Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

INGEGNERIA BIOMEDICA, ELETTRICA E DEI SISTEMI

Ciclo 34

Settore Concorsuale: 01/A6 - RICERCA OPERATIVA

Settore Scientifico Disciplinare: MAT/09 - RICERCA OPERATIVA

OPTIMIZATION ALGORITHMS FOR ENERGY-EFFICIENT TRAIN OPERATIONS IN REAL-TIME RAIL TRAFFIC MANAGEMENT

Presentata da: Federico Naldini

Coordinatore Dottorato

Michele Monaci

Supervisore

Michele Monaci

Co-supervisori

Daniele Vigo

Antonio di Carmine (Alstom)
Rosalba Suffritti (Alstom)

Esame finale anno 2022

Abstract

This thesis collects our relevant research activities addressing the problem of energy-efficient train operations in real-time rail traffic management. This is a relatively new field of research in Rail Transport Optimization that especially concerns dispatchers, who are in charge of traffic management. Due to traffic perturbations, dispatchers have to re-schedule train operations repeatedly to maintain circulation as smooth as possible, therefore minimizing the total delay. Along with re-scheduling decisions, train speed profiles have to be determined in such a way that a feasible train schedule is preserved and the total energy consumption is minimized. Altering the speed profile of one train not only alters its schedule, but may also invalidate those of the other trains circulating in the same infrastructure. Therefore, it must be ensured that the speed profile of a train is compatible with that of the other trains. A trade-off between energy-efficiency and other re-scheduling objectives, such as delay minimization, is usually to be made. Research on decision support tools for this application can lead to more advanced Traffic Management Systems (TMS), bringing a twofold benefit to rail transport: an increased performance as a mode of transport, and a reduced environmental impact. In the included papers, we consider the following optimization problems.

First, we address a single-train real-time Energy-Efficient Train Control (EETC) problem, envisioned as a sub-problem of a hypothetical multiple-train Energy-Efficient Train Timetabling (EETT) problem for real-time traffic management applications. Precisely, we focus on determining an energy-optimal speed profile for a single train with a given schedule. We propose three algorithms: a constructive heuristic; a multi-start randomized constructive heuristic; and a Genetic Algorithm. We run experiments on real-life case studies provided by our industrial partner ALSTOM, which is a world leader in rail transport.

Second, we address a real-time multiple-train EETT problem known as the real-time Energy Consumption Minimization Problem (rtECMP), which is a subproblem of the real-time Rail Traffic Management Problem (rtRTMP). The rtECMP asks for deciding the speed profiles of multiple trains circulating in a given network during a given time window. The objective is to minimize the weighted sum of total delay and energy consumption. Rail infrastructures are represented with a microscopic level of detail including the interlocking system and signals. We propose a graph-based rtECMP model and three meta-heuristic algorithms: Ant Colony Optimization (ACO); Iterated Local Search (ILS); and Greedy Randomized Adaptive Search Procedure (GRASP). We run experiments on two real-life case studies, considering mixed dense traffic subject to perturbations.

Acknowledgements

First of all, I would like to thank *Alstom* for funding this research project, making my PhD possible as a member of the *Operations Research Group*, Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi" (DEI), Università di Bologna. In particular, I want to thank the following *Alstom*'s members: Antonio Di Carmine, Simone Marzola, Luca Prezioso, Massimo Rosti & Rosalba Suffritti.

Also, I would like to thank the current and former professors of the *Operations Research Group* of the University of Bologna: Roberto Baldacci, Valentina Cacchiani, Andrea Lodi, Enrico Malaguti, Silvano Martello, Michele Monaci (my supervisor), Paolo Toth, Paolo Tubertini & Daniele Vigo (my co-supervisor);

Then, I would like to thank *Directeur de Recherche* Paola Pellegrini and *Directeur de Recherche* Joaquin Rodriguez of the *Université Gustave Eiffel* (ex IFSTTAR), with whom I collaborated as a visiting researcher at COSYS-ESTAS, in the campus of Lille, France.

Finally, I would like to thank my fellow young researchers of the *Operations Research Group*: Luca Accorsi, Carlos Rodrigo Rey Barra, Carlos Contreras-Bolton, Cristiano Fabbri, Naga Venkata Chaitanya Gudapati, Giacomo Lanza, Henri Lefebvre, Paolo Paronuzzi & Antonio Punzo.

In addition, I would like to thank the visiting researchers we hosted:
Benedikt Bienhüls, Torkil Kollsker, Jonas Prünte & Jianguo Qi.

Florence, Italy, October 23^{rd} 2021

Federico Naldini

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context & Motivation

Nowadays, the threat of global warming is more tangible than ever before. Recent studies forecast a global temperature increase of 2°C by 2060 w.r.t. the pre-industrial reference level. A temperature increase of this magnitude will cause disastrous effects on the environment and, in turn, on the world's economy (CotEU, 2021).

Transportation plays an enormous role in terms of environmental impact given its nature and ever increasing demand. In Europe, approximately 25% of the overall greenhouse emissions are caused by transportation alone (EC, 2021a).

The European Union is actively involved in reducing environmental impacts in many sectors, including transportation. Indeed, the European Commission has set the target of reducing the total greenhouse gas emissions of at least 55% by 2030, compared to 1990 levels. This target is part of a broader plan, the so-called *European Green Deal*, that aims to completely eradicate greenhouse gas emissions by 2050 (EC, 2021b).

Rail is widely recognized as one of the most sustainable, reliable and safe mode of transportation. Its great sustainability is not only due to it being mostly electrified but also to other intrinsic aspects, such as a low wheel-rail interface friction (Scheepmaker et al., 2017). Given the zero-emission target and the virtues of railway systems, the European union is pursuing a twofold goal: (OBEU-1) increasing the user base of

rail so to improve the whole transport sector from an environmental perspective, and (OBEU-2) reducing the intrinsic environmental impact of rail itself even further. In this context, 2021 has been declared the European Year of Rail, as part of an awareness campaign of the European Union to promote mass adoption of rail (EYoR, 2021).

In this thesis, we present our research in the realm of railway optimization whose outcomes are in line with both the objectives of the European Union in promoting rail transport. Coherently with OBEU-1, we contribute to the evolution of the decision support systems that can make rail transport even more efficient. This can further boost rail attractiveness, leading more and more users to shift to it from less sustainable modes of transport. Coherently with OBEU-2, we perform research that directly addresses environmental matters: minimization of energy consumption.

We remark that, beside the environmental benefits discussed so far, saving energy yields a significant cost reduction for train operators, further boosting the competitive edge of rail transport. Indeed, an improvement in energy efficiency as small as 1% can save up to several million euros for most railway systems each year (UIC, 2021).

The scope of this thesis is detailed in Section 1.2.

1.2 Scope

In this thesis, we focus on the problem of optimizing energy consumption for train movements as they require very large amounts of energy if not adequately managed. In this setting, minimizing energy consumption results in a twofold positive impact for rail systems. First, it reduces greenhouse gas emissions. Second, it cuts energy costs for train operators, further boosting rail competitiveness (Martínez Fernández et al., 2019). Such positive impacts are in line with the European Union environmental objectives discussed in Section 1.1

The literature on minimizing energy consumption of train movements is usually declined into two important problems: Energy-Efficient Train Control (EETC) and Energy-Efficient Train Timetabling (EETT). Precisely:

- EETC aims to determine energy-efficient speed profiles for one or multiple trains with a given schedule/timetable. A speed profile expresses the train speed as a function of time or distance along its entire journey. By following a speed profile obtained by EETC methods, a train will spend minimal energy while fulfilling its planned timing.

- EETT integrates the train timetabling problem (Section 2.2.2) and EETC to some degree, depending on the specific problem formulation and solution method. Indeed, EETT aims to determine train timetables where energy-efficiency and other objectives are jointly optimized. EETT methods make scheduling decisions, such as arrival, departure, passing-through and dwelling times, for each train while concurrently determining energy-efficient speed profiles to achieve the schedule itself. Eventually, such methods achieve a trade-off between energy consumption and the other objectives.

EETC and EETT do not indicate either specific mathematical formulations or solution methods. Indeed, they are two rather broad families of quantitative approaches aiming at supporting both tactical and strategic service planning (Section 2.2) in achieving energy-efficient train operations. A fairly wide range of models and algorithms has been proposed in the literature. However, most problems in EETC, and therefore in EETT, are modeled with similar principles, which are outlined in Section 2.3.2.

For a thorough review on EETC and EETT, we refer the reader to Scheepmaker et al. (2017). For an extensive review focused on the *Optimal Control Theory* aspects in EETC, we refer the reader to Albrecht et al. (2016a,b). Moreover, each of the included papers contains a literature review focused on the respective topics (Chapters 3, 4 & 5).

From the literature, it emerges that a relevant research gap lies in optimizing energy consumption of train movements during the operational horizon, to support traffic control (*dispatching*) decisions, which are made in *real-time* as traffic flows. This research area is only partially explored and very challenging. Indeed, due to its focus on train operations and energy consumption minimization, this research area not only demands very short computing times, for real-time applicability, but also an accurate representation of train motion, that involves non-linear dynamic constraints typical of optimal control problems (see Section 2.3.2). During train operations (Section 2.2.3), traffic is subject to perturbations causing delays (e.g., a broken door on a car, an unforeseen peak of demand increasing a dwelling time at a station). As a result, the traffic controller (*dispatcher*) must re-schedule trains frequently over the operational horizon to minimize delay propagation. Naturally, control decisions have to be taken as quickly as possible, i.e., in real time, to prevent newly arisen delays from propagating and thereby disrupting the service. This concerns two main areas of research, which are subsets of EETC and EETT. We call them *real-time EETC* (rtEETC) and *real-time EETT* (rtEETT).

This thesis collects our relevant publications that contribute to rtEETC and rtEETT research branches (Section 1.4). We address two problems in particular: the first belongs to the rtEETC family, the second to the rtEETT one. Details are provided in Sections 1.2.1 & 1.2.2.

1.2.1 real-time Energy-Efficient Train Control (rtEETC)

In this thesis, we tackle rtEETC in the form of *single-train EETC* (Howlett et al., 1994a), i.e., computing an energy-efficient speed profile for one train at a time with given schedule. We especially focus on developing optimization algorithms to be used as a module for a rail Traffic Management Systems (TMS) to account for energy consumption in real-time train (re-)scheduling decisions. Therefore, infrastructural details and train interactions are not directly modeled. However, it is envisioned that such details and interactions are handled by a TMS that, in this configuration, relies upon the rtEETC module for computing energy-efficient speed profiles for single trains.

As a result, the TMS can make energy-aware train re-scheduling decisions for multiple trains. Indeed, for each desired running time between two stations, the TMS can determine a realistic energy-efficient speed profile, along with its energy expenditure. Obviously, due to the unforeseen events continuing affecting traffic, the schedule of the train may change during operations. Therefore, the algorithms are required run in a very short computing time.

1.2.2 real-time Energy-Efficient Train Timetabling (rtEETT)

In this thesis, we tackle rtEETT in the form of the *real-time Energy Consumption Minimization Problem* (rtECMP) (Montrone et al., 2018). The rtECMP requires finding energy-efficient speed profiles for multiple trains running in a network fulfilling the operational principles of railway systems, e.g., speed limits and safety distances between trains. The goal of the problem is minimizing total energy consumption along with delay given by traffic perturbations. The rtECMP accurately models both the interactions between trains and infrastructural details of the considered railway network and traffic state. A microscopic level of detail is adopted for achieving a suitable accuracy in representing actual traffic in networks. Indeed, the interlocking and signaling systems are modeled (see Section 2.3.1.1).

1.3 Scope limitations

In this thesis, we address the domain of rtEETC and rtEETT (Section 1.2) with the following limitations and assumptions.

First, the proposed rtEETC/rtEETT methods are heuristic in nature: a sub-optimal solution is sought. A short computing time is prioritized over exactness given the nature of the application. In fact, as discussed in Section 1.2, we focus on energy-efficient train operations in real-time traffic management, which require fast algorithms to support dispatcher's complex decisions. Moreover, highly accurate models of train motion have to be considered along with railway operational constraints. As of today, given the size of realistic instances and the tight computing time requirements,

rtEETC/rtEETT are pretty much intractable with purely exact methods, see, e.g, Montrone et al. (2018), Luan et al. (2018a,b) and Goverde et al. (2021).

Second, the models dealt are subject to simplifying assumptions. Precisely:

- We represent speed profiles as combinations of the four *optimal* driving regimes resulting from the *Pontryagin's Maximum Principle* optimality conditions (Scheepmaker et al., 2017). These are *maximum acceleration; cruising; coasting* and *maximum braking*. For further details, see Section 2.3.2.

- As common in this field, we only explore a reduced discrete subset of all the possible speed profiles for each train. In particular, we make simplifying assumptions on both the switching points and driving regime sequences that constitute train speed profiles. This largely reduces the search space and, thus, eases the computational burden for the solution algorithms proposed (Scheepmaker et al., 2017). By discretizing the search space, we therefore eliminate unneeded complexity. Indeed, for instance, modeling switching point coordinates (time, speed, distance) as a continuum of high-precision decimals might be more detrimental than useful, leading to search space with a large number of similar speed profiles. Furthermore, a human train driver is naturally sensitive only to coarse values of speed, distance and time, effectively leading to a discretization of such values when implementing driving actions. Discretization policies are under our control. Therefore, they let us achieve a trade-off between a model's accuracy and its complexity, matching the precision of the model's solutions to the specific application. See, e.g., Wang et al. (2016); Zhou et al. (2017); Montrone et al. (2018); Luan et al. (2018a,b).

  For an overview of speed profiles, switching points, and their related assumptions see Section 2.3.2. In addition, detailed problem-specific assumptions can be found in each included papers (Chapters 3, 4 & 5).

Third, for simplicity, we neglect the management of energy recuperation through *regenerative brake*. For details on the subject, we refer the interested reader to, e.g., González-Gil et al. (2013); Scheepmaker et al. (2017).

1.4 Included Papers

This thesis collects our relevant papers that contribute to rtEETC (Section 1.2.1) and rtEETT (Section 1.2.2) fields. Such papers are:

Paper-1 Cacchiani, V., Di Carmine, A., Lanza, G., Monaci, M., Naldini, F., Prezioso, L., Suffritti, R., and Vigo, D. (2019). Energy-efficient train control. In *Advances in Optimization and Decision Science for Society, Services and Enterprises*, pages 57–68. Springer

Paper-2 Naldini, F., Pellegrini, P., and Rodriguez, J. (2021). Real-time optimization of energy consumption in railway networks. *Transportation Research Procedia. Accepted, in press.*

Paper-3 Naldini, F., Pellegrini, P., and Rodriguez, J. (TBD). Energy optimization of multiple trains in real-time rail traffic management considering signals and the interlocking system. *Under review for a major journal.*

The papers are available in Chapters 3 (Paper-1), 4 (Paper-2) & 5 (Paper-3). Their abstracts are reported in Sections 1.4.1, 1.4.2 & 1.4.3.

1.4.1 Abstract of Paper-1

This research presents a practical application of the *Energy Efficient Train Control* (EETC) problem, which involves a collaboration between the Operations Research group of the University of Bologna and ALSTOM Ferroviaria Spa. The work is carried out within the framework of project *Swift*, funded by the Emilia-Romagna regional authority. Given a train running on a certain line, the problem requires to determine a speed profile that minimizes the traction energy consumption. In particular, we consider the setting of a real-time application, in which the speed profile has to be recomputed due to changes in the schedule caused by unpredictable events. We introduce three solution methods: a constructive heuristic, a multi-start randomized constructive heuristic, and a Genetic Algorithm. Numerical experiments are performed on

real-life instances. The results show that high quality solutions are produced and that the computing time is suitable for real-time applications.

1.4.2 Abstract of Paper-2

In railway traffic, perturbations may give rise to conflicts, causing delays. As a countermeasure, effective re-scheduling and re-routing decisions can be taken by addressing the real-time Rail Traffic Management Problem (rtRTMP). One of its subproblems is the real-time Energy Consumption Minimization Problem (rtECMP). The latter enforces the train routing and precedences computed by a rtRTMP solver and defines train timings and speed profiles. The objective is to minimize the weighted sum of train energy consumption and total delay. In this paper, we propose an Ant Colony Optimization algorithm for the rtECMP and we test it on the French Pierrefitte-Gonesse control area with dense mixed traffic. The results show that, in a very short computing time, a remarkable exploration of the search space is performed before convergence.

1.4.3 Abstract of Paper-3

Rail is widely recognized to be one of the most sustainable transport mode. However, its efficiency is often limited by the fact that perturbation-induced delays can propagate in a snow-ball effect. This propagation can be mitigated with sensible train re-scheduling and re-routing decisions, i.e., facing the real-time Railway Traffic Management Problem (rtRTMP). To optimize energy consumption and hence preserve sustainability, we consider a subproblem of the rtRTMP named the real-time Energy Consumption Minimization Problem (rtECMP). The rtECMP decides upon the speed profiles and timings for multiple interacting trains running in a network to minimize the weighted sum of total energy consumption and delay. In this paper, we propose a graph-based model for the rtECMP, considering detailed interlocking and signaling systems. Moreover, we propose three meta-heuristic algorithms to deal with it: Ant Colony Optimization (ACO); Iterated Local Search (ILS); Greedy Randomized Adaptive Search Procedure (GRASP). We perform experimental analyses on two French infrastructures with dense mixed traffic. High quality solutions are achieved in a very

short computing time. The ACO algorithm achieves the best performance, with rather small but statistically significant differences.

1.5 Contributions

In the included Paper-1 (Chapter 3), we contribute to the rtEETC research. Precisely, we propose fast and accurate heuristic and meta-heuristic algorithms to compute energy-efficient single-train speed profiles with a given schedule. In the considered setting, a train travel a line with distance varying-speed restrictions. Both track and rolling-stock features affect the motion of the train. Experiments are run on case studies built with real-life rolling stock, schedule and line data, provided by our industrial partner, Alstom ferroviaria SpA.

The main contributions of this paper w.r.t. the related literature at the time it was being written are the following:

- It addresses an application critical to *Alstom*. The model, algorithms and solution quality assessment are in line with their industrial needs;
- It addresses new real-life case studies, including high-speed trains;
- It adopts an original fast constructive heuristic for generating speed profiles;
- Two of the proposed algorithms builds on top of the constructive heuristic: a multi-start and a genetic algorithms.
- For increased accuracy of train dynamics model, an actual *braking curve* is adopted instead of the more commonly used uniform deceleration rate;
- Computing time is suitable for real-time traffic management applications.

In the included Paper-2 & Paper-3 (Chapter 4 & Chapter 5), we contribute to rtEETT research. Precisely, we propose a graph-based model and fast and accurate heuristic & meta-heuristic algorithms for the rtECMP. Complex railway network and their infrastructural characteristics are considered. Both track and rolling-stock features influence speed profiles. In the route of each train, distance-varying speed

restrictions are to be enforced. Experiments are run on case studies built with real-life rolling stock, timetable and line data. Two real life infrastructures with mixed and dense traffic are used as case studies. Train movements are realistically operated in a microscopically detailed network in which the interlocking system and signals is are modeled. For details on the microscopic level of detail in representing rail infrastructures, on the interlocking and on signals refer to Section 2.3.1.1. Both Paper-2 and Paper-3 (Chapter 4 & Chapter 5) extend the work of Montrone et al. (2018), in which the rtECMP was first introduced.

Paper-2 (Chapter 4) proposes an novel graph-based model for the rtECMP, capable of dealing with additional constraints, and it introduces an Ant Colony Optimization algorithm to address the model.

The main contributions of this paper w.r.t. the related literature at the time it was being written are the following:

- A novel graph representation is introduced for the rtECMP, which extends the MILP formulation of Montrone et al. (2018) by supporting more complex signaling systems.
- A real-life case study is addressed: the Pierrefitte-Gonesse junction. It is a complex junction with a quite intricate layout. Many potentially conflicting train routes are available. Each instance derived from the Pierrefitte-Gonesse infrastructure includes either 15 or 16 trains (high-speed, regional, intercity and freight) over a 1-hour time window at peak-hour. The rtECMP instances that we derived from the traffic in Pierrefitte-Gonesse are larger w.r.t. those addressed the literature of the time.
- The proposed graph model is tackled effectively using an Ant Colony Optimization algorithm (ACO). To the best of our knowledge, this is the first application of ACO to the rtECMP.
- Computing time is suitable for real-time traffic management applications.

Paper-3 (Chapter 5) extends Paper-2 (Chapter 4) in several ways: the rtECMP graph-based model definition is slightly improved; a highly effective Local Search is

devised; two additional meta-heuristic algorithms are proposed along with the ACO of Paper-2; a new way of initializing the ACO is adopted; a thorough experimental analysis is performed considering also railway oriented indicators; a comparison of the three meta-heuristics is made; a statistical significance analysis of the experiment results is carried out; a new real-life case study, leading to very large instances, is considered.

The main contribution of Paper-3 (Chapter 5) w.r.t. the related literature at the time it was being written are the following:

- Besides the Pierrefitte-Gonesse Junction, an additional real-life case study is considered: a section of the Paris-Le Havre line. It is a long bi-directional corridor with several stations, some of which with multiple stopping platforms. Each instance includes either 15 or 16 trains (high-speed, regional, intercity and freight) over a 1-hour time window at peak-hour, when traffic is most dense.
- The rtECMP is tackled with Iterated Local Search (ILS) and Greedy Randomized Constructive Procedure (GRASP). To the best of our knowledge, this is the first application of such methods to the rtECMP.
- An effective Local Search algorithm is developed, and shared among the three algorithms.
- Computing time is suitable for real-time traffic management applications.

1.5.1 Overview of the papers

As discussed in Section 1.2, the papers included in this thesis consider two related problems, one of the rtEETC class (Paper-1: Chapter 3), the other of the rtEETT one (Paper-2 & Paper-3: Chapters 4 & 5, respectively). Despite the similarities, different methods and modeling assumptions are adopted. In this section, we summarize their main similarities and differences.

Let us introduce the following features for classifying the papers included in this thesis:

AT1 computing time is suitable for real-time rail traffic management;

AT2 multiple interacting trains running in a network are handled;

AT3 microscopic details of rail infrastructures are considered (see Section 2.3.1.1);

AT4 scheduling/timetabling decisions are taken with EETC in mind;

AT5 realistic case studies are tractable;

AT6 reduction of total delay is considered along with energy efficiency.

Using the aforementioned features, we summarize the content of each paper in Table 1.1.

	Paper-1 Chapter 3	Paper-2 Chapter 4	Paper-3 Chapter 5
AT1 (real-time)	✓	✓	✓
AT2 (multi-train interactions)		✓	✓
AT3 (microscopic detail)		✓	✓
AT4 (Scheduling/Timetabling)		✓	✓
AT5 (realistic instances)	✓	✓	✓
AT6 (delay minimization)		✓	✓

Table 1.1: Overview of the papers.

Table 1.2 summarizes the algorithms proposed in each included paper and the case studies dealt. The following notation is adopted: Ant Colony Optimization (ACO); Iterated Local Search (ILS), Greedy Randomized Adaptive Search Procedure (GRASP); Constructive Heuristic (CH); Multi-start Constructive Heuristic (MCH); Genetic Algorithm (GA); Local Search (LS). Line A & B stand for undisclosed actual lines, crucial to our industrial partner Alstom. The Pierrefitte-Gonesse Junction and Paris-Le Havre line are well-known infrastructures located in France, characterized by mixed dense traffic.

	Algorithm(s)	Case studies
Paper-1 (Chapter 3)	CH; MCH; GA	Line A, approx 260 km Line B, approx 270 km
Paper-2 (Chapter 4)	ACO	Pierrefitte-Gonesse Junction
Paper-3 (Chapter 5)	ACO; ILS; GRASP; LS	Pierrefitte-Gonesse Junction Paris-Le Havre line (a section)

Table 1.2: Proposed algorithms and tested case studies.

1.6 Structure of the thesis

The rest of this thesis is laid out in the following fashion:

- Chapter 2 provides a brief overview of the domain-specific knowledge of rail transport that are relevant to the included papers. First, the decision making process is declined in its time horizons: strategical, tactical and operational. For each planning horizon, the major optimization problems are discussed and literature references are provided. Second, it presents the modeling principles for rail infrastructures and speed profiles.

- Chapter 3 contains Paper-1 (see Section 1.4): "*Energy Efficient Train Control: a practical application*".

- Chapter 4 contains Paper-2 (see Section 1.4): "*Real-Time Optimization of Energy Consumption in Railway Networks*".

- Chapter 5 contains Paper-3 (see Section 1.4): "*Energy Optimization of Multiple Trains in real-time Rail Traffic Management considering Signals and the Interlocking System*".

- Chapter 6 reports our conclusions.

Chapter 2

Railway System

2.1 Introduction

In this chapter, we outline the main problems arising in the design, deployment and management of railway systems. These concerns planning decisions that have to be taken in three horizons: strategic, tactical and operational (Section 2.2).

In the rest of the Chapter (Section 2.3), we outline the modeling principles for two important areas: the railway infrastructure (microscopic and macroscopic levels) and train movements (in terms of train dynamics and energy consumption computation).

2.2 Decision Making Process

In this section, we outline the decisions involved in railway service planning, i.e., the design, deployment and management of railway systems. As for most public transport system, these activities are complex and costly. Railway systems, in particular, are not only extremely extended in terms of surface area, but also characterized by extremely high volume of transported goods and passengers. In addition, high reliability and safety standards have to be enforced through accurate control and regulation. This is due to the great number of vehicles, moving at considerable speed on a shared railroad network. Moreover, human actors, continuously entering and leaving the system, are a critical liability to be addressed. Therefore, numerous components, either static

or moving, must be effectively orchestrated on a daily basis also through accurate planning.

Transportation sector is becoming more and more competitive and, therefore, a high level of efficiency must be achieved and sustained through sensible railway service planning decisions. Attention to performance-critical indicators such as, e.g., punctuality, is tightly correlated to customer satisfaction and is a requirement for a state-of-the-art integrated, multi-modal transportation system. Moreover, due to the large infrastructural and rolling-stock related investment, railway service planning must also strive to maximize the utilization of the available capacity to increase profit.

Commonly, the planning process involves the interaction between two types of actor: the *infrastructure managers* (IM) and the *railway undertakings* (RU). An IM is usually a firm in charge of an infrastructure in terms of maintenance, traffic management, signaling and control. A RU is usually a public or private provider of transportation services on rail for goods and/or passengers.

In its modern envisioning, the planning process is split into three phases: strategic (Section 2.2.1), tactical (Section 2.2.2), operational (Section 2.2.3). The strategic level mostly concerns demand forecasting and procurement decisions, i.e., securing the acquisition of the required resources, such as, e.g., purchasing enough rolling stock to satisfy the demand forecast. The tactical level is centered around resource allocation. The operational level focuses on the day-by-day utilization of the resources in order to achieve and maintain a high level of service throughout the year.

A major output of the service planning process is the so-called *annual railway service plan*. It consists of a document with multiple sections, each describing the plan for different aspects of the railway system. The three planning levels mentioned above can be related to the start date of the annual rail plan and the so-called day of operations. In particular, the strategic level covers a horizon of 5 to 15 years before the start date of the annual plan; the tactical level covers a horizon of 5 years before the start of the annual plan; the operational level focuses on the development of the annual plan.

In the rest of this section, we offer an overview of the main decision problems arising in railway service planning at each horizon level. Precisely, Section 2.2.1, 2.2.2 and

2.2.3 detail the strategic, tactical and operational horizons respectively. Figure 2.1 summarizes these optimization problems by classifying them by time horizon (strategic, tactical and operational) and by decision maker (IM, RU or both). It also expresses the nominal flow of activities in the form of arrows. Such activities, in real applications, may be subject to significant variations. In addition, backtracking or concurrent solution of more than on problem may be necessary since problems of different levels and/or decision can be mutually dependent to some extent.

2.2.1 Strategic

As shown in Figure 2.1, two main problems concern the strategic level: Network design and Line planning. These are detailed below.

Network Design This problem involves making plans for building or altering pre-existing infrastructures. Usually, this kind of decisions is in charge to an IM in cooperation with the public authorities. However, RUs can take part in the process as well and provide useful insights on customer behavior and demand forecasting. The deployment of a new infrastructure from scratch or the alteration of an existing one are usually motivated by significant changes in travel needs, increased demand, or a desire to innovate or adhere to improved standards. For further details, we refer the reader to Magnanti and Wong (1984), Guihaire and Hao (2008) and Canca et al. (2019).

Line Planning This problem requires defining train lines on an existing network, which can be given by addressing a network design model. Precisely, a line is an itinerary that connects an origin-destination pair, usually two major stations of interest. Such itinerary potentially includes several minor stations in between. A line plan must also specify the stop pattern, the type of rolling stock (engine model, number of wagons, etc.) and the frequency of the line itself. Usually, the objective is satisfying passenger demand while complying with network characteristics and capacity limits. This problem is primarily faced by the RUs, which, however, may benefit from a collaboration with the IM and, to some extent, with the regional or national transportation authorities. For further details, we refer the reader to Goossens et al.

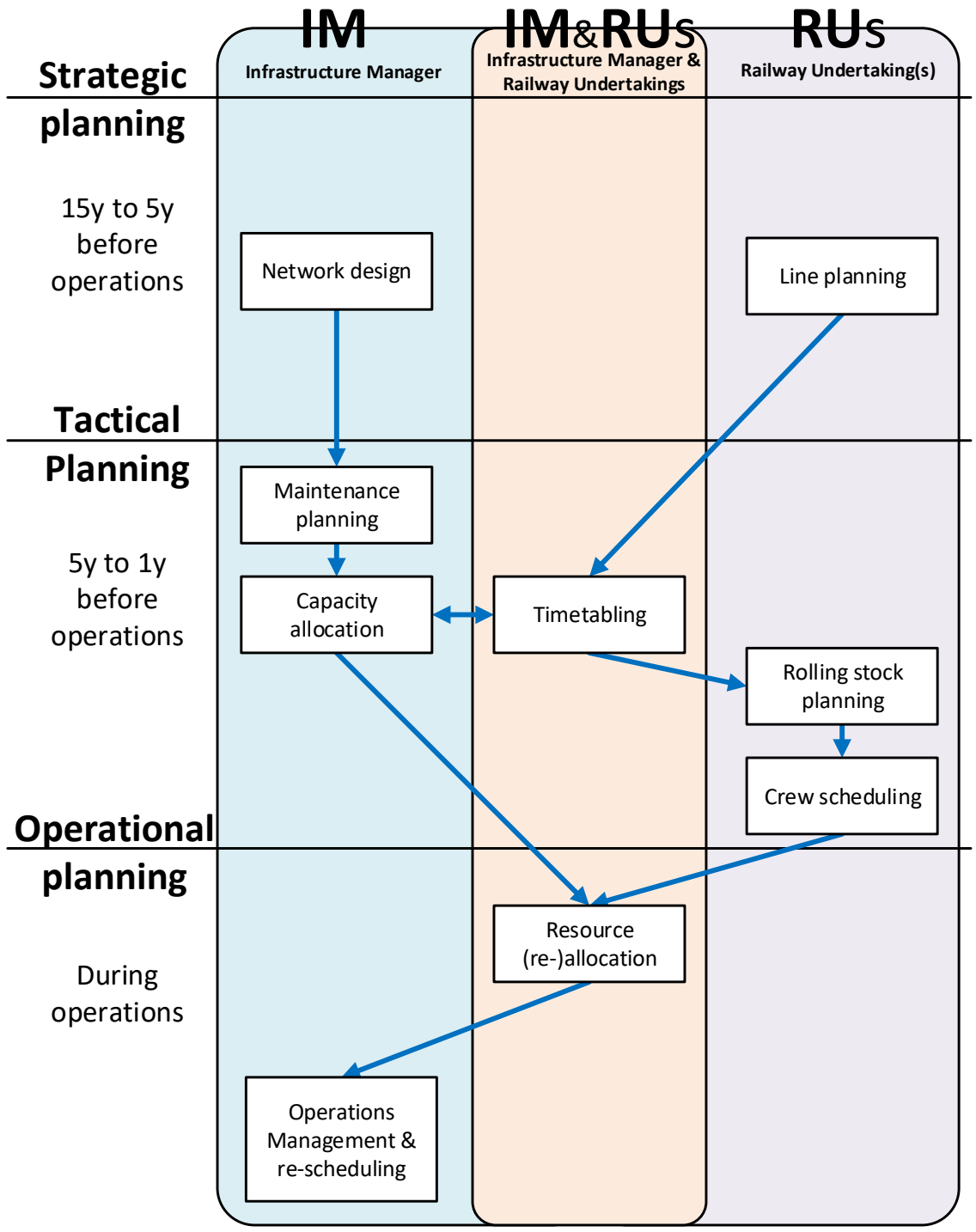


Figure 2.1: Railway planning decisions: strategic, tactical & operational horizons. The rows indicate each planning phase. The columns indicate each actor involved in the decisions.

(2006), Schöbel (2012) and Fu et al. (2015).

2.2.2 Tactical

As shown in Figure 2.1, five main problems concern the tactical level: Maintenance planning, Timetabling, Station capacity allocation, Rolling stock planning, Crew scheduling. These are detailed below.

Maintenance planning This problem is usually addressed by the IM, who is required to prepare a maintenance schedule matching the networks needs. When maintenance operations are performed, portions of the network may become unavailable. This may greatly affect train operations and, therefore, the maintenance plan should be known prior to making timetabling decisions. For further details, refer to Higgins (1998), Caprara et al. (2006) and Forsgren et al. (2013).

Timetabling At the tactical level of railway service planning the IM has to define a timetable to regulate traffic circulation on the network in terms of arrival/departure times at points of interest along each line. The timetable therefore expresses the nominal running time for each train and the dwelling times at each planned stop. The IM receives train path and timing requests from several RUs and, based on these, he constructs the timetable trying to accommodate each request as much as possible. The requests are generally mutually incompatible and a compromise has to be reached. Several constraints might be taken into account such as, e.g., headway times, speed limits, and connections between services. A tipical objective is minimizing the deviation w.r.t. to the RU requests. This problem, which is also known as train scheduling, is widely studied in the literature. See, e.g, Cordeau et al. (1998), Cacchiani and Toth (2012) and Goverde et al. (2016).

Capacity allocation This problem requires to define a capacity allocation plan to ensure the fruition of all services scheduled in the timetable. Such plan assigns routes (given the actual infrastructure layout) and station platforms in such a way that each train schedule is fulfilled. The capacity allocation problem is in typically in charge to the IM. However, in case no feasible allocation is possible, the IM interacts

with the RUs to re-negotiate their requests. In literature, this problems is referred with several names, such as, train routing problem, track allocation problem, train path allocation problem and train platforming problem. For further details, see, e.g, Caprara et al. (2011), Lusby et al. (2011) and Borndörfer et al. (2012).

Rolling stock planning This problem requires the computation of the minimum-cost assignment between rolling stock and the services scheduled in the timetable. Empty runs, shunting movements and vehicle unavailability due to rolling stock maintenance are also considered. Usually, this kind of planning aims at reducing the overall number of vehicles employed as it is strongly correlated with the total cost of meeting the timetabled services. RUs are in charge of producing the rolling stock plan. For further details, see, e.g., Abbink et al. (2004), Fioole et al. (2006), Steinzen et al. (2010) and Cacchiani et al. (2012).

Crew scheduling This problem requires generating cost-efficient duty combinations for railway crews to cover all the scheduled trains over a time horizon. The resulting crew schedules must comply with regulations and operational requirements. Each RUs is in charge of scheduling its own crew. Relevant contributions are, e.g., Caprara et al. (1997), Abbink et al. (2005), Jütte and Thonemann (2012) and Heil et al. (2020).

2.2.3 Operational

As shown in Figure 2.1, two problems concern the strategic level: Resource (re-) Allocation and Operations Management and Rescheduling. These are detailed below.

Pre-operations resource (re)allocation The need of (re)allocating resources at the operational level may occur for several reasons. For instance, additional freight rains due to extra convoys, unplanned maintenance operations due to vehicle malfunctions, unforeseen variations in crew rotations due, e.g., to strikes, etc. Usually the time horizon for dealing with such unforeseen disruptions is of a few days. When (re)allocation needs concern railway capacity, as for unplanned maintenance actions or requests for additional train services, the IM is in charge of it. When the issue lies,

instead, on resources such as rolling stock or crew members, the RUs are responsible of dealing with the (re)allocation needs. For further details, see, e.g., Jütte and Thonemann (2012), Chu and Oetting (2013).

Operations management & re-scheduling Traffic must be overseen, coordinated and managed in real time, because railway operations are subject to unforeseen events that might negatively impact circulation (such as hardware failure, human errors, forcast errors, etc.). Precisely, traffic perturbations cause so-called conflicts w.r.t. the nominal schedule. A conflict is an event where one or more trains are forced to slow down or even stop due to other trains no longer adhering to their original schedule. This causes delays that can propagate to the whole system hindering service. As a countermeasure, several train re-routing and re-scheduling decisions must be made rapidly and sensibly by the IM. If addressed quantitatively, this means solving the so-called real-time Rail Traffic Management Problem (rtRTMP), an NP-hard combinatiorial optimization problem. See, e.g., D'Ariano et al. (2007), Corman et al. (2011) and Pellegrini et al. (2014). The rtRTMP can be considered a real-time version of the timetabling problem (Section 2.2.2). The output of a rtRTMP solver is, indeed, a whole new train schedule implying a new timetable: arrival/departure times, passing through times and dwelling times (for stops) at points of interest. Typically, the rtRTMP is modeled with a highly detailed representation of the railway networks to accurately schedule each train on the available physical routes.

The aforementioned papers refer to frequent, low-magnitude traffic perturbations. A related research branch deals with infrequent, very large perturbations (disruptions), that may hinder circulation for up to several hours. In this case, other traffic management decisions are needed, e.g., short-turning (Zhan et al., 2015), rolling-stock re-allocation (Nielsen et al., 2012) and crew rescheduling (Veelenturf et al., 2017). Some of these may involve negotiation between RUs and the IM. For further details on disruption management, see also Jespersen-Groth et al. (2009), Dollevoet et al. (2012), Corman et al. (2017) and Hong et al. (2021). For a review on the rtRTMP and disruption management, see Cacchiani et al. (2014).

2.3 Modeling principles

This section briefly outlines the modeling principles that are relevant to the railway optimization problems considered in this thesis. First, the railway infrastructure and its modeling principles are discussed in Section 2.3.1. Second, train speed profile and energy consumption modeling is addressed in Section 2.3.2.

2.3.1 Infrastructure

The railway infrastructure is a complex system including a network of tracks on which train movements occur. Tracks guide a train motion so that no active steering is required by means of a standardized wheel-flange interface. Tracks intersect at so-called *junctions*, some of which are stations and, hence, passenger loading/unloading may occur there. At junctions, switches (points) and crossings are located where one or more tracks intersect. Both switches and crossings are mechanical devices that are used to physically configure tracks according to each train's scheduled route. At a switch/crossing, each train is guided in the proper direction, according to plans, preventing them from accessing the wrong track portions.

In the literature, railway infrastructures are represented with different levels of detail commonly known as: macroscopic (least detailed); microscopic (most detailed, closely represents real-life railway infrastructures); and mesoscopic (in-between microscopic and macroscopic) model. The two extremes, microscopic and macroscopic, are outlined in this section as they are relevant to the papers collected in this thesis. Microscopic modeling of railway infrastructure is introduced in Section 2.3.1.1. This level of detail is is used in Paper-2 and Paper-3 (Chapters 4 and 5). Microscopic modeling of railway infrastructure is discussed in Section 2.3.1.2. This level of detail is used in Paper-1 (Chapter 3).

2.3.1.1 Microscopic representation

In this section, we offer a brief overview of the real-life railway system aspects that are usually included in the microscopic modeling of railway infrastructures. At this

level of detail, the following infrastructural elements are usually considered: track-circuits, block sections, signaling and interlocking systems. This allows for an accurate descriptions of train movements while ensuring their feasibility w.r.t. rail operational principles. These elements are discussed in the following and examples are provided.

For real-time traffic management purposes, where the microscopic representation is the most suitable (Pellegrini et al., 2014), the railway system is divided into portions of network called *control areas*. Those are in charge to a *dispatcher* or controller, who has to actively maintain a smooth traffic flow so that timetabled target times are met by all trains as much as possible. A control area contains a set of possible origin-destination pairs and many potential train routes connecting them. To overcome delay caused by perturbations arising during train operations, a dispatcher can make the following decisions: train re-routing (re-assigning each train a route connecting its origin-destination pair), re-ordering (deciding the sequence of trains traversing a given route) and re-timing (deciding train travel times). Such decisions must be implemented as quickly as possible. For additional details on real-time traffic management, the interested reader is referred to Section 2.2.3.

In the infrastructure contained in a control area, tracks are divided into *track-circuits*: track stretches equipped with electrical devices capable of sensing train presence. For our purposes, track-circuits can be considered both fundamental and atomic resources of railway networks. Indeed, utilization time of track-circuits by each train have to be properly scheduled onto track-circuits to let traffic flow safely and in compliance with the timetable. A track-circuit must be utilized by only one train at a time for safety.

Track circuits are grouped into *block sections*, that are track sections delimited by two signals: one at the entrance and one at the exit. For safety, only one train at a time can be in a block section. A *signal* protects the entrance of its corresponding block section by conveying instructions affecting the pace of trains approaching the block section itself: a speed reduction or a full stop may be demanded. Each train in a control area is assigned a route that it must follow. A route is an ordered sequence of block sections and, therefore, an ordered sequence of track-circuits for a train to traverse.

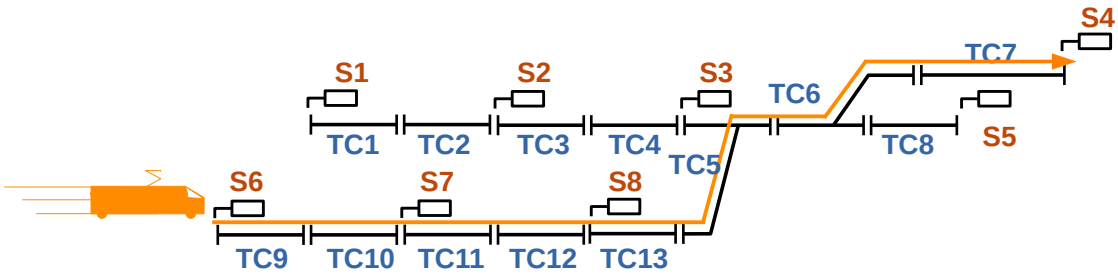


Figure 2.2: Example railway infrastructure.

Figure 2.2 displays an example infrastructure including: track-circuits TC1 to TC13; signals S1 to S8. The signals delimit the following block sections: S1-S2 (containing TC1 and TC2); S2-S3 (containing TC3 and TC4); S3-S4 (containing TC5, TC6 and TC7); S3-S5 (containing TC5, TC6 and TC8); S6-S7 (containing TC11 and TC12); S8-S4 (containing TC13, TC5, TC6 and TC7); S8-S5 (containing TC13, TC5, TC6 and TC8). A possible train route is displayed as an orange arrow. The route is composed of block sections S6-S7, S7-S8, S8-S4 (composed of track-circuits TC9, TC10, TC11, TC12, TC13, TC5, TC6, TC7).

Railway signals are design to maintain a safe distance between trains circulating in a control area. Indeed, a signal is in essence a traffic light that conveys instructions on whether a block section can be accessed or not by an approaching train. Moreover, when a block section can be accessed, its entrance signal may impose a speed restriction to a train approaching it if needed. The specific instruction displayed by a signal to an approaching train depends on the utilization state of the track-circuits belonging to the block sections following the one in which the train is currently located.

Let us consider a train approaching a signal while traveling on its planned route. In the simplest case (a 3-aspect signaling system), the signal can display three colored aspects to the train approaching it: green, yellow, red. Red indicates that the block section after the signal must not be accessed, being already under utilization, or the emergency brake will be triggered. Yellow indicates that the train must proceed with caution. In fact, if the traffic state does not improve, the train will have to stop before the next signal. Therefore, yellow may require the train to slow down as soon

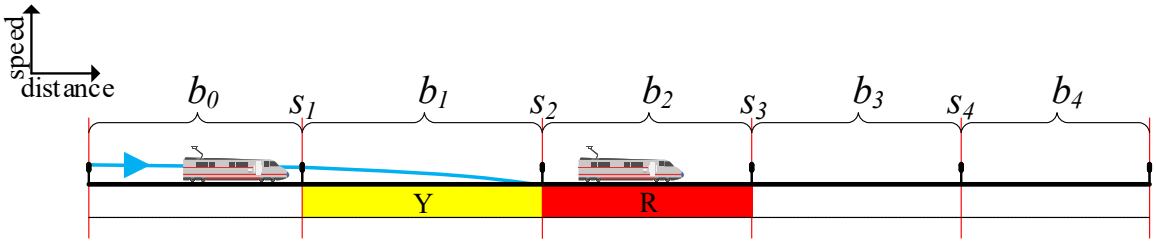


Figure 2.3: Example: the train on the left meets a yellow (Y) on a 3-aspect signal (s_1). Track-circuits are not shown here.

as possible. The yellow aspect is triggered by the the next signal being red. Green indicates that the train should proceed at the planned speed. For a 3-aspect signal to display a green aspect, the next two block sections must be available, i.e. every track-circuit in them must not be utilized by any other train.

Figure 2.3 exemplifies the functioning of a 3-aspect signal. It shows two trains sharing the same route, composed of block sections $b_0, \ldots, b_4$ and signals $s_1, \ldots, s_4$. Signal s_1 is a 3-aspect signal. Due to the train on the right occupying block section b_2, the train on the left, which is about to enter b_1, meets a yellow (Y) aspect on s_1. This instructs the train to prepare for a possible stop within the length of block section b_1 in order to maintain safety distances. This is due to the aspect displayed on s_2 being red (R) as b_2 is physically occupied by another train. The blue curve represents one of many possible speed profiles for the train on the left complying with the speed restriction imposed by signal s_1.

Theoretically, each signal can have any number n of aspects, including green, yellow and red. In case of n-aspect signals, additional restrictive aspects, between green and red, are used to instruct drivers to gradually reduce speed, if needed. This progression of speed restrictions mediated by signals is conceived to enable a train to stop before a red signal that is located farther ahead on the route. For a signal to display a green aspect, the next $(n-1)$ block sections must be available. For instance, a 4-aspect signal, can display the following aspects: green, flashing-yellow, yellow, red. Green is displayed only when the next three block sections are available. Flashing-yellow warns of a possible requirement to stop within two block sections. Yellow and red meaning is unvaried.

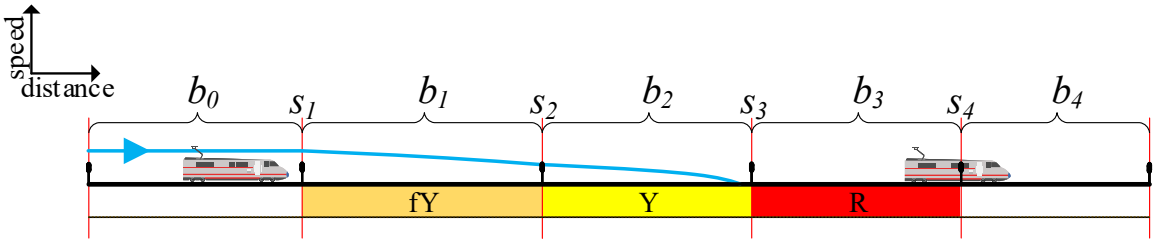


Figure 2.4: Example: the train on the left meets a flashing-Yellow (fY) on a 4-aspect signal (s_1). Track-circuits are not shown here.

Figure 2.4 exemplifies the functioning of a quite common 4-aspect signaling system. It shows two trains sharing the same route, composed of block sections $b_0, \ldots, b_4$ and signals $s_1, \ldots, s_4$. Signal s_1 is a 4-aspect signal. This time, due to the train on the right occupying both block section b_3 and b_4, the train on the left, which is about to enter b_1, meets a flashing-yellow (fY) aspect on s_1. This aspect tells the driver that a full stop could be required within two block sections, i.e., by the end of b_2 and, therefore, the speed of the train on the left, entering b_1, must be adjusted accordingly. If, for instance, the train on the right stood still, the train on the left would eventually meet a yellow aspect on s_2 and a red one on s_3. The blue curve represents one of many possible speed profiles for the train on the left to fulfill the displayed aspects on s_1, s_2 and s_3.

Another fundamental infrastructure component is the *interlocking system* that manages the proper train access to track-circuits and block sections for a safe and coordinated circulation. It is in charge of handling the reservation, set up and release of track-circuits so that each train route is formed as planned and traversed only when it is safe to do so. The most basic rule the interlocking system enforces is that only one train at a time can utilize the same track-circuit. Two block sections sharing one or more track-circuits are called *incompatible* (with this definition, block sections are considered incompatible with themselves). Two train routes sharing at least one track-circuit are said to be *conflicting*. Figure 2.5 shows two conflicting routes assigned to two distinct trains (orange and green arrows) located in the example infrastructure of Figure 2.2. The green route is composed of block sections: S1-S2; S2-S3; S3-S5. The

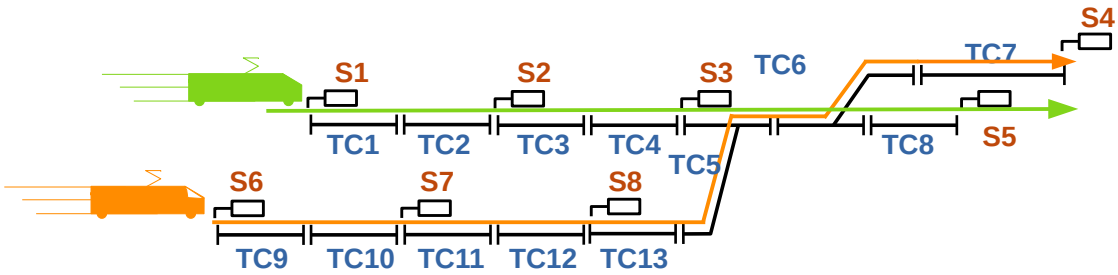


Figure 2.5: Example infrastructure with two trains having conflicting routes.

orange route is composed of block sections: S6-S7; S7-S8; S8-S4. Block sections S3-S5 (green route) and S8-S4 (orange route) are incompatible as they share track-circuits TC5 and TC6.

The interlocking system regulates the flow of trains in incompatible block sections so that the safety rules are fulfilled and proper route formation is ensured. To achieve its purpose, the interlocking system continuously coordinates the signals with the actual utilization of track-circuits in the control area. This requires exploiting the train detection capabilities of track-circuits. With a red signal at the entrance of a block section b, the interlocking system states that it is not safe for a train t to proceed in it, as one or more track-circuits are currently under utilization in b. Moreover, the block sections b^* that are incompatible with b are made unavailable by the interlocking system. Eventually, block section b is made available again as soon as its track-circuits are cleared and released. This implies that not only the train's tail has cleared b, but also that enough time has elapsed for it to be considered released due to technical reasons. Block sections b^*, instead, depending on the interlocking system type, can be released even before that b becomes available again.

Figure 2.6 displays the routes and positions of two trains, A and B, on a simple infrastructure. The route of B is depicted in orange and includes block sections S1-S2, S2-S3, S3-S4. The route of A is depicted in blue and includes block sections S5-S6, S6-S7, S7-S8. We assume a 3-aspect signaling system and that only trains A and B are circulating in the infrastructure. In Figure 2.6, train B is currently occupying a track-circuit that is shared between block sections S2-S3 and S6-S7, that are incompatible

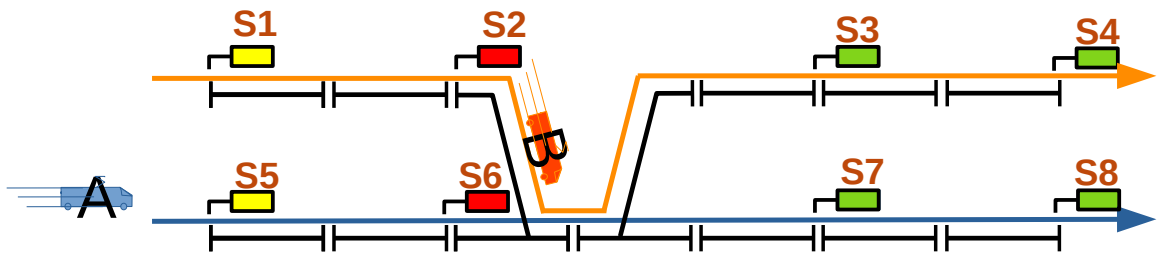


Figure 2.6: Example of *route-lock* and 3-aspect signaling. Train A: blue route. Train B: orange route.

block sections. As a result, train A is denied access to S6-S7. Indeed, S6 displays a red aspect for A. In addition, train A will encounter a yellow aspect on S5 when approaching S5-S6, due to S5 being a 3-aspect signal. Signals S1 and S2 respectively display yellow and red as S2-S3 is still under utilization.

Two interlocking system types are available: *route-lock route-release* and *route-lock sectional-release*. Both systems have the same *route-lock* functioning, in which, as discussed, a block section is considered inaccessible until each track-circuit in it is no longer utilized (c.f. Figure 2.6). What differentiates the two interlocking system types is the policy with which incompatible block sections are made available as shared track-circuits are progressively released.

With the route-lock route-release interlocking system, all the track-circuits in a block section are released only after that a train has completed their utilization. Then, a new train can reserve the track-circuits in the block section if applicable. If this kind of interlocking is deployed, track-circuit utilization details can be neglected and block sections become the fundamental resources onto which train utilization time is allocated. However, groups of mutually incompatible block sections must be know and this information can be derived only from the infrastructural layout at the track-circuit level of detail. A qualitative example of route-lock route-release is presented in Figure 2.7 in a similar setting to Figure 2.6. Here, train B is located in S3-S4. Assuming that enough time has elapsed to consider S2-S3 released and S6-S7 set, train A can access block section S6-S7 with no restrictions. This scenario is reflected

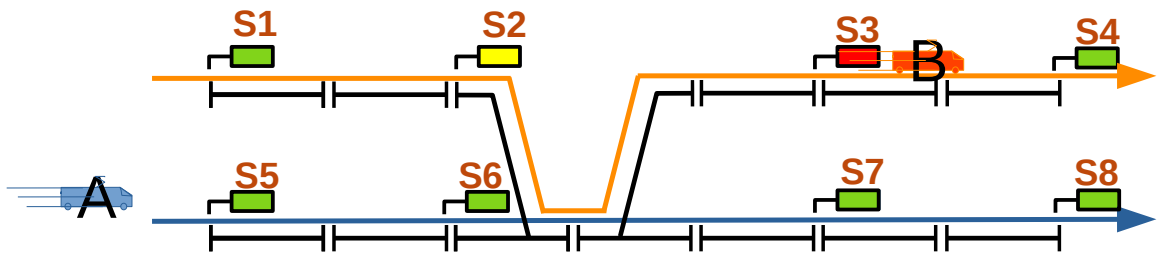


Figure 2.7: Example of *route-lock route-release* with 3-aspect signals.

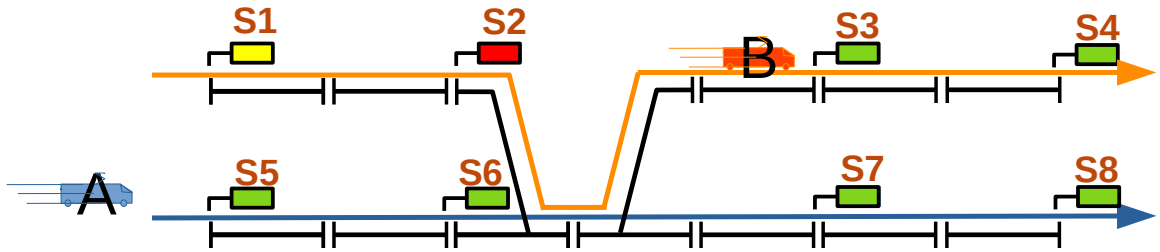


Figure 2.8: Example of *route-lock sectional-release with 3-aspect signals.*

in the signal aspect displayed: S2 is yellow, S3 is red, all the others are green.

With the route-lock sectional-release interlocking system, the track-circuits in a block section b that are being traversed by a train t are released as soon as their utilization by t itself ceases. In this way, such track-circuits can be made available to other trains scheduled to traverse block sections other than b. Indeed, b remains inaccessible until its last track-circuit has been released by train t. This kind of interlocking requires knowing the utilization time of track-circuits by trains and can yield a better saturation of the infrastructure's capacity. A qualitative example of route-lock sectional-release is presented in Figure 2.8 in a similar setting to Figure 2.6. Here, train B is still located in S2-S3. However, it is now located on the last track-circuit, right before signal S3. Assuming that enough time has elapsed to consider the other track-circuits in S2-S3 released and S6-S7 set, train A can access block section S6-S7, even if B has not released the last track-circuit in S2-S3 yet. This scenario is reflected in the signal aspect displayed: S1 is yellow, S2 is red, all the others are green.

Generally, implementing train movements in a control area requires complying with the constraints imposed by both the interlocking and signaling systems. As discussed so far, modeling this kind of constraints requires an accurate consideration of the utilization times of each track-circuit (and block sections) by each train. Generally, the utilization time does not coincide with exclusively the time a train physically occupies a track-circuit. The *Blocking theory* (Pachl, 2009) is often applied to determine the utilization time of track-circuits (and block sections), especially in the context of real-time traffic management (Pellegrini et al., 2014).

Figure 2.9 illustrates the utilization time resulting by applying the Blocking Theory to block section S8-S4 (track-circuits TC13, TC5, TC6, TC7). The horizontal and vertical axes indicate distance and time, respectively. The two orange oblique lines represent the train's head and tail distance traveled as a function of time. A route-lock sectional-release interlocking and a 3-aspect signaling systems are assumed to be in place. For each track-circuit, the utilization time equals the sum of the following time intervals:

1. *formation time*: it is the sums of (i) the time to set-up a block section (in terms of configuring its moving elements, switches and crossings, and its entry signal) and (ii) the time needed by the driver to see and react to the signal aspect at the block section's entrance;

2. *reservation time*: the time required by the train to traverse the preceding block section where the reservation of track-circuits begins to ensure that a green aspect is encountered on S8 (in case of 3-aspect signaling);

3. *running time*: the time it takes the train's head to traverse a track-circuit, which is tightly related to the chosen speed profile;

4. *clearing time*: the time it takes the train's tail to leave a track-circuit, which is tightly related to the chosen speed profile;

5. *release time*: the time that must elapse to make a track-circuit available again, for technical reasons, after it has been cleared by the train.

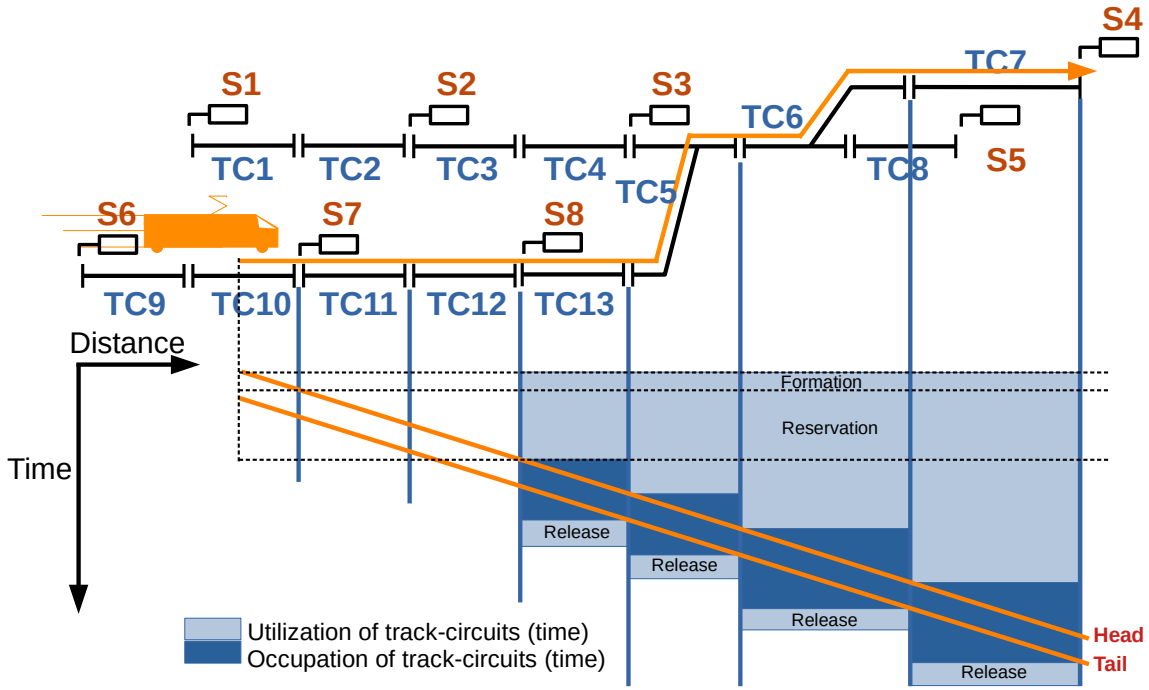


Figure 2.9: Utilization of block section S8-S4 by a train traveling the orange route. Route-lock sectional-release interlocking. Reservation time is considered (3-aspect signaling).

Each clear blue rectangle in Figure 2.9 represents the utilization time of each track-circuit by a train. The dark blue rectangles, contained in the clear blue ones, represent the physical occupation time of each track-circuit, i.e., running time plus clearing time. The utilization times of a track-circuit by two trains must never overlap.

Figure 2.9 reflects the utilization time definition of many real-time traffic management approaches. See, e.g., Pellegrini et al. (2014). In this setting, conflict-free schedules are directly sought. A schedule is said to be conflict-free if it implies that every train crosses exclusively green signals along its entire trip (if no traffic perturbations occur). To obtain conflict-free schedules, track-circuits must be reserved in advance as a train approaches a block section. In this setting, reservation time (Figure 2.9) becomes a part of utilization time. In case of n-aspect signals, the reservation of a track-circuit must start before a train occupies the first track-circuit in the $(n-2)^{th}$ preceding block section, as the formation time must elapse first. In the example of Figure 2.9, since we have $n = 3$, the reservation time concerns the previous block

section only ($n - 2 = 3 - 2 = 1$). In any case, the train can actually enter the block section only after the formation time has elapsed (Figure 2.9).

When obtaining a conflict-free schedule is not a hard constraint, the reservation of track-circuits can be neglected. However the other components of utilization times must still be computed. This is the case considered by Montrone et al. (2018), Paper-2 and Paper-3 (Chapters 4 and 5). Figure 2.10 illustrates the components of track-circuit utilization time as it is modeled in Montrone et al. (2018), Paper-2 and Paper-3. For example purpose, the setting of Figure 2.9 is reprised. However, now, a route-lock route-release interlocking system is in adopted and reservation times are neglected. Utilization times and occupation times are represented as clear and dark green rectangles, respectively. As usual, the utilization times of a track-circuit by two trains must never overlap. As shown in Figure 2.10, the utilizations of TC13, TC5, TC6 and TC7 last the same amount of time in this case. Each track-circuit's utilization time is the sum of: formation time; release time; the time interval between the physical occupation of the first and the last track-circuits. For further details, see Montrone et al. (2018).

The interlocking system discussed so far reflects the specifics of the *fixed-block interlocking system* from the *European Rail Traffic Management System* standard (ERTMS, 2021), which is currently deployed in most control areas around the world. The specific signal aspects described in this section, although reflecting the French signaling system, are representative of most implementations worldwide. For further details on the interlocking and signaling systems, we refer the reader to Theeg et al. (2009). Additional details are also provided in Paper-2 and Paper-3 (Chapters 4 & 5).

2.3.1.2 Macroscopic representation

In this section, we outline of the so-called microscopic representation of railway systems. At this level of detail, infrastructural components such as track-circuits, block sections and the interlocking and signaling systems (Section 2.3.1.1) are neglected.

A macroscopic model usually consists of a graph whose nodes symbolize junctions or stations of interest and whose arcs represent existing rail connections between such

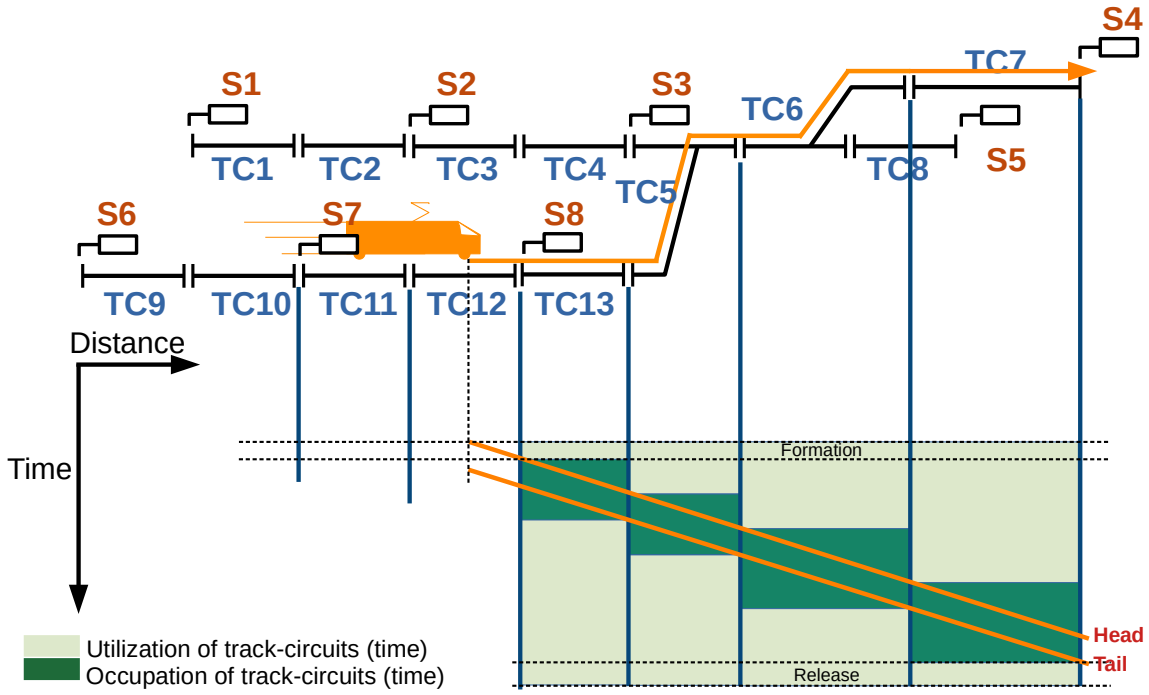


Figure 2.10: Utilization of block section S8-S4 by a train traveling the orange route. Route-lock route-release interlocking. Reservation time is neglected.

nodes. As a result, they are more natural and intuitive compared to microscopic models. In addition, they are much easier to obtain, not requiring describing every technical element composing the layout of an infrastructure.

In case of train scheduling, microscopic models account for only the total travel times of the trains running between each pair of nodes. Each node is often assigned a maximum and minimum number of trains that can be handled in terms of available capacity. In addition, train separation is enforced by means of minimum headway constraints. These impose a minimum distance (spatial or temporal) between two consecutive utilizations of an arc by two trains. In reality, though, as discussed in Section 2.3.1.1, train separation is dynamically managed through the interlocking system and signals.

The reduced level of detail implies that microscopic models, in most cases, allow for addressing very large rail networks preserving tractability. However, adopting macroscopic models yields a largely inaccurate representation of train movements as neither the actual track-circuits traversed nor the interlocking and signaling systems

are considered. Therefore, such models are best suited for those problems that do not require either infrastructural details or accurate modeling of train operations. However, due to the fact that microscopic models often enable tackling very large rail network instances, they are frequently used in several railway problems belonging to the strategic and tactical planning horizons (Section 2.2.1 and 2.2.2).

For instance, the Train Timetabling Problem of the tactical planning horizon (Section 2.2.2) is usually tackled with macroscopic models. This allows for rapid solutions of large railway networks. However, the obtained timetables might be either inefficient, in terms of infrastructure capacity utilization, or infeasible due to the violation of operational rules. In fact, for example, certain headway and running times may be either underestimated or overestimated by a timetable, leading to under-utilization of resources or to insufficient train separation and, hence, violation of safety rules. Moreover, in this context, the model of train motion is usually extremely simplified, i.e., the actual physical properties of the route, rolling-stock and, in particular, the engine characteristics are ignored. This, along with no consideration of signal and interlocking constraints, leads to infeasible timetables from a microscopic point of view, which, nonetheless, are good enough for the tactical planning horizon. Assessing the feasibility and capacity utilization that a timetable actually yields can be done ex-post, e.g, with a rail *simulator* running on the corresponding microscopic model for the considered network. This is to remark even further the importance of microscopic infrastructural details when seeking accurate train operations modeling.

In Paper-1 (Chapter 3), a microscopic model is adopted for a single-train single-line rtEETC application. The addressed problem calls for finding an energy-optimal speed profile for a train running between two consecutive stops in a given line (a train route). The line instances tackled are up to approximately 260km long. Each line is corresponds to a specific route between an initial and final station where the train respectively departs and arrives. The intermediate stations between the departure and arrival stations are neglected as intermediate stops are not planned in the considered instances. Naturally, track-circuits, block sections, the interlocking system and signals are neglected. This macroscopic model includes a description of the enforced speed

restriction, radius of curvature and slope as piece-wise functions of distance. Such line features are crucial for modeling train dynamics and, hence, computing speed profiles with accuracy (see Section 2.3.2). The so mentioned piece-wise functions are derived from the properties of each track-circuit composing the line. Similar models are used in many EETC approaches (Scheepmaker et al., 2017).

For details on track and rolling-stock physical features and their relationship with train speed profiles, see Section 2.3.2.

For further details on macroscopic models, see, e.g., Caprara et al. (2002) and Schlechte et al. (2011).

2.3.2 Train speed profiles & energy consumption

In this section, we provide a brief overview of the modeling principles adopted when an energy-efficient speed profile is sought for a single train. Such principles are relevant to most EETC approaches in the literature. Therefore, what we discuss in this section is also crucial for the EETT class of problems, being indeed tightly related to EETC. However, we remark that both EETC and EETT indicate two families of models and algorithms. Despite usually sharing similar modeling principles that are rooted in optimal control theory, EETC and EETT can take quite different forms depending on the specific context (Scheepmaker et al., 2017). Only the modeling aspects that are relevant to the papers collected in this thesis will be discussed here.

Let us focus on the motion of a single train whose state will be represented using speed and distance as dependent variables and time as the independent one. Although different variables can be chosen depending on the application, we believe this setting to be the most intuitive for the purpose of this section. To accurately represent the motion of a train one must consider characteristics of both the the train and infrastructure it traverses. First of all, a train is frequently approximated as a point mass and Newton's second law of motion is used to describe the relation between its speed v and time t (Brunger and Dahlhaus, 2009). The train motion can be described

through the following system of differential equations:

$$\begin{cases} F(t) - F_r(t) = m f_p \dfrac{dv(t)}{dt} \\ v(t) = \dfrac{ds(t)}{dt} \end{cases} \tag{2.1}$$

where

- $v(t)$ is the train speed at t
- $s(t)$ is the train distance along its route at t
- $F(t)$, depending on its sign, is either an accelerating or decelerating force whose magnitude and sign are under the control of the train driver at time instant t
- $F_r(t)$ is the resistance force at t;
- m is the train mass
- f_p is a correction factor to account for the rotating parts of the train

The engine of a train is capable of exerting a maximum force of $F_T(v) \geq 0$, that is a non-increasing function of speed v. In most cases, $F_T(v)$ is effectively approximated as a piece-wise hyperbolic function.

A maximum braking force $F_B < 0$ can be exerted to decelerate the train or to stop it. F_B is often approximated as a constant value and its most frequently used values are: *service braking* and *sharp braking* force (Brunger and Dahlhaus, 2009). Sharp braking yields a higher deceleration rate and is hence meant for emergencies, while service braking is preferred in regular operation.

The term $F(t)$, which is the controlled variable, is bounded between the maximum braking force $F_B < 0$ and the maximum tractive force $F_T(v(t)) > 0$.

To move, a train must overcome the resistance force F_r. A quite common definition for resistance F_r is the following:

$$F_r(t) = L_r(\theta(s(t))) + C_r(\rho(s(t))) + M_r(v(t)), \tag{2.2}$$

where

- $L_r(\theta(s(t)))$ is the *line resistance*, which depends on the track slope angle θ at distance $s(t)$ along the track;
- $C_r(\rho(s(t)))$ is the *curve resistance*, which depends on the radius of curvature ρ at distance $s(t)$ along the track;
- $M_r(v(t))$ is the *vehicle resistance*, which depends on train speed v at distance $s(t)$ along the track.

Line resistance (L_r) accounts for the effects of gravity on inclined track portions. It is computed as

$$L_r = m\ g\ sin(\theta) \approx m\ g\ tan(\theta), \tag{2.3}$$

where g is the standard gravitational acceleration for the surface of the earth. For small values of θ, we have that $sin\theta \approx tan\theta$.

Curve resistance (C_r) accounts for the resistive effects of track curvature on train motion. It can be computed with different formulas. A quite common one is the *Roeckl formula* (Wang et al., 2016):

$$C_r = \begin{cases} m\frac{6.3}{\rho-55}, & if \quad \rho \geq 300 \text{ meters} \\ m\frac{4.91}{\rho-30}, & if \quad \rho < 300 \text{ meters} \end{cases} \tag{2.4}$$

In Paper-2 and Paper-3 (Chapters 4 and 5) we adopted the Roeckl formula, while in Paper-1 (Chapter 3) we adopted a different formula that better matched the needs of our partner Alstom.

Vehicle resistance (M_r) combines both rolling resistance and air resistance. It depends on speed and on rolling stock features and is usually approximated with the *Davis formula* (J.W. Davis Jr, 1926). Precisely:

$$M_r(v) = A + Bv + Cv^2, \tag{2.5}$$

where A, B and C are non-negative empirical constants that depends on the specific rolling stock.

The values of θ and ρ are dependent only on the position $s(t)$ of the train along its route. In practice, $\rho(s)$ and $\theta(s)$ are usually defined piece-wise constant function of distance s along the considered route. This implies that a train route is comprised of a sequence of track intervals having a constant ρ and θ. In each of these intervals, F_r becomes a function of speed only. As a result, for some applications, it is often convenient to use distance s as the independent variable (Albrecht et al., 2016c). Knowing that $v = ds/dt$ by definition, (2.1) becomes:

$$\begin{cases} F(s) - F_r(s) = m f_p v(s) \dfrac{dv(s)}{ds} \\ \dfrac{dt(s)}{ds} = \dfrac{1}{v(s)} \end{cases} \tag{2.6}$$

In this section, given its introductory nature, we maintain the more intuitive system of differential equation (2.1).

To move along its route, a train requires a certain amount of mechanical energy E, which is computed as follows:

$$E = \int_{\tau_0}^{\tau_1} F^+(t) v(t) dt \tag{2.7}$$

with $(\tau_0 - \tau_1)$ being the travel time of the train: starting at time τ_0 and ending at τ_1. The term $F^+(t)$ equals $F(t)$ if $F(t) > 0$ and equals 0 if $F(t) \leq 0$. Therefore, only the positive values of F are considered in the computation of energy as the negative ones do not imply energy consumption. Indeed, when $F < 0$, the brake is being engaged. By definition, $F^+(t)v(t)$ is the mechanical power $P(t)$ associated with the train motion state at t.

Let us now discuss a simple example of a single-train EETC problem that includes all the modeling aspects introduced so far in this section. A train is expected to traverse a D-long route in a scheduled travel time $(\tau_1 - \tau_0)$. We want to find a speed profile $v(t)$ with $t \in [\tau_0, \tau_1]$ that minimizes the total energy consumption. The system of differential equations (2.1) represents the law of motion for the train and boundary

conditions have to be specified for it. The whole trip must start and end at times τ_0 and τ_1, respectively. This implies that the initial and final distances are such that $s(\tau_0) = 0$ and $s(\tau_1) = D$, respectively. Let us assume that the initial and final speed values are equal to zero: $v(\tau_0) = 0$ and $v(\tau_1) = 0$. In addition, let us consider a distance-dependent speed restriction constraint: $v(t) \in [0, \tilde{V}(s(t))]$, where $\tilde{V}(s)$ is the maximum allowed speed at position $s(t)$. In this setting, a train speed profile is a function $v(t)$, resulting from the chosen control law $F(t)$, that solves the system of differential equation (2.1) and that complies with the constraints and boundary conditions described so far. To compute an energy-optimal speed profile $v^*(t)$, one wants to find a control law $F^*(t)$ such that it minimizes (2.7). The example problem discussed so far, is summarized in Equations (2.8)-(2.14). This constitutes a common *optimal train control problem*, i.e., *optimal control theory* (Lee and Markus, 1967) applied to train motion. According to Scheepmaker et al. (2017), optimal control problems, such as (2.8)-(2.14), are addressed in the literature with either *indirect exact methods*, *direct exact methods* (Betts, 2010; Rao, 2014) or with heuristic/meta-heuristic algorithms based on simplifying assumptions. The latter are the target of this thesis. Optimal control theory is beyond our scope.

$$\min_{F(t)\in\Omega(t)} \left\{ \int_{\tau_0}^{\tau_1} F^+(t)v(t)dt \right\} \tag{2.8}$$

$$\Omega(t) := [F_B, F_T(v(t))], \quad t \in [\tau_0, \tau_1] \tag{2.9}$$

$$F(t) - F_r(t) = m f_p \frac{dv(t)}{dt} \tag{2.10}$$

$$v(t) = \frac{ds(t)}{dt} \tag{2.11}$$

$$s(\tau_0) = 0; \quad s(\tau_1) = D \tag{2.12}$$

$$v(\tau_0) = 0; \quad v(\tau_1) = 0 \tag{2.13}$$

$$v(t) \in \left[0, \tilde{V}(s(t))\right] \tag{2.14}$$

In the EETC literature, many authors focused on indirect methods, which require finding the necessary conditions for a speed profile to be energy-optimal. By using the *Pontryagin's Maximum Principle*, researchers have shown that an energy-optimal train speed profile is necessarily a sequence of the four so-called *optimal driving regimes*. The four optimal driving regimes are:

1. *maximum acceleration* (also known as acceleration or maximum traction);
2. *cruising* (also known as speed holding);
3. *coasting*;
4. *maximum braking* (also known simply as braking).

Maximum traction is obtained by applying the maximum tractive effort, i.e., $F = F_T(v(t))$ in System (2.1), to accelerate a train. Generally, this is the most energy consuming among the four optimal driving regimes.

Cruising is obtained by maintaining the net force acting on the train equal to zero, i.e. $F = F_r$, so that the train moves with uniform speed. Although coasting consumes energy, it is generally preferable than maximum acceleration.

Coasting is obtained by neither engaging the brake nor the engine so that a train simply moves by inertia. This driving regime does not consume energy.

Maximum braking is obtained by exerting the maximum braking force $F = F_B$, i.e., $F = F_T(v)$ in System (2.1) to decelerate the train. This driving regime does not consume energy.

For further details on optimal control formulations concerning train movements, typical solution methods (such as direct & indirect methods) and the optimal driving regimes see, e.g., Albrecht et al. (2015, 2016c); Scheepmaker et al. (2017); Goverde et al. (2016).

A large number of EETC/EETT methods in the literature consists of heuristic/meta-heuristic algorithms. They are usually based on the knowledge of the four optimal driving regimes just discussed plus additional simplifying assumptions to reduce the set of feasible speed profiles (Scheepmaker et al., 2017). In this context, the problem

of finding an energy-efficient speed profile for a train is converted into to the problem of finding a sequence of driving regimes and its corresponding sequence of switching points. In general, one needs to find:

1. a sequence $(d_0, d_1, \ldots, d_k, \ldots, d_L)$ of driving regimes, containing L of the four optimal driving regimes, with L>0 being an unknown integer;

2. a sequence $(P_0, P_1, \ldots, P_k, \ldots, P_L)$ of switching points, where each P_k is a coordinate point (e.g., spatial or temporal) at which the switching to driving regime d_k occurs along the train route.

Figure 2.11 shows a distance-speed plot in which a train speed profile (blue curve) is obtained with six ($L = 6$) switching points $P_0, \ldots, P_5$. The driving regimes are the following: cruising from P_0; coasting from P_1; maximum acceleration from P_2; braking from P_3; coasting from P_4; braking from P_5. The profile starts and ends with non-zero speeds. The red vertical lines indicate the distance intervals in which line properties are uniform (e.g., speed limit, radius of curvature and slope angle). The green horizontal lines indicate the speed limit values along the line.

In addition, to reduce complexity, further assumptions are usually made, especially when computing speed is as crucial, as in this thesis. Precisely, the speed profile representation is usually simplified to make the search space smaller (Scheepmaker et al., 2017). An example could be Montrone et al. (2018), in which the number of switching points along with their spatial coordinates are fixed beforehand and switching points take only discrete speed values. Another example could be Luan et al. (2018a,b), in which coasting is neglected, maximum acceleration curves are approximated and speed values at switching points are discrete, as in Montrone et al. (2018).

The papers included in this thesis address rtEETC and rtEETT problems (Section 1.2 & 1.3) via meta-heuristic and heuristic algorithms based on the knowledge of the four optimal driving regimes. As common practice, we made reasonable assumptions to reduce the size of the search space so to deal with large realistic instances in a short computing time. The details are discussed in the following paragraphs.

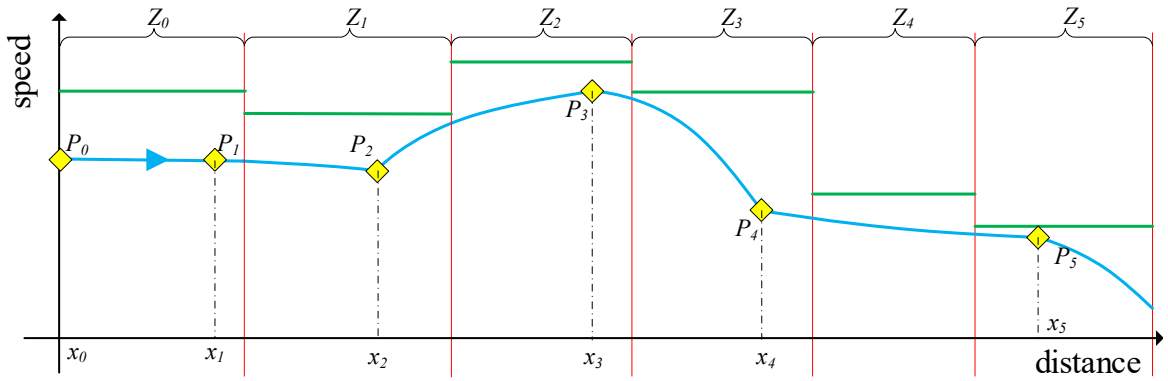


Figure 2.11: Example of train speed profile (blue curve) and switching points (yellow diamond shapes).

In Paper-1 (Chapter 3), only one train and a route are considered. The route length is divided into a fixed number of subsections containing at most four switching points. For each subsection, only the following ordered sequence of driving regimes is permitted: (1^{st} switching) maximum acceleration, (2^{nd} switching) cruising, (3^{rd} switching) coasting and (4^{th} switching) maximum braking. Driving regimes could be omitted but the order stays unchanged. Switching points are not fixed, except for the spatial coordinate of the first one in each subsection, being located at each subsection's entrance. Switching points take discrete speed values. If needed, only a subset of the four allowed driving regimes may appear in a subsection. For instance, let us suppose that maximum acceleration is to be avoided in a subsection for some reason. To do so, the algorithms will have the second switching point overlap the first one so that the undesired driving regime does not occur at all (maximum acceleration would be held for either zero meters or zero seconds). A similar set of assumptions was used in Li and Lo (2014b).

In Paper-2 and Paper-3 (Chapters 4 & 5), multiple interacting trains are considered in a microscopically modeled infrastructure (Section 2.3.1.1). Each block section is divided into a fixed number of subsections with equal length. Only at the beginning of each subsection, a change of driving regime may happen. In each speed profile, the switching between two regimes occurs at fixed positions within each block section. Unlike Paper-1 (Chapter 3), the sequence of driving regimes is not fixed. For each

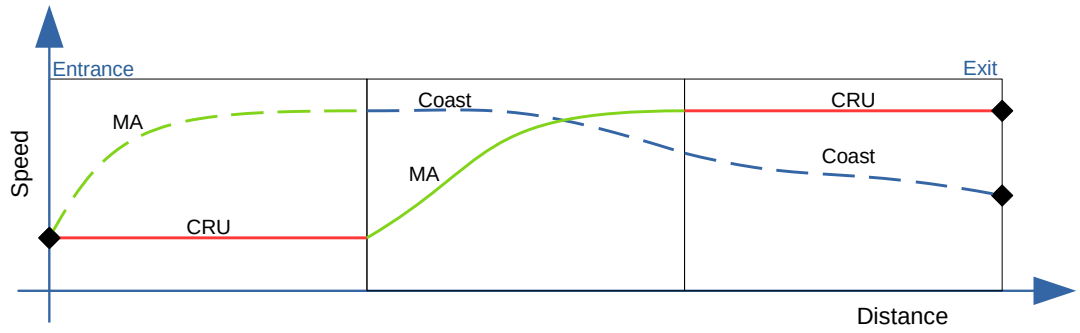


Figure 2.12: Speed profile representation in Paper-2 and Paper-3. X-axis: distance; Y-axis: speed; Vertical black lines: borders of the subsections in which the block section is divided into; MA: maximum acceleration; Coast: coasting; CRU: cruising;

block section, several speed profiles are pre-computed (for those trains scheduled to traverse that specific block section). Lists of initial-final speed pairs and of driving regime sequences are used. As a result, the rtECMP is modeled as a combinatorial optimization problem. For each train, the algorithms described in Paper-2 & Paper-3 (Chapters 4 & 5) perform a selection of a pre-computed speed profile for each block section in the train's route. This process builds a full speed profile for each train. The set of available block section speed profiles is pre-computed using a version of the *TDRC algorithm* (Montrone et al., 2018) that we slightly improved.

Figure 2.12 displays two possible speed profiles in a block section (solid and dashed curves). They are obtained according to the assumptions of Paper-2 and Paper-3 (Chapters 4 & 5). The block section shown has been split into three subsections of equal length. The dashed speed profile is composed of: maximum acceleration; coasting; coasting. The solid speed profile is composed of: cruising; maximum acceleration; cruising. As discussed, the switching of driving regimes occurs at the boundaries between subsections (on the black vertical lines). Both speed profiles share the same initial speed.

Chapter 3

Paper-1: Energy-Efficient Train Control: a practical application

3.1 Introduction

One of the major costs for railway companies is given by energy consumption. For this reason, the development into a more mature and competitive market makes an efficient energy management imperative for reducing the operating costs. Ecological awareness is also a major driver for energy efficiency in railway systems in an effort towards reducing air pollutants, e.g., carbon dioxide, whose emissions are one of the causes of global warming (Luijt et al., 2017).

The construction of energy-efficient driving profiles has attracted large attention from researchers in the recent years, being an effective way of saving energy. The resulting problem is known as *Energy-Efficient Train Control* (EETC), and is sometimes also referred to as *eco-driving* or *train trajectory planning problem*. This problem aims at finding the most energy-efficient driving profile for a given train traveling on a certain line, while satisfying a number of operational constraints to ensure a safe and punctual journey. Recent surveys on EETC have been proposed by Yang et al. (2016) and Scheepmaker et al. (2017). A complete review on the Optimal Train Control Theory has been given in Albrecht et al. (2016a,b).

Analyses based on the *Pontryagin Maximum Principle (PMP)* have shown that an

optimal driving strategy consists of a sequence of four *driving regimes*, namely, maximum acceleration/traction (MT), cruising/speed-holding (SH), coasting (CO) and maximum braking (MB), see, e.g., Howlett et al. (1994b) and Albrecht et al. (2016a,b). An example speed profile consisting of MT-SH-CO-MB is shown in Figure 3.1.

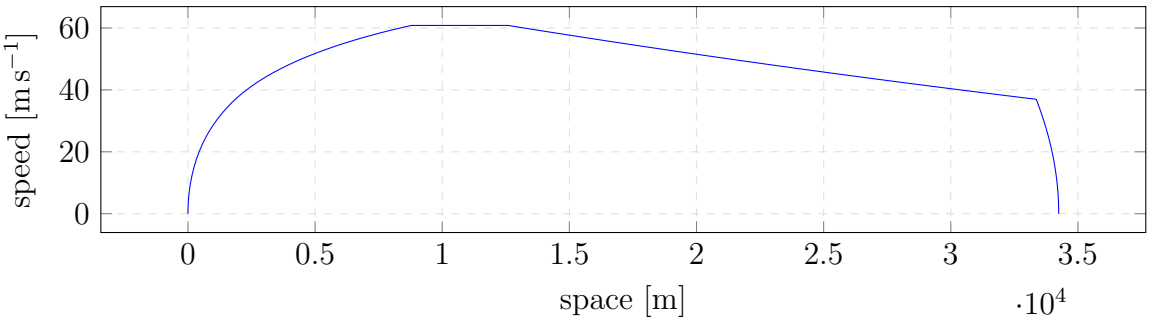


Figure 3.1: *A profile consisting of MT, SH, CO and MB. The train runs on a track approximately 35 kilometers long, and starts and ends at zero speed.*

Based on this result, most of the algorithms proposed in the literature (Albrecht et al. (2016b), Yang et al. (2016) and Scheepmaker et al. (2017)) define the driving profile as a sequence of these four driving regimes, identifying suitable switching points between consecutive driving regimes.

In this paper we focus on the EETC in its basic version, arising when the driving profile of a single train has to be determined. We do not consider railway traffic management, as we assume that the schedule of the train has already been optimized by a (possibly online) scheduling algorithm (see, e.g., Bettinelli et al. (2017) and Fischetti and Monaci (2017)). In this context, safety requirements impose that the schedule of the train is given on input and cannot be changed. In addition, we do not consider possible interactions between trains in terms of power exchange, as it could happen in complex networks equipped with appropriate infrastructures and energy storage systems.

This paper is organized as follows. In Section 3.2 we define the problem, and in Section 3.3 the solution approaches are outlined. The results of the computational testing on real-world instances are given in Section 3.4. Finally, Section 3.5 draws some conclusions.

3.2 Problem Definition

We are given a rail track having length D, and a train running on the track. The train has a fixed schedule, imposing its travel time be equal to exactly T time units, and is characterized by some known rolling-stock properties (e.g., its weight). The track geometry, e.g., the position-varying slope and radius of curvature, is also known. Finally, there are given speed limits that the train must respect, and that vary along the track. The problem requires to determine a driving profile for the train such that (i) the travel time of the train is exactly T; (ii) speed limits are respected; and (iii) the total amount of energy required for running the train is minimized.

We represent the track as a segment $[0, D]$ and assume that the train travels on the track from time instant 0 to time instant T.

For the sake of simplicity, we do not consider the case of *steep* uphill/downhill tracks, i.e., we assume that the train, regardless of its speed, is always capable of negotiating uphills/downhills or speed-holding. Moreover, we assume a continuous control and neglect any form of energy recovery, as it happens with the so-called *Regenerative Brake*. For details on steep uphills/downhills and on Regenerative Brake, the reader is referred to Howlett et al. (1994b); Albrecht et al. (2016a,b).

In our setting, the motion of a train is approximated using the following point-mass model

$$\frac{dv}{dt} = u(t) - R(v(t)) + G(x(t)) \tag{3.1}$$

$$\frac{dx}{dt} = v(t), \tag{3.2}$$

where time $t \in [0, T]$, is the independent variable, while speed $v(t)$, position $x(t)$ are coordinates of the dynamical system and $u(t)$ is the controlled acceleration.

Term $R(v)$ represents the so-called *Basic Resistance*, taking into account several resistive phenomena depending only on the given rolling-stock. In particular, it includes rolling resistances, mechanical resistances that are proportional to speed (e.g., rotation of axels and shaft, mechanical transmission, etc.), and air resistance (see, Profillidis, 2016).

Basic resistance is typically approximated using the well-known *Davis Equation*

$$R(v) = r_0 + r_1\, v + r_2\, v^2, \tag{3.3}$$

where r_0, r_1 and r_2 are positive empirical constants depending on the rolling-stock (see, J.W. Davis Jr (1926)).

Term $G(x)$ is the *Line Resistance*, which measures the position-dependent forces acting on the train along its route. This resistance considers the horizontal component of gravity (depending on the position-varying track slope along the track) and the *curve resistance* (depending on the position-varying radius of curvature along the track). The analytic expression of $G(x)$ has been provided by our industrial partner as a piece-wise function through the empirical formula:

$$G(x) := -g\ \sin\left[\arctan\left(\frac{0.8}{\rho(x)} + \tan\theta(x)\right)\right], \tag{3.4}$$

$\theta(x)$ and $\rho(x)$ being the (position-dependent) track grade and radius of curvature respectively, and g the gravitational acceleration (see, Fayet, 2008; Sapronova et al., 2017).

The controlled acceleration $u(t)$ is bounded by two functions, i.e.

$$-u_L(v) \le u(t) \le u_U(v). \tag{3.5}$$

where bounds $u_L(v) \ge 0$ and $u_U(v) \ge 0$ are the *maximum deceleration* and the *maximum acceleration*, respectively, that the train engine can handle. In the following, we will always assume that u_L and u_U are decreasing non-linear functions of the speed.

As mentioned above, speed limits impose an upper bound on the maximum speed

$$0 \le v(t) \le \bar{V}(x(t)) \tag{3.6}$$

where the upper bound value $\bar{V}(x(t))$ depends on the geometry of the track. In addition, the following boundary conditions

$$x(0) = 0, \qquad x(T) = D, \qquad v(0) = v_{init}, \qquad v(T) = v_{final}, \tag{3.7}$$

are imposed to fix the position and speed of the train at time instants $t = 0$ and $t = T$, with v_{init} and v_{final} given on input.

Finally, the objective is to minimize the total energy spent by the train, given by

$$E = \int_0^T \frac{u(t) + |u(t)|}{2} v(t) dt. \tag{3.8}$$

Equation (3.8) is derived from the definition of *work* done on a point mass. Since we do not account for energy recovery, negative values of u must be ignored. We therefore introduce $(u + |u|)$ so that energy contribution is zero whenever $u < 0$. The reader is referred to Albrecht et al. (2016a) and Scheepmaker et al. (2017) for further details.

Observe that the Equations (3.1)-(3.8) constitute a well-known Optimal Control Formulation for the EETC, see Albrecht et al. (2016a,b) and Scheepmaker et al. (2017).

3.2.1 Overview of the Solution Approach

In most practical cases, track geometry and speed-limits are described as piece-wise constant functions of the space. Therefore, it is possible to partition the track $[0, D]$ into a sequence of n consecutive sections $S_k = [x_o^{(k)}, x_f^{(k)}]$, with $x_o^{(0)} = 0$, $x_f^{(n)} = D$ and $x_f^{(k)} = x_o^{(k+1)}$, $(k = 0, 1, \ldots, n-1)$, that have constant slope, radius of curvature and speed limit (see Haahr et al. (2017); Wang et al. (2012, 2013)). We will refer to those sections as *segments*. This results in having a fixed *Line Resistance* along with a fixed speed limit. In particular $\forall\, k = 1 \ldots n$:

$$G(x) = g^{(k)} \qquad\qquad \bar{V}(x) = \bar{V}^{(k)}. \tag{3.9}$$

As mentioned in the introduction, similar to what is done in Li and Lo (2014a), our approach is based on the Optimal Train Control theory and assumes that a concatenation of at most four given driving regimes is used to define the speed profile in each section. The driving regimes are characterized by the following conditions:

$$\text{Maximum Traction, MT:} \qquad u_1 = u_U(v) \tag{3.10}$$

$$\text{Speed Holding (or Cruising), SH:} \qquad u_2 = R(v) - g^{(k)} \tag{3.11}$$

$$\text{Coasting, CO:} \qquad u_3 = 0 \tag{3.12}$$

$$\text{Maximum Braking, MB:} \qquad u_4 = -u_L(v), \tag{3.13}$$

where u_1, u_2, u_3 and u_4 are the accelerations in every regime. Within a segment, each regime can be executed at most once, and regimes must appear in fixed order, namely, MT, SH, CO, and MB. Furthermore, some regimes may not be used in a segment.

As a consequence, Equations (3.1) and (3.2) can be simplified, because the term u corresponds to one of the u_j $(j = 1, \dots 4)$ shown in Equations (3.10)-(3.13) and $G(x) = g^{(k)}$ in each segment k.

We observe that, within our settings, energy is only consumed during the MT and SH driving regimes. Let $E^{(k)}_{MT}$ and $E^{(k)}_{SH}$, respectively, be the energy consumed in the MT and SH regimes for segment k $(k = 1 \dots n)$. Then Equation (3.8) reduces, for MT, to

$$E^{(k)}_{MT} = \int_{t_1^{(k)}}^{t_2^{(k)}} u_1(v(t))\, v(t)\, dt, \tag{3.14}$$

where $t_1^{(k)}$ and $t_2^{(k)}$ are the start and end time for MT in segment k. Similarly, for regime SH, we have that (3.8) simplifies to

$$E^{(k)}_{SH} = u_2(v_2^{(k)})\, v_2^{(k)}(t_3^{(k)} - t_2^{(k)}), \tag{3.15}$$

where $v_2^{(k)}$ is the constant speed value, while $t_2^{(k)}$ and $t_3^{(k)}$ are respectively the start and end times for SH regime.

We note that, Equations (3.1), (3.2) and (3.14), can be pre-computed, for each segment and driving regime, by using numerical methods or closed-form expressions under some assumptions on u as in Ye and Liu (2017).

3.3 Heuristic Solution Approaches

In our application, train control has to be determined in short computing times, in order to take into account possible changes in the schedule of the train due to some unpredictable events. For this reason, we propose heuristic approaches for its solution. In this section, we first introduce a constructive heuristic (CH), then, we present a multi-start randomized constructive heuristic (RCH) and a Genetic Algorithm (GA).

3.3.1 Constructive Heuristic

The proposed constructive heuristic is an iterative procedure that starts from an infeasible solution and, at each iteration, tries to reduce infeasibility, until a feasible solution is produced.

At the beginning, the algorithm computes the so-called *allout* speed profile, i.e. the solution obtained by running the train at its maximum allowed speed. When computing this speed profile, train motion laws and speed limits are taken into account.

Observe that the running time τ_{AO} associated with this solution provides a lower bound on the running time for the train in any feasible solution. If this running time is not equal to the required time T, the current solution is infeasible. To recover feasibility we consider the actual maximum speed $\tilde{V}^{(k)}$ reached in every segment k, and slow the train down by artificially reducing the speed limits. This speed reduction is applied in one segment at a time.

For each segment k, we reduce its maximum speed by a given amount s (which is a parameter of the algorithm), and re-compute the associated speed profile and running time τ. If $\tau > T$, then the last update of the maximum speed is canceled. If, instead $\tau < T$, the next segment is considered. This procedure continues until a feasible solution is obtained. The pseudo-code of the algorithm is shown in Algorithm 1.

Algorithm 1: CH

```
1 Compute the allout speed profile for the given route. Store travel time in
    τ = τ_AO. Store in Ṽ^(k) the maximum speeds that are actually reached by the
    allout speed profile in each segment k;
2 Set w_k := Ṽ^(k), ∀k = 1...n;
3 while τ < T do
4     for k = 1...n do
5         w_k := w_k − s;
6         Re-compute speed profile using the current speed limits in each segment
            (i.e. w_1, ... w_n). Store travel time in τ;
7         if τ > T then
8             Undo the change, namely w_k := w_k + s
```

3.3.2 Multi-start Randomized Constructive Heuristic

The constructive heuristic of the previous section is used in a multi-start fashion to determine a pool of feasible solutions. This allows to possibly find better solutions and to determine an initial population of the genetic algorithm (see Section 3.3.3).

The multi-start randomized constructive heuristic performs a fixed number of iterations, changing a set of parameters in a random way. At each iteration, γ_1 segments are selected (where γ_1 is a parameter of the algorithm), and the maximum speed of the selected segments is reduced by a random value. If the travel time of the allout profile associated with the current w'_k values is not larger than T, the constructive heuristic of Section 3.3.1 is applied. Otherwise, the speed limits are re-defined, and the process is repeated.

The algorithm produces one feasible solution per iteration and is executed for NI (say) iterations. The best solution found among all iterations is then returned. The pseudo-code is reported in Algorithm 2.

Algorithm 2: RCH

1 Compute the *allout* speed profile for the given route. Store in $\tilde{V}^{(k)}$ the maximum speeds that are actually reached by the *allout* speed profile in each segment k;
2 Set $w_k := \tilde{V}^{(k)}$, $\forall k = 1 \dots n$;
3 **for** $i = 1 \dots, NI$ **do**
4 **repeat**
5 set $w'_k = w_k$ for each segment k;
6 **for** $c = 1 \dots \gamma_1$ **do**
7 Randomly select a segment $\tilde{k}$ according to a uniform random distribution;
8 Define $w'_{\tilde{k}}$ as a random number within $(0, \gamma_2\, w_{\tilde{k}}]$ with $0 \leq \gamma_2 \leq 1$;
9 Re-compute speed profile using the current speed limits w'_k;
10 Store travel time in τ;
11 **until** $\tau \leq T$;
12 Set $w_k := w'_k$, $\forall k = 1 \dots n$, and apply lines 3-11 of **Algorithm 1**;

3.3.3 Genetic Algorithm (GA)

In this section, we present a Genetic Algorithm, inspired by the work of Li and Lo (2014a).

In our algorithm, each chromosome is associated with a solution and is represented by a vector $\mathbf{V}$ containing the values of the initial speed for each driving regime. In particular we have

$$\mathbf{V} = \left[\; v_j^{(k)} \; \right]_{\substack{k=1\dots n \\ j=1\dots 4}}. \tag{3.16}$$

The population contains M individuals. During each iteration, by means of crossover and mutation operators, the current population might grow larger than M. The selection operator restores the number of individuals to M by eliminating the lowest ranking ones from the current population. A fixed number h of *Elite* individuals is kept intact over the iterations.

The evaluation of each chromosome c of a current population is carried out by means of the following fitness function

$$f(c) = k_1 E(c) + k_2 H(c), \tag{3.17}$$

where k_1 and k_2 are tuning parameters, $E(c)$ represents the traction energy that is spent driving the train according to solution c, and $H(c)$ measures the diversity of chromosome c with respect to the rest of the current population (thus making the fitness function biased towards favoring diversity, instead of considering energy efficiency only). $H(c)$ is computed as a mean difference between c and the other individuals.

The main components of the GA are reported in the following list, according to the order in which chromosomes are processed in each iteration:

1. **Mutation Operator:** this operator is applied, with probability P_m, to each non-elite chromosome, producing new individuals. It consists of applying random variations of speed to a set of randomly selected segments, according to a uniform random distribution. The maximum variation amount is given as a percentage m (parameter of the algorithm) of the maximum speed in a each segment. Preliminary experiments showed that better performances are achieved when these operators are not applied to elite chromosomes;

2. **Crossover Operator:** it implements a single-point crossover to produce two children chromosomes from a couple of parents, by recombining their genome after splitting on a randomly selected segment boundary. The crossover break point is randomly selected among all those genes that correspond to the end of some segment. This operator has a probability P_c of affecting each chromosome, excluding Elites and those chromosomes produced by mutation at the current iteration;

3. **Repair Operator:** it restores feasibility of solutions after mutation and crossover, if needed. It implements a procedure similar to the constructive algorithm introduced in Section 3.3.1

4. **Selection Operator:** it ranks each chromosome c of the current population by means of the fitness function.Then, it deletes the lowest ranking individuals

exceeding the maximum population size M, while preserving a group of h Elites intact.

3.4 Computational Experiments

The algorithms described in the previous section were implemented in C and executed on a 2.9-Ghz Intel Core i7-7500U with 16 GB of RAM. Our benchmark is composed by real-world instances provided by ALSTOM. All instances are associated with two railway lines, denoted as ROUTE 1 and ROUTE 2. These lines have lenghts equal to 250 and 270 kilometers, respectively, and are composed by 42 and 67 segments, respectively. For each line, we considered five different train models, denoted with letters A to E in the following, and having different characteristics. This produced a benchmark of 10 different instances. To evaluate the quality of the obtained solutions, we used the following measure of efficiency, commonly used by ALSTOM's practitioners:

$$e = 100 \cdot \frac{E_A - E}{E_A}, \tag{3.18}$$

where E_A represents the energy consumption of the aforementioned allout profile, while term E corresponds to the energy consumption associated with the given solution.

The RCH algorithm (see Algorithm 2), which also initializes the GA, was iterated 32 times in each experiment in a multi-start fashion. Parameter s was set to 1 (m/s), $\gamma_1 = 20$ and $\gamma_2 = 40\%$. The GA was configured according to the following parameter values: $P_c = 0.97$, $P_m = 0.95$, $M = 9$, $h = 5$ and $m = 5\%$. A maximum number of 300 generations was imposed. Moreover the GA was set to terminate after 11 non-improving consecutive generations.

Table 3.1 shows the results associated with ROUTE 1. For each train model we report the fixed travel time T, the level of efficiency computed according to (3.18), and the associated CPU time (in seconds).

The results show that, when the first line is considered, the constructive heuristic is able to reduce the energy consumption, with respect to the allout profile, by around 10% on average. The associated computing times are rather small, being always below 2 seconds. By executing the randomized multistart constructive heuristic and

		Efficiency			CPU time (seconds)		
Train	T (s)	CH	RCH	GA	CH	RCH	GA
A	4560	3.2	5.4	7.2	0.2	1.5	4.4
B	4560	8.2	10.4	11.6	0.1	1.2	2.5
C	6525	6.6	11.5	25.3	0.3	4.9	10.6
D	10920	10.9	12.5	12.7	0.1	0.6	2.2
E	7079	18.8	20.2	21.6	0.2	0.7	2.1

Table 3.1: Results on ROUTE 1 (250 km, 42 segments).

the genetic algorithm we obtain further savings, about 12% and 15%, respectively. Although these improvements could be expected, as the algorithms are built "on-top" of each other, the additional savings that we obtained are quite significant. While the increase in CPU time for the randomized multistart constructive heuristic is rather limited, the genetic algorithm requires significantly larger computing times for some instances: in these cases, a time limit can be imposed in order to use this algorithm in a real-time system.

		Efficiency			CPU time (seconds)		
Train	T (s)	CH	RCH	GA	CH	RCH	GA
A	7285	6.6	13.7	23.7	0.5	9.6	15.8
B	7142	8.7	14.8	29.0	0.6	11.4	21.7
C	7933	9.9	13.4	27.2	0.8	15.9	29.4
D	11729	12.7	13.2	13.9	0.2	0.9	2.4
E	7595	18.2	18.9	22.2	0.3	1.7	5.2

Table 3.2: Results on ROUTE 2 (270 km, 67 segments).

The results for the second line, reported in Table 3.2, confirm those obtained for the first line, although the computing times are larger than before. However, we observe that energy reduction in Table 3.2 is also higher than in Table 3.1.

3.5 Conclusions and future research

In this paper we studied the Energy-Efficient Train Control (EETC) problem. This problem was solved using three approaches: a constructive heuristic, a multi-start randomized constructive heuristic, and a genetic algorithm.

Computational results on real-life instances show that, in most cases, the computing times of the algorithms are short enough to allow their use in a real-time application.

Future research directions will be to consider real-time traffic management aspects, related to the possibility of slightly changing the schedule of the train, as well as the adoption of a system which allows energy recovery through regenerative brake.

Chapter 4

Paper-2: Real-Time Optimization of Energy Consumption in Railway Networks

4.1 Introduction & literature review

Managing traffic perturbations in railway networks to reduce the overall delay is known as *real-time railway traffic management problem* (rtRTMP) (Pellegrini et al., 2014). Its solution is a feature of modern rail *traffic management systems* (TMS). The task of managing traffic to reduce delay is entrusted to a dispatcher, who is in charge of a control area, i.e., a limited size railway network. The *real-time Energy Consumption Minimization Problem* (rtECMP), as introduced by Montrone et al. (2018), has the objective of minimizing the weighted sum of train energy consumption and total delay by deciding speed profiles in a given control area and time horizon. It takes as input the decisions on train routing and precedences made by a rtRTMP solver. In addition, having to define energy-efficient speed profiles for multiple interacting trains, it takes into accounts infrastructure characteristics, operational constraints and train dynamics. The rtECMP ouputs include arrival, departure, passing through and dwelling times along with speed profiles.

Minimization of energy consumption in train operations is the main concern of

the literature on Energy-Efficient Train Control (EETC) and Energy-Efficient Train Timetabling (EETT) of which Scheepmaker et al. (2017) provided a comprehensive review. EETC methods focus on minimizing the traction energy consumption of one or multiple trains given a timetable to fulfill and, therefore, they aim at computing energy-efficient speed profiles. ETTT methods focus on scheduling one or multiple trains. Here, energy saving by EETC is considered along with either timetabling objectives, in case of nominal traffic, or re-scheduling objectives, in case of perturbed traffic. For perturbed traffic, we refer the reader to, e.g., Wang and Goverde (2017), Galapitage et al. (2018), Luan et al. (2018b), Montrone et al. (2018), Li et al. (2020). For nominal traffic, we refer the reader to, e.g., Goverde et al. (2016), Wang and Goverde (2019), Xu et al. (2020).

Montrone et al. (2018) proposed a mixed-integer linear programming formulation for the rtECMP. However, the solution of such formulation quickly becomes impractical for realistic instances, due to too long computing times. In this paper, we extend the research of Montrone et al. (2018) by proposing a graph-based rtECMP model that we solve with an Ant Colony Optimization (ACO) algorithm, to which we refer as ACO-rtECMP. An experimental analysis is conducted on a real life railway infrastructure (the Pierrefitte-Gonesse junction, located in France) subject to various traffic perturbations.

The rest of the paper is structured as follows: Section 4.2 defines the problem by also introducing operational constraints and characteristics of train speed profiles; Section 4.3 describes our graph-based model for the rtECMP; Section 4.4 discusses the proposed ACO algorithm; Section 4.5 explains our experimental setup; Section 4.6 analyzes the results; Section 4.7 exposes our conclusions.

4.2 Problem

As discussed in the introduction, solving the rtECMP requires defining an energy-efficient speed profile for each train when traffic perturbations occur. The objective is to jointly minimize the overall delay and energy consumption. It is crucial for the chosen speed profiles to be compliant with traffic management decisions (routing and

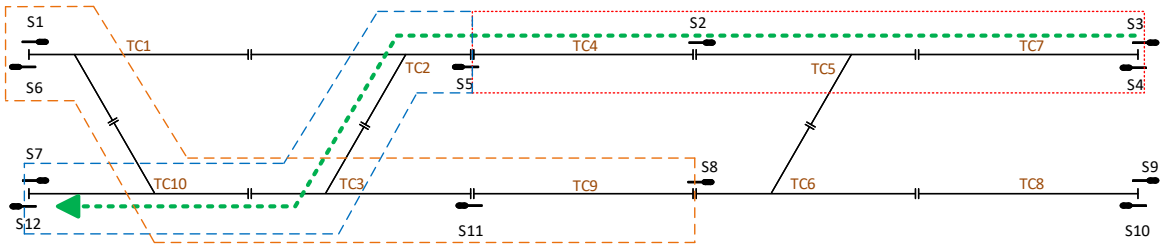


Figure 4.1: Example of a simple infrastructure in which TC denotes track-circuits, S denotes signals. The dashed green arrow highlights a train route composed of block sections S4-S5 (dotted red) and S5-S12 (dashed blue). Block section S1-S8 (dashed orange) is incompatible with S5-S12 as they share TC3 and TC10.

precedences) and railway safety requirements. The problem constraints depend on two main aspects: infrastructure characteristics and train physical capabilities impacting possible speed profiles, and operational rules.

For what concerns possible speed profiles, we consider four alternative driving regimes which have been proven to be convenient based on the *Pontryagin's maximum principle*: maximum traction, cruising, coasting and maximum braking. For further details, see, e.g., Scheepmaker et al. (2017).

For what concerns the modeling of operational rules, we consider a microscopic representation of rail infrastructures. Specifically, the infrastructure in a control area is composed of *track-circuits*. These are track elements equipped with an electric device for sensing the presence of a train. Track-circuits are grouped into *block sections*, which are track stretches that can be traversed by only one train at a time to maintain safe distancing. Every train is assigned a route in the infrastructure by a TMS. More precisely, we define a route as a sequence of block sections linking an origin-destination pair. Control areas are equipped with an *interlocking system* that ensures a coherent route formation. To indicate whether a block section can be accessed or not, the interlocking system employs signals, which are designed to give visual instructions to drivers. In particular, every block section is delimited by two signals placed at its entrance and exit location (see for instance Pachl 2002). Figure 4.1 shows an example of a simple infrastructure.

In the simplest case, a signal can display three so-called *aspects*: green, yellow and

red. A red aspect forbids the access to the following block section. A signal displaying yellow allows the access and demands a slow down so that a full train stop is possible before the following signal. Green grants the access to a block section and implies that a driver can safely enter the following one at full speed, if suitable. Each signal in a route may have a different number of aspects, up to some $n \geq 3$. This means that, when a n-aspect signal is crossed with a green, at least $(n-2)$ block sections are available in front of it. In addition to red, yellow and green, n-aspect signals can display further *restrictive aspects*. They progressively demand greater speed reductions to be reached by the end of the block section considered, up to a stop before a red signal. The interlocking system ensures that the length of $(n-2)$ block sections is sufficient for a train to brake after a signal with n aspects, provided that the train type is authorized to use the infrastructure.

Given a block section b, we designate all the block sections sharing one or more track-circuits with b as *incompatible* (in Figure 5.1, block sections S5-S12 and S1-S8 are incompatible as they share track-circuits TC3 and TC10). When b is available, having its opening signal displaying an aspect different from red, a train can enter it.

In addition to safety, the interlocking system also allows the enforcement of traffic management decisions. As for routing decisions, it is in charge of setting block sections for trains so that the planned route is indeed traversed. As for precedence decisions, it allows unlocking block sections along a route only if all the included track-circuits are to be used, first of all, along the routes itself.

4.3 Model

In this section, we propose a graph-based model for the rtECMP as defined in Section 4.2. In the model definition, a train speed profile is defined as a combination of *partial speed profiles*. For each train and for each block section in the train's route, we pre-compute a discrete set of partial speed profiles.

We are given a set of trains $i \in T$ and set of block sections composing a route for each of the trains B_i. We refer to a generic block section in the route of train i as $j \in B_i$. Let S be the set containing train-block pairs $(i,j) : i \in T, j \in B_i$, such that

either j is the last block section in the route of i (before the train leaves the control area), or j is where a scheduled stop for i is planned. For each $(i, j) \in S$, we denote the scheduled departure time as $\lambda_{i,j}$. For each $i \in T$ and $j \in B_i$, we are provided with train precedence in the form of a set $\mathcal{P}_{i,j}$ containing train-block pairs (i^*, j^*). Only when every train i^* has released its block j^*, i can access j. Let $G = (V, A)$ be a graph with V and A set of vertices and arcs respectively. For each block section $j \in B_i$, we pre-compute a discrete set of partial speed profiles with the following attributes: an initial speed; a final speed; a value of energy consumed; a running time. Except for two special vertexes, *source* (s) and *terminal* (t) (which indicate the beginning and end of every route), each vertex in graph G represents a partial speed profile for a train-block pair. The attributes of a generic $v \in V$ are denoted by: i_v (train to which the node refers); j_v (the considered block section of B_{i_v}); k_v (initial speed); k'_v (final speed); r_v (running time of i_v in j_v); E_v (energy consumption).

The set of arcs A of G is defined as follows:

- For all $a, b \in V$ such that $i_a = i_b$, arc (a, b) exists if all following statements are true: j_a directly precedes j_b in the train's route; the final speed associated to node a is equal to the initial speed associated to node b, i.e., $k'_a = k_b$.
- For all $a, b \in V$ such that $i_a \neq i_b$, arc (a, b) exists if one of the following conditions is verified: (i_a, j_a) is in $\mathcal{P}_{i_b,j_b}$; j_a and j_b are compatible.
- For all $v \in V$, arc (s, v) exists if train-block pair (i_v, j_v) verifies the following conditions: j_v is the first block section in the route of i_v; no other train precedes i_v in j_v, i.e., $\mathcal{P}_{i_v,j_v} = \emptyset$.
- For all $v \in V$, arc (v, t) exists if j_v is the last block section of the route of train i_v.

An example $G(V, A)$ is shown in Figure 4.2, where two trains, denoted as T1 and T2, are considered. The route of T1 comprises block sections A and B. The route of T2 comprises block sections C and D. Let us suppose that only block sections D and B are incompatible, and that T2 must have cleared D before T1 can enter B, i.e., $(T2, D) \in \mathcal{P}_{T1,B}$. Nodes are partitioned in four clusters (colored rectangles),

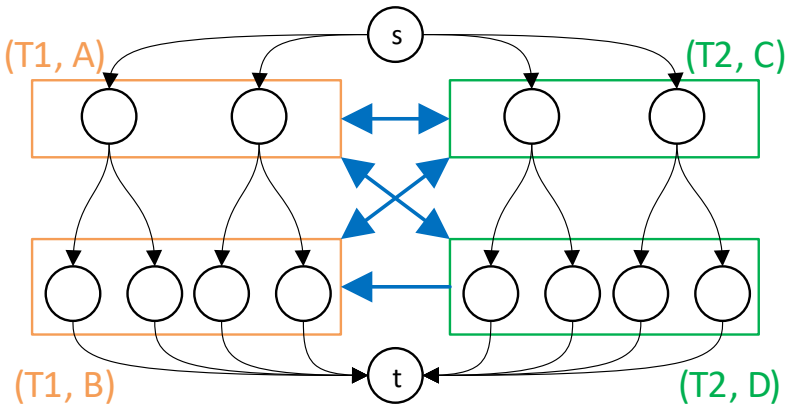


Figure 4.2: Example $G(V, A)$ described in Section 4.3.

each associated to a given train-block pair: (T1, A); (T1, B); (T2, C); (T2, D). The black arcs connect compatible block section speed profiles for a train in terms of initial/final speed. In addition, they connect the source to the first block sections' nodes. Similarly, black arcs connects the last block sections' nodes to the terminal. As A and C are compatible, each vertex in (T1,A) is bidirectionally connected to each vertex in (T2,C). The corresponding arcs are replaced by a single blue bidirectional arc connecting the corresponding rectangles, for readability. The same applies for the bidirectional arcs between: (T1, A)-(T2, D), (T1, B)-(T2, C), and (T1, A)-(T2, D). Due to the precedence constraint, nodes in (T2, D) are connected to nodes in (T1, B), but not viceversa. The blue arrow linking the two rectangles is hence monodirectional.

A path h in G that satisfies the following set of constraints is a feasible solution of the rtECMP:

C1 For each $(i, j) : i \in T, j \in B_i$, exactly one vertex such that $(i_v, j_v) = (i, j)$ must be in h. This makes sure that a unique choice of partial speed profile is effected per train-block pair.

C2 For each $v, w \in h$ referring to two consecutive block sections j_v and j_w that have to be traversed by the same train ($i_v = i_w$), it must be true that $k'_v = k_w$. This ensures speed profile continuity on the boundaries between every pair of consecutive block sections traversed by a train.

C3 For each $v \in h$, k'_v must be coherent with the signal aspect opening block section j_v when i_v enters it. This constraint models the interlocking system functioning as discussed in this section.

The optimal solution is the one minimizing the weighted sum of normalized energy consumption and delay.

4.4 Algorithm

We propose an Ant Colony Optimization (ACO) algorithm (Dorigo and Stützle, 2004) to tackle the graph model introduced in Section 4.3. Precisely, we adopt a $\mathcal{MAX}$–$\mathcal{MIN}$ *ant system* ($\mathcal{MM}$AS) algorithm (Stützle and Hoos, 2000), to which we refer as *ACO-rtECMP*.

ACO is a population-based meta-heuristic modeled after the pheromone-based cooperative behavior of ants. In ACO, ants are computational agents implementing a stochastic constructive heuristic for the problem at study. The elements of a solution are the nodes of a so-called *construction graph*. To build a solution, each ant moves across the graph, one node at a time, selecting, at each decision step, a node in the neighborhood of the current one. The selection of each node is done through the so-called pseudo-random proportional rule. It is biased by two factors: *heuristic information* and *pheromone trails*. The former is a problem-specific quantity associated to nodes or arcs that represents a greedy measure of their attractiveness. The latter is a quantity representing a collective memory on the quality of previously visited solutions containing the node or arc on which pheromone is deposited. Pheromone constitutes an indirect mean of communication between ants as it is deposited by them throughout a run of the algorithm and it guides further ants' individual search. The algorithm is iterative: at each iteration, a group of ants, named colony, builds solutions. Based on these solutions, the pheromone trail is updated to lead future ants towards promising regions of the search space.

The rtECMP, by nature, is well suited to be addressed with a constructive, incremental approach such as ACO. Precisely, from a chronological standpoint, the initial

decisions made have a strong impact on the subsequent ones both in terms of feasibility and in terms of resulting quality. In an ACO algorithm, decisions can easily be made following both the chronological order and the one imposed by the precedence constrains. This allows ants to choose the speed profile of each block section being aware of the signal aspect opening it. Therefore, only the candidate profiles matching the possibly necessary braking are considered and infeasible solutions are avoided entirely. Furthermore, the rtECMP involves many complex space-time relations arising between groups of interacting trains. Here, another advantage of ACO is that it does not require these relations to be formalized into *a priori*. Indeed, they are autonomously learned through the evolution of the pheromone trails.

In the ACO-rtECMP, we use G as the construction graph as we tackle the model in Section 4.3. Hence, an ant is required to construct a path h in G subject to constraints **C1**-**C3**.

Each vertex $v \in V$ is assigned a heuristic information value (denoted as η_v) defined as:

$$\eta_v = w_0 \left(\frac{\epsilon_{max}^{(j_v)} - E_v}{\epsilon_{max}^{(j_v)} - \epsilon_{min}^{(j_v)}} \right) + w_1 \left(\frac{\gamma_{max}^{(j_v)} - r_v}{\gamma_{max}^{(j_v)} - \gamma_{min}^{(j_v)}} \right) \qquad \forall v \in V, \tag{4.1}$$

where

- $\epsilon_{min}^{(j_v)}$ and $\epsilon_{max}^{(j_v)}$ are respectively the minimum and maximum energy-consumption values w.r.t. all the available vertices associated to block section j_v and train i_v,
- $\gamma_{min}^{(j_v)}$ and $\gamma_{max}^{(j_v)}$ are respectively the minimum and maximum running-time values w.r.t. all the available vertices associated to block section j_v and train i_v.
- parameters w_0 and w_1 are the weights of relative importance of, respectively, energy and delay. The same weights are used in the objective function.

The so defined η varies between 0 and 1. It favors vertices representing faster and less energy-consuming partial speed profiles.

A pheromone update is applied after each iteration. First, a *pheromone evaporation* phase takes place, causing a fraction $\rho < 1$ of the current pheromone trails τ to be removed. Second, a *pheromone deposit* is performed on the arcs of the *best-so-far* (*bsf*)

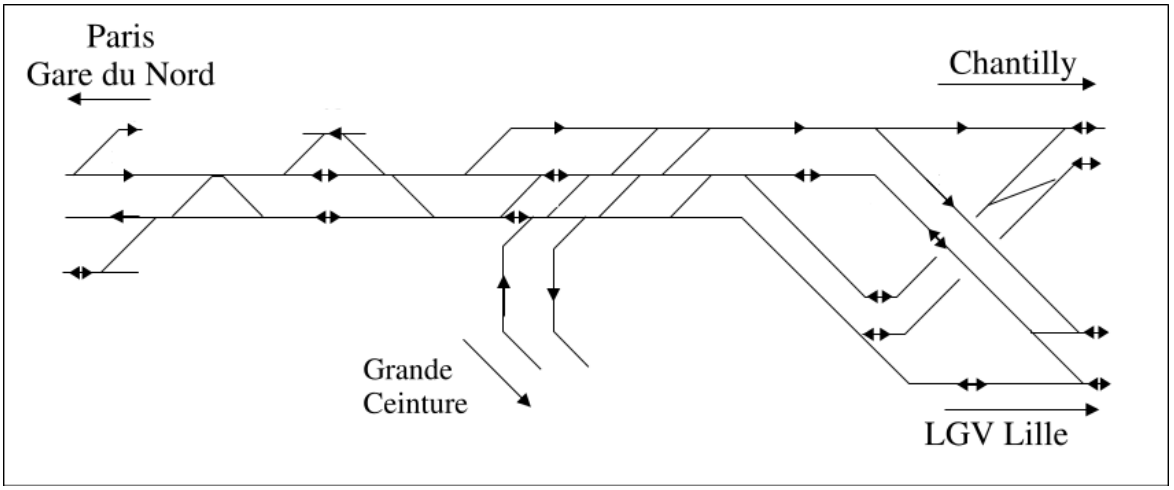


Figure 4.3: Network layout of the infrastructure in the Pierrefitte-Gonesse control area.

or the *iteration-best* (*ib*) solution. The use of one or the other is driven by parameter f_{ib}. Moreover, every τ is bounded between τ_{min} and τ_{max} (Stützle and Hoos, 2000).

4.5 Experimental setup

We test the ACO-rtECMP on the French Pierrefitte-Gonesse control area (Figure 4.3). It consists of an infrastructure with 89 track-circuits grouped into 79 block sections. The possible train routes are 39 and the control area does not contain stations. The traffic is dense: 336 trains are scheduled to traverse the control area in a classic week-day timetable. Three kinds of trains are used here, with different rolling-stock features. To construct a set of rtECMP instances, we first generate random perturbations of the timetable and construct the corresponding rtRTMP instances. To do so, following the literature (Pellegrini et al., 2015), we select a random sample containing 20% of the trains planned in a real week-day timetable. We then create 100 daily perturbations by delaying the nominal entry time in the control area, for each of the sampled trains, of a random amount between 5 and 15 minutes. For each perturbation, we consider the same 1-hour time horizon (06:00-07:00AM), which represents the morning peak time. This leads to a set of 100 rtRTMP instances comprising either 15 or 16 trains. By

employing the rtRTMP solver RECIFE-MILP (Pellegrini et al., 2015) to address the 100 perturbations, we finally obtain the routing and precedences required to construct the corresponding 100 rtECMP instances. They correspond to the best (possibly proven optimal) set of routes and schedules to minimize total delays found in the available computational time.

To ensure real-time applicability, the solution of both the rtRTMP and the rtECMP must be completed within a short computing time. Considering that, three minutes are often allocated for real-time traffic management (Samà et al., 2016), we assign 30 seconds to ACO-rtECMP imagining that the remaining 150 seconds are devoted to the rtRTMP. Such computing times, although to some extent arbitrary, are in line with the currently envisaged deployment of a TMS (Shift2Rail, 2019). In particular, it is in line with human-in-the-loop specifications: as dispatchers are meant to assess and possibly modify optimization decisions before their implementation, some minutes are to be appointed to decision making. Three of these minutes can reasonably be used for optimization.

To choose the appropriate parameter setting of ACO-rtECMP, we use IRACE (López-Ibáñez et al., 2016). It implements a set of machine learning tools for automatically configuring algorithms. Half of the rtECMP instances is used to tune the algorithm's parameters. The values tested for these parameters are: exponential weight of η and τ in the pseudo-random proportional rule in $\{1, \ldots, 10\}$, the number of ants in $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ and $\rho \in \{0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.4, 0.5, 0.6, 0.8\}$. We perform one *ib*-based pheromone update every 50 iterations, i.e., $f_{ib} = 1/50$. We allow a total of $181,500$ algorithm executions. The chosen algorithm configuration confirms our intuition on the aimed algorithm behavior. Indeed, the heuristic measure is a very good lead for ants' decisions. Its weight in the pseudo-random proportional rule is hence rather high compared to the pheromone's one (10 vs 2). Moreover, pheromone evaporation rate is low (0.05) so as to give time to colonies to build on the common knowledge cumulated throughout several iterations. A high number of ants (100) compose each colony to preserve a remarkable level of exploration.

Experiments are run on the 50 instances that were not used used for tuning. We test three configurations of weights (w_0, w_1) in the objective function: $(0.9999, 0.0001)$

where mostly the energy-consumption term matters; $(0.5, 0.5)$ where both energy and delay are equally weighted; $(0.0001, 0.9999)$ where mostly the delay term matters. We perform the tuning considering weights $(0.5, 0.5)$ and we consider the selected configuration for all cases. Preliminary analysis showed that the configuration performance is not very sensitive to these weights. We benchmark the results we obtain against those of a random search performed on the same instances and with the same computing time limit.

We implement ACO-rtECMP in C++ and run the experiments on a AMD Ryzen™ 7 2700X 8-core, 3.7GHz CPU with 64 GB of ram and Ubuntu 18.10 operating system.

4.6 Results

The results of our experimental analysis are reported in Table 4.1, in which the following aggregate indicators (w.r.t. the 50 tackled instances) are used:

- $\bar{E}$: mean energy consumption in the final solution, expressed in megajoules;
- $\bar{D}$: mean delay in the final solution, expressed in seconds;
- $\bar{I}$: mean percentage improvement of the final solution w.r.t. the corresponding first *ib* solution;
- *iter*: mean iteration at which ACO-rtECMP finds the best solution;
- *time*: mean time at which ACO-rtECMP finds the best solution;
- $\bar{B}$: mean percentage improvement of the final solution w.r.t. the best solution found by random search.

As expected, we can see in Table 4.1 that energy consumption decreases as more weight is given to it in the objective function. The opposite holds for delay. The highest value of $\bar{I}$ is obtained with $w_0 = 0.9999$, i.e., when all importance is given to energy consumption. This strongest improvement within the run of the algorithm does not depend on the total number of iterations performed. Indeed, on average, the total number of iterations completed are 252, 190 and 170 for $w_0 = 0.9999$, 0.5 and 0.0001, respectively. However, the best solution is found sooner in the first case, i.e.,

w_0	w_1	$\bar{E}$ $[10^3 J]$	$\bar{D}$ $[s]$	$\bar{I}$	*iter*	*time* $[s]$	$\bar{B}$
0.9999	0.0001	855	13,409	64%	92 (252)	12s	86%
0.5	0.5	2,119	8,248	28%	108 (190)	19s	48%
0.0001	0.9999	6,946	3,291	20 %	104 (170)	21s	26%

Table 4.1: ACO-rtECMP results on 50 instances (with 3 different objective function weight (w_0, w_1) configurations) and benchmarks against the corresponding random-search results. Time limit: 30 seconds.

at iteration 92 on average. We can observe a trend on $\bar{I}$, *iter* and *time*: the highest the importance of delay, the smaller the improvement of the algorithm with respect to the first *ib* solution and the latest the final solution is found, both in terms of number of iterations and of time elapsed. In all cases, the algorithm does not find any improving solution for about one third of the computational time available, hence longer runs would be useless. Column $\bar{B}$ shows that ACO-rtECMP always outperforms the random search, suggesting that the instances are not trivial. However, the difference is again smaller when delay is more important.

The lower $\bar{I}$ values for $w_0 = 0.5$ and $w_0 = 0.0001$ can be explained by observing the values of the heuristic information η with the different weight configurations. Indeed, this measure assumes very similar values when most weight is given to delay. As an example, Figure 4.4 reports a scatter plot of the values of η across all nodes for each chosen setting of w_0 in a representative instance. In Figure 4.4a, where $w_0 = 0.9999$, η varies in the whole range $[0, 1]$ and its values tend to be noticeably different from each other, although some denser clusters can be seen around values 0.35 and 0.65. In Figure 4.4b, where $w_0 = 0.5$, the range of η is compressed towards the upper bound so that η varies between approximately 0.5 and one for almost all nodes. Although being less evident, the same clusters appearing in Figure 4.4a are visible, thanks to the still remarkable importance of energy consumption which allows discriminating between nodes. In Figure 4.4c, in which $w_0 = 0.0001$, the extent of the compression is even more severe. Despite the presence of slightly more values lower than 0.5, the majority

of points are shifted towards one.

Apparently, for $w_0 = 0.5$ and $w_0 = 0.0001$, the reduced variation of η impairs its capability of guiding the search as it fails to bias the node selection probabilities. This forces each ant to perform a somehow more uniformly random selection of nodes until enough pheromone has been deposited and some convergence is achieved. This is confirmed by the fact that the increase of w_1 corresponds to a very large number of explored arcs (denoted as n_a): in case of $(w_0, w_1) = (0.9999, 0.0001)$, $n_a = 5$ millions arcs; in case of $(w_0, w_1) = (0.5, 0.5)$, $n_a = 18$ millions arcs; in case of $(w_0, w_1) = (0.0001, 0.9999)$, $n_a = 23$ millions arcs.

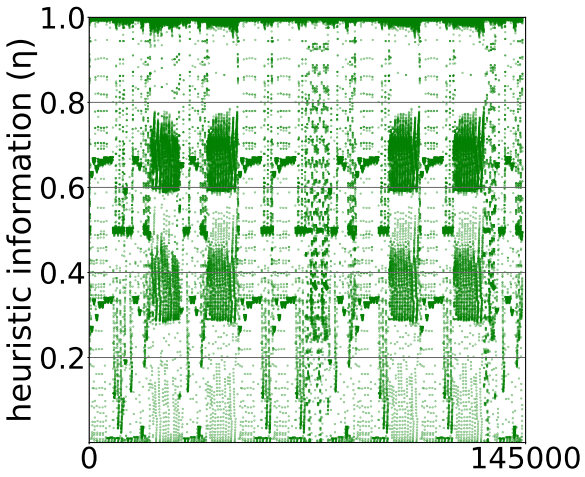
4.7 Conclusion

In this research, we developed a graph model and a meta-heuristic algorithm to address the rtECMP, which is the problem of minimizing the weighted sum of total delay and energy consumption given fixed train routing and precedences. In real-time traffic management, whenever operations are perturbed, an rtRTMP solver computes new routing and precedences that can be fed into an rtECMP solver. The latter computes a speed profile for each train traversing a control area in a given time horizon.

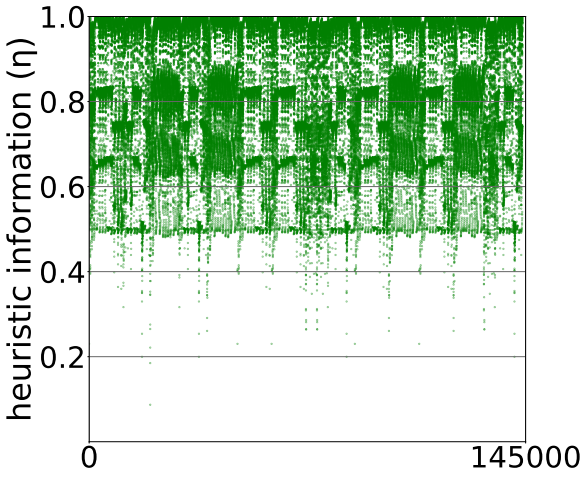
We modeled the rtECMP as a graph. A path that fulfills a set of constraints is a feasible solution. Each node of the graph is associated to a pre-computed partial speed profile for a train in a block section.

To solve the problem, we introduced ACO-rtECMP, an ant colony optimization algorithm that employs the aforementioned graph to construct solutions.

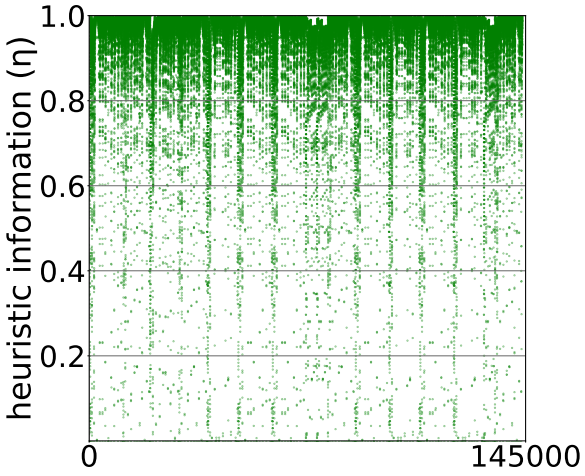
We tested ACO-rtECMP on instances derived from the peak-hour traffic in the French Pierrefitte-Gonesse control area. For each instance, we studied the improvement of the best ACO-rtECMP solution w.r.t. the best one found in the first iteration with a time limit of 30 seconds. First, to indicate the non-trivial nature of this problem, we showed how ACO-rtECMP results significantly outperform those of a random-search. Second, we assessed the impacts of shifting the optimization priority from energy consumption to delay reduction. By focusing on energy consumption minimization, the most remarkable improvements are obtained. This seems to be due



(a) $w_0 = 0.9999$



(b) $w_0 = 0.5$



(c) $w_0 = 0.0001$

Figure 4.4: Scatter plots of the heuristic information (η) in a representative instance for each configuration of objective function weights.

to the range of values assumed by the heuristic information used to guide ants' exploration of the search space. This is due to the fact that running times associated to many train-block section speed profiles being very similar to each other. This implies the inability of the algorithm to spot better and better feasible space regions throughout the run, and thus its stagnation in a low quality local optimum.

In future research, we will address whether a different definition of the heuristic information may improve the algorithm performance.

Another research direction consists of extending ACO-rtECMP with a local-search algorithm, which is generally known to improve solution quality in conjunction with meta-heuristics as ACO.

Future work will also be devoted to benchmarking the ACO-rtECMP results against an exact rtECMP solver, such as TDRC-MILP (Montrone et al., 2018), on different infrastructures.

Finally, an actual multi-objective definition of the rtECMP may be considered, instead of relying on a weighted sum of its two objectives.

Chapter 5

Paper-3: Energy Optimization of Multiple Trains in real-time Rail Traffic Management Considering Signals and the Interlocking System

5.1 Introduction

The European Union (EU) is committed in promoting environmental sustainability, especially to prevent global warming (EEA, 2021). Since transportation plays a large part in producing dangerous emissions, such as carbon-dioxide (CO_2) ones, environmental measures of the EU target the transportation sector as well. Railway is greener than most other transport modes as it is largely electrically powered and intrinsically energy-efficient, e.g., it is characterized by a low friction coefficient between wheels and rails (Scheepmaker et al., 2017). Apart from environmental sustainability, railway transport is also highly competitive in terms of safety and cost for both professional and private users. Europe is approximately covered by a 200,000-km railway network, making rail transport widespread and available for large portions of the population (EYoR, 2021). The EU is therefore pushing for greater adoption of rail transport for both goods and passengers. In this context, 2021 has been declared the *European Year*

of Rail (EYoR, 2021), and an awareness campaign has been launched by the EU aimed at promoting the mass use of rail transport. Moreover, the Community of European Railway and Infrastructure Companies has set a target for emission reduction of 30% by 2030 and another of 100% by 2050 (CER, 2021). Thus, not only does rail transport seeks to increase its users in order to preserve the environment, but it also strives to further reduce its carbon footprint in line with EU policies.

During operations, railway systems are subject to unforeseen events causing traffic perturbations. The latter give often rise to *conflicts*, that occur when at least one train encounters a restrictive signal that demands an unplanned deceleration. This causes delays w.r.t. to the nominal timetable and these delays can propagate to the whole system in a snow-ball effect. The task of managing traffic to reduce delay is entrusted to a *dispatcher*, who is in charge of a *control area*. This is a limited size railway network. Dispatchers can decide upon train-route assignment (routing) and train ordering along common passing locations (scheduling). These decisions have to be taken in a very short time frame to cope with unexpected perturbations. The problem faced by dispatchers is commonly referred to as *real-time railway traffic management problem* (rtRTMP) (Pellegrini et al., 2014) and its solution is a feature of modern rail *traffic management systems* (TMS).

The *real-time Energy Consumption Minimization Problem* (rtECMP), as introduced by Montrone et al. (2018), has the objective of minimizing both train energy consumption and total delay by deciding the speed profiles in a given control area and time horizon. It takes as input the decisions on train routing and precedences made by a rtRTMP solver. In addition, to define energy-efficient speed profiles for multiple interacting trains, it takes into accounts infrastructure characteristics, operational constants and train dynamics. The rtECMP ouputs include arrival, departure, passing through and dwelling times along with speed profiles. As the rtRTMP, given its real-time nature, the rtECMP has to be solved in a short computing time which starts when the TMS is called to address newly arisen perturbations. Under this computing time limit, one first needs to solve a rtRTMP instance, then solve the corresponding rtECMP instance.

In this paper, we extend the research of Montrone et al. (2018) by proposing a graph-based rtECMP model that we solve with three meta-heuristic algorithms: an Ant Colony Optimization (ACO), an Iterated Local Search (ILS) and a Greedy Randomized Adaptive Search Procedure (GRASP). These algorithms exploit a set of common algorithmic components: namely they use the same solution construction and local search procedures.

A thorough experimental analysis is conducted on two real life railway infrastructure (the Pierrefitte-Gonesse junction and a portion of the Paris-Le Havre line, both located in France) subject to various traffic perturbations.

The rest of the paper is structured as follows: Section 5.2 reviews the related literature; Section 5.3 defines the problem by also introducing the operational constraints and the characteristics of train speed profiles; Section 5.4 describes our graph-based model for the rtECMP;

Section 5.5 discusses the common components of the proposed algorithms, which are detailed in Sections 5.6-5.8; Section 5.9 explains our experimental setup; Section 5.10 analyzes the results; Section 5.11 exposes our conclusions.

5.2 Literature review

Minimization of energy consumption in train operations is the main concern of the literature on Energy-Efficient Train Control (EETC) and the Energy-Efficient Train Timetabling (EETT), of which Scheepmaker et al. (2017) provided a comprehensive review.

EETC methods focus on minimizing the traction energy consumption of one or multiple trains given a timetable, and therefore aim at computing energy-efficient speed profiles. It is also known as optimal train control, train trajectory optimization or optimal train trajectory planning.

ETTT methods focus on scheduling one or multiple trains in such a way that energy saving by EETC is considered along with either timetabling objectives, in case of nominal traffic, or re-scheduling objectives, in case of perturbed traffic.

In this section, we present the relevant EETC/EETT literature in which the operation of multiple interacting trains and their speed profiles are considered. We omit here the vast stream of research that is mostly focused on maximizing the effectiveness of energy recuperation though regenerative braking by synchronizing accelerations and brakings of nearby trains. For that, we refer the interested reader to, e.g., Li and Lo (2014b), He et al. (2020) and the review paper by Yang et al. (2016). Papers on nominal traffic are discussed in Section 5.2.1, while those on perturbed traffic are discussed in 5.2.2.

5.2.1 Nominal traffic

When energy consumption is to be optimized considering nominal traffic, the problem consists in optimizing a train speed profile while respecting the defined timetable.

Yang et al. (2012) proposed an EETT approach with the objective of finding a trade-off between total traction energy and total travel time in the railway network. A Genetic Algorithm (GA), which incorporated an EETC simulation module, was used to optimize the control of multiple trains running in a network by taking into account headways, speed limits and riding comfort.

Li et al. (2013) developed a timetable optimization model to minimize energy consumption, passenger travel times and greenhouse gas emissions. The model was solved with fuzzy non-linear optimization techniques. The solution approach was tested on instances with multiple trains running on a 10 station corridor.

Su et al. (2014) obtained a multi-train schedule in a metro system trough an iterative algorithm that tried different values of cycle time and fleet size. The algorithm incorporates an EETC module, based on a single-train integrated EETC-EETT model that the authors themselves proposed.

Wang et al. (2014) considered the EETC for a leading-following train pair with the objective of minimizing the total energy consumption for both trains.

Goverde et al. (2016), in a three-level timetabling framework, adapted a dynamic programming algorithm from Binder and Albrecht (2013) to adjust the departure and arrival times at the intermediate stops on lines between major stations for regional

trains. The algorithm jointly finds the time supplement distribution and the optimal speed profiles.

Zhou et al. (2017) proposed a model that integrates EETC and EETT. It is formulated as a path finding problem on a space-time-speed (STS) graph, obtained by discretizing space, time and speed values. The model considers a macroscopic representation of rail corridors with multi-platform stations. The proposed solution method is a dynamic programming algorithm.

Albrecht et al. (2018), extending their previous research (Albrecht et al., 2015), addressed the two-train separation problem. It consists in jointly finding the energy-efficient speed profiles for two trains traveling on the same track, in the same direction, subject to prescribed intermediate clearance times (enforced by a signaling system) and scheduled running times.

Wang and Goverde (2019) proposed an EETT approach based on a Multiple Train Trajectory Optimization (MTTO) model. Energy-efficiency was achieved by re-allocating the running times of a timetable given as input. The model was solved with the Pseudospectral method. The MTTO model relied on the optimal control formulation previously used in Wang and Goverde (2016) and Wang and Goverde (2017), discussed in Section 5.2.2.

Xu et al. (2020) proposed an integrated micro-macro EETT approach as a Mixed Integer Linear Program (MILP) that they reformulated as a STS network. The model was solved with a two-step algorithm. First the EETC was computed for each train individually. Second, potential conflicts were resolved. The infrastructure representation was macroscopic and speed profiles were simplified.

5.2.2 Perturbed Traffic

When energy is to be optimized in perturbed traffic, speed profiles must be decided so as to minimize delay propagation and possibly quickly recovering the nominal situation.

Wang and Goverde (2016) addressed the EETC for two successive trains with consideration of delays. Signaling constraints were incorporated in the Train Path

Envelope (TPE), which is a set of prescribed time and speed windows to be fulfilled by a speed profile at specific track positions. By considering multiple delays of the first train, the speed profile of the second train was computed with the objective to minimize a trade-off between delay and energy consumption. The Pseudospectral method was adopted to find optimal train trajectories.

Wang and Goverde (2017) proposed a MTTO model for single-track corridors with a focus on delay recovery. Precisely, they dealt with two trains on a single-track line which meet at multiple-track stations. The MTTO method computed multiple, conflict-free speed profiles simultaneously and chose meeting locations for opposite trains with the aim of energy saving and delay reduction. The MTTO was formalized as a multi-phase optimal control problem and was solved with the Pseudospectral method.

Galapitage et al. (2018) developed a system that combines real-time driving advice calculation with real-time junction scheduling, with the joint objective of reducing delay (at a junction entrance) and saving energy when traffic perturbations occur. Junctions were represented with great approximation, as nodes with a set of inbound/outbound traffic flows.

Luan et al. (2018b) proposed two models to integrate rtRTMP and EETC with a simplified speed profile representation. Energy consumption (which may benefit of regenerative brake) was accounted for as an additional objective along with delays. Both models were solved heuristically. The solution algorithm for the first model pre-computed a discrete set of speed profiles for each block section, which were then combined by a commercial MILP solver into a feasible schedule. The solution method for the second model consisted in a two-level heuristic approach that used a Genetic Algorithm and a commercial Non Linear Programming (NLP) solver. Experiments were carried out considering multiple trains in a corridor.

Li et al. (2020) applied the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to find energy-efficient delay recovery strategies for metro systems. When a train was delayed, both its running time and those of the subsequent trains had to be revised, along with the dwell times. Experiments were run on Chengdu Metro in China.

Montrone et al. (2018) formalized the rtECMP (presented in Section 5.1) as a MILP which requires pre-computation of a discrete set of energy-efficient speed profiles for each train-block pair. The infrastructure was modeled microscopically: a route-lock route-release interlocking system with 3-aspect signals was studied. Experiments were run on a control area with multiple trains traveling in a one-hour time horizon and intersecting routes. Several instances of re-scheduled traffic, including either 15 or 16 trains, were tested. Solutions were obtained with a commercial MILP solver.

We consider the rtECMP (Montrone et al., 2018) to be very relevant for the competitiveness of rail transport, which nowadays has earned even more attention due to its low environmental impact (Section 5.1). In particular, the rtECMP focuses on two crucial aspects of real-time traffic management, delay and energy consumption minimization. Furthermore, it considers accurate interlocking and signaling systems, frequently approximated or neglected entirely. From the reviewed literature, we believe that further research should be devoted to extend Montrone et al. (2018) as their proposed approach, a MILP addressed with a commercial solver, is not suitable for relevant-size instances. Therefore, in this paper, we propose three meta-heuristic algorithms to deal with the rtECMP. To do so, we first introduce a graph-based model for the problem. As an evolution w.r.t. the formulation in Montrone et al. (2018), we consider a more realistic interlocking system in which more than three signal aspects may exist, as described in Section 5.3.2.

5.3 Problem

As discussed in the introduction, the rtECMP asks for minimizing the overall energy consumption and delay of railway traffic when perturbations occur, in a control area in a given time horizon. To do so, it decides upon the speed profile of each train involved. Intuitively, slower profiles yield a lower energy consumption, but may increase delay. Conversely, faster profiles may help reduce delay at the cost of more energy. Furthermore, it is crucial for the chosen speed profiles to be compliant with traffic management decisions and railway safety requirements. In particular, they must be

compliant with the routing and precendences identified in an rtRTMP solution.

The problem constraints depend on two main factors: infrastructure characteristics and train physical capabilities impacting possible speed profiles, and operational rules. In the following, we focus on these two factors (Sections 5.3.1 and 5.3.2).

5.3.1 Speed profile definition

In this section, we report a brief outline of some of the major principles behind the computation of energy-efficient train speed profiles.

The characteristics of the engine and of the braking systems of a train are crucial for computing a speed profile. A train engine is capable of exerting a maximum force of $F_T(v) > 0$, which is a positive, non-increasing function of speed v. Usually, $F_T(v)$ is approximated as a piece-wise hyperbolic function. A maximum braking force $F_B < 0$, which is negative as it opposes train motion, can be exerted to decelerate the train or to stop it. F_B is often approximated as a constant value.

Train motion is opposed by a resistive force that we denote as F_r, which accounts for several factors such as rolling resistance, air resistance, track slope angle, weather phenomena, presence of tunnels and curved sections of track.

The most commonly used components to express F_r are: *vehicle resistance*, which accounts for both rolling resistance and air resistance and depends on speed and on rolling stock features; *line resistance*, which depends on the track slope angle; *curve resistance*, which accounts for the effects of curved track portions and depends on the track radius of curvature. Vehicle resistance is usually approximated using the *Davis Formula*, that is $A + Bv + Cv^2$, where A, B and C are non-negative constants that depend on the specific rolling stock.

In the literature, a train is usually approximated as a point mass. Newton's second law of motion is used to describe the relation between its speed v and time t or distance s. Here, we adopt time t as the independent variable. This results in a system of differential equations thanks to which we can compute the time spent by a train in a track portion depending on the speed profile $v(t)$ adopted, i.e., indirectly, on the force applied. This time corresponds to the so-called *running time* of a track portion, i.e.,

the time in which the head of the train travels within it. The same computation allows quantifying the *clearing time*, i.e., the time in which the head of the train has left the track portion but its tail is still there. We define *travel duration* the time required by a train for traveling from its origin to its destination.

To overcome resistance F_r and move along its route according to a speed profile $v(t)$, a train requires mechanical energy E, which is obtained by integrating the train's traction power $P_T(t)$ over time. In detail, the traction power and the mechanical energy can be respectively expressed as

$$P_T(t) = \frac{1}{2}\, F^+(t)\, v(t) \qquad (5.1) \quad \text{and} \quad E = \int_{\tau_0}^{\tau_1} P_T(t)dt, \qquad (5.2)$$

where $F^+ > 0$ is the traction force acting on the train at time t.

Finding a speed profile $v(t)$ that minimizes (5.2) constitutes a typical *optimal control problem*. Analyses based on the *Pontryagin's maximum principle* performed on the latter lead to the conclusion that an energy-optimal speed profile, when regenerative braking is neglected, is necessarily a sequence of the following *optimal driving regimes*:

- *maximum traction* regime: application of the maximum tractive effort $F_T(v(t))$ to accelerate a train;
- *cruising* regime: preservation of a net force equal to zero acting on a train, so that it moves with uniform speed;
- *coasting* regime: neither brake nor engine engagement, so that a train simply moves by inertia;
- *maximum braking* regime: excertion of maximum braking force F_B to decelerate the train.

We define a location where a switch from one driving regime to another occurs as a *switching point*. In general, an energy-efficient speed profile results from a driving-regime sequence having several switching points, in which each optimal driving regime may appear several times.

We refer the interested reader to, e.g, Albrecht et al. (2016c), Scheepmaker et al. (2017) and Goverde et al. (2020) for further details on optimal train control.

5.3.2 Operational rules description

For describing operational rules, we consider a microscopic representation of the infrastructure of a control area. Specifically, the infrastructure in a control area is composed of *track-circuits*. These are track elements fitted with an electric device capable of sensing the presence of a train. Track-circuits are grouped into *block sections*, which are track stretches that can be traversed by only one train at a time to maintain safe distancing. As mentioned in the introduction, every train is assigned a route in the infrastructure by a TMS. More precisely, we define a route as a sequence of block sections linking an origin-destination pair. Hence, a route can also be seen as a sequence of track-circuits. Figure 5.1 depicts an example of train routes in a simple infrastructure. Origin and destinations can either be a track-circuit at the limit with an adjacent control area or a station platform. Stations can be inside the control area.

To ensure the coherence of route formation, control areas are equipped with an *interlocking system*, which manages all necessary safety conditions to set a route and afterward allows a movement to take place. The safety conditions are complex relations between the state of different signaling system items and the route setting process of incompatible routes. Once the route conditions are met, the signal protecting the route can be cleared. Then, the signaling system comes into play by controlling signal aspects to ensure train separation with block sections: every block section is delimited by two signals placed at its entrance and exit location (see for instance Pachl 2002).

A signal is a device which conveys visual instructions to allow the driver to drive in full safety. Instructions are visualized in the form of so-called *aspects*. In the simplest case, a signal is capable of displaying three different aspects: green, yellow and red. A red aspect forbids the access to the following block section. A signal displaying yellow allows the access and demands a slow down so that a full train stop is possible before the following signal. Green grants the access to a block section and implies that a driver can safely enter the following one at full speed, if suitable.

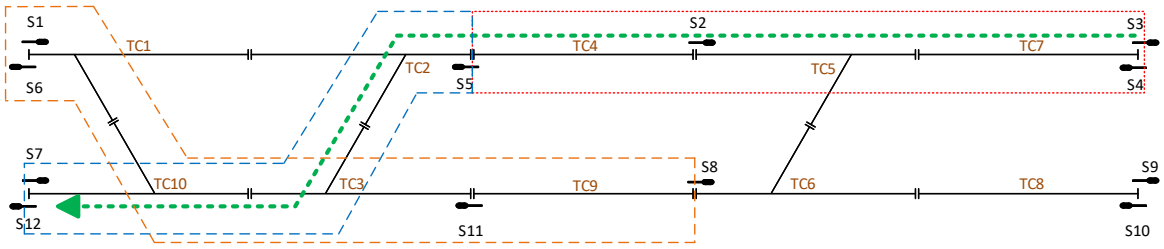


Figure 5.1: Example of a simple infrastructure in which TC denotes track-circuits, S denotes signals. The dashed green arrow highlights a train route composed of block sections S4-S5 (dotted red) and S5-S12 (dashed blue). Block section S1-S8 (dashed orange) is incompatible with S5-S12 as they share TC3 and TC10.

Each signal in a route may have a different number of aspects, up to some $n \geq 3$. This means that, when a n-aspect signal is crossed with a green, at least $(n-1)$ block sections are available in front of it (c.f. example E4 of Figure 5.2, $n = 5$). In addition to red, yellow and green, n-aspect signals can display further *restrictive aspects*. They gradually demand greater speed reductions to be reached by the end of the block section considered, up to a stop before a red signal. The signaling system ensures that the length of $(n-2)$ block sections is sufficient for a train to brake after the first restrictive signal, provided that the train type is authorized to use the infrastructure.

As an example, let $n = 5$, where green is separated from red by 3 restrictive aspects: *flashing-green, flashing-yellow, yellow*. A 5-aspect signal requires to consider the four blocks that come after it. Let us consider a sequence of blocks $(b_0, b_1, b_2, b_3, b_4, b_5)$ in a train's route, as shown in Figure 5.2. At the block sections' boundaries we have signals s_1, s_2, s_3, s_4, s_5. Signal s_1, which is between b_0 and b_1, is a 5-aspect signal. Suppose that a train is currently in b_0, approaching b_1. When the train is close enough (at visibility distance), the driver will be able to see s_1 at the beginning of b_1, i.e., at the end of b_0. When the train crosses s_1, its speed profile must be adjusted so that the following statements are true:

1. If green (G) is seen, no speed reduction is required (all blocks up to b_4 are accessible);
2. If flashing-green (fG) is seen, the train may have to fully stop before s_4 (b_4 cannot

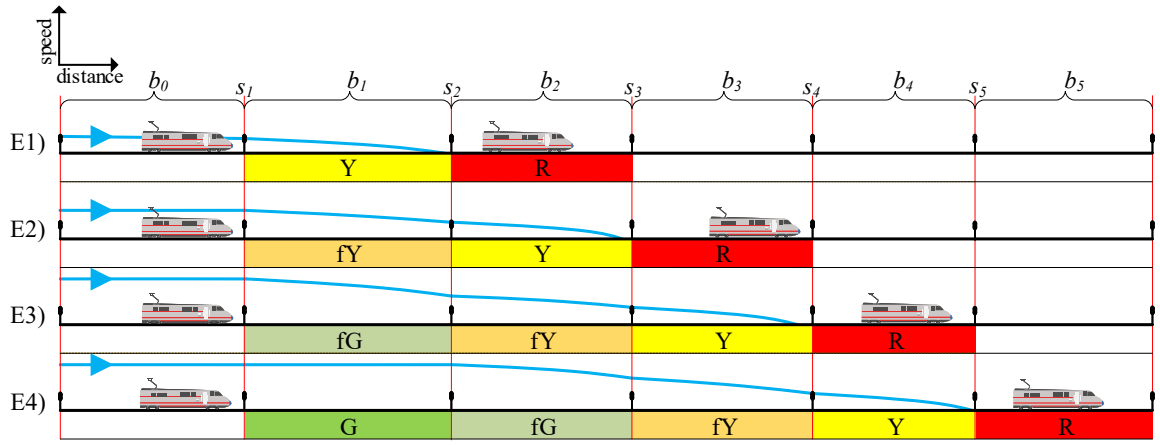


Figure 5.2: Example of speed profiles on a route composed of block sections $b_0, b_1, b_2, b_3, b_4, b_5$ separated by signals s_1, s_2, s_3, s_4, s_5. Four signaling scenarios are presented considering Red (R), Yellow (Y), flashing-Yellow (fY), flashing-Green (fG), Green (G). The aspects shown refer to the time at which s_1 is crossed.

be accessed), hence its maximum speed cannot exceed 160 km/h at the end of b_1 (relevant if the maximum speed of the line is higher than this threshold);

3. If flashing-yellow (fY) is seen, the train may have to fully stop before s_3 (b_3 cannot be accessed);
4. If yellow (Y) is seen, the train may have to fully stop before s_2 (b_2 cannot be accessed);
5. If red (R) is seen, the train brakes to full stop before s_1 (b_1 cannot be accessed).

These statements reflect the current French system. Some difference exists in the signaling systems in different countries, as for example the specific meaning of an aspect or the specific sequence of restrictive aspects. However, the same basic principle of progressive speed reduction is always applied.

Given a block section b, we refer to all block sections sharing at least one track-circuit with b as *incompatible* (see the example Figure 5.1, in which block section S5-S12 and S1-S8 are incompatible as they share track-circuits TC3 and TC10). When b is accessible, having its opening signal displaying an aspect different from red, a train can enter. Depending on the specific deployment, two kinds of interlocking systems

are possible: *route-lock route-release* and *route-lock sectional-release*. The difference lies in how incompatible block sections are made available to other trains.

- **route-lock route-release:** It implies that all the track-circuits of b have been released before the incompatible block sections of b itself are made available.
- **route-lock sectional-release:** It implies that, once a track-circuit of b is released, all block sections incompatible with b itself and not including any not-yet-released track-circuits are made available. When route-lock sectional-release is adopted, infrastructure capacity can be more extensively exploited.

For further details on interlocking systems, we refer the reader to Theeg et al. (2009).

In this paper, we consider the route-lock route-release interlocking system, following the seminal paper on the rtECMP by Montrone et al. (2018). However, the proposed approach can easily be extended to deal with the route-lock sectional-release system.

To implement traffic management decisions, we must take into account both the restrictions due to the interlocking and signaling systems. The interlocking system is in charge of setting block sections for trains so that the chosen route is indeed traversed in coherence with train precedence constraints. The signaling system indicates whether a block section is accessible at a given time, by displaying aspects other than red. Moreover, the signaling system demands speed restrictions to preserve safety distances between trains by means of restrictive signals. For instance, if a train t meets an aspect other than red entering a block section b, it must be true that: b is in the route of t; b is available to t according to the route-lock route-release policy; the precedence constraints of t w.r.t. b are satisfied and therefore t can traverse b if applicable. The signal that t meets entering b either displays a green aspect or a restrictive one depending on the traffic state.

In conclusion, solving an instance of the rtECMP means defining the speed profile of each train involved in the considered traffic situation (w.r.t. a given time horizon and control area). The problem's constraints, related to (I) train precedence, (II) routing and (III) safety constraints, are obtained by modeling the functioning of a control

area's interlocking and signaling systems, which enforce both traffic management decisions and railway safety regulations. The aim is to minimize both a function of delay and a function of the energy required to drive each train according to its chosen speed profile.

5.4 Model

In this section, we propose a model for the rtECMP as defined in Section 5.3. We formalize the problem as that of finding a path on a graph subject to a set of constraints. The notation and the graph definition are given in Section 5.4.1. The constraints are discussed in Section 5.4.2. Finally, the objective function is described in Section 5.4.3.

In the model definition, we consider the following assumptions:

A1 [*zero signal visibility distance*]: With respect to the railway signaling system as discussed in Section 5.3.2, as long as a signal is in a driver's field of vision, after a short reaction time, the driver can react to any change of aspect. In particular, under typical speed control systems, he can react to any change reducing the level of restriction imposed. This leads to an extremely large number of possible speed profiles (Montrone et al., 2018). For instance let us consider a train that has entered a block section with a yellow signal. The train decelerates for being able to stop before the following signal, as a red aspect is possible. At some point, the following signal is close enough to be visible and, due to the next block section having been released, the aspect is green. As a consequence, a stop is not required anymore and the train can re-accelerate. This translates into a double choice of speed profile for a single block section entered with the same restrictive signal aspect: one at the entrance and one at the latest between the moment in which the following signal is visible and the one in which it changes aspect. The location in which the train is when this second moment occurs also depends on the first speed profile chosen, as a faster one may imply the train to be closer to the next signal when its aspect changes.

As in Montrone et al. (2018), to contain the model's complexity, we assume

signal visibility distance to be zero. This implies that a signal's aspect can only be acknowledged as the train is about to enter a block section. This means that only one speed profile is to be chosen for each block section and, precisely, at the beginning of it. Let us consider the previous example again, this time applying the zero-visibility assumption. Incapable of seeing the signal aspect changing from red to green, the driver of the train simply brakes to a full stop at the block section's end. Only when the train is fully stopped, the green signal aspect is acknowledged and the train accelerates. This somehow mimics the functioning of specific speed control systems as, for example, the French KVB, in which a train that entered a block section with a yellow signal cannot exit it at more than 40 km/h regardless the aspect of the following signal. This assumption implies then a large energy consumption, due to the acceleration from null speed, and a long running time, which may imply delays. As both objective functions considered are penalized by the fact of crossing a restrictive signal, this assumption will result in the attempt to avoid as many of them as possible. Hence, a zero-visibility distance can be considered a reasonable assumption which does not defeat the purpose of energy efficiency and delay reduction. However, when restrictive signals are unavoidable, more costly brakings and accelerations will penalize the objective functions more than what happens in reality.

A2 [*discretization*]: We define a train speed profile as a combination of *partial speed profiles*. For each train and for each block section in the train's route, we pre-compute a discrete set of partial speed profiles. We then construct a graph (see Section 5.4.1) in which each node is a partial speed profile. This lets us define the rtECMP as a combinatorial optimization problem, in which deciding a speed profile for a train means choosing a node for each block section in the train's route. We remark that the model and a solution method we propose are entirely independent from the algorithm with which the partial speed profiles are pre-computed.

A3 [*prompt restart*]: A train that has stopped in a block section restarts as soon as the following block in its route becomes available. With this assumption and

the previous ones, the dwelling time of a train at a stop is fully determined by the selection of the nodes in the graph. In case of a scheduled stop, a minimum dwelling time and the scheduled departure times have to be considered.

5.4.1 Notation and graph definition

We are given a set of trains $i \in T$ and a route B_i for each of the trains. We refer to a generic block section in the route of train i as $j \in B_i$. Let S be the set containing $(i, j) : i \in T, j \in B_i$ such that either j is the last block section in the route of i (before the train leaves the control area), or j is where a scheduled stop for i is planned. For each $(i, j) \in S$, we denote the corresponding scheduled departure time as $\lambda_{i,j}$. For each $i \in T$ and $j \in B_i$, we are provided with train precedence in the form of a set $\mathcal{P}_{i,j}$ containing train-block pairs (i^*, j^*) in which block sections j^* are incompatible with j, therefore possibly including j itself, as a block section is incompatible with itself. Precisely, if one or more trains are to traverse j before i, one or more j^*'s will be equal to j. Only when every train i^* has released its corresponding block j^*, i can access j.

Let $G = (V, A)$ be a graph with V and A set of nodes and arcs respectively.

As stated in Assumption A2, we represent a speed profile as a concatenation of partial speed profiles, one for each block section in a train's route. For each block section $j \in B_i$, we pre-compute a discrete set of partial speed profiles having the following attributes: an initial speed; a final speed; a value of energy consumed; a running time; a clearing time.

Except for two special nodes, *source* (s) and *terminal* (t) (which are connected to the beginning and end of every path), each node in graph G represents a partial speed profile for a certain train-block pair. The attributes of a generic $v \in V$ are denoted by: i_v (train to which the node refers); j_v (the considered block section of B_{i_v}); k_v (initial speed); k'_v (final speed); r_v (running time of i_v in j_v); c_v (clearing time of i_v in the block section that precedes j_v); E_v (energy consumption).

The set of arcs A of graph G is defined as follows:

- For all $a, b \in V$ such that $i_a = i_b$, arc (a, b) exists if all following statements are true:

1. j_a directly precedes j_b in the train's route, and
2. the final speed associated to node a is equal to the initial speed associated to node b, i.e., $k'_a = k_b$.

- For all $a, b \in V$ such that $i_a \neq i_b$, arc (a, b) exists if, according to precedence constraints, one of the following statements is true:
  1. the decision on the event i_a traversing j_a is necessary and sufficient to establish the conditions at which i_b may traverse j_b, or
  2. the decision on the event i_a traversing j_a is independent on the event i_b traversing j_b.
- For all $v \in V$, arc (s, v) exists if the train-block pairs (i_v, j_v) verifies the following conditions:
  1. j_v is the first block section in the route of i_v;
  2. no other train precedes i_v in j_v, i.e., $\mathcal{P}_{i_v,j_v} = \emptyset$.
- For all $v \in V$, arc (v, t) exists if j_v is the last block section of the route of train i_v.

Let us consider the example in Figure 5.3. Two trains denoted as T1 and T2, are considered. The route of T1 comprises block sections A, B and C. The route of T2 comprises block sections D, E and F. The two routes intersect on blocks C and F, sharing two track-circuits. Hence, C and F are incompatible. Let us assume the following precedence constraint: train T2 must traverse F before T1 is allowed to traverse C. Let us assume a 3-aspect signaling system for the whole infrastructure.

An example $G(V, A)$, modeling the scenario in Figure 5.3, is shown in Figure 5.4. The nodes are partitioned into six clusters (colored rectangles), each associated to a given train-block pair: (T1, A); (T1, B); (T1, C); (T2, D); (T2, E); (T2, F). The black arcs connect compatible block section speed profiles for a train in terms of initial/final speed. In addition, they connect the source to the first block sections' nodes. Similarly, black arcs connect the last block sections' nodes to the terminal. For readability, we

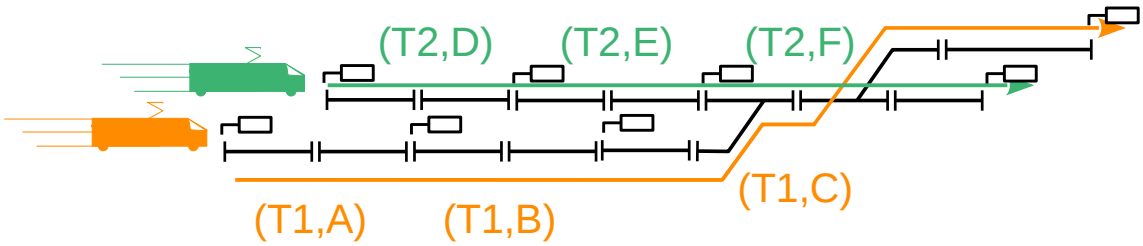


Figure 5.3: Example intersection described in Section 5.4.1 concerning the graph in Figure 5.4

introduce the blue arcs connecting clusters. A blue arc between two clusters means that each node in the first cluster is connected to each node in the second one. The blue arcs can either be unidirectional or bidirectional. In the example, the nodes in clusters (T1, A) and (T2, D) can be chosen in any order: a bidirectional blue arrow links the clusters. The same holds for the other clusters of T2 w.r.t. (T1, A). Instead, a unidirectional arc links (T2, F) and (T1, B).

Indeed, the aspect crossed by T1 entering B will be either yellow or green depending on whether block C is available when T1 is entering B. Due to the precedence constraint, block C becomes available only after that F has been cleared and released by T2. The time at which this happens is known only when the speed profile of T2 along its route, D-E-F, is fully known. According to the graph definition in this section, an arc from (T2, F) to (T1, C) could be included. However, it would never be used and thus it is not drawn in Figure 5.4. Indeed, deciding on a profile for (T1, C) requires first deciding on one for (T1, B), which is a decision only available after a profile has been decided for (T2, F).

A path h in G that satisfies a set of constraints (see Section 5.4.2) represents a solution of the problem. Indeed, h entirely describes the speed profile of each $i \in T$ as well as the dwelling time in the blocks where stops occur (see assumption A3). For each train-block pair $(i, j) : i \in T, j \in B_i$, we denote the actual departure time of i from j as $l_{ij}(h)$, which is a function of the chosen path h. While the scheduled departure time λ_{ij} is a fixed parameter for each instance, $l_{ij}(h)$ is a variable that depends on the chosen speed profiles.

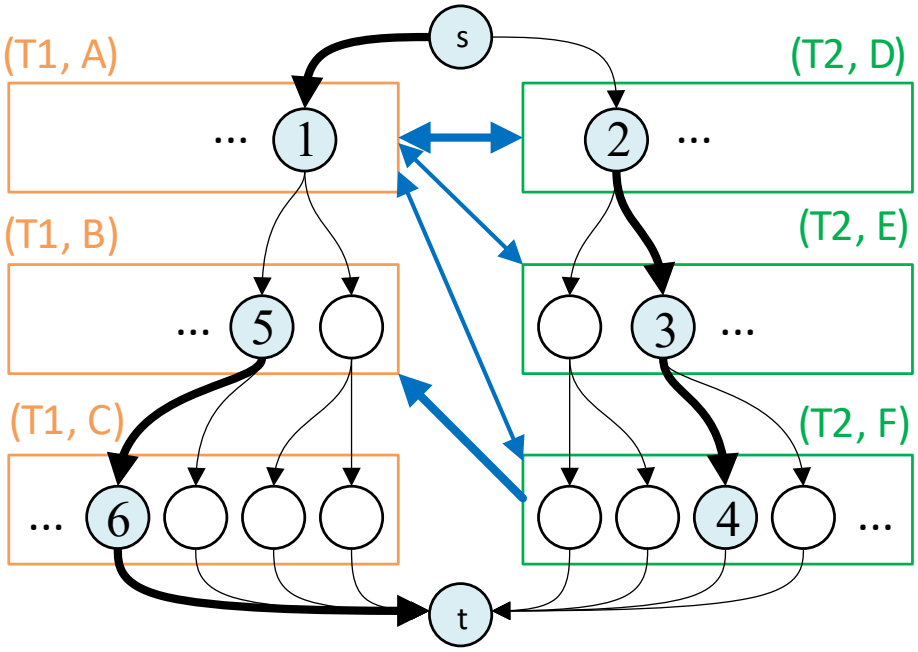


Figure 5.4: Example $G(V, A)$ described in Section 5.4.1. An example path s-1-2-3-4-5-6-t is displayed with its arcs and nodes highlighted.

Figure 5.4 also reports an example path s-1-2-3-4-5-6-t with thicker arcs. The path nodes are numbered and highlighted with a blue fill. Clearly, the first and last nodes are s and t. The other nodes, from 1 to 6, are respectively in clusters (T1, A), (T2, D), (T2, E), (T2, F), (T1, B) and (T1,C). This arrangement is consistent with the requirement of deciding the speed profiles for T2 in D, E and F before that of T1 in B.

5.4.2 Constraints

To be a feasible solution of the rtECMP, a path h in G must satisfy the following constraints:

C1 [*Partial speed profile selection*]: For each $(i, j) : i \in T, j \in B_i$, exactly one node v such that $(i_v, j_v) = (i, j)$ must be in h. This ensures that a choice of a partial speed profile for a train-block pair is made and is unique. In addition, a node must be in h at most once for each (i, j).

C2 [*Speed profile continuity*]: For each $v, w \in h$ referring to two consecutive block sections j_v and j_w that have to be traversed by the same train ($i_v = i_w$), it must be true that $k'_v = k_w$. This ensures speed profile continuity on the boundaries between every pair of consecutive block sections traversed by a train.

C3 [*Signaling*]: For each $v \in h$, k'_v must be coherent with the signal aspect opening block section j_v when i_v enters it. This constraint models the functioning of the signaling system as discussed in Section 5.3.

5.4.3 Objective function

To account for the two objectives of the rtECMP, i.e., energy efficiency and delay reduction, we *minimize* the objective function in (5.3). It is the weighted sum of two terms, E_h and D_h, which respectively account for the normalized total energy consumption and delay. The terms $w_0, w_1 \in [0, 1]$ such that $w_0 = 1 - w_1$ are weight parameters that control the importance of one objective against the other.

$$\min_h \{w_0 E_h + w_1 D_h\}, \qquad h \text{ is a path in } G(V, A) \text{ subject to } \mathbf{C1} \text{ to } \mathbf{C3}. \tag{5.3}$$

The energy consumption term in (5.3) is given by the following expression:

$$E_h = \frac{1}{\psi} \left(\sum_{v \in h} \frac{E_v}{\omega_{j_v}} \right) \tag{5.4}$$

where E_v is the energy consumption associated to node v (Section 5.4.1), ω_{j_v} is the maximum energy consumption value among the nodes associated to block section j_v and ψ is defined as $\sum_{i \in T} |B_i|$, where $|B_i|$ is the number of block sections in B_i. The terms ω and ψ are used to normalize E_h.

The delay term in (5.3) sums train delays at their exit from the control area and at each scheduled stop. It is given by the following expression:

$$D_h = \frac{1}{\phi} \left(\sum_{(i_v, j_v) \in S} \max \{\, 0, \;\; l_{i_v j_v}(h) - \lambda_{i_v, j_v} \,\} \right) \tag{5.5}$$

where ϕ is a normalization factor for D_h, defined as $\phi = \hat{d}\,|T|$ with $\hat{d}$ being a conventional value representing the maximum possible delay.

Remark that trains running ahead of schedule, for which $l_{i_v j_v}(h) - \lambda_{i_v,j_v} < 0$, do not contribute to the delay term D_h.

The term $l_{i_v j_v}(h)$ is equal to the earliest time at which train i_v can exit block section j_v, which is either the last one of its route or one in which it has a scheduled stop. For each block section, the earliest time at which the respective train can exit it is the minimum between the sum of the time at which the train itself enters the block section plus the running time corresponding to the speed profile represented by the node, and the time defined by the prompt restart assumption.

5.5 Algorithmic components

In this section, we detail the algorithmic components for the meta-heuristic algorithms proposed in Section 5.6, 5.7 and 5.8. Such components serve as shared building blocks composing the logic of such algorithms.

In Section 5.5.1, we present two constructive procedures for building feasible solutions.

In Section 5.5.2, we discuss a local search algorithm for improving solutions.

5.5.1 Constructive procedures

In this section, we propose a Stochastic Greedy (SG) and a greedy algorithm used as building blocks in the proposed meta-heuristic algorithms. The SG algorithm is a stochastic construction heuristic that incrementally builds a feasible path in G. Each node $v \in V$ in G is assigned a profit score σ_v. It is a problem-dependent quantity representing an heuristic measure of attractiveness of a node v. The higher the σ_v, the more likely it is for v to be added to a path being constructed by the SG algorithm. We define it as the relative gain in energy and running time w.r.t. the alternative nodes. Precisely:

$$\sigma_v = w_0 \left(\frac{\epsilon_{max}^{(j_v)} - E_v}{\epsilon_{max}^{(j_v)} - \epsilon_{min}^{(j_v)}} \right) + w_1 \left(\frac{\gamma_{max}^{(j_v)} - r_v}{\gamma_{max}^{(j_v)} - \gamma_{min}^{(j_v)}} \right) \qquad \forall v \in V, \tag{5.6}$$

where $\epsilon_{min}^{(j_v)}$ and $\epsilon_{max}^{(j_v)}$ are respectively the minimum and maximum energy-consumption values w.r.t. all the available nodes associated to block section j_v. The terms $\gamma_{min}^{(j_v)}$ and $\gamma_{max}^{(j_v)}$ are respectively the minimum and maximum running-time values w.r.t. all the available nodes associated to block section j_v and train i_v. The aforementioned ϵ and γ coefficients are introduced to normalize (5.6) so that σ varies between 0 and 1. The first term in round brackets is associated to the energy consumption of v, while the second to its running time. The value of σ increases as E_v or r_v decrease. As a consequence, σ indicates the greedy attractiveness of v and favors nodes representing faster and less energy-consuming partial speed profiles. In coherence with the objective function in (5.3), weights w_0 and w_1 are here used to determine the importance of each component of σ.

The SG algorithm selects a sequence of feasible nodes connecting the source node (s) to the terminal node (t). It adds one node $v \in V$ at a time. Each v is selected at random according to a non-uniform probability distribution, biased by σ_v. The probability of selecting a node v among the candidate nodes reachable from a parent node w is given by:

$$p_{w,v} = \frac{\sigma_v^\beta}{\sum_{l \in \mathcal{N}_w} \sigma_l^\beta} \tag{5.7}$$

where $\mathcal{N}_w$ is the neighborhood of w, v is in $\mathcal{N}_w$ and p_v is the probability of v being selected. The term β is a parameter of the algorithm that controls the impact of the difference between σ's on the difference of probabilities. Remark, only the nodes that contribute to satisfying the model constraints (Section 5.4) belongs to $\mathcal{N}_w$.

We also define a greedy algorithm as a deterministic variant of the SG algorithm. In this paper, we use it to initialize the proposed meta-heuristic algorithms. Precisely, the greedy algorithm construct a path in G by repeatedly selecting the node v with the highest σ_v among the available candidates. Therefore, as for the SG algorithm, (5.7) drives the selection of nodes in the greedy algorithm too. Due to its deterministic nature, the greedy returns an unique solution given an instance, β, w_0 and w_1. Therefore, it is suitable to be used as a mean of comparison in our experiments. In fact, we assess the performance of each proposed algorithm as the percentage improvement of the best solution w.r.t. to greedy solution. For further details see Section 5.10.

5.5.2 Local Search

In this section, we propose the Local Search (LS) algorithm used as a building block in the proposed meta-heuristic algorithms. It is detailed in Algorithm 3.

Algorithm 3: Local Search algorithm

Input: original path; k; first-improvement flag; fullSet; first-improvement

$h \leftarrow$ original path;
$N \leftarrow$ selection of k nodes in h to be replaced;
foreach $v \in N$ **do**
    $h^{(v)} \leftarrow \emptyset$;
    $f_{best} \leftarrow f(h)$;
    $W_v \leftarrow$ selection of the candidate replacement nodes for v according to fullSet;
    foreach $w \in W_v$ **do**
        $h' \leftarrow$ move(h,v,w);
        if h' *is infeasible* **then**
            $h' \leftarrow$ repair(h');
        end
        if $f(h') < f_{best}$ **then**
            $h^{(v)} \leftarrow h'$;
            $f_{best} \leftarrow f(h')$;
            If *first-improvement* then **Break**;
        end
    end
    If $h^{(v)}! = \emptyset$ **then** $h \leftarrow h^{(v)}$
end

Output: h

The LS algorithm, first of all, selects a set N of k nodes to be replaced in a given path h. The selection is based on a selection criterion (SC). They are alternatively the nodes: (SC1) resulting from a uniform random selection; (SC2) suffering the largest

delay at their exit from the infrastructure; (SC3) having the worst η value; (SC4) referring to the trains with highest criticality score. This score is a proxy of the number of restrictive signals of which a train may be responsible. For a train i, it is the sum of a criticality index associated to each of its blocks b: this is equal to the number of trains that follow i, i.e., the number of sets $\mathcal{P}_{i',j'}$ to which (i, b) belongs.

For each node v in N, the algorithm makes a list W_v of candidates for replacing it. These can either be all the nodes reachable from the node preceding v in h or a subset of them. In the latter case, the subset contains only the nodes whose exit speed is the same as the one of v. This is set through a binary parameter of the algorithm (*fullSet*).

We define a move of the LS algorithm as the replacement of one node in a path h. A function (*move*) is used to swap a node $v \in N$ with a replacement candidate $w \in W_v$.

A single move can make a path infeasible. Precisely, one or more of the nodes following a replaced node v could violate the problem constraints. If needed, a repair function is used. Starting from the first node after v, it sequentially replaces each infeasible node. Each replacement node for repair is chosen in the same way as the SG algorithm determines its moves (Section 5.5.1).

For each $v \in N$ to be replaced, the LS algorithm can either select the first $w \in W_v$ yielding an improvement in solution quality or the w yielding the best improvement overall. In the latter case, all the potential moves on v have to be evaluated. A binary parameter of the algorithm regulates whether the first or the second strategy is adopted (first-improvement).

5.6 Ant Colony Optimization

In this section, we propose an Ant Colony Optimization (ACO) algorithm to tackle the rtECMP model in Section 5.4. Precisely, we adopt a $\mathcal{MAX}$–$\mathcal{MIN}$ *ant system* ($\mathcal{MM}$AS) algorithm (Stützle and Hoos, 2000). We refer to it as ACO-rtECMP.

ACO is a well know meta-heuristic in which a colony of artificial ants cooperate in finding good quality solutions to hard combinatorial problems in a short computing

time. An artificial ant is a computational agent that implements a stochastic constructive heuristic for the problem at hand. Precisely, an ant figuratively moves from node to node in an ad-hoc *construction graph* adding, as it moves, each selected node to a partial solution being built. The ant can only select one of the nodes in the neighborhood of the last chosen node. The selection of each move is stochastic with a probability biased by two factors: *heuristic information* (η) and *pheromone trails* (τ). The former is a problem-specific quantity associated to nodes or arcs and representing a greedy measure of their attractiveness. The latter is a quantity representing the cumulated knowledge on the quality of previously visited solutions including the node or arc on which the pheromone is deposited. Artificial pheromone constitutes a mean of communication between ants: it is deposited by them throughout a run of the algorithm and it guides further ants' individual search. The algorithm is iterative: at each iteration, a group of artificial ants, named colony, stochastically builds solutions. Based on these solutions, the pheromone trail is updated to lead future ants towards promising regions of the search space.

In ACO-rtECMP, we adopt G as the construction graph. Here, ants construct paths in G subject to constraints C1-C3. They use the SG algorithm (Section 5.5.1) to build solutions, with a slight variation of the equation defining the probability of selecting nodes (7). Specifically, this probability is defined as:

$$p_{w,v} = \frac{(\tau_{w,v})^{\alpha}(\eta_v)^{\beta}}{\sum_{l \in \mathcal{N}_w} (\tau_{w,l})^{\alpha}(\eta_l)^{\beta}} \tag{5.8}$$

Here, the classic $\mathcal{MM}$AS product of heuristic information and pheromone trail is considered.

The heuristic information (η_v) is associated to nodes $v \in V$ in G, and coincides with σ_v of (5.6).

Instead, the pheromone trail $\tau_{w,u}$ is associated to arcs, to better capture the dependency of subsequent choices.

Similar to β, α is a parameter of the algorithm that weights pheromone trails. Equation (5.8) is commonly referred to as pseudo-random proportional rule. For details, see Dorigo and Stützle (2004).

Before the first iteration, we initialize all pheromone trails with the value τ_0 given by

$$\tau_0 = \frac{1}{\rho\chi\hat{Z}} \tag{5.9}$$

where $\hat{Z}$ indicates the objective function value of the solution returned by the greedy algorithm (Section 5.5.1).

After each iteration, a pheromone update is applied. First, a *pheromone evaporation* phase takes place, causing a fraction $\rho < 1$ of the current pheromone trails τ to be removed. Value ρ is a parameter of the algorithm. It is called pheromone evaporation rate.

After evaporation, a *pheromone deposit* is performed only on the arcs composing, alternatively, either the *best-so-far* (*bsf*) solution found or the *iteration-best* one (*ib*). Specifically, for each concerned $(w, v) \in A$, the corresponding $\tau_{w,v}$ is increased by $1/Z$ with Z denoting the objective function value of the solution being used. The frequency with which *bsf* and *ib* are used is controlled by a parameter of the algorithm, denoted as f_{ib}, and typically has an impact on the convergence behavior of the algorithm. The *pheromone matrix* we adopt for ACO-rtECMP is symmetric whenever possible, i.e., whenever both $(v, w) \in A$ and $(w, v) \in A$ exist.

As specified by the $\mathcal{MM}$AS paradigm, every τ is bounded between a τ_{min} and τ_{max} (Stützle and Hoos, 2000). The two bounds depend on the *bsf* solution. When a new *bsf* is found, the new values for τ_{min} and τ_{max} are given by

$$\tau_{max} = \frac{1}{\rho\,\chi\,Z^{bsf}} \tag{5.10}$$

$$\tau_{min} = \tau_{max}\,\frac{(1 - \sqrt[n]{0.05})}{(avg - 1)(\sqrt[n]{0.05})} \tag{5.11}$$

where, Z^{bsf} is the objective function value of the *bsf* itself and *avg* is a proxy of the average number of available choices for an ant at each step of solution construction. Terms τ_{max} and τ_{min} must be in $[0, 1]$. As our objective function Z is by definition smaller than one, we introduce term $\chi = 10,000$.

To improve solution quality of ACO-rtECMP at each iteration, we employ the LS algorithm introduced in Section 5.5.2, which we apply to the *ib* solution of each iteration. When the LS algorithm is used with ACO-rtECMP, it uses (5.8) instead of

using (5.7) to compute the selection probability of a candidate node. This holds for the repair procedure as well.

5.7 Iterated Local Search

In this section, to tackle the rtECMP model in Section 5.4, we propose an Iterated Local Search (ILS) algorithm (Stützle T., 2018). We refer to it as ILS-rtECMP.

ILS is a well-known meta-heuristic paradigm that iteratively applies both a subsidiary local search and a perturbation algorithm. This is meant to exploit the capabilities of a local search while preventing the premature convergence to a low quality local optimum. ILS starts by constructing an initial solution and then applies a local search algorithm to it. Then, the algorithm performs the following three steps during each iteration. First, the current solution is perturbed by applying moves in a larger neighborhood w.r.t. the one of the LS algorithm. Second, the perturbed solution is passed to the LS algorithm. Third, before the iteration ends, an acceptance criterion decides whether the next iteration will continue from either the solution returned by the LS or from the one before perturbation. When termination conditions are met, the best-so-far is returned.

For building the initial solution, we use the SG algorithm (Section 5.5.1).

As the subsidiary local search, we use Algorithm 3 (in Section 5.5.2).

As the acceptance criterion, we take all the solutions whose value is better than the best-so-far value. Solutions that are, on the contrary, worse than the best-so-far are accepted only if they satisfy a threshold of maximum percentage worsening w.r.t. the best-so-far value. Such threshold is a parameter of the algorithm (w_T).

As the perturbation, we propose a simple algorithm that, given a path h and an integer parameter k_p, repeats the following steps k_p times: (I) a random node in h is chosen and then replaced with another node, which is selected as in the SG algorithm (Section 5.5.1); (II) if needed, the repair function of Algorithm 3 is used; (III) finally, the procedure restarts from step (I) using the current h.

5.8 Greedy randomized adaptive search procedure

In this section, to deal with the rtECMP model in Section 5.4, we propose a Greedy Randomized Adaptive Search Procedure (GRASP) (Resende and Ribeiro, 2010). We refer to it as GRASP-rtECMP.

GRASP is a meta-heuristic whose iterations consist of, first, constructing a solution with a randomized greedy algorithm and, second, improving it through a local search.

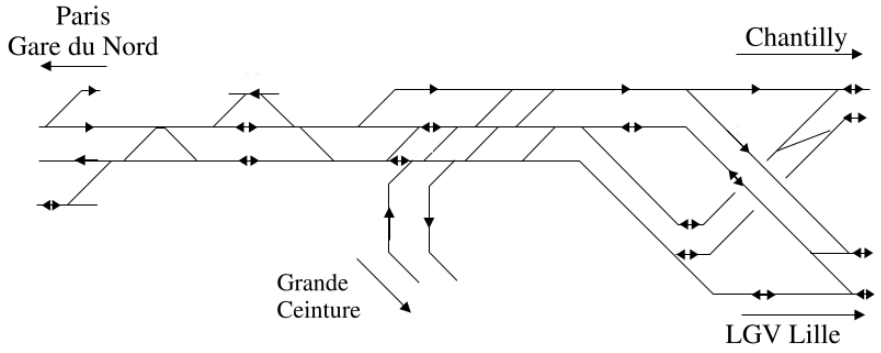
GRASP-rtECMP uses the SG and LS algorithms proposed in Section 5.5.1 and 5.5.2 respectively. The first algorithm constructs the solutions, while the second improves them at each iteration. The greedy algorithm (Section 5.5.1) is used in the first iteration of GRASP-rtECMP rather than the SG one.

5.9 Experimental setup

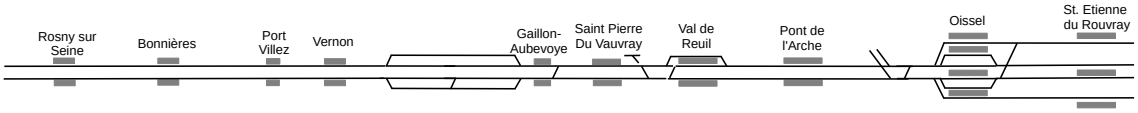
We test ACO-rtECMP, ILS-rtECMP and GRASP-rtECMP on two French control areas: the Pierrefitte-Gonesse junction (Figure 5.5a) and a section of the Paris-Le Havre line (Figure 5.5b). The Pierrefitte-Gonesse junction consists of an infrastructure with 89 track-circuits grouped into 79 block sections. The control area does not contain stations. The section of the Paris-Le Havre line we consider here comprises 236 track-circuits grouped into 150 block sections. It approximately covers an 80-km length between the stations of Rosny sur Seine and St. Etienne du Rouvray. In the rest of the paper, we will refer to this control area as Rosny-St. Etienne. The control area contains ten stations, some of which having multiple platforms that can be used in both directions.

The traffic is dense in both infrastructures. Precisely, the number of trains scheduled to circulate, in a typical week-day timetable, is around 330 and 220 for Gonesse and Rosny-St. Etienne respectively. Very different trains circulate in both control areas: high-speed, conventional passenger and freight trains.

To construct a set of rtECMP instances, we first generate random perturbations of the timetable and solve the corresponding rtRTMP instances. To do so, for each of the two considered control areas, we select a random sample containing 20% of the



(a) Pierrefitte-Gonesse control area.



(b) Rosny-St. Etienne control area.

Figure 5.5: Infrastructure layout of the considered control areas.

trains planned in a real week-day timetable. We then create a daily perturbation by delaying the nominal entry time in the control area, for each of the sampled trains, of a random amount between 5 and 15 minutes. By repeating this procedure, we obtain 100 perturbations per control area. For each perturbation, we consider the same 1-hour time horizon, i.e., 06:00AM-07:00AM, which represents the morning peak time. This leads to a set of 100 rtRTMP instances per control area, comprising either 15 or 16 trains. By employing the rtRTMP solver RECIFE-MILP (Pellegrini et al., 2015), we address the 200 perturbations. As a result, we obtain the routing and precedences required to construct the corresponding 200 rtECMP instances: 100 for Gonesse and 100 for Rosny-St. Etienne. They correspond to the best (possibly proven optimal) set of routes and schedules to minimize total delays found in the available computing time.

To ensure real-time applicability, the solution of both the rtRTMP and the rtECMP must be completed within a short computing time. Since three minutes are often allocated for real-time traffic management scenarios (Samà et al., 2016), we assign 150 seconds to RECIFE-MILP and the remaining 30 seconds to one of the three

algorithms (Sections 5.6, 5.7 & 5.8).

Such computing times are in line with the currently envisaged TMS deployment in some European countries (Shift2Rail, 2019). In particular, it is in line with human-in-the-loop specifications: as dispatchers are meant to assess and possibly modify optimization decisions before their implementation, some minutes are to be appointed to decision making. Three of these minutes can reasonably be used for optimization. Short term traffic prediction is exploited as a baseline for optimization (Quaglietta et al., 2016).

To compute the set of possible speed profiles for every train-block pair (i, j) : $i \in T, j \in B_i$, we utilize the Train Driving Regime Combination (TDRC) algorithm by Montrone et al. (2018). TDRC splits each block section j into q subsections of equal length, in which exactly one driving regime is to be used, if the infrastructure characteristics allows doing so. Given $(i, j) : i \in T, j \in B_i$, an initial speed k and a permutation of optimal driving regimes $\mathcal{C}$ containing exactly q regimes, TDRC computes a speed profile by applying each driving regime in C to its corresponding subsection. As a result, we obtain the attributes of each $v \in V$ of G (Section 5.4). In TDRC, we set $q = 3$ as it yields tractable values of $|V|$ and reasonably good solution values as shown in Montrone et al. (2018). Moreover, we exclude all profiles implying a running time higher than 10 minutes per block section as they would not be reasonable in practice. In our experiments, $|V|$ amounts to about $150,000$ and $230,000$ for Gonesse and Rosny-St. Etienne respectively.

The tuning of the three algorithms is carried out on half of the available instances for each control area as detailed in Section 5.9.1. Therefore, the experiments are run on the 100 instances that are not used for tuning: 50 for Gonesse and 50 for Rosny-St. Etienne.

We test the following three configuration of weights (w_0, w_1) of the objective function: $(0.9999, 0.0001)$ where mostly the energy term in (5.3) matters, denoted as "Energy"; $(0.5, 0.5)$ where both terms in (5.3) are equally weighted, denoted as "Both"; $(0.0001, 0.9999)$ where mostly the delay term in (5.3) matters, denoted as "Delay". Let us recall that we normalize the energy consumption and delay values so as each objective function component is within 0 and 1.

Initially, we measure solution quality of each proposed meta-heuristic algorithm by comparing it with the corresponding solution of the greedy algorithm of Section 5.5.1. For each combination of (w_0, w_1), the greedy algorithm is run on the same instance of each meta-heuristic algorithm and with the same value of parameter β. We consider the performance of the greedy algorithm as a baseline benchmark, which each meta-heuristic is expected to outperform. Then, we compare ACO-rtECMP, ILS-rtECMP and GRASP-rtECMP with each other to identify the best performing one.

Finally, we assess the impact of the best performing algorithm through a set of railway oriented indicators and we present a visualization of the speed profiles obtained.

We implement all algorithms in C++ and run the experiments on a AMD Ryzen™ 7 2700X 8-core, 3.7GHz CPU with 64 GB of ram and Ubuntu 18.10 operating system.

5.9.1 Parameter tuning

The tuning of ACO-rtECMP, ILS-rtECMP and GRASP-rtECMP is performed by using the IRACE package (López-Ibáñez et al., 2016), which implements a set of machine learning tools for automatically configuring algorithms. Half of the rtECMP instances referring to Gonesse and Rosny-St. Etienne is used to tune the parameters of the algorithms. We configure IRACE to perform a total of $25,000$ runs per algorithm.

As discussed in Sections 5.6-5.8, the three proposed algorithms use the same LS algorithm. Therefore, the LS parameters are common to all meta-heuristic algorithms. The LS parameters we tune are: b_I, which is set to "best" if the best improving candidate is selected for each $v \in N$, "first" otherwise; *fullSet*, which is set to "true" if all potential candidate replacement nodes are to be considered, "false" otherwise; the node selection criteria in {SC1, SC2, SC3. SC4}; $k \in \{20, 40, 80, 100\}$. Moreover, as all meta-heuristic algorithms use the SG one or a slight variation of it, its parameter β is in the set of common ones too, and its possible values are the integer numbers between zero and 20.

In addition to the common parameters, for ACO-rtECMP, IRACE tests the following parameter values: $\alpha \in \{1, \ldots, 20\}$; the number of ants in $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$; $\rho \in \{0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.4, 0.5, 0.6, 0.8\}$. We perform one *ib*-based

	α	ρ	ants	β	b_I	k	sel. crit.	*fullSet*
Gonesse	10	0.8	10	20	best	100	SC1	true
Rosny-St. Etienne	5	0.6	20	20	best	100	SC1	true

Table 5.1: ACO-rtECMP tuning.

	w_T	k_p	β	b_I	k	sel. crit.	*fullSet*
Gonesse	0	0	10	best	100	SC1	true
Rosny-St. Etienne	0	0	10	best	80	SC1	true

Table 5.2: ILS-rtECMP tuning.

pheromone update every 50 iterations, i.e., $f_{ib} = 1/50$, and we apply the LS algorithm to each *ib* solution of ACO-rtECMP.

For ILS-rtECMP, IRACE tests the following parameter values: $k_p \in \{0, 1, 2, 3, 4, 10, 20, 30, 40\}$; $w_T \in \{0\%, 1\%, \ldots, 15\%\}$.

GRASP-rtECMP has no additional parameters to be tuned after considering the common ones.

We run preliminary experiments on the Gonesse instances to understand the need for adopting a different tuning for each (w_0, w_1) combination. For them, we use ACO-rtECMP as it is rather sensitive to its configuration due to its numerous parameters (Pellegrini and Birattari, 2011). First, we tune ACO-rtECMP for each (w_0, w_1) combination. Second, for each of the three resulting parameter configurations, we run ACO-rtECMP with all (w_0, w_1) configurations. We observe no relevant difference in performance between the three tuning results. Specifically, no statistically significant difference emerges according to the *Wilcoxon Signed-Rank Test* with a 95% confidence level.

After observing this scarce sensitivity, we run control area-specific tuning for each meta-heuristic algorithm only considering $(w_0, w_1) = (0.5, 0.5)$, and we use the configuration obtained for all weight combinations. These configurations are reported in Tables 5.1, 5.2 and 5.3 for ACO-rtECMP, ILS-rtECMP and GRASP-rtECMP, respectively.

	β	b_I	k	sel. crit.	*fullSet*
Gonesse	20	best	100	SC1	true
Rosny-St. Etienne	19	best	100	SC1	true

Table 5.3: GRASP-rtECMP tuning.

5.10 Results

In this section, we discuss the results of the experiments performed in the setting presented in Section 5.9. First, in Section 5.10.1, we compare the results obtained with the three proposed meta-heuristic algorithms with the greedy algorithm of Section 5.5.1 and with each other, showing that ACO-rtECMP typically achieves the best results. Second, in Section 5.10.2, we discuss the impact of ACO-rtECMP from a railway point of view.

5.10.1 Comparison between the algorithms

As discussed in Section 5.9, the experiments are run for three weight combinations: "Energy" ($w_0 = 0.9999, w_1 = 0.0001$); "Both" ($w_0 = 0.5, w_1 = 0.5$); "Delay" ($w_0 = 0.0001, w_1 = 0.9999$).

Tables 5.4 and 5.5 report the results achieved by each meta-heuristic algorithm for Gonesse and Rosny-St. Etienne instances respectively. The following aggregate indicators, w.r.t. the 50 tackled instances for each control area, are shown in the columns:

- $\bar{I}$: mean percentage improvement of the best solution w.r.t. the corresponding solution of the greedy algorithm;
- $\bar{E}$: mean total energy consumption in the final solution, expressed in megajoules;
- $\bar{D}$: mean total delay in the final solution, expressed in seconds;
- Time: mean time at which the algorithm finds the best solution, expressed in seconds.

As discussed in Section 5.5.1, we adopt the solution of the greedy algorithm as a mean of comparison in assessing solution quality. This is done through the indicators $\bar{I}$ in Table 5.4 and 5.5. The comparison between the proposed algorithms is done through $\bar{I}$ as well.

Column $\bar{I}$ of Table 5.4 and 5.5 shows that ACO-rtECMP, ILS-rtECMP and GRASP-rtECMP always outperform the greedy algorithm. This shows the pertinence of using elaborated algorithms as the meta-heuristic ones, as the problem is not trivial and cannot be effectively solved by the one-shot greedy approach. As expected, we can see from Table 5.4 and 5.5 that $\bar{E}$ decreases progressively as more relevance is given to energy consumption in the objective function, i.e., when we move from Delay to Energy. The opposite holds for $\bar{D}$.

For the Gonesse instances (Table 5.4), the highest value of $\bar{I}$ is obtained with Energy, i.e., when all importance is given to energy consumption. We can observe a trend on $\bar{I}$: the highest the importance of delay, the smaller $\bar{I}$. While GRASP-rtECMP finds the best solution almost immediately, the other two algorithms do so after about 25 seconds. With Energy and Both, for ACO-rtECMP and ILS-rtECMP, the last solution improvements occur closer and closer to the time limit as we shift focus towards delay. In fact, with Delay, the best solution value keeps improving up to the time limit itself. This may suggest that, by increasing the time limit, better performance might be reached.

Table 5.4 shows that the difference between the performance of the three algorithms are quite small. However, they are rather regularly in favor of ACO-rtECMP, with ILS-rtECMP ranking second. These differences are statistically significant according to the *Wilcoxon Signed-Rank Test* with a 95% confidence level. Therefore, additional experiments would likely have ACO-rtECMP ranking first. However, the differences in the absolute value of the returned solutions especially for ACO-rtECMP and ILS-rtECMP, would likely be very small.

In the results reported in Table 5.4, $\bar{I}$ decreases as the importance of delay increases. This point was addressed in a similar context by Naldini et al. (2021) and is discussed in the following. As delay gains relevance in the objective function, the variation range

Algorithm	Weights	$\bar{I}$	$\bar{E}$ [MJ]	$\bar{D}$ [s]	Time [s]
ACO-rtECMP	Energy	48 %	155	4,265	20
	Both	15 %	180	3,863	23
	Delay	7 %	3,364	1,941	29
ILS-rtECMP	Energy	46 %	154	4,229	23
	Both	10 %	226	4,046	26
	Delay	7 %	3,454	1,947	30
GRASP-rtECMP	Energy	36 %	185	4,728	3
	Both	7 %	254	4,124	1
	Delay	3 %	4,522	2,015	1

Table 5.4: Results of the experiments for Gonesse instances.

of σ (or η) narrows progressively. As a result, σ becomes less effective in steering the stochastic search to better regions of the search space and, therefore, the improvement is less marked as w_1 increases. The narrowing of σ can be explained as follows. As w_1 is increased, the delay term in (5.6) gradually becomes more important and constitutes more of the value of σ. The variation range of the delay term depends on the one of the running time of the nodes in G, which is quite narrow on average. Hence, the range of σ narrows when the delay term is more relevant. For a thorough analysis, refer to Naldini et al. (2021).

For the Rosny-St. Etienne instances (Table 5.5), the highest value of $\bar{I}$ is obtained with Both rather than with Energy. Here, Energy yields the second best value of $\bar{I}$. Delay yields the minimum $\bar{I}$ as for Gonesse. Except for GRASP-rtECMP, the best solution value keeps improving up to the time-limit for Both and Delay as in Gonesse. With Energy, ACO-rtECMP and ILS-rtECMP find the last solution one to two seconds away from the time limit. Again, as in Gonesse, GRASP-rtECMP is the first to stop improving its solution, although this happens later in Rosny-St. Etienne. Table 5.5 shows that the differences between the performance of the three meta-heuristic algorithms are the same as in Gonesse. However, they are here larger in absolute value

Algorithm	Weights	$\bar{I}$	$\bar{E}$ [MJ]	$\bar{D}$ [s]	Time [s]
ACO-rtECMP	Energy	40 %	5,610	40,820	28
	Both	54 %	12,160	12,928	30
	Delay	9 %	19,658	8,479	30
ILS-rtECMP	Energy	39 %	5,695	42,609	29
	Both	53 %	14,133	12,595	30
	Delay	8 %	19,794	8,495	30
GRASP-rtECMP	Energy	33 %	6,340	42,787	5
	Both	36 %	12,894	20,925	8
	Delay	5 %	21,309	8,821	2

Table 5.5: Results of the experiments for Rosny-St. Etienne instances.

and therefore more marked. These differences are statistically significant according to the *Wilcoxon Signed-Rank Test* with a 95% confidence level, hence, confirming the algorithm ranking as in Gonesse: ACO-rtECMP, ILS-rtECMP, GRASP-rtECMP.

As mentioned in Table 5.5, $\bar{I}$ is best for Both in the for Rosny-St. Etienne results, which is the most marked difference w.r.t. the Gonesse results, where $\bar{I}$ is best for Energy. This point is discussed in the following.

To explain this difference, let us first observe that stops are scheduled for most trains in the Rosny-St. Etienne instances. As delay at each scheduled stop contributes to the objective function, multiple positive delay terms appear if a train traverses the whole infrastructure late. Due to this multiplicity, the objective function is very sensitive to delay. This is true especially for delays occurring at early stops in a train route as they can impact the punctuality at the subsequent stops as well. For example, let us consider a train with n_s scheduled stops in its route and let us assume no running time buffers exist. In such setting, if a delay occurred at the first stop, it would be summed n_s times in the objective function, greatly worsening the total delay term. If buffers are present, as it is for Rosny-St. Etienne, delays can be absorbed to some extent. However, the multiplicative impact of a stop delay on the total delay term

cannot be eliminated. For what concerns energy consumption, scheduled stops have a sort of flattening impact. Indeed, trains, must stop and re-accelerate at least some times due to scheduled stops, which limits the flexibility in terms of energy.

The better value of $\bar{I}$ that is observed for Rosny-St. Etienne with Both is due to a poor total delay value of the greedy solutions. Indeed, decisions in the greedy algorithm are entirely based on σ (Section 5.5.1), where the running time in a block section is used as a proxy of the total delay term of the objective function. In Rosny-St. Etienne, the multiplicative impact of delay on the objective function value is not represented at all by running times. Therefore, variations of running times may not be very representative of their impact on total delay. With Both, the greedy algorithm tends to select rather slow running times, as σ also rewards energy efficiency. This often leads to extremely negative impacts on total delay, which are not predicted at all by the running times themselves. In contrast, with Delay and Energy, the poor-quality proxy is not an issue. With Delay, σ only rewards minimal running times, which is indeed an effective strategy for the greedy to minimize total delay. With Energy, delay is ignored and so is the low representativeness issue.

In contrast, in Gonesse, the variation of delay and energy in the objective function is reversed to some extent. The former is much thinner as no stations are present, and therefore no scheduled stops. If conflicts allow it, to only minimize energy, a train may coast along its entire trip, while, to minimize delay, it could select the fastest driving regime available in each block section. Therefore, energy consumption can vary largely in Gonesse. Instead, the differences in travel time, and hence in delay, are quite small, as the routes are much shorter than those in Rosny-St. Etienne.

Since ACO-rtECMP and ILS-rtECMP, often, find the best solution either near or at the time limit, we run additional experiments with an extended computing time: five minutes instead of 30 seconds. In Gonesse, the results show a slight improvement of $\bar{I}$. With ACO-rtECMP, we have that $\bar{I}$ is 54%, 18% and 9% for respectively Energy, Both and Delay. With ILS-rtECMP, we have that $\bar{I}$ is 55%, 13% and 9%. In Rosny-St. Etienne, with both ACO-rtECMP and ILS-rtECMP, we have that $\bar{I}$ is 41%, 58% and 9% for respectively Energy, Both and Delay. In all these experiments, the best solution

is found between about 100 and 200 seconds. Again, we run a Wilcoxon Signed-Rank Test with a 95% confidence level. This time, however, the differences in performance do not result statistically significant. Therefore, no statement can be made on which algorithm would likely yield the best results if additional experiments were made. Since, in this paper, we aim at short computing times, we consider ACO-rtECMP the best as per the 30-second experiment results.

In our opinion, the rtECMP, by nature, is well suited to be addressed with a constructive, incremental approach such as ACO. Indeed, from a chronological point of view, the initial decisions made have a strong impact on the subsequent ones both in terms of feasibility and in terms of resulting quality. In an ACO algorithm, decisions can easily be made following both the chronological order and the one imposed by the precedence constraints. This allows ants to choose the speed profile of each block section being aware of the signal aspect opening it. Therefore, only the candidate profiles matching the necessary braking are considered and infeasible solutions are avoided entirely. Trajectory based algorithms, as ILS and GRASP, being based on the perturbation of solutions, cannot directly aim at feasibility in their exploration of the search space, implying a strong recourse to the repair mechanism. This hardly allows them to exploit the chronological and precedence relations between speed profiles. Furthermore, the rtECMP involves many complex space-time relations arising between groups of interacting trains. Another advantage of ACO is that it does not require these relations to be formalized *a priori* to be taken into account. Indeed, they are autonomously learned through the evolution of pheromone trails.

5.10.2 Practical impact analysis

In this section, we discuss the results of ACO-rtECMP from a railway oriented point of view. We consider the following average indicators, computed over all instances:

- acc, cru, coa, brk: number of subsections (Section 5.9) in which each driving regime is used: acceleration (acc), cruising (cru), coasting (coa) and braking (brk);

- stops: total number of unscheduled stop;
- v^{min}, v^{Max}: minimum and maximum speed held by the trains, expressed in meters per second;
- G, Y, fY, fG: total occurrences of each aspect when trains cross signals: Green (G), Yellow (Y), flashing-Yellow (fY), flashing-Green (fG). The yellow signals caused by scheduled stops are ignored. Since Rosny-St. Etienne does not include signals with either four or five aspects, only G and Y are shown.

These indicators are shown in Table 5.6 and 5.7 respectively for Gonesse and Rosny-St. Etienne. We group the data by objective function weights as in Section 5.10.1.

From Table 5.6, we can observe that the number of accelerations and cruisings increase as the importance of delay increases, while the number of coasting decreases. This is sensible as faster profiles contribute to a lower total delay. The number of brakings and stops is approximately the same in each weight configuration. No scheduled stops are planned in Gonesse, as no stations are present. Being the number of stops imposed by traffic very low, the minimum speed averaged for all trains is grater than zero.

These results show lower number of brakings with Both compared to Energy and Delay. With Energy, most trains tend to coast their whole travel. Trains entering the control area first are undisturbed and generally they choose the slowest running times. Therefore, the trains entering later have to brake and then coast to maintain a safe distancing. This occurs more frequently with Energy rather than with Both, which explains the two additional brankings. With Both, trains tend to go faster, in average applying twice the cruising regime, rather than once. Finally, in Delay, trains tend to go as fast as possible and, therefore, they require longer braking distances to meet changing speed limits along their routes or to simply respect restrictive signals. As expected, with Delay the number of braking is hence the highest. Moreover, to shorten running times, the fastest regimes (acceleration and cruising) are used more often.

From Table 5.7, for Rosy-St. Etienne, we observe more or less the same trends

	acc	cru	coa	brk	stops	v^{min}	v^{Max}	G	Y	fY	fG
Energy	2	1	547	16	1	11	44	177	1	6	4
Both	3	2	546	14	1	13	44	178	1	5	4
Delay	134	29	385	18	1	22	56	185	1	2	1

Table 5.6: Railway indicators for Gonesse (average over 50 instances).

	acc	cru	coa	brk	stops	v^{min}	v^{Max}	G	Y
Energy	40	56	855	69	5	0	38	271	5
Both	110	138	686	86	3	0	44	271	3
Delay	307	92	511	111	3	0	44	272	3

Table 5.7: Railway indicators for Rosny-St. Etienne (average over 50 instances).

as in Gonesse for most indicators. In contrast to Gonesse, the number of cruisings is maximum with Both. In addition, the number of brakings increases and the one of unplanned stops tends to slightly decrease as the relevance of delay increases. The higher number of stops with Energy, may be counter intuitive: as after stops, since trains necessarily have to accelerate, energy consumption is not negligible. However, consumption would be even higher if these stops were to be avoided: the algorithm chooses to exploit very rarely acceleration and cruising, although this imply additional stops.

The most marked difference w.r.t. Gonesse concerns the number of occurrences of the cruising regime: here it reaches a peak with Both. This peak is due to the total delay term dominating the objective function value, as discussed in Section 5.10.1. In fact, ACO-rtECMP mostly leverages on total delay minimization to improve the objective function value. To do so, it tends to return solutions in which trains travel rather fast: they accelerate after stops and then cruise at relatively high speeds. These speeds, imply rather long braking distances to deal with lowering speed limits and scheduled stops, hence requiring brakings in several block sections. Even higher speeds are chosen with Delay, which bring to more accelerations and brakings and fewer cruisings.

In both control areas (Table 5.6 & 5.7), we observe that green is markedly the most crossed aspect. This is due to the trains self-organizing into platoons, to some extent. This happens when groups of interacting trains run at a similar speed to maintain proper distancing. This let them avoid restrictive signals as much as possible, keeping traffic flow smooth. As a result, constraints C3 are very rarely active. Hence, they do not restrict the available speed profiles to be selected in each block section most of the times. Such behavior is especially noticeable in the Rosny-St. Etienne results, where the occurrence of green is practically unaffected by the weights of energy consumption and total delay in the objective function. Speed typically decreases when more weight is given to energy consumption (Table 7) but it is quite uniform across trains.

Figures 5.6 and 5.7 show the speed profile plot (distance-speed) of two trains selected from a representative instance of the Rosny-St. Etienne line. The speed is expressed in kilometers per hour and the distance in kilometers from the train entrance in the control area. The green vertical lines indicate where signals are located along each train's route. In each figure, three alternative speed profiles of the train are presented. The profiles respectively correspond to those obtained with Delay, Both and Energy, from top to bottom. As we can observe, the more relevant the delay (the upper the plot in the figure), the highest the average speed in each profile. In fact, especially in Delay, the trains maintain on average a faster pace to preserve punctuality w.r.t. the timetable. Indeed, as energy is not important, they tend to accelerate as much as possible if they need to recover some delay. In some cases, the presence of buffer allows not suffering any delay even if the speed is reduced eventually. An example of this can be seen in Figure 5.6 (top), where the train slows down before the third stop. As energy consumption importance is almost negligible here, the train slows down and then re-accelerates rather than using some less energy demanding regime. When energy is not neglected, as in the middle plot of Figure 5.6, cruising and coasting are preferred. As a corollary remark, let us note that the braking of the train before the second stop in the top and middle plots of Figure 5.6 are due to the fact that the train must go through a portion of track with a lower speed limit at Km 23. This is not visible in the bottom plot as the train keeps coasting and cruising after the initial

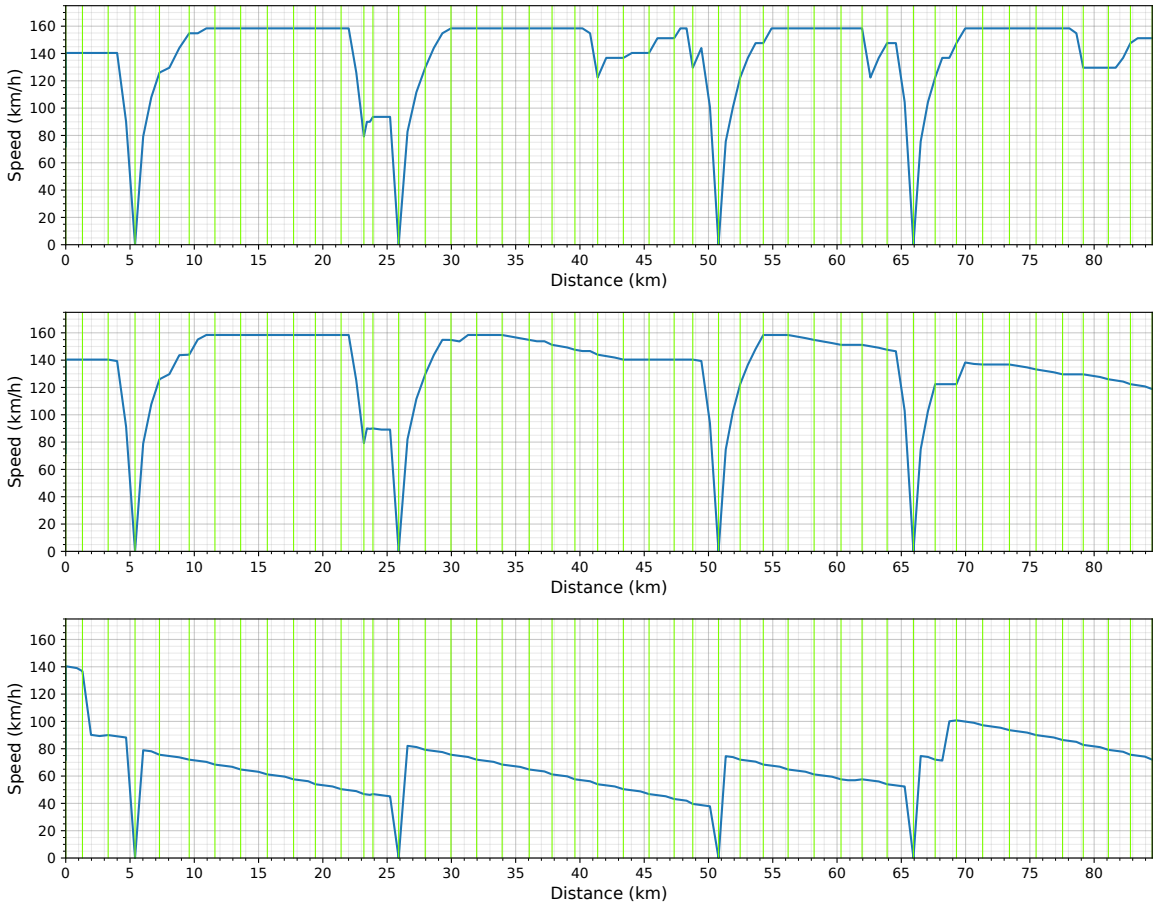


Figure 5.6: Speed profile (distance-speed): Delay (top), Both (middle), Energy (bottom).

acceleration at departures, and gets to this point at a low enough speed. In general, in the two figures we see that as energy is prioritized (the lower the plot), the speed profile progressively shift from having high speed cruisings and frequent accelerations to having more extended coasting and low speed cruising. When the optimization focus is entirely on energy, coasting makes up for the majority of the speed profile.

Despite differently in various speed profiles, Figures 5.6 and 5.7 show quite clearly the effect of the decision decomposition in terms of block sections and subsections. A quite evident case is in the top plot of Figure 5.6, where the train brakes to immediately re-accelerate at about Km 41. Right after that, the train performs a series of repeated acceleration-cruising sequences within the next four block sections. Another case is in

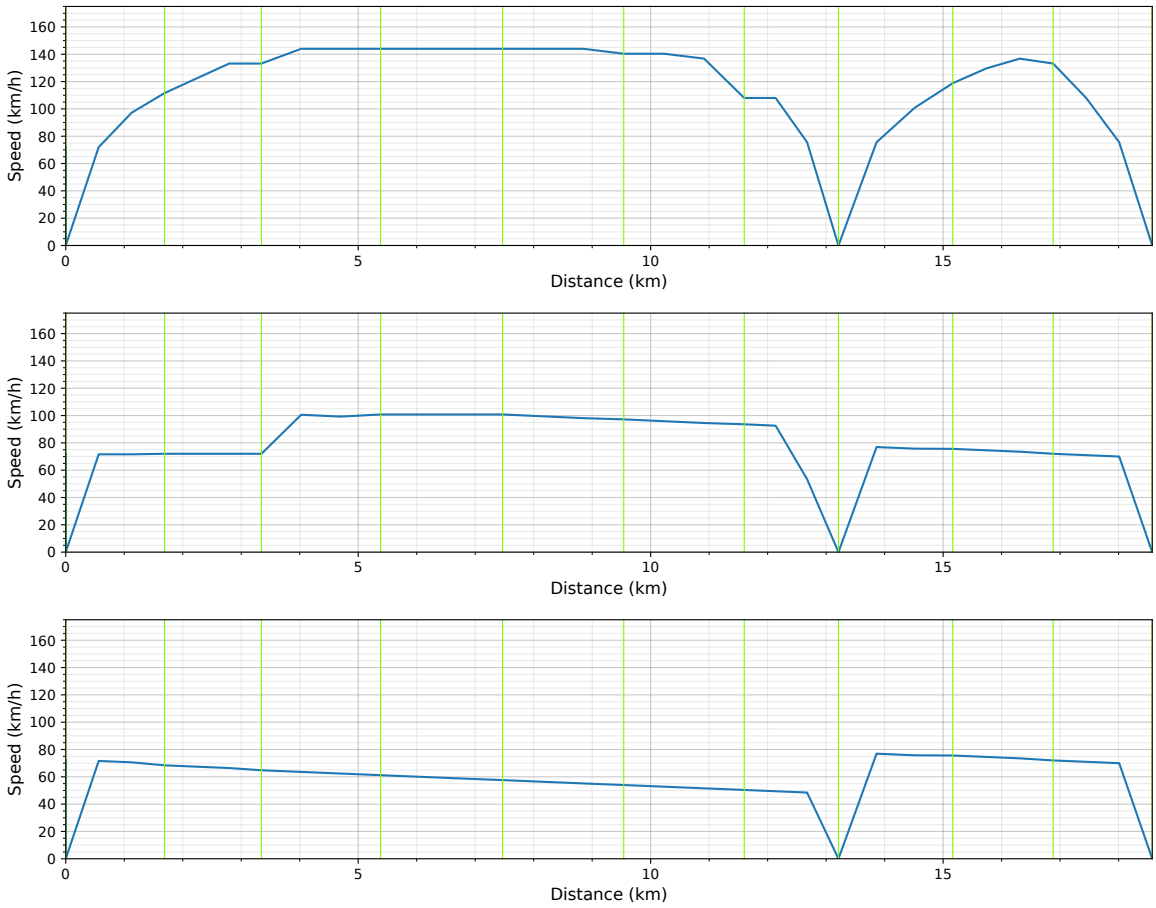


Figure 5.7: Speed profile (distance-speed): Delay (top), Both (middle), Energy (bottom).

the top plot of Figure 5.7, where the train, between the second and the third block section, switches from acceleration to coasting and then back to acceleration again. Indeed, these are rather impractical speed profiles for a train, which in general will follow a more uniform profile characterized by one phase of cruising followed by one phase of coasting. Indeed, the more fragmented profile returned by ACO-rtECMP may be the best feasible one in some cases, due to traffic, but often the more uniform alternative would be possible. Typically, a uniform and a fragmented profile will be equally evaluated by the algorithm as far as their impact on delay and energy consumption is equal. A possibility to obtain more uniform profiles would be to include the number of regime switches in the objective function to be minimized. However, we think this would excessively burden the algorithm, worsening the solution quality w.r.t. the current objectives. Another possibility is the identification of track sections where speed profiles may be smoothed through a post-processing without impacting traffic.

5.11 Conclusion

In this research, we designed a graph model and three meta-heuristic algorithms to address the rtECMP (Montrone et al., 2018). This problem requires minimizing the weighted sum of total delay and energy consumption given fixed train routing and precedences. In real-time traffic management, whenever operations are perturbed, an rtRTMP solver computes new routing and precedences that can be fed into an rtECMP solver. The latter computes a speed profile and timing for each train traversing a control area in a given time horizon, considering their interactions. This research aims at supplying decision support tools to boost the performance of the railway system and, by doing so, to increase the sustainability and the efficiency of railway mobility.

In this paper, we modeled the rtECMP as a graph of which a path fulfilling a set of constraints is a feasible solution. Each node of the graph is associated to a pre-computed partial speed profile for a train-block pair.

To solve the graph model, we introduced three meta-heuristic algorithms: ACO-rtECMP, ILS-rtECMP and GRASP-rtECMP. We tested them on instances derived

from the peak-hour traffic in two French control areas. In both control areas and with all combination of objective function weights we tested, ACO-rtECMP achieved the best results in general. The difference is often quite small in absolute terms, but it is statistically significant.

After identifying ACO-rtECMP as the best algorithm, we observed the practical relevance of the solutions it returned. The analysis displays the clear difference of speed profiles followed by trains in different situations. In all cases, ACO-rtECMP manages to find profiles with which traffic flows quite smoothly across both control areas.

In future research, we will benchmark ACO-rtECMP against optimal solutions. To do so, we first need to extend the exact method TDRC-MILP (Montrone et al., 2018) to support a signaling system as general as the one considered in this paper. Then, we will use the optimal solutions it returns as a term of comparison.

Furthermore, we will introduce a multi-objective definition of the rtECMP, instead of relying on a weighted sum of its two objectives.

Focusing on the practical applicability of rtECMP solutions, we will also study the possibility of applying an *a posteriori* technique to obtain speed profile as uniform as possible. Indeed, currently our solutions may propose fragmented regimes, changing in all block sections, while train drivers typically prefer to change regime only few times between stops. However, we think the practical interest of a uniform profile rather than a fragmented one will largely decrease when a higher automation will be put in place and train control will not require the driver to actively implement regimes.

Finally, we will study the impact on the problem definition of two planned technical evolutions of the railway system, namely moving block and virtual coupling. These evolutions change the ratio at the basis of train safe separation, and may require a major change of the problem definition.

Chapter 6

Conclusions & future research

Saving energy is not only very important from an environmental point of view, but it also results in a significant cost reduction for transport companies, especially for railway ones. For this reason, it is important to take energy efficiency into account when scheduling trains. During train operations, traffic is subject to perturbations that may require adjustments to train schedule and routing in real-time. In fact, due to unforeseen events, such as, e.g., a train door not opening or a modest underestimation of passenger demand, one or more trains will run behind schedules forcing other trains to slow down or even make unplanned stops in turn. With this mechanism, delay propagates. Traffic controllers (dispatchers) are in charge of making sensible re-routing and re-scheduling decisions to minimize the overall delay in the control area they are assigned. In the literature, this is embodied by the so-called real-time Rail Traffic Management Problem (rtRTMP). In itself, the rtRTMP is a very hard problem and thus energy consumption is usually neglected, especially when we deal with several trains and more complex networks. Research in decision support to achieve energy-efficient train operations in real-time traffic management is an extremely important but immature field. It can be declined into two classes of problems: real-time Energy Efficient Train (rtEETC) Control and real-time Energy Efficient Train Timetabling (rtEETT). Contributions in this field are instrumental for the development of next-generation railway Traffic Management Systems (TMS) that can improve both the competitiveness and environmental sustainability of rail transport.

In this thesis, we addressed both classes of problems: rtEETC and rtEETT.

In Chapter 1, we first introduced the scope, presented the papers we included in this thesis, outlined their content and contribution, emphasized the scope limitations.

In Chapter 2, we presented an introduction to railway systems. We first discussed the phases of rail transport planning and its decision making process, declined in strategic, tactical and operational phases. For each phase, we presented the most relevant problems from the broader literature of rail transport.

In Chapters 3, 4 & 5, we collected our relevant papers on rtEETC and rtEETT.

In the first paper (Chapter 3), we dealt with an optimization problem of the rtEETC class: a single-train energy-efficient train control problem for real-time applications. It asks for finding a speed profile that minimizes energy consumption for a train with a give schedule. Realistic case studies are considered. The instances are tractable in real time and results are promising.

In the remaining two papers (Chapters 4 and 5), we dealt with an optimization problem of the rtEETT class: the real-time Energy Consumption Minimization Problem (rtECMP). It minimizes the weighted sum of the total energy consumption and delay by deciding the speed profiles for multiple trains. More in detail, the paper in Chapter 4 proposes an Ant Colony Optimization (ACO) algorithm and a graph-based model for the rtECMP. These results are extended in the last paper (Chapter 5) by introducing a Local Search (LS), an Iterated Local Search (ILS) and a Greedy Randomized Search Procedure (GRASP) algorithms. A detailed computational analysis is performed, considering also an additional infrastructure w.r.t. Chapter 4. The best performing algorithm turned out to be ACO. The instances from both infrastructures are tractable in real time and solution quality is promising.

In Chapter 3, we stated that our future research on the rtEETC will be focused on designing a way of using the proposed single-train approach as a module in a multiple-train context to obtain a rtEETT method. In particular, we will allow slight variations of the schedule of each train to achieve a trade-off between energy efficiency and punctuality. Moreover, energy recuperation by regenerative brake will be tackled.

In Chapters 4 and 5, we highlighted the following future research directions. First,

we will compare the quality of ACO solutions against optimal solutions by extending the MILP proposed by Montrone et al. (2018). Second, we will explore a multi-objective definition of the rtECMP. Third, we will study the impact on the rtECMP of two planned technical evolution for railway systems: moving block and virtual coupling. Fourth, we will evaluate whether a different formulation of the ACO heuristic information may lead to increased performances.

Bibliography

Abbink, E., Fischetti, M., Kroon, L., Timmer, G., and Vromans, M. (2005). Reinventing crew scheduling at netherlands railways. *Interfaces*, 35(5):393–401.

Abbink, E., Van den Berg, B., Kroon, L., and Salomon, M. (2004). Allocation of railway rolling stock for passenger trains. *Transportation Science*, 38(1):33–41.

Albrecht, A., Howlett, P., Pudney, P., Vu, X., and Zhou, P. (2015). Energy-efficient train control: The two-train separation problem on level track. *Journal of Rail Transport Planning & Management*, 5(3):163 – 182.

Albrecht, A., Howlett, P., Pudney, P., Vu, X., and Zhou, P. (2016a). The key principles of optimal train control-part 1: Formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points. *Transportation Research Part B: Methodological*, 94:482–508.

Albrecht, A., Howlett, P., Pudney, P., Vu, X., and Zhou, P. (2016b). The key principles of optimal train control-part 2: Existence of an optimal strategy, the local energy minimization principle, uniqueness, computational techniques. *Transportation Research Part B: Methodological*, 94:509–538.

Albrecht, A., Howlett, P., Pudney, P., Vu, X., and Zhou, P. (2016c). The key principles of optimal train control—part 1: Formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points. *Transportation Research Part B: Methodological*, 94:482 – 508.

Albrecht, A., Howlett, P., Pudney, P., Vu, X., and Zhou, P. (2018). The two-train separation problem on non-level track—driving strategies that minimize total re-

quired tractive energy subject to prescribed section clearance times. *Transportation Research Part B: Methodological*, 111:135 – 167.

Bettinelli, A., Santini, A., and Vigo, D. (2017). A real-time conflict solution algorithm for the train rescheduling problem. *Transportation Research Part B: Methodological*, 106:237 – 265.

Betts, J. T. (2010). *Practical methods for optimal control and estimation using nonlinear programming*. SIAM.

Binder, A. and Albrecht, T. (2013). Timetable evaluation and optimization under consideration of the stochastical influence of the dwell times. In *Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transportation Systems 2013*, pages 471–481.

Borndörfer, R., Schlechte, T., and Swarat, E. (2012). Railway track allocation-simulation, aggregation, and optimization. In *Proceedings of the 1st International Workshop on High-Speed and Intercity Railways*, pages 53–69. Springer.

Brunger, O. and Dahlhaus, E. (2009). Railway timetable and traffic: analysis, modelling and simulation, chapter running time estimation (4). *Hamburg, Germany*, pages 58–82.

Cacchiani, V., Caprara, A., Galli, L., Kroon, L., Maróti, G., and Toth, P. (2012). Railway rolling stock planning: Robustness against large disruptions. *Transportation Science*, 46(2):217–232.

Cacchiani, V., Di Carmine, A., Lanza, G., Monaci, M., Naldini, F., Prezioso, L., Suffritti, R., and Vigo, D. (2019). Energy-efficient train control. In *Advances in Optimization and Decision Science for Society, Services and Enterprises*, pages 57–68. Springer.

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. Feature Clusters.

Canca, D., De-Los-Santos, A., Laporte, G., and Mesa, J. A. (2019). Integrated railway rapid transit network design and line planning problem with maximum profit. *Transportation Research Part E: Logistics and Transportation Review*, 127:1–30.

Caprara, A., Fischetti, M., and Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations research*, 50(5):851–861.

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. L. (2006). A lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics*, 154(5):738–753. IV ALIO/EURO Workshop on Applied Combinatorial Optimization.

CER (2021). The community of european railway and infrastructure companies. `https://www.cer.be/`. Last accessed 2021-06-17.

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(3):54–67. Robust Rescheduling and Capacity Use.

Cordeau, J.-F., Toth, P., and Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation science*, 32(4):380–404.

Corman, F., D'Ariano, A., Hansen, I. A., and Pacciarelli, D. (2011). Optimal multiclass rescheduling of railway traffic. *Journal of Rail Transport Planning & Management*, 1(1):14–24. Robust Modelling of Capacity, Delays and Rescheduling in Regional Networks.

Corman, F., D'Ariano, A., Marra, A. D., Pacciarelli, D., and Samà, M. (2017). Integrating train scheduling and delay management in real-time railway traffic control. *Transportation Research Part E: Logistics and Transportation Review*, 105:213–239.

CotEU (2021). Council of the european union. `https://www.consilium.europa.eu/en/policies/climate-change/`. Last accessed 2021-07-21.

Dollevoet, T., Huisman, D., Schmidt, M., and Schöbel, A. (2012). Delay management with rerouting of passengers. *Transportation science*, 46(1):74–89.

Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization*. The MIT Press.

D'Ariano, A., Pacciarelli, D., and Pranzo, M. (2007). A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2):643–657.

EC (2021a). Council of the european union. `https://ec.europa.eu/clima/policies/transport_en`. Last accessed 2021-07-21.

EC (2021b). Council of the european union. `https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en`. Last accessed 2021-07-21.

EEA (2021). European environement agency. `https://www.eea.europa.eu/`. Last accessed 2021-06-17.

ERTMS (2021). The european rail traffic management system. `https://www.era.europa.eu/activities/european-rail-traffic-management-system-ertms_en`. Last accessed 2021-06-17.

EYoR (2021). The european year of rail. `https://europa.eu/year-of-rail/index_en`. Last accessed 2021-06-17.

Fayet, P. (2008). *Modélisation des réseaux électriques ferroviaires alimentés en courant alternatif*. PhD thesis.

Fioole, P.-J., Kroon, L., Maróti, G., and Schrijver, A. (2006). A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, 174(2):1281–1297.

Fischetti, M. and Monaci, M. (2017). Using a general-purpose mixed-integer linear programming solver for the practical solution of real-time train rescheduling. *European Journal of Operational Research*, 263:258 – 264.

Forsgren, M., Aronsson, M., and Gestrelius, S. (2013). Maintaining tracks and traffic flow at the same time. *Journal of Rail Transport Planning & Management*, 3(3):111–123. Robust Rescheduling and Capacity Use.

Fu, H., Nie, L., Meng, L., Sperry, B. R., 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.

Galapitage, A., Albrecht, A. R., Pudney, P., Vu, X., and Zhou, P. (2018). Optimal real-time junction scheduling for trains with connected driver advice systems. *Journal of Rail Transport Planning & Management*, 8(1):29 – 41.

González-Gil, A., Palacin, R., and Batty, P. (2013). Sustainable urban rail systems: Strategies and technologies for optimal management of regenerative braking energy. *Energy Conversion and Management*, 75:374–388.

Goossens, J.-W., van Hoesel, S., and Kroon, L. (2006). On solving multi-type railway line planning problems. *European Journal of Operational Research*, 168(2):403–424. Feature Cluster on Mathematical Finance and Risk Management.

Goverde, R. M., Bešinović, N., Binder, A., Cacchiani, V., Quaglietta, E., Roberti, R., and Toth, P. (2016). A three-level framework for performance-based railway timetabling. *Transportation Research Part C: Emerging Technologies*, 67:62 – 83.

Goverde, R. M., Scheepmaker, G. M., and Wang, P. (2020). Pseudospectral optimal train control. *European Journal of Operational Research*.

Goverde, R. M., Scheepmaker, G. M., and Wang, P. (2021). Pseudospectral optimal train control. *European Journal of Operational Research*, 292(1):353–375.

Guihaire, V. and Hao, J.-K. (2008). Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice*, 42(10):1251–1273.

Haahr, J. T., Pisinger, D., and Sabbaghian, M. (2017). A dynamic programming approach for optimizing train speed profiles with speed restrictions and passage points. *Transportation Research Part B: Methodological*, 99:167–182.

He, D., Yang, Y., Chen, Y., Deng, J., Shan, S., Liu, J., and Li, X. (2020). An integrated optimization model of metro energy consumption based on regenerative energy and passenger transfer. *Applied Energy*, 264:114770.

Heil, J., Hoffmann, K., and Buscher, U. (2020). Railway crew scheduling: Models, methods and applications. *European Journal of Operational Research*, 283(2):405–425.

Higgins, A. (1998). Scheduling of railway track maintenance activities and crews. *Journal of the Operational Research Society*, 49(10):1026–1033.

Hong, X., Meng, L., D'Ariano, A., Veelenturf, L. P., Long, S., and Corman, F. (2021). Integrated optimization of capacitated train rescheduling and passenger reassignment under disruptions. *Transportation Research Part C: Emerging Technologies*, 125:103025.

Howlett, P., Milroy, I., and Pudney, P. (1994a). Energy-efficient train control. *Control Engineering Practice*, 2(2):193 – 200.

Howlett, P., Milroy, I., and Pudney, P. (1994b). Energy-efficient train control. *Control Engineering Practice*, 2:193–200.

Jespersen-Groth, J., Potthoff, D., Clausen, J., Huisman, D., Kroon, L., Maróti, G., and Nielsen, M. N. (2009). Disruption management in passenger railway transportation. In *Robust and online large-scale optimization*, pages 399–421. Springer.

J.W. Davis Jr (1926). The tractive resistance of electric locomotives and cars. *General Electric Review*, 29:2–24.

Jütte, S. and Thonemann, U. W. (2012). Divide-and-price: A decomposition algorithm for solving large railway crew scheduling problems. *European Journal of Operational Research*, 219(2):214–223.

Lee, E. B. and Markus, L. (1967). Foundations of optimal control theory. Technical report, Minnesota Univ Minneapolis Center For Control Sciences.

Li, W., Peng, Q., Wen, C., Wang, P., Lessan, J., and Xu, X. (2020). Joint optimization of delay-recovery and energy-saving in a metro system: A case study from china. *Energy*, 202:117699.

Li, X. and Lo, H. K. (2014a). An energy-efficient scheduling and speed control approach for metro rail operations. *Transportation Research Part B: Methodological*, 64:73 – 89.

Li, X. and Lo, H. K. (2014b). Energy minimization in dynamic train scheduling and control for metro rail operations. *Transportation Research Part B: Methodological*, 70:269 – 284.

Li, X., Wang, D., Li, K., and Gao, Z. (2013). A green train scheduling model and fuzzy multi-objective optimization algorithm. *Applied Mathematical Modelling*, 37(4):2063 – 2073.

Luan, X., Wang, Y., De Schutter, B., Meng, L., Lodewijks, G., and Corman, F. (2018a). Integration of real-time traffic management and train control for rail networks - part 1: Optimization problems and solution approaches. *Transportation Research Part B: Methodological*, 115:41 – 71.

Luan, X., Wang, Y., De Schutter, B., Meng, L., Lodewijks, G., and Corman, F. (2018b). Integration of real-time traffic management and train control for rail networks - part 2: Extensions towards energy-efficient train operations. *Transportation Research Part B: Methodological*, 115:72 – 94.

Luijt, R. S., van den Berge, M. P., Willeboordse, H. Y., and Hoogenraad, J. H. (2017). 5 years of dutch eco-driving: Managing behavioural change. *Transportation Research Part A: Policy and Practice*, 98:46 – 63.

Lusby, R. M., Larsen, J., Ehrgott, M., and Ryan, D. (2011). Railway track allocation: models and methods. *OR spectrum*, 33(4):843–883.

López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., and Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43 – 58.

Magnanti, T. L. and Wong, R. T. (1984). Network design and transportation planning: Models and algorithms. *Transportation science*, 18(1):1–55.

Martínez Fernández, P., Villalba Sanchís, I., Yepes, V., and Insa Franco, R. (2019). A review of modelling and optimisation methods applied to railways energy consumption. *Journal of Cleaner Production*, 222:153–162.

Montrone, T., Pellegrini, P., and Nobili, P. (2018). Real-time energy consumption minimization in railway networks. *Transportation Research Part D: Transport and Environment*, 65:524 – 539.

Naldini, F., Pellegrini, P., and Rodriguez, J. (2021). Real-time optimization of energy consumption in railway networks. *Transportation Research Procedia. Accepted, in press.*

Naldini, F., Pellegrini, P., and Rodriguez, J. (TBD). Energy optimization of multiple trains in real-time rail traffic management considering signals and the interlocking system. *Under review for a major journal.*

Nielsen, L. K., Kroon, L., and Maróti, G. (2012). A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220(2):496–509.

Pachl, J. (2002). Spacing trains. *Railway Operation & Control*, pages 38–90.

Pachl, J. (2009). Railway operations processes. *Railway Signaling and Interlocking*, 25:39–60.

Pellegrini, P. and Birattari, M. (2011). Out-of-the-box and custom implementation of metaheuristics. a case study: The vehicle routing problem with stochastic demand. In *Intelligent Computational Optimization in Engineering: Techniques & Applications*, pages 273–295. Springer.

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. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2609–2619.

Pellegrini, P., Marlière, G., and Rodriguez, J. (2014). Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *Transportation Research Part B: Methodological*, 59:58 – 80.

Profillidis, V. (2016). *Railway management and engineering*. Routledge.

Quaglietta, E., Pellegrini, P., Goverde, R. M., 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: Emerging Technologies*, 63:23–50.

Rao, A. V. (2014). Trajectory optimization: a survey. In *Optimization and optimal control in automotive systems*, pages 3–21. Springer.

Resende, M. G. and Ribeiro, C. C. (2010). Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In *Handbook of metaheuristics*, pages 283–319. Springer.

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: Methodological*, 85:89 – 108.

Sapronova, S., Tkachenko, V., Fomin, O., Kulbovskiy, I., and Zub, E. (2017). Rail vehicles: the resistance to the movement and the controllability.

Scheepmaker, G. M., Goverde, R. M., and Kroon, L. G. (2017). Review of energy-efficient train control and timetabling. *European Journal of Operational Research*, 257(2):355 – 376.

Schlechte, T., Borndörfer, R., Erol, B., Graffagnino, T., and Swarat, E. (2011). Micro–macro transformation of railway networks. *Journal of Rail Transport Planning & Management*, 1(1):38–48. Robust Modelling of Capacity, Delays and Rescheduling in Regional Networks.

Schöbel, A. (2012). Line planning in public transportation: models and methods. *OR spectrum*, 34(3):491–510.

Shift2Rail (2019). X2rail-4: Advanced signalling and automation system - completion of activities for enhanced automation systems, train integrity, traffic management evolution and smart object controllers. https://cordis.europa.eu/project/id/881806/results. Last accessed April 6,2021.

Steinzen, I., Gintner, V., Suhl, L., and Kliewer, N. (2010). A time-space network approach for the integrated vehicle-and crew-scheduling problem with multiple depots. *Transportation Science*, 44(3):367–382.

Stützle, T. and Hoos, H. H. (2000). Max–min ant system. *Future Generation Computer Systems*, 16(8):889 – 914.

Stützle T., R. R. (2018). Iterated local search. In Martí R., Pardalos P., R. M., editor, *Handbook of Heuristics*, chapter 19, pages 579–606. Springer, Cham.

Su, S., Tang, T., Li, X., and Gao, Z. (2014). Optimization of multitrain operations in a subway system. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):673–684.

Theeg, G., Maschek, U., and Nasedkin, O. (2009). Interlocking principles. *Railway Signalling and Interlocking. International Compendium*, pages 61–112.

UIC (2021). International union of railways. `https://uic.org/sustainability/energy-efficiency-and-co2-emissions/`. Last accessed 2021-06-17.

Veelenturf, L. P., Kroon, L. G., and Maróti, G. (2017). Passenger oriented railway disruption management by adapting timetables and rolling stock schedules. *Transportation Research Part C: Emerging Technologies*, 80:133–147.

Wang, P. and Goverde, R. M. (2016). Multiple-phase train trajectory optimization with signalling and operational constraints. *Transportation Research Part C: Emerging Technologies*, 69:255 – 275.

Wang, P. and Goverde, R. M. (2017). Multi-train trajectory optimization for energy efficiency and delay recovery on single-track railway lines. *Transportation Research Part B: Methodological*, 105:340 – 361.

Wang, P. and Goverde, R. M. (2019). Multi-train trajectory optimization for energy-efficient timetabling. *European Journal of Operational Research*, 272(2):621 – 635.

Wang, Y., De Schutter, B., van den Boom, T. J., and Ning, B. (2014). Optimal trajectory planning for trains under fixed and moving signaling systems using mixed integer linear programming. *Control Engineering Practice*, 22:44 – 56.

Wang, Y., Ning, B., Van den Boom, T., De Schutter, B., et al. (2016). *Optimal trajectory planning and train scheduling for urban rail transit systems*. Springer.

Wang, Y., Schutter, B. D., van den Boom, T. J., and Ning, B. (2012). Optimal trajectory planning for trains under operational constraints using mixed integer linear programming. *IFAC Proceedings Volumes*, 45:13 – 18. 13th IFAC Symposium on Control in Transportation Systems.

Wang, Y., Schutter, B. D., van den Boom, T. J., and Ning, B. (2013). Optimal trajectory planning for trains – a pseudospectral method and a mixed integer linear programming approach. *Transportation Research Part C: Emerging Technologies*, 29:97 – 114.

Xu, Y., Jia, B., Li, X., Li, M., and Ghiasi, A. (2020). An integrated micro-macro approach for high-speed railway energy-efficient timetabling problem. *Transportation Research Part C: Emerging Technologies*, 112:88 – 115.

Yang, L., Li, K., Gao, Z., and Li, X. (2012). Optimizing trains movement on a railway network. *Omega*, 40(5):619 – 633.

Yang, X., Li, X., Ning, B., and Tang, T. (2016). A survey on energy-efficient train operation for urban rail transit. *IEEE Transactions on Intelligent Transportation Systems*, 17:2–13.

Yang, X., Li, X., Ning, B., and Tang, T. (2016). A survey on energy-efficient train operation for urban rail transit. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):2–13.

Ye, H. and Liu, R. (2017). Nonlinear programming methods based on closed-form expressions for optimal train control. *Transportation Research Part C: Emerging Technologies*, 82:102 – 123.

Zhan, S., Kroon, L. G., Veelenturf, L. P., and Wagenaar, J. C. (2015). Real-time high-speed train rescheduling in case of a complete blockage. *Transportation Research Part B: Methodological*, 78:182–201.

Zhou, L., Tong, L. C., Chen, J., Tang, J., and Zhou, X. (2017). Joint optimization of high-speed train timetables and speed profiles: A unified modeling approach using space-time-speed grid networks. *Transportation Research Part B: Methodological*, 97:157 – 181.