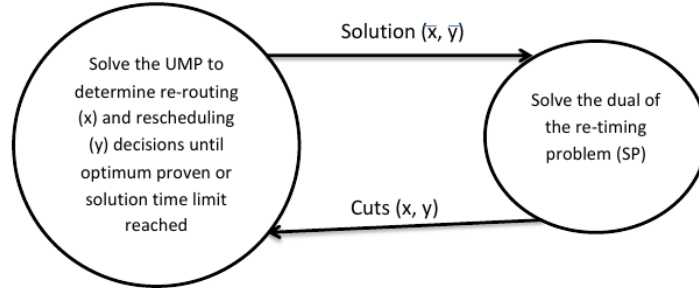


Figure 8.1 Standard Benders Algorithm in which the **RECIFE-MILP** formulation is decomposed into two problems.

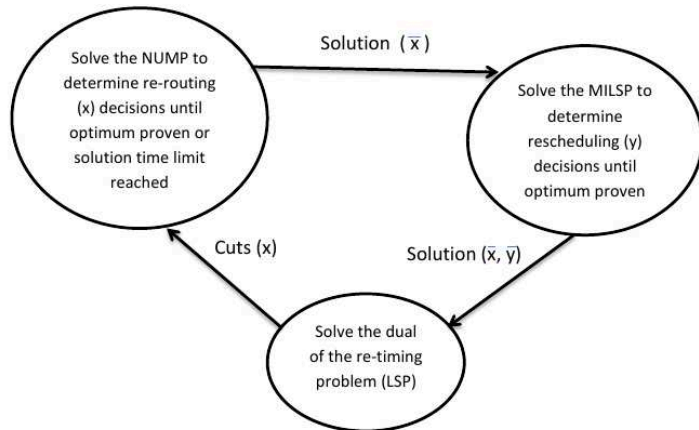


Figure 8.2 Three-step Benders Algorithm in which the **RECIFE-MILP** formulation is decomposed into three problems.

8.2.1 New Unrestricted Master Problem (NUMP)

The **NUMP** includes only the re-routing variables $x_{t,r}$ and a dummy variables z . Its formulation is a follows.

$$\min z. \tag{8.1}$$

8 Three-step Benders Algorithm for RECIFE-MILP

$$\sum_{r \in R_t} x_{t,r} = 1 \quad \forall t \in T \quad (8.2)$$

$$x_{t,r} \in \{0, 1\} \quad \forall t \in T, r \in R_t \quad (8.3)$$

$$z \geq 0. \quad (8.4)$$

8.2.2 Mixed Integer Linear Sub-Problem (MILSP)

When the re-routing variables are fixed in the **NUMP** (let them be $\bar{x}_{t,r}$), then we know the route used, $\bar{r} \in R_t$, by each train t . As done in Section 6.2.1, we can write the reduced formulation of the **MILSP** as follows.

$$\min \sum_{t \in T} w_t D_t. \quad (8.5)$$

$$o_{t,\bar{r},tc} \geq \text{init}_t \quad \forall t \in T, tc \in TC^{\bar{r}} : p_{\bar{r},tc} = tc_0 \quad (8.6)$$

$$-o_{t,\bar{r},tc} + o_{t,\bar{r},p_{\bar{r},tc}} + l_{t,\bar{r},p_{\bar{r},tc}} = -rt_{\bar{r},ty,p_{\bar{r},tc}} \quad \forall t \in T, tc \in TC^{\bar{r}} : tc \neq tc_0 \quad (8.7)$$

$$D_t - o_{t,\bar{r},tc_\infty} \geq -\text{exit}_t \quad \forall t \in T \quad (8.8)$$

$$sU_{t,tc} - o_{t,\bar{r},ref_{\bar{r},tc}} \leq -\text{for}_{bs_{\bar{r},tc}} \quad \forall t \in T, tc \in TC^{\bar{r}} \quad (8.9)$$

$$\begin{aligned} eU_{t,tc} - o_{t,\bar{r},ref_{\bar{r},tc}} - \sum_{tc' \in TC(ref_{\bar{r},tc},tc,\bar{r})} l_{t,\bar{r},tc'} &\geq rel_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \\ &+ \sum_{tc' \in TC(ref_{\bar{r},tc},tc,\bar{r})} rt_{\bar{r},ty,tc'} \quad \forall t \in T, tc \in TC^{\bar{r}} \end{aligned} \quad (8.10)$$

$$eU_{t,tc} - sU_{t',tc} \leq M - My_{t,t',\hat{tc}} \quad \forall t, t' \in T, t < t', tc, \hat{tc} \in TC^{\bar{r}} \cap TC^{t'} : tc \in \hat{TC}_{t,t',\hat{tc}} \quad (8.11)$$

8.2 Three-step Benders Algorithm: 3BA

$$eU_{t',tc} - sU_{t,tc} \leq My_{t,t',\hat{c}} \quad \forall t, t' \in T, t < t', tc, \hat{c} \in TC^{\bar{r}} \cap TC^{\bar{r}'} : tc \in \hat{TC}_{t,t',\hat{c}}. \quad (8.12)$$

Note that in the **MILSP**, Constraints (8.11) and (8.12) contain the binary variables $y_{t,t',\hat{c}}$. Hence, we cannot formulate the dual of this problem and identify the cut to be added to the **NUMP**. To do so, we solve the **MILSP** to get the optimal rescheduling decisions $\bar{y}_{t,t',\hat{c}}$, then we construct the **LSP** given $\bar{y}_{t,t',\hat{c}}$ and the re-routing decisions $\bar{x}_{t,r}$ fixed in the **NUMP**. The reduced formulation of the **LSP** is as follows.

8.2.3 Linear Sub-Problem (LSP)

Note that when we know the re-routing decisions made in the **NUMP** (let them be $\bar{x}_{t,r}$) and the corresponding rescheduling decisions made in the **MILSP** (let them be $\bar{y}_{t,t',\hat{c}}$), the remaining problem is a re-timing one. The **LSP** is formulated as follows.

$$\min \sum_{t \in T} w_t D_t. \quad (8.13)$$

$$o_{t,\bar{r},tc} \geq \text{init}_t \quad \forall t \in T, tc \in TC^{\bar{r}} : p_{\bar{r},tc} = tc_0 \quad (8.14)$$

$$-o_{t,\bar{r},tc} + o_{t,\bar{r},p_{\bar{r},tc}} + l_{t,\bar{r},p_{\bar{r},tc}} = -rt_{\bar{r},ty,p_{\bar{r},tc}} \quad \forall t \in T, tc \in TC^{\bar{r}} : tc \neq tc_0 \quad (8.15)$$

$$D_t - o_{t,\bar{r},tc_\infty} \geq -\text{exit}_t \quad \forall t \in T \quad (8.16)$$

$$sU_{t,tc} - o_{t,\bar{r},ref_{\bar{r},tc}} \leq -\text{for}_{bs_{\bar{r},tc}} \quad \forall t \in T, tc \in TC^{\bar{r}} \quad (8.17)$$

$$\begin{aligned} eU_{t,tc} - o_{t,\bar{r},ref_{\bar{r},tc}} - \sum_{tc' \in TC(ref_{\bar{r},tc},tc,\bar{r})} l_{t,\bar{r},tc'} &\geq rel_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \\ &+ \sum_{tc' \in TC(ref_{\bar{r},tc},tc,\bar{r})} rt_{\bar{r},ty,tc'} \quad \forall t \in T, tc \in TC^{\bar{r}} \end{aligned} \quad (8.18)$$

8 Three-step Benders Algorithm for RECIFE-MILP

According to the value of the rescheduling decisions $\bar{y}_{t,t',\hat{t}c}$ made in the **MILSP** we add to the **LSP** one of the following two sets of constraints:

$$eU_{t,tc} - sU_{t',tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}'} \quad (8.19)$$

$$eU_{t',tc} - sU_{t,tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}'}. \quad (8.20)$$

If $\bar{y}_{t,t',\hat{t}c} = 1$ then we add Constraints (8.19) to the **LSP** otherwise (8.20) is added.

Let $\alpha_{t,\bar{r},tc}$, $\lambda_{t,\bar{r},tc}$, θ_t , $\phi_{t,tc}$, $\omega_{t,tc}$, $\eta_{t,t',tc}$, $\psi_{t,t',tc}$, be the dual variables associated with Constraints (8.14)-(8.20) respectively. As such, $\alpha_{t,\bar{r},tc} \geq 0$, $\lambda_{t,\bar{r},tc} \in \mathbb{R}$, $\theta_t \geq 0$, $\phi_{t,tc} \leq 0$, $\omega_{t,tc} \geq 0$, $\eta_{t,t',tc} \leq 0$ and $\psi_{t,t',tc} \leq 0$. The **DLSP** is written as follows.

8.2.4 Dual of the Linear Sub-Problem (DLSP)

$$\begin{aligned} & \max \sum_{t \in T} \sum_{tc \in TC^{\bar{r}}} \text{init}_t \alpha_{t,\bar{r},tc} - rt_{\bar{r},ty,p_{\bar{r},tc}} \lambda_{t,\bar{r},tc} + \\ & + \sum_{t \in T} \sum_{tc \in TC^{\bar{r}}} (\text{rel}_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \sum_{tc' \in TC(\text{ref}_{\bar{r},tc},tc,\bar{r})} rt_{\bar{r},ty,tc'}) \omega_{t,tc} - \text{for}_{bs_{\bar{r},tc}} \phi_{t,tc} - \\ & - \sum_{t \in T} \text{exit}_t \theta_t. \end{aligned} \quad (8.21)$$

$$\alpha_{t,\bar{r},tc} + \lambda_{t,\bar{r},tc} - \phi_{t,tc} - \omega_{t,tc} \leq 0 \quad \forall t \in T, tc \in TC^{\bar{r}} : p_{\bar{r},tc} = tc_0 \quad (8.22)$$

$$\lambda_{t,\bar{r},tc} - \lambda_{t,\bar{r},p_{\bar{r},tc}} - \phi_{t,tc} - \omega_{t,tc} \leq 0 \quad \forall t \in T, tc \in TC^{\bar{r}} \quad (8.23)$$

$$\lambda_{t,\bar{r},tc} - \omega_{t,tc} \leq 0 \quad \forall t \in T, tc \in TC^{\bar{r}} \quad (8.24)$$

$$\alpha_{t,\bar{r},tc_\infty} - \lambda_{t,\bar{r},tc_\infty} - \theta_t \leq 0 \quad \forall t \in T \quad (8.25)$$

$$\theta_t \leq 1 \quad \forall t \in T \quad (8.26)$$

8.2 Three-step Benders Algorithm: 3BA

If $\bar{y}_{t,t',\hat{c}} = 1$ then we add Constraints (8.27) and (8.28) to the DLSP

$$\phi_{t',tc} - \eta_{t,t',tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}'}$$
 (8.27)

$$\omega_{t,tc} + \eta_{t,t',tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}'}$$
 (8.28)

Constraints (8.27) and (8.28) can be rewritten as follows.

$$\phi_{t',tc} + \omega_{t,tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}'}$$
 (8.29)

If $\bar{y}_{t,t',\hat{c}} = 0$ then we add (8.30) and (8.31) to the DLSP.

$$\phi_{t,tc} - \psi_{t,t',tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}'}$$
 (8.30)

$$\omega_{t,tc} + \psi_{t,t',tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}'}$$
 (8.31)

Constraints (8.30) and (8.31) can be also rewritten as follows.

$$\phi_{t,tc} + \omega_{t',tc} \leq 0 \quad \forall t, t' \in T, t < t', tc \in TC^{\bar{r}} \cap TC^{\bar{r}'}$$
 (8.32)

As the MILSP finds the optimal schedule given the routing decisions, the re-timing problem with such schedule and routing is feasible by definition. Thus, the DLSP is always bounded, and all Benders cuts returned are optimality ones. Remark that, as neither the NUMP nor the DLSP include y -variables, these cuts are formulated as follows:

$$\begin{aligned} & z - \left(\sum_{t \in T} \sum_{\bar{r} \in R_t} \sum_{tc \in TC^{\bar{r}}} x_{t,\bar{r}} (\text{init}_t \alpha_{t,\bar{r},tc} - rt_{\bar{r},ty,p_{\bar{r},tc}} \lambda_{t,\bar{r},tc}) + \right. \\ & + \sum_{t \in T} \sum_{\bar{r} \in R_t} \sum_{tc \in TC^{\bar{r}}} x_{t,\bar{r}} [(\text{rel}_{bs_{\bar{r},tc}} + ct_{\bar{r},ty,tc} + \sum_{t' \in TC(\text{ref}_{\bar{r},tc,t,c,\bar{r}})} rt_{\bar{r},ty,t'}) \omega_{t,tc} - \text{for}_{bs_{\bar{r},tc}} \phi_{t,tc}] - \\ & \left. - \sum_{t \in T} \text{exit}_t \theta_t \right) \geq 0 \\ & ((\alpha, \lambda, \theta, \phi, \omega) \in P_S) \end{aligned} \quad (8.33)$$

8 Three-step Benders Algorithm for RECIFE-MILP

Table 8.1 Performance of **3BA** vs **RECIFE-MILP** and **BA-RBR-F3** over the two control areas (time limit 180 seconds).

Algos	Gonesse			Rouen		
	#Opt	Delay (s)	Gap (%)	# Opt	Delay (s)	Gap (%)
BA-RBR-F3	7	188	36	4	437	80
3BA	9	179	29	22	29	27
RECIFE-MILP	30	120	0	2	419	92

8.3 Computational experiments

In this section, we compare the performance of **3BA** to the **RECIFE-MILP** one on the two control areas considered in the previous chapters of this thesis: Gonesse and Rouen. We use the 30 instances utilized in Section 5.4 for each control area. In addition, we compare **3BA** to **BA-RBR-F3**.

In Table 8.1, we report the results achieved on the two case studies, setting the computational time available to three minutes. The first column indicates the name of the algorithm. The two following sets of three columns contain the number of instances solved to the optimum within the available computational time, the average delay in seconds and the average optimality gap achieved by each algorithm, respectively.

Removing the big-M from the cuts, together with the guidance of the re-routing decisions through the optimal rescheduling, appear to be a good approach: the average returned objective function value of **3BA** is lower than the one of **BA-RBR-F3** in the two control areas, while the number of optimal solutions proven is higher. In the computational time available, the number of cuts generated for **3BA** is lower than the one for **BA-RBR-F3**. In particular, **BA-RBR-F3** generates 142 and 309 cuts in average versus 114 and 61 for **3BA** on the Gonesse and Rouen instances, respectively. The lower number for **3BA** can be explained by the longer time needed for the generation of one cut, which includes the solution of a **MILP** problem. However, the fewer cuts do not penalize **3BA** with respect to **BA-RBR-F3**, and this allows conjecturing that the quality of the cuts added is better. Regarding the performance of **3BA** compared to **RECIFE-MILP**, the results show that the former clearly outperforms the latter in Rouen instances. **3BA** solves 22 instances to the optimum, finds 29 seconds as the average delay and 27% as the average optimality gap. The corresponding figures for **RECIFE-MILP** are 2, 419 and 92%, respectively. However, on the Gonesse instances, although the performance of **3BA** is better than the one of **BA-RBR-F3**, it is still not comparable to the one of **RECIFE-MILP**.

8.3 Computational experiments

Table 8.2 Comparison of **3BA** and **RECIFE-MILP** performance on the Gonesse instances when varying the maximum number of routes per train (time limit 180 seconds).

Max routes	# x	# y	3BA		RECIFE-MILP	
			# Opt	Av-delay (s)	# Opt	Av-delay (s)
1	15	68	30	188	30	188
2	31	308	29	138	30	135
3	46	522	23	131	30	123
4	62	1053	14	129	30	123
5	77	1438	12	165	30	121
6	86	1609	10	173	30	120
7	91	1609	11	165	30	120
8	96	1698	10	165	30	120
9	101	1698	10	165	30	120
10	105	1698	9	179	30	120
11	110	1759	8	173	30	120
12	115	1759	7	177	30	120
13	120	1802	9	179	30	120

To find an explanation to the different relative performance in the two control areas, we conjecture that **3BA** is appropriate for the instances on which the number of alternative routes available per train is important. Specifically, the Rouen instances contain more alternative routes per train than the Gonesse ones: the average number of routes per train for Rouen is 68, while it is 8 for Gonesse. To support our conjecture, we study in the following how the results on each control area change when we vary the maximum number of alternative routes per train. To do so, we solve the same instances considered above limiting the number of routes per train to different values. Specifically, we consider the first m available routes for each train, and we vary m between 1 and the maximum number of routes in the original instances.

The results of these experiments for each control area are reported in Tables 8.2 and 8.3, setting the computational time available to three minutes. Column 1 reports the maximum number of routes per train in each instance. Columns 2 and 3 indicate the average number of x and y -variables per instance, respectively. Columns 4-5 and 6-7 report the number of instances solved to the optimum within the available computational time and the average delay in seconds in the returned solution for each algorithm, respectively.

The graphical analysis of results in Tables 8.2 and 8.3 is done in Figures 8.3 and 8.4. In Figures 8.3, the x -axis indicates the maximum number of routes per train and the y -axis shows the average delay in seconds for Gonesse and Rouen. In Figure 8.4, the x -axis indicates the maximum number of routes per train and the y -axis shows the number of instances solved to

8 Three-step Benders Algorithm for RECIFE-MILP

Table 8.3 Comparison of 3BA and RECIFE-MILP performance on the Rouen instances when varying the maximum number of routes per train (time limit 180 seconds).

Max routes	# x	# y	3BA		RECIFE-MILP	
			# Opt	Av-delay (s)	# Opt	Av-delay (s)
1	12	71	30	459	30	459
2	23	230	30	22	30	22
3	31	253	30	18	30	18
4	37	253	30	18	30	18
5	43	275	30	18	30	18
6	49	309	30	18	30	18
7	55	309	30	18	30	18
8	60	329	30	18	30	18
9	66	329	30	18	30	18
10	72	357	30	18	30	18
20	122	411	25	21	30	18
40	216	486	25	19	25	33
60	310	502	25	18	24	74
80	384	507	25	24	18	95
100	451	507	25	23	13	176
120	488	507	22	31	12	243
140	524	507	23	34	9	296
160	561	507	23	36	9	344
180	598	507	23	31	9	279
200	628	507	24	23	5	341
220	648	507	22	35	5	343
240	668	507	24	27	3	332
260	688	507	24	26	5	316
280	708	507	23	29	4	394
300	728	507	23	35	2	403
320	748	507	22	35	1	428
340	768	507	23	36	3	422
360	788	507	23	28	1	459
384	812	507	22	29	2	419

8.3 Computational experiments

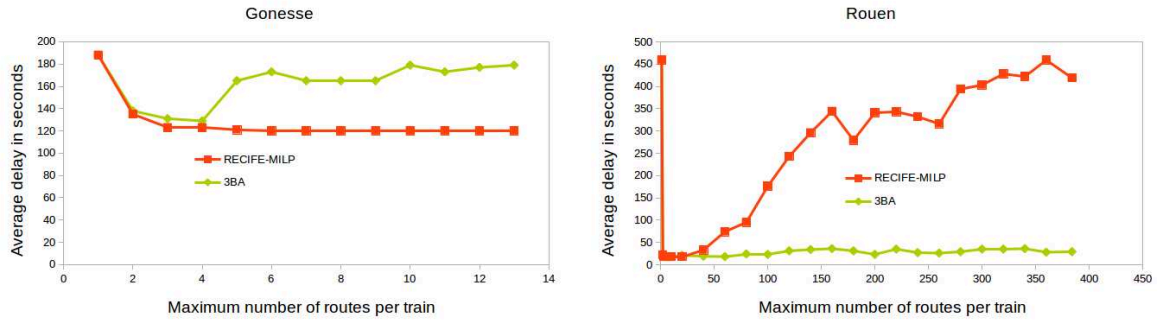


Figure 8.3 Comparison of the average delay achieved by RECIFE-MILP and 3BA on the two control areas depending the maximum number of routes per train.

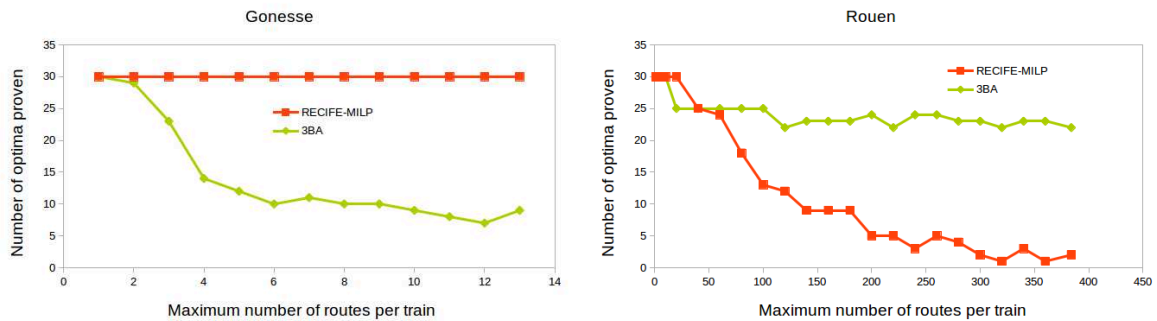


Figure 8.4 Comparison of the number of instances solved to the optimum by RECIFE-MILP and 3BA on the two control areas depending the maximum number of routes per train.

the optimum within the available computational time, again for the two control areas. These figures show the following results.

For the Rouen instances (see Table 8.3, and Figures 8.3 and 8.4, bottom) the higher the maximum number of routes per train is, the more the average delay found by RECIFE-MILP increases. Instead, the average delay found by 3BA remains stable and tends to decrease. When the maximum number of routes per train is in the interval [1,10], the two algorithms achieve the same performance. When this value is 20 RECIFE-MILP solves all the instances to the optimum, while 3BA fails to do so in 5 cases. However, for 4 of them, 3BA reaches the optimal solution although it does not manage to prove its optimality. To assess the statistical significance of the difference between the two algorithms, we perform the Wilcoxon rank-sum test with a confidence level of 0.95. This test does not show any significant difference between the algorithms run on Rouen, when the maximum number of routes per train is in [1, 60]. Nevertheless, for a maximum number of routes per train going from 80 to the extreme value of 384, the difference is statistically significant in favor of 3BA. However, different observations are suggested for Gonesse (see Table 8.2 and Figures 8.3 and 8.4, top). Specifically, the

8 Three-step Benders Algorithm for RECIFE-MILP

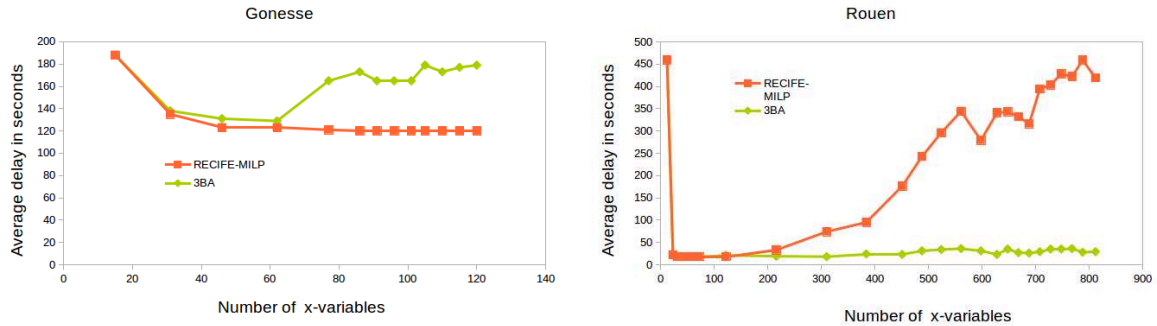


Figure 8.5 Comparison of the average delay achieved by [RECIFE-MILP](#) and [3BA](#) on the two control areas depending the number of x -variables.

experiments on Gonesse do not allow to remark any significant difference when the maximum number of routes per train is smaller than or equal to 2. From 3, this difference is statistically significant in favor of [RECIFE-MILP](#). Moreover, in terms of the number of optima proven, [RECIFE-MILP](#) solves all the instances to the optimum, while the number of instances solved to the optimum by [3BA](#) tends to decrease when the maximum number of routes per train increases.

As a result, we cannot state that the difference of performance between the two algorithms is due to the maximum number of alternative routes available per train. To further investigate the reason of this difference, we study the performance depending to the total number of x and y -variables separately. The graphical analysis of this study is done in Figures 8.5 - 8.8.

In Figure 8.5, we draw the behavior of the two algorithms on the two control areas in terms of average delay in seconds in the returned solution (y -axis) depending the number of x -variables (x -axis). We remark that on the Gonesse instances, the average delay in seconds found by [RECIFE-MILP](#) tends to stabilize after an initial diminution as a function of the number of x -variables, while this value for [3BA](#) tends to increase, also in this case after an initial diminution. On the Rouen instances, we do not observe the same behavior. Remark that the total number of x -variables is larger for Rouen (812 in average) than for Gonesse (120 in average). Although it is not clear when observing the whole range of the number of these variables, when we consider the same scale of the total number of x -variables on Gonesse and Rouen (between 12 and 120), we observe that the average delay found by both algorithms tends to stabilize after an initial diminution (see Tables 8.2 and 8.3). However, on Rouen, we observe that when the total number of x -variables increases, [RECIFE-MILP](#) suffers from this augmentation and its average delay in seconds tends to increase, while the average delay in seconds found by [3BA](#) tends to decrease after an initial slight augmentation.

8.3 Computational experiments

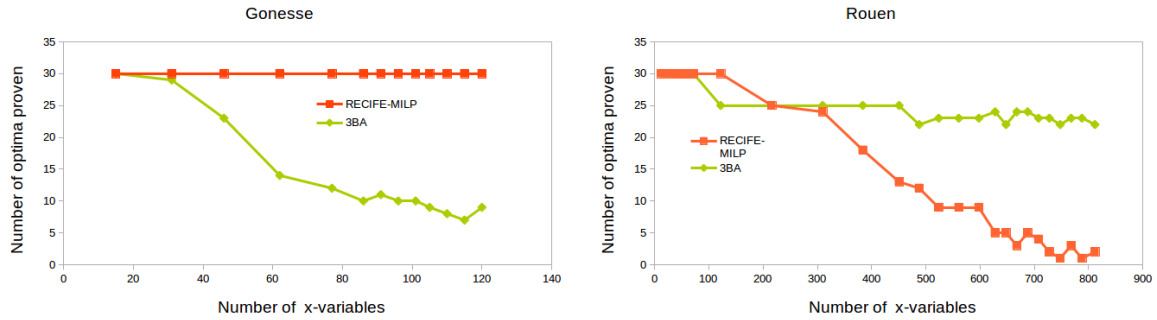


Figure 8.6 Comparison of the number of instances solved to the optimum by [RECIFE-MILP](#) and [3BA](#) on the two control areas depending the number of x -variables.

In Figure 8.6, We analyze the behavior of the two algorithms on the two control areas in terms of the number of instances solved to the optimum (y -axis) by each algorithm depending the total number of x -variables (x -axis). For Gonesse, we observe that [RECIFE-MILP](#) solves all instances to the optimum, while [3BA](#) fails to do so as soon as more than one route is considered per train (all the points following the first one). Instead, for Rouen, when the total number of x -variables is in the interval $[12, 72]$, the number of instances solved to the optimum within the available computational time by the two algorithms is the same. When the value is 122 (the first point where the lines get separated in Figure 8.6, bottom), the number of instances solved to the optimum within the available computational time decreases for [3BA](#) and remains constant for [RECIFE-MILP](#). However, when the total number of x -variables increases [RECIFE-MILP](#) suffers from this augmentation and its number of instances solved to the optimum tends to decrease, while these figures for [3BA](#) tends to stabilize.

The observation that we can draw on Figures 8.5 and 8.6 is that, when an instance of the [rtRTMP](#) contains “many” alternative train routes, meaning “many” routes are available for many trains, [3BA](#) seems to be the best option. However, this observation does not fully explain the different performance. To further investigate, we consider the total number of y -variables. In particular, we analyze the behavior of the two algorithms on the two control areas in terms of the average delay in seconds and the number of instances solved to the optimum by each algorithm (y -axis) depending the total number of y -variables (x -axis) in Figures 8.7 and 8.8. Note that the number of y -variables is constant for Rouen throughout the increase of the number of routes available per train, when this number is above 80 (see Table 8.3): the last points in Figures 8.7 and 8.8 bottom have the same abscissa. This is due to the characteristics of the additional routes available for the trains: they are new combinations of parts of routes which are already covered by existing y -variables, since they were present in other trains routes. Moreover, remark that the characteristics of the train routes in the two control areas imply that,

8 Three-step Benders Algorithm for RECIFE-MILP

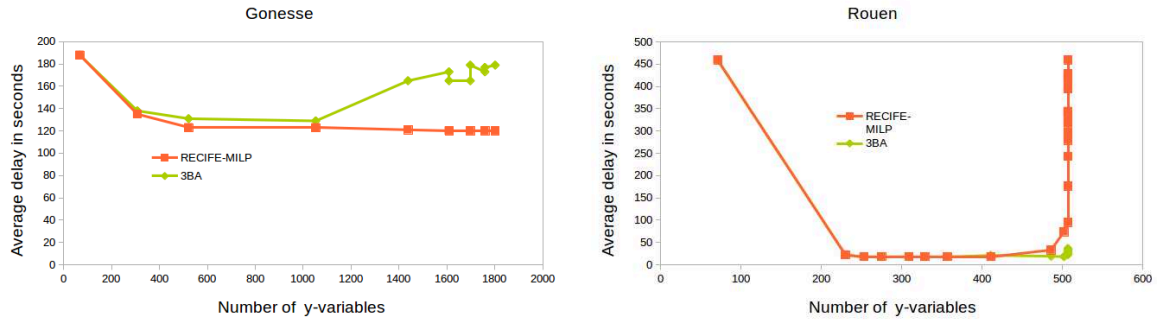


Figure 8.7 Comparison of the average delay achieved by RECIFE-MILP and 3BA on the two control areas depending the number of y-variables.

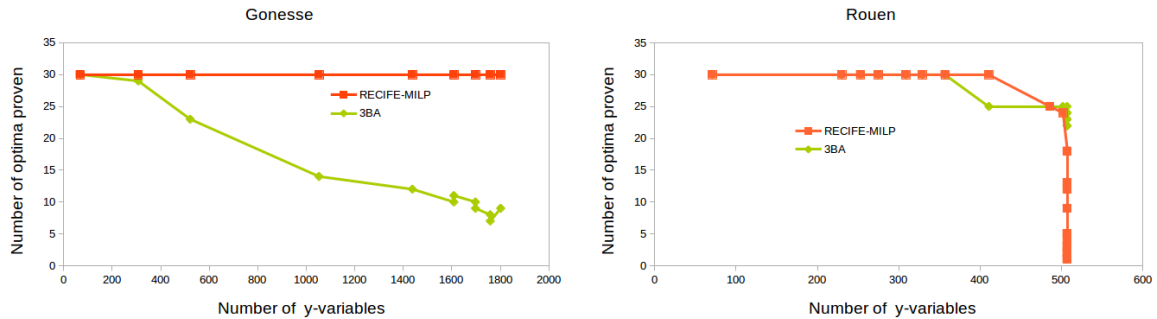


Figure 8.8 Comparison of the number of instances solved to the optimum by RECIFE-MILP and 3BA on the two control areas depending the number of y-variables.

when the maximum number of routes per train increases, the number of y-variables for Gonesse increases much more than for Rouen.

For Rouen, we observe that the average delay found by both algorithms decreases at first, as a number of y-variables. However, when this number gets high (486), 3BA tends to stabilize while RECIFE-MILP returns solutions with higher and higher average delay. Even when the number of y-variables remains constant (507), the average delay keeps increasing, due to the effect of the higher number of routes. Also for Gonesse we observe an initial delay diminution. However, when the number y-variables increases 3BA suffers from this augmentation and its average delay in seconds tends to increase, while this value for RECIFE-MILP tends to stabilize. Reasoning in terms of number of instances solved to the optimum, for Rouen, in the interval [71, 357], the two algorithms are successful for all instances. When this value is 411, the number of instances solved by 3BA decreases, while for RECIFE-MILP it remains stable. When the total number of y-variables reaches 507, the number of instances solved by 3BA tends to stabilize, while this value for RECIFE-MILP strongly decreases following the increase of the number of routes per train. Instead, for Gonesse, as soon as more than one

route is available for each train (second point in the series), **3BA** suffers from the noticeable augmentation of y -variables and the number of instances it solves to the optimum decreases, while for **RECIFE-MILP** it remains stable.

The conclusion which we draw from these analyzes is that, when an instance of the **rtRTMP** contains “many” x -variables and “few” y -variables as in the case of Rouen with many alternative routes per train, **3BA** achieves a better performance than **RECIFE-MILP**. Instead, when the number of y -variables is “large” and the number of x -ones is “small”, as in Gonesse, **RECIFE-MILP** outperforms **3BA**. These results suggest that, depending on the characteristics of the **rtRTMP** instances, either **3BA** or **RECIFE-MILP** can be the best option. Specifically, when a **rtRTMP** instance contains “many” train routes that do not generate “many” rescheduling variables $y_{t,t',\hat{c}}$ **3BA** proves to be efficient. Instead, disregards the number of train routes, if these routes generates “many” rescheduling variables $y_{t,t',\hat{c}}$, then **RECIFE-MILP** seems to be the best.

The fact that **3BA** outperforms **RECIFE-MILP** when there are “many” routing decisions (i.e., x -variables) and “few” scheduling ones (i.e., y -variables) may be due to several reasons.

One possible reason is related to the computational time necessary to solve the rescheduling problem (i.e., the **MILSP**) in **3BA**. When the **MILSP** contains “many” y -variables this computational time may increase. As a result, **3BA** generates less Benders cuts than when the **MILSP** is easy to solve. We remark that for Gonesse, where the **MILSP** contains “many” y -variables “few” Benders cuts are added to the **NUMP**: for example, when all the routes of the original instances are available for the trains, the average computational time per cut added is 1.33 seconds. Instead, for Rouen, where the **MILSP** contains “few” y -variables, “many” Benders cuts are added to the **NUMP**: the average computational time per cut added is 0.64 seconds. Indeed, the quality of the search process being equal, the more solutions are explored, and hence the more cuts are generated, the better the solution quality is. This may explain the better performance for Rouen than for Gonesse.

Moreover, we believe that another reason for the different performance is related to the characteristics of the control area considered. In particular, it is related to the fact that different control areas may feature different route inter-dependencies. Let us call two routes interdependent if two trains using them incur in strong potential conflicts, and independent otherwise. A proxy for routes interdependency is the number of y -variables which derive from their use: the more the variables, the higher the interdependency¹. Although routes interdependency is difficult to quantify, intuitively it can be observed that Rouen, with its line structure, allows the definition of more independent routes than Gonesse, that is a junction rich of switches

¹Remark that this is only a proxy since what really matters for the reasoning which follows is the presence of potential conflicts, which also depend on the trains timing. However, we think the proxy is meaningful here.

8 Three-step Benders Algorithm for RECIFE-MILP

and bidirectional tracks. For Rouen, given a set of trains, a few of which are perturbed, it is often possible to identify an allocation of routes such that the perturbed trains seldom cross the others. Depending on the fact that these routes are chosen or not, the total delay and hence the objective function value of the **LSP** will change quite a lot. For Gonesse, this is still possible if the number of trains is low, but it becomes very unlikely for large train sets. As a consequence, the objective function value of the **LSP** corresponding to different routes allocations will often be similar for this control area. This is indeed what we observe throughout **3BA** runs. Getting now to the different behavior of **3BA** on the two control areas, we think that, in general, largely different **LSP** (and hence **DLSP**) objective function values will imply largely different coefficients for the x -variables involved. In turn, this will imply that different route assignments, when evaluated in the **NUMP** already including some cuts, will have quite different impact on the value of variable z , and hence on the **NUMP** objective function. Thus, the quality of different route assignments may become visible already after a few cut generations, and this may indeed improve the performance of the solution process. Hence for Rouen, where routes are more independent, **3BA** soon manages to make wise route choices and reach good performance, whereas for Gonesse, where routes are strongly interdependent, **3BA** struggles to identify promising assignments and behaves quite poorly.

8.4 Conclusions

In this chapter, we proposed **3BA**. It is an algorithm based on the decomposition of the **RECIFE-MILP** formulation into three problems: the **NUMP**, the **MILSP** and the **LSP**. The **NUMP** contains only the binary re-routing variables $x_{t,r}$. Given the re-routing decisions made in the **NUMP**, we build the **MILSP** containing the binary rescheduling variables $y_{t,t',\hat{c}}$ and the continuous re-timing variables. We obtain the optimal scheduling decisions by solving the **MILSP**, and we construct the **LSP** with the re-routing decisions made in the **NUMP** and the corresponding optimal rescheduling decisions made in the **MILSP**, to define the cuts to be added to the **NUMP**. This reformulation allows the generation of Benders cuts without the big-M parameter and the avoidance of the exploration of rescheduling infeasible or sub-optimal solutions. By doing so, we obtain an algorithm capable of reaching significantly better performance than **RECIFE-MILP** for the Rouen instances. Unfortunately, this is not the case for Gonesse ones, although **3BA** outperforms **BA-RBR-F3** which we proposed in Chapter 7. We conjecture that the different performance on the two control areas is due to the characteristics of the **rtRTMP** instances to be tackled: we think that when a **rtRTMP** instance contains “many” train routes that do not generate “many” rescheduling variables **3BA** is the best option to achieve high quality results, while **RECIFE-MILP** is to be preferred when there

are “few” train routes even if these routes generate “many” rescheduling variables. In future research we will try to verify the general validity of this conjecture by tackling other case studies, which unfortunately are not available today.

Chapter 9

Conclusions and future works

9.1 Conclusions

In this thesis, we focused on the **rtRTMP**. This is the problem of re-routing and rescheduling trains in case of perturbation, to minimize delay propagation. More specifically, we proposed a **BD** to deal with this problem. To facilitate the comprehension of this thesis, in Chapter 2 we introduced main concepts and procedures characterizing the railway system. In particular, we started by defining the decision process for traffic planning. Then, we described the railway infrastructure elements and the two main approaches used to represent them: macroscopic and microscopic. Finally, we gave extended descriptions of the **rtRTMP**, as it is the problem tackled in this thesis.

Before addressing the details on our **BAs** proposed in Chapters 5-8 to tackle the **rtRTMP**, we presented a literature review on this problem in Chapter 3. Indeed, as mentioned in Section 1.2, we found that relevant contributions exist in the operations research literature which propose algorithms, including decomposition ones, to address the **rtRTMP**. However, many of them appear suitable to cope with the problem considering cases in which the size of instances tackled are relatively small. Moreover, several approaches neglect the train re-routing possibility. These and other strong assumptions often limit the actual applicability of the proposed algorithms. Hence, our contribution to the existing literature consisted in trying to increase the applicability of an existing algorithms in the practice by overcoming some of these limits. Namely, we consider the **RECIFE-MILP** algorithm that we presented in Chapter 4. **RECIFE-MILP** is a **MILP** based heuristic which allows modeling specific circulation constraints that can accurately emulate the railway traffic in real circumstances without penalizing the quality of the results. However, it has been shown that the performance of the **RECIFE-MILP** algorithm may strongly worsen when tackling large instances in the short time allowed by the real-time nature of the

problem. For the **MILP** formulation at the basis of this algorithm, we hence proposed **BD** approaches to improve the performance when tackling large instances.

In particular, in Chapter 5 we presented the **SBR** of the **RECIFE-MILP** formulation. It consists in splitting the **MILP** formulation presented in Chapter 4 into the **UMP** and the **SP**. The **UMP** is the real-time train routing and scheduling problem. It contains the re-routing and rescheduling variables (binary variables) of the overall problem, and one dummy variable representing the contribution of continuous variables to the **UMP** objective function. The **SP** contains the continuous variables which represent train passing and stopping times, and deduces the delay suffered by each train. The decomposed problem is solved with the **BA** in which we add progressively Benders **OCs** and **FCs** to the **UMP**, obtaining them through the iterative solution of the **DSP**. After presenting the reformulation, we performed computational experiments on two case studies representing traffic in two control areas in France: the Pierrefitte–Gonesse junction and a line section around the Rouen–Rive–Droite station. The results showed that the **SBR** is not appropriate for the **rtRTMP**.

In Chapter 6, to improve the performance of the **SBR** presented in Chapter 5, we proposed the **RBR** for the **RECIFE-MILP** formulation. Unlike the classical decomposition approach (i.e., the **SBR**), the **RBR** exploits the nature of the **RECIFE-MILP** formulation. In our experiments on the same two case studies used in Chapter 5, the results showed that the **RBR** is better than the **SBR**. However, despite the improvement achieved, the performance of **RBR** was still significantly worse than the one of **RECIFE-MILP**.

With the same performance improvement goal, in Chapter 7 we proposed some inequalities to boost the performance of **RBR** presented in Chapter 6. The proposed inequalities were added to the **UMP**, to strengthen the consistency of routing and scheduling decisions. After an algorithm configuration phase in which we identified the most convenient inequalities to add to the **UMP**, we tackled instances representing traffic at the two control areas tackled in Chapters 5 and 6. The results showed that, although adding inequalities to the **UMP** improves the **RBR** performance, the improvement of the proposed algorithm compared to **RECIFE-MILP** remains modest.

Finally, in Chapter 8 we proposed the **3BA**. It is an algorithm based on the decomposition of the **RECIFE-MILP** formulation into three problems: the **NUMP**, the **MILSP** and the **LSP**. The **NUMP** contains only the binary re-routing variables. Given the re-routing decisions made in the **NUMP**, we build the **MILSP** containing the binary rescheduling variables and the continuous re-timing variables. We obtain the optimal scheduling decisions by solving the **MILSP**, and we construct the **LSP** with the re-routing decisions made in the **NUMP** and the corresponding optimal rescheduling decisions made in the **MILSP**, to define the cuts to be added to the **NUMP**. This reformulation allows the generation of Benders cuts without the big-M parameter

9 Conclusions and future works

and the avoidance of the exploration of rescheduling infeasible or sub-optimal solutions. By doing so, we obtained an algorithm capable of reaching significantly better performance than **RECIFE-MILP** for the Rouen instances. Unfortunately, this is not the case for Gonesse ones, although **3BA** outperforms the algorithm which we proposed in Chapter 7. We conjecture that the different performance on the two control areas is due to the characteristics of the **rtRTMP** instances to be tackled: we think that when a **rtRTMP** instance contains “many” train routes that do not generate “many” rescheduling variables **3BA** is the best option to achieve high quality results, while **RECIFE-MILP** is to be preferred when there are “few” train routes even if these routes generate “many” rescheduling variables. Hence, we conclude that if we have large instances it seems necessary to reduce the number of routes in some sensible way. However, until an appropriate procedure to do so is available, **3BA** may be an option to effectively limit delay propagation in railway traffic.

9.2 Future works

In this thesis, the application of the **BD** to the **MILP** formulation used in **RECIFE-MILP** revealed some issues that deserve investigation in future works.

The first issue we identify concerns the unsuitability of **BD** when applied to the **RECIFE-MILP** formulation as it is. Specifically, we can point out three research directions which we consider particularly relevant.

- Indeed, the formulation of the **UMP** is such that the objective function does not contain binary variables (i.e., the re-routing and rescheduling decisions), but only the dummy variable representing in some sense the impact of binary decisions on continuous variables. Hence, the objective function is only driven by the cuts which are progressively added during the **BA** run. This implies that, until a remarkable number of cuts are added, no promising areas of the solution space can be identified and the search in the **UMP** cannot follow any smart direction. We think that the identification of a suitable objective function for the **UMP** may be possible, and it is likely to quite significantly improve the performance of our algorithms.
- Moreover, looking at the performance issues of the algorithms, we think that they are likely, at least in part, to be linked to the fact that the master problem which determines the value of the binary variables has no constraints to force their consistency. In other words, there is no link between the re-routing decisions and the rescheduling ones in the **UMP**, or between different rescheduling decisions. Although we proposed some inequalities in Chapter 7 that impose this consistency in several circumstances, they were

not sufficient to overcome this problem. On the one hand, in future works we will try to identify other, possibly stronger, inequalities. On the other hand, we will study the possibility of adding them dynamically to the branch-and-bound tree for exploiting their merits without adding excessive computational burden to the search process.

- Finally, we will consider the opportunity to decompose stronger reformulations of **RECIFE-MILP**, as the one recently proposed by Pellegrini et al. (2017). Indeed, this reformulation exploits the link between scheduling and routing decision, for decreasing the number of binary variables of the **MILP** formulation. From computational experiments on Pierrefitte-Gonesse junction, this reformulation seems to bring a clear performance improvement to **RECIFE-MILP**, and we think it may be the case also for our algorithms.

The second issue is related to the choice of the suitable decomposition method to be used to tackle the **rtRTMP**. The mathematical decomposition based on the **BD** principles which we proposed in thesis does not seem to be particularly efficient when the size of the instances increases. Specifically, although the performance is interesting when the control area considered have particular features related to the number and characteristics of the alternative routes of the trains, it is not practicable to tackle a whole national network, for example. To be able to do so, we think the spatial decomposition is the only viable option, i.e., a decomposition in which a whole control area is divided into subset of tractable dispatching zone. Specifically,

- we think that it is possible to imagine a combined microscopic and macroscopic approach, in which the former is used to tackle the **rtRTMP** in critical locations (e.g., stations or junctions) and the latter to deal with the more general problem in the whole network. Indeed an interactive algorithm may be defined to obtain coherent traffic management decisions, but the way this shall be done is not clear yet. The main issue to be tackled is the fact that the optimal solutions at different critical locations are indeed interdependent but are likely to be incoherent. What is the suitable way for solve the overall problem in cases as this is not agreed in the community, and how to get to the optimal or to a good sub-optimal solution once the objective is defined shall be object of future research.

The last issue concerns the improvement of the performance of the **RECIFE-MILP** algorithm. Indeed, several other possible approaches can be used in addition to the **BD** considered in this paper. In particular,

- a promising possibility is linked to the strengthening of the linear relaxation of **RECIFE-MILP** formulation. Indeed, when studying the results of our experiments, we realized that the lower bound obtained from the linear relaxation of the **RECIFE-MILP** formulation is sometimes very far from the optimal solution, and in the most unlucky instances

9 Conclusions and future works

increasing the value of this lower bound takes a very long time. An intuition behind the weakness of the linear relaxation is the fact that if the precedence binary variables appearing in the disjunctive constraints are far from integrity, then the trains can run one on the other without accumulating any delay. Other sources of weakness probably contribute to the low quality of the lower bound in some instances, and in future research it will be interesting to deeply analyze this issue to try to improve the performance of [RECIFE-MILP](#).

Bibliography

- Abbink, E., den Berg, B. V., Kroon, L., and Salomon, M. (2004). Allocation of railway rolling stock for passenger trains. *Transportation Science*, 38(1):33–41.
- Abbink, E., Fischetti, M., Kroon, L., Timmer, G., and Vromans, M. (2005). Reinventing crew scheduling at netherlands railways. *Interfaces*, 35(5):393–401.
- Acuna-Agost, R., Michelon, P., Feillet, D., and Gueye, S. (2011a). A MIP-based local search method for the railway rescheduling problem. *Networks*, 57(1):69–86.
- Acuna-Agost, R., Michelon, P., Feillet, D., and Gueye, S. (2011b). SAPI: Statistical analysis of propagation of incidents. A new approach for rescheduling trains after disruptions. *European Journal of Operational Research*, 215(1):227–243.
- Arenas Pimentel, L. (2016). Contributions on microscopic approaches to solve the train timetabling problem and its integration to the performance of infrastructure maintenance activities. *Ph.D.thesis, University of VALENCIENNES ET DU HAINAUT CAMBRESIS*.
- Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252.
- Binato, S., Pereira, M., and Granville, S. (2001). A new benders decomposition approach to solve power transmission network design problems. *IEEE TRANSACTIONS ON POWER SYSTEMS*, 16(2):235–240.
- Birattari, M. (2009). Tuning metaheuristics. In *A Machine Learning Perspective*, Springer.
- Bussieck, M., Kreuzer, P., and Zimmermann, U. (1997). Optimal lines for railway systems. *European Journal of Operational Research*, 96(1):54–63.
- Cacchiani, V., Caprara, A., Galli, L., Kroon, L., Maròti, G., and Toth, P. (2012). Railway rolling stock planning: Robustness against large disruptions. *Transportation Science*, 46(2):217–232.
- Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., and Wagenaar, J. (2014). An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15–37.
- Cacchiani, V. and Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737.

Bibliography

- Caimi, G., Chudak, F., Fuchsberger, M., Laumanns, M., and Zenklusen, R. (2011). A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling. *Transportation Science*, 45(2):212–227.
- Caimi, G., Fuchsberger, M., Laumanns, M., and Lüthi, M. (2012). A model predictive control approach for discrete-time rescheduling in complex central railway station approach. *Computers & Operations Research*, 39:2578–2593.
- Caprara, A., Fischetti, M., Toth, P., Vigo, D., and Guida, P. L. (1997). Algorithms for railway crew management. *Mathematical Programming*, 79(1):125–141.
- Caprara, A., Galli, L., and Toth, P. (2011). Solution of the train platforming problem. *Transportation Science*, 45(2):246–257.
- Caprara, A., Monaci, M., Toth, P., and Guida, P. (2006). A lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics*, 154(5):738–753.
- Chu, F. and Oetting, A. (2013). Modeling capacity consumption considering disruption program characteristics and the transition phase to steady operations during disruptions. *Journal of Rail Transport Planning & Management*, 3:54–67.
- Claessens, M., van Dijk, N., and Zwaneveld, P. (1998). Cost optimal allocation of rail passenger lines. *European Journal of Operational Research*, 110(3):474–489.
- Cordeau, J., Soumis, F., and Desrosiers, J. (1975). A benders decomposition approach for the locomotive and car assignment problem. *Transportation Science*, 34:133–149.
- Cordeau, J., Toth, P., and Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation Science*, 32(4):380–404.
- Corman, F., D’Ariano, A., Pacciarelli, D., and Pranzo, M. (2009). Evaluation of green wave policy in real-time railway traffic management. *Transportation Research Part C*, 17:607–616.
- Corman, F., D’Ariano, A., Pacciarelli, D., and Pranzo, M. (2010). A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B*, 44:175–192.
- Corman, F., D’Ariano, A., Pacciarelli, D., and Pranzo, M. (2012a). Bi-objective conflict detection and resolution in railway traffic management. *Transportation Research Part C*, 20:79–94.
- Corman, F., D’Ariano, A., Pacciarelli, D., and Pranzo, M. (2012b). Optimal inter-area coordination of train rescheduling decisions. *Transportation Research Part E*, 48:71–88.
- Corman, F., D’Ariano, A., Pacciarelli, D., and Pranzo, M. (2014). Dispatching and coordination in multi-area railway traffic management. *Computers & Operations Research*, 44:146 – 160.
- Corman, F. and Meng, L. (2015). A review of online dynamic models and algorithms for railway traffic management. *Intelligent Transportation Systems, IEEE Transactions on*, 16(3):1274–1284.
- Corman, F. and Quaglietta, E. (2015). Closing the loop in real-time railway control: Framework design and impacts on operations. *Transportation Research Part C: Emerging Technologies*, 54:15–39.

- Dantzig, G. and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, 8:101–111.
- D’Ariano, A., Corman, F., Pacciarelli, D., and Pranzo, M. (2008). Reordering and local rerouting strategies to manage train traffic in real-time. *Transportation Science*, 42(4):405–419.
- D’Ariano, A., Pacciarelli, D., and Pranzo, M. (2007a). A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183:643–657.
- D’Ariano, A. and Pranzo, M. (2009). An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, 9:63–84.
- D’Ariano, A., Pranzo, M., and Hansen, I. (2007b). Conflict resolution and train speed coordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):208–222.
- Dessouky, M., Lu, Q., Zhao, J., and Leachman, R. (2006). An exact solution procedure to determine the optimal dispatching times for complex rail networks. *IIE Transactions*, 38(2):141–152.
- Dollevoet, T., Corman, F., D’Ariano, A., and Huisman, D. (2014). An iterative optimization framework for delay management and train scheduling. *Flexible Services and Manufacturing Journal*, 26(4):490–515.
- Fan, B., Roberts, C., and Weston, P. (2012). A comparison of algorithms for minimising delay costs in disturbed railway traffic scenarios. *Journal of Rail Transport Planning & Management*, 2:23–33.
- Fang, W., Yang, S., and Yao, X. (2015). A survey on problem models and solution approaches to rescheduling in railway networks. *Intelligent Transportation Systems, IEEE Transactions on*. To appear.
- Fisher, M. and Jaicumeur, R. (1978). A decomposition algorithm for large scale vehicle routing. *Technical Report 78-11-05, Department of decision sciences, university of Pennsylvania*.
- Florian, M., Guerin, G., and Bushel, G. (1976). The engine scheduling problem in a railway network. *INFOR J*, 14:121–138.
- Forsgren, M., Aronsson, M., and Gestrelus, S. (2013). Maintaining tracks and traffic flow at the same time. *Journal of Rail Transport Planning & Management*, 3(3):111–123.
- Fu, H., Nie, L., Meng, L., Sperry, B., and He, Z. (2015). A hierarchical line planning approach for a large-scale high speed rail network: The china case. *Transportation Research Part A: Policy and Practice*, 75:61–83.
- García-Archilla, B., Lozano, A., Mesa, J., and Perea, F. (2013). Grasp algorithms for the robust railway network design problem. *Journal of Heuristics*, 19(2):399–422.

Bibliography

- Geoffrion, A. (2005). Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10:237–260.
- Ghaemi, N., Cats, O., and Goverde, R. (2017). Microscopic model for optimally short-turning trains in case of complete blockage. *RailLille 2017, Lille, France*.
- Goossens, J., van Hoesel, S., and Kroon, L. (2006). Cost optimal allocation of rail passenger lines. *European Journal of Operational Research*, 162(2).
- Higgins, A. (1998). Scheduling of railway track maintenance activities and crews. *Operational Research Society*, 49:1026–1033.
- Hooghiemstra, J., Kroon, L., Odijk, M., Salomon, M., and Zwaneveld, P. (1999). Decision support systems support the search for win-win solutions in railway network design. *Interfaces*, 29(2):15–32.
- Huisman, D., Kroon, L., Lentik, R., and Vromans, M. (2005). Operations research in passenger railway transportation. *Technical report, Erasmus Research Institute of Management*.
- IBM (2017). Ibm ilog cplex concert technology for c++. <https://developer.ibm.com/docloud/blog/2016/11/11/whats-in-cos-12-7/>.
- Jütte, S. and Thonemann, U. (2012). Divide-and-price: A decomposition algorithm for solving large railway crew scheduling problems. *European Journal of Operational Research*, 219(2):214–223.
- Kallas, S. (2011). White paper: Roadmap to a single european transport area—towards a competitive and resource-efficient transport system. *COM(2011) 144 final of 28 March 2011*.
- Kecman, P., Corman, F., D’Ariano, A., and Goverde, R. (2013). Rescheduling models for railway traffic management in large-scale networks. *Public Transport*, 5(1-2):95–123.
- Kroon, L., Romeijn, H., and Zwaneweld, P. J. (1997). Routing trains through railway networks: Complexity issues. *European Journal of Operational Research*, 68:485–498.
- Lamorgese, L. and Mannino, C. (2015). An exact decomposition approach for the real-time train dispatching problem. *Operations Research*, 63(1):48–64.
- Lamorgese, L. and Mannino, C. (2016). A non-compact formulation for job-shop scheduling problem in transportation. *SINTEF Tech. Report F27646, April*.
- Lamorgese, L., Mannino, C., and Piacentini, M. (2016). Optimal train dispatching by benders’-like reformulation. *Transportation Science*, 50(3):910–925.
- Lusby, R., Larsen, J., Ehrgott, M., and Ryan, D. (2011). Railway track allocation: models and methods. *OR Spectrum*, 33(4):843–883.
- Lusby, R., Larsen, J., Ehrgott, M., and Ryan, D. (2012). A set packing inspired method for real-time junction train routing. *Computers & Operations Research*, 40(3):713–724.
- Magnanti, T. and Wong, R. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29:464–484.

- Magnanti, T. and Wong, R. (1984). Network design and transportation planning. *Transportation Science*, 18(1):1–55.
- Maróti, G. (2006). Operations research models for railway rolling stock planning. *PhD thesis. Eindhoven University of Technology, Eindhoven, The Netherlands.*
- Mascis, A. and Pacciarelli, D. (2002). Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143:498–517.
- Mascis, A., Pacciarelli, D., and Pranzo, M. (2004). Scheduling models for short-term railway traffic optimization. In *9th international conference on computer-aided scheduling of public transport, Lecture notes in economics and mathematical systems, San Diego*, pages 9–11.
- Meng, L. and Zhou, X. (2014). Simultaneous train rerouting and rescheduling on an n-track network: A model reformulation with network-based cumulative flow variables. *Transportation Research Part B: Methodological*, 67:208–234.
- Nielsen, L., 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.
- Pachl, J. (2009). Railway Operations Processes. In Theeg, G. and Vlasenko, S., editors, *Railway Signaling and Interlocking*, volume 25, pages 39–60. International Compendium.
- Papadakos, N. (2008). Practical enhancements to the magnanti-wong method. *Operations Research Letters*, 36:444–449.
- Pellegrini, P., Marlière, G., Pesenti, R., and Rodriguez, J. (2015). Recife-milp: An effective milp-based heuristic for the real-time railway traffic management problem. *Intelligent Transportation Systems, IEEE Transactions on*, 16(5):2609–2619.
- Pellegrini, P., Marlière, G., Pesenti, R., and Rodriguez, J. (2017). Boosting the performance of railway traffic management through the reformulation of recife-milp. *RailLille 2017, Lille, France.*
- Pellegrini, P., Marlière, G., and Rodriguez, J. (2014). Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *Transportation Research Part B*, 59:58–80.
- Pellegrini, P., Marlière, G., and Rodriguez, J. (2016). A detailed analysis of the actual impact of real-time railway traffic management optimization. *Journal of Rail Transport Planning and Management*, 6(5):13–31.
- Quaglietta, E., Pellegrini, P., Goverde, R., Albrecht, T., Jaekel, B., Marlière, G., Rodriguez, J., Dollevoet, T., Ambrogio, B., Carcasole, D., Giaroli, M., and Nicholson, G. (2016). The on-time real-time railway traffic management framework: A proof-of-concept using a scalable standardised data communication architecture. *Transportation Research Part C*, 3(4):137–149.
- Richardson, R. (1976). An optimization approach to routing aircraft. *Transportation Science*, 10:52–71.

Bibliography

- Rodriguez, J. (2007). A constraint programming model for real-time train scheduling at junctions. *Transportation Research Part B*, 41:231–245.
- Rodriguez, J., Delorme, X., Gandibleux, X., Marlière, G., Bartusiak, R., Degoutin, F., and Sobieraj, S. (2007). RECIFE: models and tools for analyzing rail capacity. *Recherche Transports Sécurité*, 95:19–36. In French.
- Sahin, G. and Yüceoglu, B. (2011). Tactical crew planning in railways. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1221–1243.
- Samà, M., Meloni, C., D’Ariano, A., and Corman, F. (2015). A multi-criteria decision support methodology for real-time train scheduling. *Journal of Rail Transport Planning & Management*. To appear.
- Samà, M., Pellegrini, P., D’Ariano, A., Rodriguez, J., and Pacciarelli, D. (2016). Ant colony optimization for the real-time train routing selection problem. *Transportation Research Part B*, 85:89–108.
- Samà, M., Pellegrini, P., D’Ariano, A., Rodriguez, J., and Pacciarelli, D. (2017). On the tactical and operational train routing selection problem. *Transportation Research Part C*, 76:1–15.
- Schöbel, A. (2012). Line planning in public transportation: models and methods. *OR Spectrum*, 34(3):491–510.
- Sobieraj, S., Marlière, G., and Rodriguez, J. (2011). Simulation of solutions of a fixed-speed model for the real-time railway traffic optimization problem. *RailRome 2011, Rome, Italy*.
- Steinzen, I., Gintner, V., Suhl, L., and Kliwer, N. (2010). A time-space network approach for the integrated vehicle-and crew-scheduling problem with multiple depots. *Transportation Science*, 44(3):367–382.
- T.K.Ho and T.H.Yeung (2001). Railway junction traffic control by heuristic methods. *Proceedings IEEE – Electric Power Applications*, 148:77–84.
- Törnquist, J. and Persson, J. (2007). N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B*, 41:342–362.
- Törnquist Krasemann, J. (2015). Computational decision-support for railway traffic management and associated configuration challenges: An experimental study. *Journal of Rail Transport Planning & Management*. To appear.
- Veelenturf, L., Potthoff, D., Huisman, D., and Kroon, L. (2012). Railway crew rescheduling with retiming. *Transportation Research Part C*, 20:95–110.
- Wong, R. (1978). Accelerating benders decomposition for network design. *Ph.D.thesis, Department of electrical engineering and computer science, Massachusetts Institute of technology*.
- Zhan, S., Kroon, L., Veelenturf, L., and Wagenaar, J. (2015). Real-time high-speed train rescheduling in case of a complete blockage. *Transportation Research Part B: Methodological*, 78:182–201.

List of acronyms

- 3BA** Three-step Benders Algorithm. vi–viii, 4, 84, 85, 90–98, 101, 102, 117, 118
- AG** Alternative Graph. 22, 24, 25, 27, 34, 57, 83
- BA** Benders Algorithm. vii, 3, 34, 46, 50–57, 62–65, 67, 81–84, 90, 98, 100–102, 116
- BD** Benders Decomposition. 2, 3, 28, 32–35, 45, 46, 50, 56–58, 66, 83, 100–103, 122
- DLSP** Dual of the Linear Sub-Problem. 84, 88, 89, 98
- DSP** Dual of the Sub-Problem. 3, 33, 34, 48–51, 56, 59, 60, 65, 66, 101, 117
- FC** feasibility cut. 3, 33, 34, 50, 51, 53–55, 59, 62–65, 67, 101, 117
- ILP** Integer Linear Programming. 22, 26
- IM** Infrastructure Manager. v, 5–9, 16, 27
- LSP** Linear Sub-Problem. 4, 84, 87, 88, 98, 101, 118
- MILP** Mixed Integer Linear Programming. 2, 3, 22–24, 32, 34–36, 42, 45, 56, 83, 90, 100–103, 116, 118, 119, 122
- MILSP** Mixed Integer Linear Sub-Problem. 4, 84–89, 97, 98, 101, 118
- MP** Master Problem. 33, 34, 49
- NUMP** New Unrestricted Master Problem. 4, 84–87, 89, 97, 98, 101, 118
- OC** optimality cut. 3, 33, 34, 50, 51, 53–55, 59, 62–65, 67, 101, 117
- RBR** Reduced Benders Reformulation. vii, 4, 58–60, 62–66, 81–84, 90, 98, 101, 117

RECIFE REcherche sur la Capacité des Infrastructures FERroviaires - Research on the Railway Infrastructures Capacity. [2](#), [36](#), [116](#)

Red-DSP Reduced Dual of the Sub-Problem. [61](#), [62](#), [67](#)

Red-SP Reduced Sub-Problem. [60](#), [61](#)

RMP Restricted Master Problem. [33](#), [49](#), [50](#), [62](#)

ROMA Railway traffic Optimization by Means of Alternative graphs. [25](#), [37](#), [44](#)

rtRTMP real-time Railway Traffic Management Problem. [1–5](#), [9](#), [12](#), [16–30](#), [32](#), [34–38](#), [42](#), [45](#), [46](#), [52](#), [56](#), [58](#), [59](#), [66](#), [83](#), [85](#), [95](#), [97](#), [98](#), [100–103](#), [115](#), [122](#)

RU Railway Undertaking. [v](#), [5–9](#)

SBR Standard Benders Reformulation. [vii](#), [3](#), [4](#), [46](#), [47](#), [50–59](#), [63](#), [65](#), [66](#), [101](#), [116](#)

SP Sub-Problem. [3](#), [32–34](#), [46](#), [47](#), [56](#), [58–60](#), [66](#), [101](#), [116](#)

UMP Unrestricted Master Problem. [3](#), [4](#), [32](#), [34](#), [46](#), [47](#), [50](#), [51](#), [53](#), [56](#), [58–60](#), [66](#), [67](#), [81](#), [82](#), [101](#), [102](#), [116](#)

Glossary

block section A sequence of track-circuits whose access is controlled by a signal. [v](#), [10–12](#), [14](#), [15](#), [18](#), [23](#), [24](#), [36](#), [39](#), [52](#)

clearing time The time elapsed between the moment when the train's head leaves a track-circuit and the moment when its tail leaves it. [15](#), [38–40](#)

conflict In the rtRTMP, a conflict exists when multiple trains travelling at the planned speed would concurrently require the same track segment. [1](#), [18](#), [24](#), [25](#), [37](#), [122](#)

dwelt time The time that a train stays in a station platform to allow passengers to alight, transfer and board the train. [8](#), [16](#), [38](#), [40](#)

formation time In addition to the reservation time, a supplementary time is needed before a train starts the occupation of a sequence of block sections. [15](#), [39](#), [40](#), [42](#)

interlocking In railway signalling, an interlocking is an arrangement of signal apparatus that prevents conflicting movements through an arrangement of tracks such as junctions or crossings. [11](#), [12](#), [14](#), [38](#), [67](#)

occupation time The time interval during which a train physically occupies a track-circuit. [17](#), [40](#), [42](#), [46](#), [47](#)

RECIFE-MILP A heuristic algorithm, it tackles the railway traffic management problems. [vi–viii](#), [2–4](#), [24](#), [35–38](#), [41–47](#), [49–56](#), [58](#), [59](#), [63–67](#), [81–85](#), [90–98](#), [100–104](#), [116–120](#), [122](#)

release time The release time corresponds to the lapse of time in which the reservation of a track-circuit is still active after a train exits it. [12](#), [15](#), [39](#), [40](#), [42](#), [52](#)

reservation time The time elapsed between the start of the reservation and the actual occupation of a track-circuit by a train. [15](#), [46](#), [47](#)

route The complete sequence of track-circuits traversed by a train during its trip is named route. [8](#), [14](#), [17](#), [18](#), [38](#), [42](#), [93](#)

running time The time spent by the head of a train on a track-circuit when traveling at the planned speed. [15–17](#), [38–40](#), [42](#)

signal A signal is a mechanical or electrical device erected beside a railway line to pass information relating to the state of the line ahead to train drivers. [v](#), [10–12](#), [14](#), [15](#), [18](#), [36](#)

track segment A sequence of track-circuit in which the order between trains cannot change. [1](#), [39](#), [43](#), [44](#), [122](#)

track-circuit A detection device on which the presence of a train is automatically detected. [v](#), [10–12](#), [14](#), [15](#), [17](#), [23](#), [35](#), [37–39](#), [42](#), [43](#), [46](#), [47](#), [56](#), [58](#), [59](#), [67–72](#), [77](#)

utilization we name utilization the sum of reservation and occupation time. [37](#), [40](#), [42](#), [43](#), [59](#)

RÉSUMÉ ÉTENDU EN FRANÇAIS

La mise en place d'une nouvelle façon de produire, consommer et transporter pour soutenir le développement harmonieux de l'économie, la société et l'environnement est devenue aujourd'hui l'une des voies auquel il faut au cours de ce nouveau millénaire pour les grands défis comme la surpopulation, la réduction des ressources naturelles, le changement climatique, etc..

Le transport ferroviaire, considéré comme un moyen de transport de masse et aussi comme un mode de transport écologique, peut jouer un rôle majeur dans la mise en place d'une nouvelle façon de transporter qui est en adéquation avec les nouvelles exigences écologiques. Cependant, le ferroviaire reste peu utilisé dans le transport de biens et de personnes comparé au mode routier bien que ce dernier soit assez énergivore. Pour promouvoir le ferroviaire, le livre blanc de l'Union Européenne sur le transport préconise un transfert modal à hauteur de 50% du transport routier des biens et des personnes sur les trajets de moyenne et longue distance vers le ferroviaire ou la voie d'eau d'ici 2050 afin de parvenir à une réduction des émissions de CO₂ et de la dépendance du pétrole (Kallas, 2011). La croissance attendue du transport ferroviaire qui résultera de ce transfert modal entraîne de nouveaux défis pour ce secteur qui sont entre autres la réduction des coûts, l'efficacité de l'exploitation et l'amélioration de la qualité des services rendus aux clients.

Actuellement, sans considérer l'éventuelle croissance du trafic ferroviaire, la capacité de l'infrastructure des réseaux de plusieurs pays européens est complètement exploitée aux heures de pointe et aux points critiques : une grande quantité de trains traversent ces points critiques dans un laps de temps très réduit. Dans cette situation, en cas de perturbation du trafic, le retard peut se propager dans le réseau à la suite de l'apparition des conflits. Un conflit est une situation d'exploitation où plusieurs trains circulant à vitesse planifiée réclament la même section de voie au même moment. Ainsi, des trains doivent être arrêtés ou décélérés pour assurer la sécurité et les retards se propagent dans le réseau. La propagation du retard serait encore pire avec la croissance attendue du trafic ferroviaire si rien n'est fait. Dans ce contexte, le trafic doit être géré au mieux pour limiter la propagation du retard.

Le problème de gestion opérationnelle du trafic ferroviaire, qui se traduit en anglais par « real-time Railway Traffic Management Problem (rtRTMP) », est le problème de modification des itinéraires et des ordonnancements entre trains afin de limiter la propagation du retard. Actuellement, ce problème est traité par des régulateurs. Ils le font manuellement, ce qui rend le résultat de leur choix généralement sous-optimal. L'utilisation d'un algorithme efficace pour aider les régulateurs dans leur prise de décision est donc cruciale afin d'assurer la gestion efficace du trafic quand les perturbations surviennent. La conception d'un tel algorithme efficace a fait l'objet d'une voie de recherche très active, dans laquelle s'inscrit cette thèse.

L'objectif de cette thèse est de proposer un algorithme de décomposition, plus spécifiquement celui de décomposition de Benders, pour résoudre le problème de gestion opérationnelle du trafic ferroviaire. Plusieurs contributions intéressantes peuvent être trouvées dans la littérature scientifique sur ce problème. Elles proposent des algorithmes, y compris de décomposition, pour traiter le problème de gestion opérationnelle du trafic ferroviaire. Cependant, beaucoup d'entre elles traitent le problème en considérant des instances de taille relativement modeste. De plus, plusieurs approches ignorent ou négligent la possibilité de reroutage. Cette hypothèse forte limite assez souvent l'applicabilité réelle des algorithmes proposés.

La contribution de cette thèse vise à améliorer l'applicabilité de l'algorithme **RECIFE-MILP**. **RECIFE-MILP** est une heuristique basée sur la programmation linéaire en nombres entiers à variables mixtes, terme abrégé en **MILP** qui correspond à son équivalent anglais et qui a été développé par [Pellegrini et al. \(2015\)](#) comme partie intégrante d'une plate-forme d'outils d'aide à la décision nommée REcherche sur la Capacité des Infrastructures FERroviaires (**RECIFE**) et introduite par [Rodriguez \(2007\)](#). **RECIFE-MILP** permet de résoudre certaines instances du problème de gestion opérationnelle du trafic ferroviaire à l'optimum tout en considérant tous les détails de l'infrastructure. En effet, **RECIFE-MILP** permet de modéliser des contraintes de circulation spécifiques qui peuvent précisément représenter les circonstances réelles du trafic. Cette caractéristique est très importante pour cette thèse comme on peut le voir dans nos expérimentations où nous traitons des cas d'étude réels qui représentent le trafic dans deux parties du réseau ferroviaire français. Néanmoins, il a été montré ([Pellegrini et al., 2015](#)) que la performance de **RECIFE-MILP** peut se détériorer quand on fait face à des problèmes de grandes tailles en temps réel. Pour la formulation **MILP** à la base de cet algorithme, nous proposons des approches de décomposition de Benders afin d'améliorer la performance dans le cas de grandes instances.

Dans ce manuscrit, nous commencerons par présenter la reformulation standard de Benders (**SBR**) de la formulation de **RECIFE-MILP**. Elle consiste à séparer la formulation de départ en un problème maître (**UMP**) et un sous-problème (**SP**). Le problème maître est un problème de routage et d'ordonnancement des trains en temps réel. Ce problème contient les variables binaires de reroutage et réordonnancement du problème global et une variable «factice» représentant la contribution des variables continues dans la fonction objectif du problème maître. Le sous-problème contient les variables continues qui représentent les temps de passage et d'arrêt des trains et déduit le retard subi par chaque train en fonction des choix de routage et ordonnancement effectués dans le problème maître. Le problème décomposé est résolu avec l'algorithme de Benders (**BA**) dans lequel nous résolvons itérativement le problème maître et le sous-problème. Le problème maître propose des valeurs pour les variables binaires. Étant données les valeurs pour ces variables de réordonnancement et de reroutage, le sous-problème

déduit les valeurs des variables continues, dans le cas où les valeurs des variables du problème maître donnent des solutions faisables. Dans le cas contraire, de nouvelles contraintes (coupes de faisabilité, **FC**) qui restreignent les valeurs des variables binaires sont ajoutées au problème maître. Si par contre la solution du problème maître est faisable pour le sous-problème et si la solution du sous-problème est sous optimale pour le problème de départ (le problème décomposé), de nouvelles contraintes (coupes d'optimalité, **OC**) qui restreignent les valeurs des variables binaires sont ajoutées au problème maître. Plusieurs coupes, **FC** et **OC**, sont obtenues à travers la résolution itérative du dual du sous-problème (**DSP**). Après avoir présenté la reformulation, nous décrivons des expérimentations sur deux cas d'études réels qui représentent le trafic dans deux zones de contrôle en France : la jonction de Pierrefitte-Gonesse et la section de ligne autour de la gare Rouen-Rive-Droite. Les résultats montrent que la reformulation standard de Benders n'est pas appropriée pour la gestion opérationnelle du trafic ferroviaire : **RECIFE-MILP** trouve toujours de meilleurs résultats que l'algorithme de Benders.

Ensuite, pour améliorer la performance de l'application de la décomposition de Benders, nous proposons la reformulation réduite de Benders (**RBR**) de la formulation de **RECIFE-MILP**. Contrairement à l'approche classique (la reformulation standard), la reformulation réduite exploite la nature de la formulation de **RECIFE-MILP**. L'idée de la reformulation réduite est de supprimer dans l'évaluation du sous-problème des variables et des contraintes qui s'avèrent inutile dès lors que les décisions de routage sont fixées dans le problème maître. En effet, quand un train n'utilise pas un itinéraire, toutes les variables et les contraintes liées aux décisions de timing sur cet itinéraire sont inutiles car elles peuvent être fixées a priori sans optimisation. Dans nos expérimentations sur les deux zones de contrôle, les résultats montrent que la reformulation réduite est meilleure que la reformulation standard. Cependant, malgré l'amélioration réalisée, la performance de la reformulation réduite reste significativement moins performante que celle de **RECIFE-MILP**.

Dans la voie de recherche suivante, nous proposons de générer des inégalités pour accroître les performances de la reformulation réduite de Benders. Les inégalités proposées sont ajoutées au problème maître de Benders, pour renforcer la cohérence entre les décisions de reroutage et de réordonnancement. Après une phase de configuration algorithmique dans laquelle nous avons identifié les inégalités les plus pertinentes à ajouter au problème maître, nous avons résolu des instances représentatives du trafic sur les deux zones de contrôle. Les résultats montrent que, bien que l'ajout des inégalités au problème maître améliore la performance de la reformulation réduite, la performance de l'algorithme de Benders reste modeste comparée à celle de **RECIFE-MILP**.

Finalement, nous proposons un algorithme de Benders en trois étapes (**3BA**). C'est un algorithme basé sur la décomposition de la formulation de **RECIFE-MILP** en trois problèmes :

le nouveau problème maître (**NUMP**), le sous-problème linéaire en nombres entiers à variables mixtes (**MILSP**) et le sous-problème linéaire (**LSP**). L'algorithme **3BA** suit le même principe de l'algorithme classique de Benders : la résolution est itérative, mais à chaque itération, au lieu d'avoir la résolution du problème maître suivi par celle du sous-problème pour générer les coupes, nous résolvons le nouveau problème maître **NUMP**, le sous-problème linéaire **MILSP** puis le sous-problème **LSP** pour générer les coupes. Dans cette décomposition, le nouveau problème maître **NUMP** ne contient que les variables binaires de reroutage. Étant donné les décisions de reroutage obtenues, nous construisons le sous-problème **MILSP** contenant les variables binaires de réordonnancement et les variables continues de re-timing. Avec les décisions d'ordonnancement optimales obtenues avec le sous-problème **MILSP** nous construisons ensuite le sous-problème linéaire **LSP** contenant uniquement les variables de re-timing pour définir les coupes à ajouter au problème maître **NUMP**. Cette reformulation permet de générer des coupes de Benders sans le paramètre big-M et de cette façon éviter l'exploration de réordonnements infaisables ou de solutions sous-optimales. En faisant ainsi, nous obtenons un algorithme capable de réaliser une performance significativement meilleure que **RECIFE-MILP** pour les instances de Rouen-Rive-Droite. Malheureusement, cela n'est pas le cas pour les instances de Gonesse, bien que l'algorithme en trois étapes surpasse celui basé sur la décomposition du problème en deux étapes sur les deux zones de contrôle. Nous conjecturons que la différence de performances de l'algorithme en trois étapes sur les deux zones de contrôle est due aux caractéristiques des instances du problème. Nous pensons que lorsqu'une instance du problème contient « beaucoup » d'itinéraires pour chaque train et « peu » de variables de réordonnement, l'algorithme en trois étapes est la meilleure option pour obtenir des solutions de bonne qualité, tandis que **RECIFE-MILP** est préférable quand il y a « peu » d'itinéraires pour chaque train et « beaucoup » de variables de réordonnement.

En conclusions, nous pouvons affirmer que dans cette thèse, l'application des approches de décomposition de Benders à la formulation **MILP** utilisée dans **RECIFE-MILP** a révélé de nouvelles voies de recherche qui pourront faire l'objet de futurs travaux.

La première problématique que nous identifions concerne l'inadaptation de la décomposition de Benders à la formulation **MILP** actuelle utilisée dans **RECIFE-MILP**. En particulier, nous pouvons mettre en avant trois axes de recherche que nous considérons comme importants.

- Tout d'abord, la fonction objectif de la formulation du problème maître ne contient pas de variables binaires (les variables de reroutage et de réordonnement), mais seulement la variable « factice » représentant en quelque sorte l'impact de variables binaires sur les variables continues. Ainsi, la fonction objectif est seulement guidée par les coupes que nous ajoutons progressivement durant l'exécution de l'algorithme de Benders. Cela implique que, jusqu'à ce qu'un nombre important de coupes soient ajoutées, aucun

espace de solutions intéressant ne peut être identifié et la recherche dans le problème maître ne peut pas suivre une bonne direction. Nous pensons que l'identification d'une fonction objectif problème maître qui intègre les variables binaires pourrait améliorer la performance de nos algorithmes.

- D'autre part, lorsque l'on examine les mauvaises performances des algorithmes, nous pensons qu'elles sont probablement liées, du moins en partie, au fait que le problème maître n'a aucune contrainte pour forcer la cohérence des valeurs des variables binaires. En d'autres termes, il n'y a pas de lien entre les décisions de reroutage et celles de réordonnement ou entre différentes décisions de réordonnement dans le problème maître. Même si nous proposons dans cette thèse quelques inégalités qui imposent cette cohérence dans plusieurs circonstances, elles ne sont pas suffisantes pour remédier à ce problème. Une première perspective de recherche de cette axe serait d'identifier d'autres inégalités fortes. Une première perspective serait la possibilité d'ajouter dynamiquement des inégalités dans un arbre de branchement et séparation afin d'exploiter leurs potentialités sans augmenter le temps de calcul du processus.
- Finalement, nous considérerons la possibilité de décomposer des reformulations plus fortes de [RECIFE-MILP](#), comme celles proposées récemment par [Pellegrini et al. \(2017\)](#). En effet, cette reformulation exploite le lien entre les décisions de routage et d'ordonnement afin de diminuer le nombre de variables binaires de la formulation [MILP](#). À partir des expérimentations sur la jonction de Gonesse, cette reformulation semble apporter des améliorations évidentes à [RECIFE-MILP](#) et nous pensons que cela pourrait être le cas aussi pour les algorithmes de décomposition que nous avons étudié dans cette thèse.

La deuxième problématique est liée au choix de la méthode de décomposition pour résoudre le problème de gestion opérationnelle du trafic ferroviaire. La décomposition mathématique basée sur les principes de Benders que nous proposons dans cette thèse ne semble pas être suffisamment efficace quand on est face à des problèmes de grandes tailles. Bien que la performance soit intéressante quand la zone de contrôle considérée possède des caractéristiques particulières relatives au nombre et à la nature des itinéraires alternatifs des trains, on ne peut pas envisager de traiter par exemple tout un réseau national. Pour en être capable, nous pensons que la décomposition spatiale reste la seule option viable, c'est-à-dire une approche de décomposition dans laquelle on décompose un ensemble de zones de contrôle en plusieurs sous-ensembles «faciles» à traiter. En particulier,

- nous pensons qu'il est possible d'envisager une approche de décomposition selon une hiérarchie macroscopique-microscopique de représentation de l'infrastructure. Le niveau

microscopique serait utilisé pour traiter le problème de gestion opérationnelle du trafic ferroviaire dans des zones critiques (les gares ou les jonctions) et le niveau macroscopique pour traiter le problème sur l'ensemble du réseau. En effet, un algorithme interactif pourrait être défini pour obtenir des décisions de gestion du trafic cohérentes entre les différentes zones critiques mais la manière de faire cela n'est pas encore clairement déterminée. Le principal problème à traiter est que les solutions dans ces zones critiques sont interdépendantes et donc que les solutions optimales localement sont probablement incohérentes entre elles. La meilleure façon de résoudre le problème global dans ce cas n'est pas encore tranchée dans la communauté, de même la façon d'obtenir la solution optimale où une solution sous-optimale de bonne qualité doit faire l'objet de futurs travaux de recherche.

La dernière problématique concerne l'amélioration de la performance de l'algorithme [RECIFE-MILP](#). En effet, plusieurs autres approches peuvent être utilisées en plus de la décomposition de Benders considérée dans cette thèse. En particulier,

- une possibilité intéressante est liée au renforcement de la relaxation linéaire de la formulation de [RECIFE-MILP](#). En effet, en étudiant les résultats de nos expérimentations, nous nous sommes rendu compte que la borne inférieure obtenue de la relaxation linéaire de la formulation de [RECIFE-MILP](#) est quelquefois très éloignée de la solution optimale et dans certaines instances augmenter la valeur de cette borne inférieure consomme beaucoup de temps. Intuitivement, on peut penser que la faiblesse de la relaxation linéaire est que si les variables binaires de précédence qui apparaissent dans les contraintes disjonctives sont assez éloignées de l'intégralité alors les trains peuvent circuler «l'un sur l'autre» sans accumuler de retard. D'autres causes sont peut-être à l'origine de la mauvaise qualité de la borne inférieure dans d'autres instances, un autre futur sujet de recherche serait alors d'analyser en profondeur cette problématique pour améliorer la performance de [RECIFE-MILP](#).

Décomposition de Benders pour la gestion opérationnelle du trafic ferroviaire

Dans plusieurs pays européens, la capacité de l'infrastructure est complètement exploitée aux heures de pointe et aux points critiques : une grande quantité de trains traversent ces points critiques dans un laps de temps très réduit. Dans cette situation le retard d'un train provoqué par un conflit de circulation peut se propager dans tout le réseau. Le problème de la gestion opérationnelle du trafic ferroviaire consiste à trouver les modifications des itinéraires et des ordonnancements des trains qui minimisent la propagation des retards. Dans cette thèse, nous proposons une approche de décomposition de Benders pour la formulation linéaire en nombres entiers à variables mixtes utilisée dans l'algorithme RECIFE-MILP. Après avoir constaté que l'approche de décomposition standard de Benders ne permet pas de trouver rapidement une solution de bonne qualité pour certaines instances du problème, nous étudions trois approches alternatives afin d'améliorer la performance de notre algorithme. Nous proposons d'abord une approche que nous appelons la reformulation réduite de Benders. Ensuite, nous introduisons des inégalités dans la formulation du problème maître de Benders. Finalement, nous scindons le processus de résolution en trois étapes au lieu de deux comme dans la décomposition standard de Benders. L'analyse expérimentale montre que la combinaison de la première et dernière approche surpasse l'algorithme original RECIFE-MILP dans la résolution de grandes instances sous certaines conditions.

Mots-clés :

Gestion opérationnelle du trafic ferroviaire, programmation linéaire en nombres entiers à variables mixtes, décomposition de Benders.

Benders decomposition for the real-time Railway Traffic Management Problem

In railway systems, during congested traffic situations, the infrastructure capacity is completely exploited for trains circulation. In these situations, when traffic is perturbed some trains must be stopped or slowed down for ensuring safety, and delays occur. The real-time Railway Traffic Management Problem (rtRTMP) is the problem of modifying trains route and schedule to limit delay propagation. In this thesis, we propose a Benders decomposition of a MILP-based algorithm for this problem, named RECIFE-MILP. After observing that the standard Benders decomposition (BD) does not allow the effective solution of rtRTMP instances, we study three possible approaches to improve the performance. Specifically, we first propose a modification of the problem reformulation which is typical of BD, obtaining what we call reduced BD. Then, we introduce some inequalities to the Benders master problem. Finally, we split the solution process in three steps rather than two as in the standard BD. As we show in a thorough experimental analysis, the combination of the first and last approaches outperforms the original RECIFE-MILP algorithm when tackling large instances with some specific features.

Mots-clefs

real-time Railway Traffic Management Problem (rtRTMP), Mixed Integer Linear Programming (MILP), Benders decomposition (BD).

