Port Rail Shunting Optimization Problems

Veronica Asta

Italian Center of Excellence on Integrated Logistics, Transports and Infrastructures (CIELI)

University of Genova

Supervisor

Professor Daniela Ambrosino

In partial fulfillment of the requirements for the degree of

PhD Course on Transports and Logistics

July 2021

Acknowledgements

There are two persons to whom I want to send my most heartfelt thanks: my academic supervisor Prof. Daniela Ambrosino and my company supervisor Dott. Luca Abatello. They both have completely belived in me since the beginning of this amazing path, giving me the strenght to face up each difficulty I met. They both have been my guide and my example in this jurney allowing and sometimes forcing me to perform an incredible and beautiful growth both human and professional.

Prof. Daniela Ambrosino, with her calm and kindness, tought me how to improve my research activities and brought me even more inside the operational research and the optimization methods applied to the transport field that appassionate me so much. She also showed me how to calm my feisty character when it could create me problems.

Dott. Luca Abatello, on the other side, with his sparkling character, stimulated me to use all my skills in a deep way bringing me completely inside the company heart. Even if it sometimes has been hard and out of my comfort zone, it has been the reason of my big human and professional growth inside the company. After my university years dedicated to study Logistics and Transports, this has been a great jurney completely in touch with the companies operating in this field.

Thanks to the balance, which sometimes it's been lost and then recovered, between the three of us, between the research and the real contexts, between scientific papers and walks inside the port area, I can say that it's been a pleasure and I am so proud of what I have done and what I can do now thanks to them.

Then, my most sincere thank to Prof. Teodor G. Crainic (Professor at UQAM, Canada), the supervisor of my visiting period at the *Centre interuniversitaire de reserche sur les réseaux d'entreprise, la logistique, et le trasport* in Montrèal (Canada). It has been a great pleasure to be able to work in such worldwide important research center thanks to him. I really have learnt so much and I will always keep everything inside of me. The visiting period allowed me to see the research world in another center and in another continent.

Moreover, I want to thank my mother and my father, who belived in me since I was a child, bulding inside of me all the resources and the securities that I needed to face up with this sometimes difficult path. They accompanied and supported me in all my studies years teaching me that *culture make us free*, and now that I am going conclude my PhD I can't do anything else to say thank to them, because without I wouldn't be here.

A special thank is for my sister, my eternal support, my eternal friend, the only that understands me when I don't say anything. She is always by my side, even when we are physically far away, and this is the best support she could give me.

Thanks to each one of my lovely friends, Erika, Silvia, Francesca, Rachele, Valeria, who carefully and sincerely allowed me to share with them both the difficulties and the successes of these years.

Thank you.

Abstract

The work focuses on a particular section of the intermodal chain of freight transportation, which is the link between rail and sea transportation modes and happens in the maritime port area. Among this field, the study deals with the management of rail operations, called here rail shunting operations, that have to be performed in the port area.

Two optimization problems arises in this context. The first concerns the scheduling of the rail shunting operations, here called Port Rail Shunting Scheduling Problem (PRSSP). The second deals with the re-scheduling of the same operations in case of unpredictable events, here called Port Rail Shunting Re-Scheduling Problem (PRSRP). After a literature overview on the concerning studies, we concentrate on an innovative way to use the well known space-time networks as solution approach structure for both the above mentioned problems. The innovative structure has been called operation-time-space network and is deeply analyzed in a dedicated chapter.

A network flow model based on an operation-time-space network for solving PRSSP has been developed. It has been tested using random generated instances providing good results. The same model has been extended in order to solve PRSRP and it has been tested giving good results as well.

Finally, the models have been used to solve the real case of a port area located in Italy in order to test the applicability of the developed models to a real context. The tests have been executed using real data and provided good results confirming the possibility to apply the proposed approach in similar real problems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context of the study

The present study is the result of a three years PhD Course in Logistics and Transports that has been done in collaboration with the company Circle SpA. This company provides process and management consulting services, innovative technological solutions and digital marketing solutions having specific vertical expertises in ports, maritime and intermodal logistics.

The work focuses on a particular section of the intermodal chain of transportation, i.e. the link between rail and sea transportation modes, that happens in maritime port area, here also called rail-sea yard. Within this particular field, the study is focused on the rail operations, called shunting operations, that have to be performed in the port area for transferring trains. The entire process is called port rail shunting process. The optimization problems addressed in this thesis are related to both the scheduling and the re-scheduling of the shunting operations with the aims of managing all the activities with the available resources.

The final objective of the study is to develop an optimization approach useful to solve and manage the port rail shunting scheduling and re-scheduling problems.

1.2 Port rail shunting process

This Section aims at detailing the rail shunting process of the port area (1.2.1) and the related shunting operations (1.2.2).

1.2.1 The process

The global freight transport is used to link the origin and the destination of the goods that have to be transferred. It might be either a simple link, when the origin and the destination are close to each other, or a complex system including more transport modes, when the origin and the destination are far away.

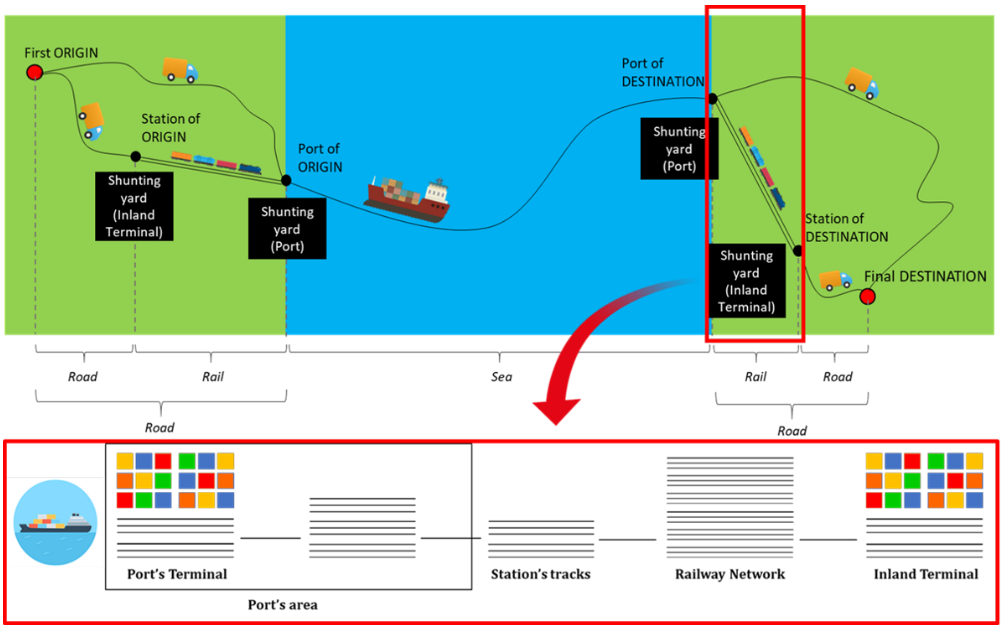

Figure 1.1: Example of intermodal freight transport system

Fig.1.1 is an example of intermodal freight transport chain including railway. On the left side of the picture there is the first origin of the goods. In this example, cargo has two possibilities to reach the port area (port of origin): either only by truck or by truck until a rail-road yard and then by train. Once arrived at the port area, goods are loaded on a vessel, which provides the maritime leg transport toward the port of destination. From there, accordingly with the example, the final destination might be reached either only by truck or using intermodality (rail and road transportation mode).

Given the general example of the intermodal system, the focus of the study is the link between the rail and the sea transportation mode. This link includes the process of transferring freight trains from the maritime terminal to the station of the rail network and viceversa.

The study refers to Italian transportation systems in which there is, usually, a shunting company with the duty to transfer the trains from the railway station outside the port area to the maritime terminals and viceversa. The main shunting activities to manage are those concerning these transfer operations.

The entire process is complex in terms of passages to execute, subjects involved, flows of information and documents to manage. The presence of many actors interacting in the intermodal transport, where rail modality is involved, causes some bottlenecks.

Let's describe in detail the import cycle with the different operators that are involved (referring to the lower part of Fig.1.1).

As explained in Ambrosino & Asta [2019], the shipping company transports the goods by ship to reach the maritime terminal of destination. The ship docks at the terminal quay and let's the discharge operations start. The maritime terminal has to manage several operations, among which the discharge operations, the goods stocking in the yard and the trains loading. Once the train is loaded, the shunting company intervenes in the process. This is the first delivery passage of the train: from the terminal manager to the shunting company. The delivery passage may includes the liability of the train depending on the rules of the specific port. The shunting company has to execute the operations related to the transfer of the loaded train from the maritime terminal to the rail station outside the port area. The shunting company may also have to carry out accessory operations, as for example the wagons discarding, the transfer of single carriages, the introduction/extraction of empty wagons. Note that, the goods loaded on the train are subjects to customs controls. Moreover, the whole train is controlled by the financial police and it is subjects to the technical verification by the railway undertaking. The technical verification may happens in different areas depending on the specialties of the port. Once the technical verification ends, the railway undertaking can manage the train on the rail national network. This is the second delivery passage of the train, from the shunting company to the railway undertaking. It usually happens when the train is positioned on the tracks of the railway station outside the port area but it may happen in other areas accordingly to the rules of the port. Since this moment, the train will travel on the rail national network, controlled by the infrastructure manager, toward the inland terminal of destination.

The export cycle is the opposite. Once the train arrives at the railway station outside the port area, the delivery passage of the train from the railway undertaking to the shunting company happens. The shunting company has to transfer the train within the port area to reach the maritime terminals. In particular, the shunting company brings the train through the infrastructure network of the port until the tracks of the terminal of destination. Then, the second delivery moment of the train, from the shunting manager to the maritime terminal, happens. At this point, the train is unloaded and the goods are stocked in the dedicated areas of the yard. Goods wait in the yard of the terminal until their loading on the vessel departing for the maritime leg of the transport.

Given the focus of the study, the next section is dedicated to the description of the port rail shunting operations for the trains transfer in this area.

1.2.2 The shunting operations

This section discusses the process from the shunting company point of view in the main Italian realities. Moreover, some problems that might be approached with operational research are identified and introducted.

Generally, the port rail shunting company receives a long term plan of arriving and departing trains for the port of interest. Then, the plan is generally updated on weekly base due to variations and suppressions. Finally, on a daily level, unpredictable events may occur. The unpredictable events might be, for example, trains delays, "last minute" suppressions, extraordinary trains to insert, and so on. The shunting company has to manage and continuously plan and re-plan its operations in order to execute all the activities, respecting both the necessities of the maritime terminals and the schedule of the trains on the national rail network. Note that, the maritime terminals necessities arise from both their internal operations and the ships schedules.

Once the shunting operator owns the information on arriving and departing trains, it has to organize and then to perform the activities to transfer the trains within the port area.

A port area, usually, includes one or more units of three distinct elements: train station, shunting zone and maritime terminal. Each train station is connected with both the railway network and one or more shunting zones. The shunting zone permits the transfer of trains from the railway network to the terminals trought the train station, and vice versa, and is composed by tracks for transferring trains and one or more parks where the trains can wait. The maritime terminal receives the train, unloads goods that later will be loaded on ships, then loads the train (with goods waiting in the yard) and, at the end, waits for the train tranfer outside the terminal itself.

Our attention focuses on the activities allowing the transfer of the trains from the entry (the train station) to the terminals and viceversa. These activities are strongly affected by the schedule of trains on the railway network and also by the schedule of the ships docked at the terminals, and can be realized using limitated resources.

In the *export cycle*, a train arriving at the station can either wait until it is moved to a shunting park or reach immediately a park. Then, the train can have to wait in the park until the terminal is ready to receive it. After that, it will be transferred to the terminal. In some particular cases, the train can be positioned in more than one shunting park before reaching the terminal; otherwise, in the opposite case, it can be directly transferred from the station to the terminal.

In the *import cycle*, the train is loaded in the terminal area and then it is transferred to a shunting park. After that, it can be transferred either directly to the rail station for its immediate departure in the railway network or in a shunting park waiting for its departure scheduled time. Note that, for import trains some security and stability checks are usually required. These controls may be realized either in the shunting zone or in the terminals. In the latter case, it is possible that the train is directly transferred from the terminal to the rail station.

It is convenience to distinguish all these activities in: waiting operations (on the tracks of either the station or the parks) and shunting operations (movement of the trains within the port area). Waiting operations starting and ending times depend on both the timing of the shunting operations and the train arrival/departure schedule.

The shunting operator uses specific resources to perform its activities. In particular, transfer operations need an engine, a shunting team and the tracks.

Here the challenge is to schedule all the activities taking into account the limitated resources necessary to realize them.

In some realities, the shunting companies manually provide a basic schedule of the activities some months in advance. Then, they reorganize the plan both week by week and day by day. That is because some variations in the plan are known some days in advance and some other during the day due to unpredictable events.

In other realities, the plan isn't provided months before. In this cases, shunting operators organize the operations day by day, after the update of the trains arriving / departing plan with the variations communicted nearby one day before. Also in this case, then, the operators have to manually replan the activities when unpredictable events occur.

1.2.3 Introduction to the optimization problems

In the just descripted process, the most relevant problem is to automatically schedule the activities to perform on the trains that are planned to arrive and depart to and from the port area. This problem consists in defining, for a given time horizon, the schedule of all the necessary activities for transferring trains from the railway network to the terminals and viceversa, respecting the time limits imposed by the railway network schedule and by the ships one, and the limits due to the finite resources available in the shunting zone. This problem is here denoted Port Rail Shunting Scheduling Problem (PRSSP). The lenght of the time horizon of the plan is generally choosen in such a way to be representative for the activities realized in the port, i.e., a week. The schedule can be repeated week by week, unless some changes are required (see Fig.1.2).

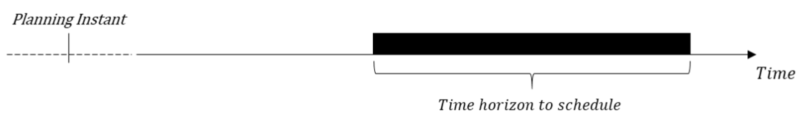

Figure 1.2: Port Rail Shunting Scheduling Problem

Beside this planning problem, an operative, sometimes in real time, rescheduling problem can occur. The shunting manager, during the time horizon considered, has to reschedule waiting and shunting operations when variations occur (see Fig.1.3). This is here denoted Port Rail Shunting Re-scheduling Problem (PRSRP).

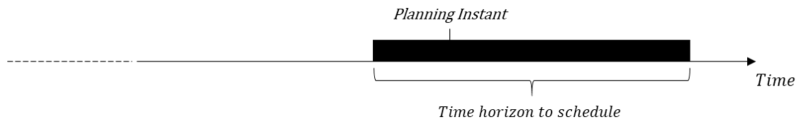

Figure 1.3: Port Rail Shunting Re-scheduling Problem

The PRSSP, PRSRP and the respective developed solution approaches are detailed in Chapters 4 and 5.

1.3 Literature overview

The port rail shunting optimization problems, which are the center of this thesis, have been rarely addressed in the literature. Therefore, the literary review is organized as in Fig.1.4 and it is focused on the papers that have aspects usefull for approaching the problems under inspection. Looking to Fig.1.4, the readers can note that the rail transportation literature can be devided into works dealing with passengers transportation and works dealing with freight transportation. Then, the square denote that the focus is on the latter. The literature on freight transportation includes studies based on the rail network and studies based on the rail terminals of the network. In particular, as shown in Fig.1.4, the rail network part includes works on both planning and real time problems. Planning problems includes general scheduling problems and specific rail scheduling problems while the real time one groups the re-scheduling problems of the rolling stock, the crew and the train timetable. In particular, the train timetable re-scheduling has been studied distnguishing between both the microscopic/macroscopic level and the type of event that occurs (disturbance/disruption). Fig.1.4 also shows the rail terminals part, which is divided depending on the type of terminal: only rail, rail-road and rail-sea (the latter two are intermodal terminals). For each terminal type are reported the main arising optimization problems. All the literature is addressed in Chapter 2.

1.4 Thesis overview

The work is articulated as explained in this section.

The present Chapter 1 has been used to introduce the work of the thesis.

Chapter 2 is dedicated to the literature analysis. In fact, it provides the review of the literature closest to PRSSP and PRSRP. The literature is organized as mentioned in 1.3 and the works are classified following their main features. In particular, focusing on papers dealing with freight rail transport, some works concern the trains circulation on the rail network, while others concern the management of freight trains in the rail terminals. The papers included in the first group are classified again depending if they consider either planning or real time management problems. Then, the papers included in the second group are organized depending on the kind of terminal on which they focus: only rail, rail-road or rail-sea yard.

Given that the choosen approach to solve PRSSP and PRSRP is based on an innovative time-space network, we used Chapter 3 to presents a detailed description of it. The first part of the Chapter reports a literature overview on papers dealing with time-space networks and their more common uses. Then, the innovative use that we have choosen to adopt is deeply analysed.

Chapters 4 and 5 are dedicated to the two problems faced in this thesis. Chapter 4 focuses on PRSSP and Chapter 5 on the PRSRP. Both the Chapters proposes the problem description and the our solution approach. The last part of these Chapters includes the computational tests and the results analysis.

Chapter 6 is dedicated to the application of the models addressed in Chapters 4 and 5 for solving PRSSP and PRSRP to a real case study. This study has been done in order to understand if the developed models could be applied in a real context characterized by a given infrastructure and management particularities. The real case concerns a port area located in Italy that we won't precise for privacy issues. This Chapter starts with the explaination of the considered system and then it is divided into two parts. The first deals with the PRSSP in the studied real case and proposes the computational tests executed with real data of a standard week. The second concerns the PRSRP in the proposed case study and shows the developed tests campaign using again the provided real data for testing different events requiring a re-scheduling.

Finally, Chapter 7 provides the conclusions of the thesis work and proposes some future works. It starts with an explaination on the reasons that brought to develop all the present study and it continues providing an overview of the developed work. At the end, the future works that might be addressed starting from this thesis are descripted.

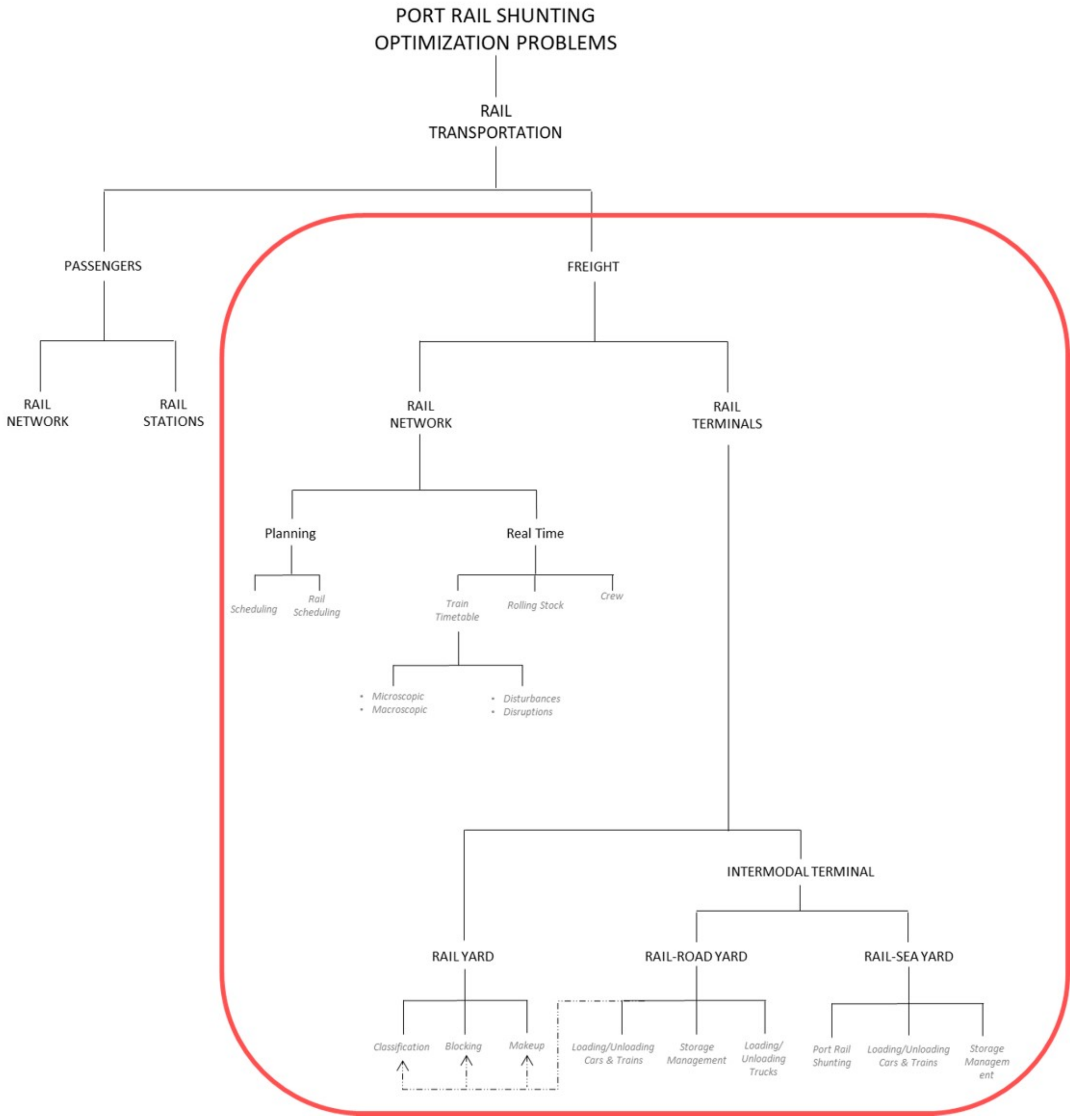

Figure 1.4: Literature organization

Chapter 2

Literature review

Summary

This Chapter gives an overview of the existing literature usefull to approach the rail shunting optimization problems introducted in Chapter 1.

2.1 Introduction

Many transportation systems are multimodal, that is, the infrastructure supports various transportation modes, such as truck, rail, air, and ocean/river navigation. Then, broadly defined, intermodal transportation refers to the transportation of either people or freight from their origin to their destination by a sequence of at least two transportation modes. Transfers from one mode to the others are performed at intermodal terminals, which may be either a sea port or an inland terminal (Bektas & Crainic [2007]). We focus on the intermodal chains including rail transportation. Although both people and freight can be transported using the intermodal chain, we consider the freight as underlined in Fig.2.1. Note that, few papers dealing also with passengers can be found in the following literature overview because we consider those specific studies important for approaching the problems under inspection in this thesis.

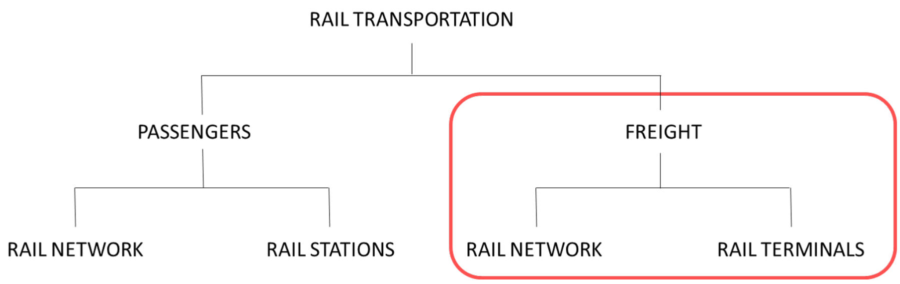

Figure 2.1: Rail transportation

An intermodal chain including railway for freight transportation is schematized in Fig.2.2. The example reported in the figure includes a logic representation of a general railway network with several terminals here represented with different shapes in accordance with their different functions. Freight trains travel on the lines of the network, i.e. the tracks, between terminals from their origin to their destination. The star shapes are rail yards, i.e. regular rail terminals where inbound trains are disassembled into cars and reassembled forming new outbound trains depending on the cargo destination. When goods have to change mode of transport in order to reach their destination, trains have to stop at intermodal terminals. The intermodal terminal may be either a rail-sea yard, i.e. the terminal for the switch between rail and sea transportation modes, or a rail-road yard, generally simply called inland terminal, for the switch between rail and road transportation ones.

The management of these kind of yards is generally different. In fact, rail-sea yards, i.e. the maritime ports, are usually managed by public authorities while inland yards could be managed by private companies.

The modal switch nodes between maritime and rail transportation represent crucial points for improving the usage of the rail transport modality, generally

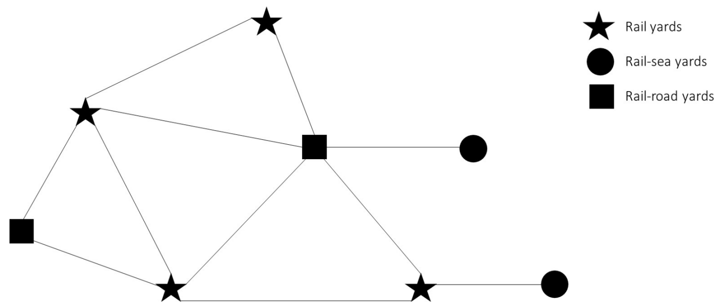

Figure 2.2: Railway network

recognised as the most environmental friendly mode of transport. One of the main problems is to balance the modal split between rail and road modality in the flows planning through maritime terminals (Iannone [2012]).

The topic of the present work, i.e. the port rail shunting optimization, has been rarely addressed in the literature. Therefore, the latter has been analysed following the works that in some way could had been relevant to solve the problems under inspection. This chapter is organized as follows. Section 2.2 proposes a general overview of the major literature on rail network topic, while section 2.3 includes both an introduction on works dealing with optimization problems in rail terminals and a focus on the existing literature about rail-sea yard problems. The reason of the focus is due to the fact that the port rail shunting operations take place in the rail-sea yard itself.

2.2 Rail network

A huge literature about the management of trains travelling on the rail network exists. In the present thesis, only the works useful to build an approach for port rail shunting optimization problems are analyzed. Fig.2.3 shows the topics classification scheme used in this section to organize the inherent papers.

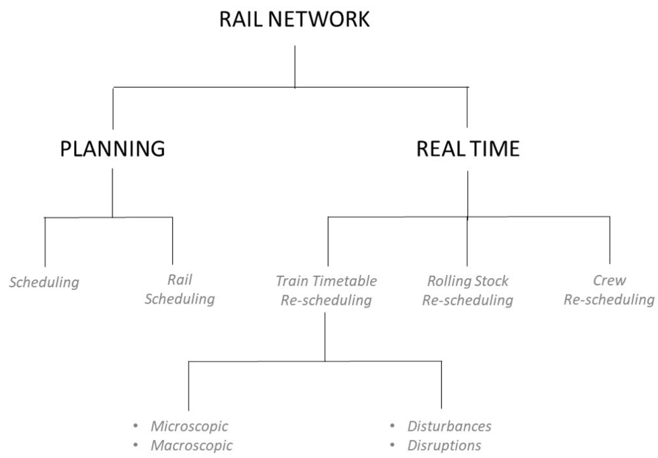

Figure 2.3: Rail network scheme

There are both works dealing with the railway service planning on the network and literature addressing the problems of real time management of the schedule when unpredicted events occurs. As far as the planning problems are considered (2.2.1), there are works on either general scheduling or specific rail scheduling problems. Concerning the real time problems, a distinction between the re-scheduling of rolling stock (2.2.2.1), crew (2.2.2.2) and train timetable (2.2.2.3) can be done, as shown in Fig.2.3. Note that, the Train Timetable Re-scheduling Problem is one of the most similar problems to the one tackled in this work. The related literature is addressed in subsection 2.2.2.3 distinguishing between the microscopic and the macroscopic approaches. In fact, microscopic models consider the railway network in detail, including, for example, the complex set-up of pieces of tracks separated by switches and signals, while macroscopic models consider the above aspects at a relative high level, and have a more aggregated representation of some resources, i.e., stations are represented by nodes of a graph and tracks by arcs. Moreover, a second distinction concerning the type of event that occurs is necessary. The events can be categorised into disturbances and disruptions, even if there is not a sharp distinction between the two in term of time length. Disturbances are generally considered relative small perturbation to the railway system that can be handled by modifying the timetable, but without modifying the duties of rolling stock and crew. Disruptions are relatively large incidents, requiring both the timetable and the duties for rolling stock and crew to be modified.

2.2.1 Planning stage

The first field of literature to look is the one dealing with the scheduling problem, which belongs to the planning stage of activities. There is a huge part of the literature related to the scheduling problem in general and in its various forms.

Generally speaking, scheduling means to assign machines and resources to tasks in order to complete all tasks under the imposed constraints. There are two main constraints in classical scheduling theory: the first is that each task is to be processed by at most one machine at a time and the second is that each machine is capable of processing at most one task at a time. Machines may be either parallel, i.e. performing the same functions, or dedicated i.e. specialized for the execution of certain tasks. Three types of parallel machines are generally distinguished depending on their speeds. If all machines have the same task processing speed, they are called identical. If the machines differ in their speeds but the speed of each one is constant and does not depend on the specific task, they are called uniform. Then, if the speeds of the machines depend on the particular task, they are called unrelated.

In case of dedicated machines there are mainly three models of processing tasks: *flow shop*, *open shop* and *job shop*. Let's assume that tasks can be grouped into sets, each set of tasks is called job and two adjacent tasks are to be performed on different machines. In *open shop* the number of tasks is the same for each job. In *flow shop* is the same and, in addition, the processing of tasks has a specific order that have to be follow for each task and for each job. In a general *job shop* system the number of tasks for each job is arbitrary. Usually, in such systems, it is assumed that buffers between machines have unlimited capacity and a job after completion on one machine may wait before its processing starts on the next one. If, however, buffers are of zero capacity, jobs cannot wait between two consecutive machines, thus, a no-wait property is assumed (Blazewicz *et al.* [2019]).

In general, tasks are characterized by:

- processing time;
- ready time for start processing;
- due date, which specifies a time limit by which the task should be completed, usually, penalty functions are defined in accordance with due dates;
- deadline, which is a "hard" limit by which the task must be completed;
- weight (priority), which expresses the relative urgency of the task;
- resource request (if any).

Given the general scheduling problems just introducted, we focus in the following on the studies dealing with the scheduling problems of rail activities, which may include also routing decisions.

Generally, the rail scheduling problems involve a rail network, usually represented using graphs in which the nodes are the stations and the arcs the existing infrastructure between them, where the trips of the trains have to be scheduled defining the arrival and the departure times at each node. Then, the constraints usually concern both the feasibility of the trips respecting the existing infrastructure and the non conflicts solutions.

D'ariano *et al.* [2007] proposes an algorithm for scheduling trains in a railway network. The scheduling problem in the railway industry is also called Train

Timetable Problem (TTP) because the concept is to build a schedule for trains to follow. The railway system is based on a timetable which consists on a set of train's trips. A trip is the train's movement from a station to another with specific departure and arrival time. The train timetable is carried out by rolling stock staffed by the appropriate crew. This latter are the resources operating in the railway system. The rolling stock refers to railway vehicles, including both powered and unpowered entities, for example locomotives, railway cars and wagons. In terms of resources, there are differences between passenger and freight trains. For the passenger transport, the rolling stock consists on a number of wagons hauled by either a locomotive or a number of coupled train units and the crew is composed by the train driver and one or more conductors. For the freight transport, the rolling stock is the locomotive plus a number of wagons and the crew is only the train driver. In both cases, the resources operate accordingly to pre-planned duties or schedules. A resource's duty is a task that must be done in a specific moment and so, the duties of every resource consists on a set of tasks that must be carried out with specific times. Every task is linked to a train's trip. In the railway system the responsible subjects are the railway infrastructure manager and the train operating companies. In fact, they are responsible for the tasks to be done in respect with the pre-planned times.

In general, given that the TTP involves the rail infrastructure and, sometimes, routing decisions, it is mainly approached by using either models or heuristics based on graphs or networks.

The train timetabling problem aims at determining a periodic timetable for a set of trains that does not violate track capacities and satisfies some operational constraints. In Caprara *et al.* [2002], the authors concentrate on the problem of a single, one-way track linking two major stations, with a number of intermediate stations in between. Each train connects two given stations along the track (possibly different from the two major stations) and may have to stop for a minimum time in some of the intermediate stations. Trains can overtake each other only in correspondence of an intermediate station, and a minimum time interval between two consecutive departures and arrivals of trains in each station is specified. They propose a graph theoretic formulation for the problem using a directed multigraph in which nodes correspond to departures/arrivals at a certain station at a given time instant. This formulation is used to derive an integer linear programming model that is relaxed in a Lagrangian way.

In this field, also Hansen & Pachl [2014] deals with the railway timetabling and operations issues including analysis, modeling, optimization, simulation and performance evaluation topics. A review of these problems is in Cacchiani & Toth [2012] and in Chapter 5 of Borndörfer *et al.* [2018]. They survey the main studies dealing with the train timetabling problem in its nominal and robust versions. The nominal version of the problem focuses in determining "good" timetables for a set of trains following the aim of optimizing an objective function that can have different meanings. Two are the main variants of the nominal problem: one is to consider a cyclic (or periodic) schedule of the trains that can be repeated every given time period, and the other one is to consider a network where a non-cyclic schedule can be performed. Then, in the recent years, many works have been dedicated to the robust version of the problem. In this case, the aim is to determine robust timetables for the trains, i.e. to find a schedule that avoids, in case of disruptions in the railway network, delay propagation as much as possible. Nowadays railway systems are highly affected by disturbances, occurring in daily operations, and causing train delays. They cause additional operational costs, since the planned schedule needs to be modified in real-time. Therefore, it is an important issue to determine robust timetables. In Chapter 5 of Borndörfer *et al.* [2018], the authors present the state-of-the-art methods that achieve robust timetables, and discuss their advantages and drawbacks.

The train timetabling problem for large geographical areas and many trains is intractable. In Gestrelius *et al.* [2017], the authors present a MILP-based heuristic that has been designed to generate good-enough timetables for large geographical areas and many trains. In the incremental fix and release heuristic (IFRH), trains are added to the timetable in batches. For each batch of trains, a reduced timetable problem is solved using a mathematical integer program and CPLEX.

2.2.2 Real time stage

As introducted in the previous subsection, the railway system is based on a timetable which consists on a set of train's trips. At the beginning, when the activities' scheduling is built, the timetable and the resources duties are without conflicts, but in the real time operations problems are unavoidable. Conflicts can occur for delays due to disturbances or disruptions. On one side, disturbances consist on small problems causing little delays, on the other side, disruptions are large complications producing big delays and cancelations. If a disturbance or a disruption occurs, the railway system's timetable has to be re-scheduled. The re-scheduling may include the rolling stock duties, the timing and the routing. Then, the result is used as input in order to re-schedule the crew's activities. This problem in literature is called Train Timetable Rescheduling (TTR). Since it is a real time scheduling problem, there are some differences in respect of the first railway system scheduling to consider:

1. the time needed for a solution is much less;
2. the scheduling is less flexible;
3. the solution space is bounded depending on the day's moment in which the problem occurs;

4. the objectives to follow can be different.

In the following, there is a brief overview of: rolling stock re-scheduling (2.2.2.1), crew re-scheduling (2.2.2.2) and train timetable re-scheduling (2.2.2.3).

2.2.2.1 Rolling stock re-scheduling

If a disruption has occurred in a railway network, the original rolling stock allocation is usually no longer feasible. Therefore, the rolling stock needs to be re-scheduled, using the updated timetable and the original rolling stock allocation as input. In the planning stage of a railway system, rolling stock has been assigned to the trips in the timetable. The decisions to take concern the assignment of the rolling stock to trips in the timetable accordingly to the constraints and the objectives of the involved railway operator. Finally, the shunting possibilities inside the stations and to the depot planning are important. The problem of rolling stock rescheduling may be formulated as a multi-commodity flow model on a graph where the nodes correspond to stations at a specific times, while the arcs correspond to trips that have to be performed according to the timetable.

The interested readers can find more details on Rolling Stock Re-scheduling in Nielsen [2011], Kroon *et al.* [2015], Lusby *et al.* [2017] and Hoogervorst *et al.* [2020].

2.2.2.2 Crew re-scheduling

The railway crew re-scheduling problem deals with assigning tasks to the train drivers and conductors after the timetable and the rolling stock have been re-scheduled. Due to the re-scheduled timetable and rolling stock, some crew duties may have become infeasible. When re-scheduling the duties, the original crew schedule is used as input, since the re-scheduled duties should not differ too much from the original one. The crew re-scheduling problem can be formulated as an

Extended Set Covering problem. In that sense, it is comparable to the crew scheduling problem in the planning stage. The main difference between the re-scheduling stage and the planning stage is that in the re-scheduling stage the original duties of the crew members have to be taken into account in order to be sure that the re-scheduled duties may not differ too much from the original ones.

Huisman [2007] defines the Crew Re-Scheduling Problem (CRSP) and shows that it can be formulated as a large-scale set covering problem. The problem is solved with a column generation based algorithm and the performance of the algorithm is tested on real-world instances of NS, the largest passenger railway operator in the Netherlands.

In Veelenturf *et al.* [2012], the authors model and solve the crew re-scheduling problem with retiming. This problem extends the crew re-scheduling problem by the possibility to slightly delay the departure of some trains, so that some more flexibility in the crew scheduling process is obtained. The algorithm is based on column generation techniques combined with Lagrangian heuristics. In order to prevent a large increase in computation time, retiming is allowed only for a limited number of trains for which it seems promising.

Then, Abbink *et al.* [2011] and Caprara *et al.* [1999] deal with the large-scale crew scheduling problems. The first work approaches the problem arising at the main Dutch railway operator, Netherlands Railways (NS), which operates about 30000 trains a week. The authors present an algorithm, called LUCIA, which can solve such huge instances without splitting. This algorithm combines Lagrangian heuristics, column generation and fixing techniques. The second describes the development of a new crew planning system set up by Ferrovie dello Stato SpA (the Italian railway company) in co-operation with the University of Bologna.

Finally, in Potthoff *et al.* [2010] an algorithm to reschedule the crews when a disruption occurs is presented. The algorithm is based on column generation techniques combined with Lagrangian heuristics.

2.2.2.3 Train timetable re-scheduling

The Train Timetable Re-scheduling Problem consists in the real-time adjustment of an existing timetable that has become infeasible due to unpredictable disturbances or disruptions. The aim of TTR is to quickly re-obtain a feasible timetable of sufficient quality. Cacchiani *et al.* [2014] presents an overview on recovery models and algorithms for real-time railway rescheduling. The main decisions concerns the trains' re-routing, the time instants of trains' departure and arrival and the order of trains on their common track sections. So, the TTR can be split into the retiming problem and the rerouting problem (Corman *et al.* [2010b]).

In general, the studies on TTR can be organized depending on two different criteria. As already introducted, the *first* concerns the kind of event they are dealing with: **disturbances** or **disruptions**. The term **disturbances** is used for relatively small perturbations influencing the railway system and the term **disruptions** for larger external incidents leading to the cancelation of a number of trips in the timetable.

The *second* is about the considered level of detail of the rail network: **microscopic** and **macroscopic**. The **macroscopic** approach considers the railway network at a higher level, in which stations can be represented by nodes of a graph and tracks by arcs, and the details of block sections and signals are not taken into account. In the **microscopic** one all the aspects are considered in detail. In particular, blocking time graphs and the underlying data are, usually, used to compute detailed running and headway times.

Microscopic approach examples

For what concerns the studies that consider a microscopic level of detail of the network, Boccia *et al.* [2013] describes two heuristic approaches to solve the re-scheduling and re-routing decisions problem based on a mixed integer linear programming formulation in the area of **disturbances** event management. Other examples of works dealing with the **disturbances** event management are D'Ariano *et al.* [2008a] and D'Ariano *et al.* [2008b]. D'Ariano *et al.* [2008a] describes the implementation of a real-time traffic management system, called ROMA (Railway traffic Optimization by Means of Alternative graphs), to support controllers in the everyday tasks of managing disturbances. They make use of a branch-and-bound algorithm for sequencing train movements, while a local search algorithm is developed for rerouting optimization purposes D'Ariano *et al.* [2008b] explaines the principle of a flexible timetable, i.e. to plan less in the timetable and to solve more inter-train conflicts during operations. The larger degree of freedom left to real-time management offers better chance to recover disturbances. The authors illustrate a detailed model for conflict resolution, based on the alternative graph formulation, and analyze different algorithms for resolving conflicts, based on simple local rules or global optimization.

Passing, then, to the **disruptions** event recovery area, an example of approach is in Corman *et al.* [2010a]. The authors start from a given set of disruption resolution scenarios, computed off-line, where disrupted train services are either cancelled, rerouted in the disrupted dispatching area or rerouted in other areas. Given a disruption resolution scenario, they adopt an advanced decision support system, ROMA (Railway traffic Optimization by Means of Alternative graphs), in order to compute efficient train schedules at the level of signal control, along with detailed performance indicators. The dispatcher has, then, to choose one schedule for implementation, with a trade-off between minimizing delays of all trains running in the network and limiting the cancellation of disrupted train services. A test case is evaluated on a large Dutch railway network with heavy traffic and strong disorder.

Macroscopic approach examples

For what concerns the **disturbances** event area, two examples are here reported: Krasemann [2012] and Kecman *et al.* [2013]. In Krasemann [2012], the authors have developed a greedy algorithm which effectively delivers good solutions within the permitted time as a complement to the previous approach fast response to the re-scheduling of railway traffic during disturbances. Then, Kecman *et al.* [2013] is dedicated to the development of new macroscopic models that are able to incorporate traffic management decisions. The objective of this paper is to investigate how different levels of detail and number of operational constraints may affect the applicability of models for network-wide rescheduling in terms of quality of solutions and computation time. The authors present four different macroscopic models and test them on the Dutch national timetable.

Some works include MIP models for the problem under inspection. In MIP models the arrival and departure times and delays are represented as continuous decision variables. This is the main difference from the IP model where rescheduling tactics can be represented by binary decision variables, such as the connection maintenance, the priority of two trains, sequences of trains, etc. Moreover, in IP models, the arrival and departure time of trains can be represented by non-binary integer values by denoting them in discrete time intervals and the delays are also treated as integer variables. Acuna-Agost *et al.* [2011], for example, presents the railway re-scheduling problem as the problem of finding a new schedule of trains after one or several incidents by minimizing some measure of the effect. They investigate the solution of this problem through a local search based on a mixed-integer programming (MIP) formulation.

An alternative model is a CP model since it incorporates techniques to reduce the variables and constraints. The main components in the CP model for formulating rescheduling are the data sets, parameters, decision variables (including continuous and binary variables), traffic constraints, and the objective function, which are similar to those in an MIP model. However, a CP model can describe the problem more naturally because of the definitions of some variables and constraints, the representation of the order of trains, and the use of tracks. These make a CP model require much fewer variables and constraints than an MIP model. An example is Rodriguez & Kermad [1998], which describes a constraint-based model of the train circulation management problem. The basic concepts of the model are activities, resources, time and allocation constraints. The problem is formulated as a joint scheduling and allocation problem

Anyway, also other approaches are used. Corman *et al.* [2012] considers the bi-objective problem of minimizing train delays and missed connections in order to provide a set of feasible non-dominated schedules to support this decisional process. The authors use a detailed alternative graph model to ensure schedule feasibility and develop two heuristic algorithms to compute the Pareto front of non-dominated schedules. The, another example is Cheng & Yang [2009], which aims to transform a train dispatcher's expertise into a useful knowledge rule using the fuzzy Petri Net approach. It is adopted to formulate the decision rules of train dispatchers in case of abnormality as the basis for future development of a dispatching decision support system.

Finally, also simulation can be used. The components of a railway network, such as railway infrastructure, train movements, dispatching process, etc., can be represented, in fact, by a computer-based simulation model. The simulation model can not only simulate the real-time status of a railway system, but also forecast the future status of the railway system and have the ability to resolve conflicts. Therefore, the rescheduling approaches can be integrated into the simulation model to support real-time dispatching.

Passing to the **disruptions** event management, two exaples are Corman *et al.* [2011], in which the authors investigate disruption handling strategies for large and busy railway networks, and Narayanaswami & Rangaraj [2013], where a MILP model that reschedules disrupted train movements on both directions of a single track layout with an objective to minimise total delay of all trains at their destinations is proposed.

Note that, due to **disturbances** and **disruptions**, a particular situation, called deadlock status, may happens. It is when there is a circular precedence among a set of trains, each one of them waiting for the access to a resource blocked by another train in the set. Chapter 12 of Borndörfer *et al.* [2018] and Dal Sasso *et al.* [2021] deal with the Train Dispatching Problem for deadlock detection and avoindance in railways traffic control.

Even if the Train Dispatching Problem can be viewed as Job Shop Scheduling problem with blocking and no-waits constraints, once a routing is fixed for each train, this problem can be effectively formulated by a disjunctive formulation based on the so called alternative graph (Chapter 12 of Borndörfer *et al.* [2018]). An alternative graph is composed by nodes, which are associated with the event that a train starts the occupation of the resource (infrastructure), fixed and directed arcs, and set of pairs of alternative directed arcs. The precedence rules for using the resources by the trains are obtained by using binary variables in order to decide which of the alternative arcs is chosen. The other variables are used to define the starting time for using the resources by each train. A MILP formulation for the scheduling problem is provided.

Dal Sasso *et al.* [2021] presents a new 0,1 linear formulation for detecting deadlocks. They propose to decompose the train movements into sequences of moves, where a move is the progress of a train's headfrom one signal along its path to one of the following signals in the same direction. They say that these moves take place at the same tick. A tick represent an (unspecified) continuous time interval in which a subset of trains start and complete a move each. With the Tick formulation that they want to model, if it exists, they obtain a sequence of moves for each train which ensures a deadlock-free plan. Note that, a train may stay still for several ticks, if it needs to wait for the railway routes to be freed by other trains. In the mathematical formulation the variables record which route arcs are occupied by each train at each tick.

2.2.2.4 Integrated approaches

Some works deal with the integration of different phases of real time railway rescheduling. These papers' aim is to determine a new schedule for the timetable, the rolling stock and crew duties when a disruption occurs. By treating several re-scheduling phases at the same time, it can be expected that better solutions are obtained. The problem is that the integrated models are usually much more complex and difficult to be solved than the models dealing with a single phase. This is relevant in the real-time rescheduling stage, where the rescheduling problems must be solved soon. In the planning stage, more computation time is available for solving complex models.

For example, in Walker *et al.* [2005] alterations to the existing train timetable and crewing roster are made simultaneously in real time using an integer programming model, while previous studies in the literature have always decoupled these two problems and solved them in series.

Again, the authors of Veelenturf *et al.* [2012] model and solve the crew rescheduling problem with retiming. This problem extends the crew rescheduling problem by the possibility to slightly delay the departure of some trains, so that some more flexibility in the crew scheduling process is obtained. They used a column generation techniques combined with Lagrangian heuristics.

2.3 Rail Terminals

This section includes an introduction on the works dealing with problems in rail terminals and a focus on the rail sea yard. Rail terminals might be: rail yard, for the rail transportation mode, rail-road yard, for both the rail and the road transportation modes, rail-sea yard, for the rail and the sea transportation modes. In Fig.2.4 the different types of rail terminals together with the main operations arising in each of them are reported.

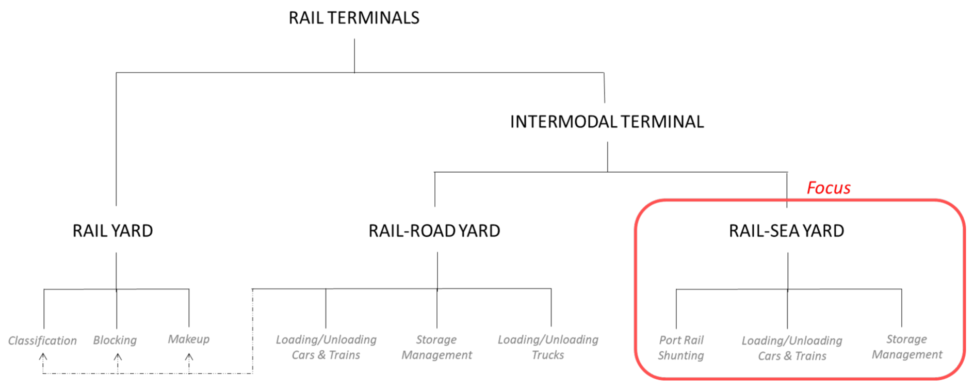

Figure 2.4: Rail terminals and related activities

For what concerns rail yards and intermodal rail-road yards, only a briefly mention of the major classes of operations and issues related to them is reported.

According to Boysen *et al.* [2012], the major operations in **rail yards** are: classification, blocking, and trains make-up. A significant amount of research exists on planning these operations. Classification problems deal with the operational tasks at a shunting yard of the assignment of railcars to classification tracks, i.e. the core of a sorting procedure. Then, the blocking problem consists in grouping railcars to blocks, which then travel together through different stages of the network, i.e.

consolidation of rail cars into blocks. Finally, the train makeup is the problem of forming blocks to trains, i.e. an assignment of railcars, for which a route is known, to trains.

In intermodal **rail-road yards** the main operations to perform are loading and unloading trains, the management of the storage area and the loading and unloading of trucks (Dotoli *et al.* [2016], Carboni & Deflorio [2020]). Note that, intermodal rail-road yards may have to deal also with the problems of classification, blocking and trains make-up, as the rail yards do (see dotted arrows in Fig.2.4).

Concerning the intermodal **rail-sea yard**, the main operations are the transfer of trains between the railway station and the maritime terminals, the loading and unloading of trains and the storage management. Depending on the specific port, sometimes trains approaching the area have to be splitted into cars and then these latter are transferred to terminals of destination (and viceversa for the import cycle). The rail operations in port area have been unfortunately rarely addressed in the exisiting literature. Given that this is the object of the paper, subsection 2.3.1 provides a focus on the literature on it. Note that, trains, as single entities, instead of single rail cars, which could be loaded with different types of cargo, will be considered.

2.3.1 Focus on rail-sea yards

A huge part of the rail freight transport has its origin or destination in sea ports. Considering the intermodal system, ports play a more important role than the pure interfaces between sea and land transport (Krämer [2019]).

Today, the port railway process is much more complex than truck and barge ones; this represents a competitive disadvantage for the rail mode. Real innovations within the rail cargo sector, especially in European area, are rare, even if good ideas do exist. As a consequence rail transport of cargo until today is in many cases very traditional and old-fashioned (Krämer [2019]).

Trains arriving in port area should spend a maximum of half an hour before the operations at maritime terminals begin. In the same way, considering the opposite cycle, after completion of the loading procedures in the terminals trains should be ready to leave the port area without any additional treatments or checks within minutes. Only if these timings will be reached, there will be a prosperous future for rail cargo transportation.

Of course, from the actual steady situations, a fast and efficient rail process could be reached following several intermediate steps, such as, for examples, simplification of operational processes, optimization of infrastructure usage with savings on future investments, reduction of operational efforts and costs through reduction of workers, and so on (Krämer [2019]).

The following sections report an overview of the main existing literature on the optimization of operational problems in the rail processes in port area.

The rail operations in a port area can be divided depending on the zone where they mainly happen, the port rail shunting operations (2.3.1.1) and the rail maritime terminals ones (2.3.1.2).

2.3.1.1 Rail operations in port shunting yard

Concerning the general rail process related to the port area, Krämer [2019] describes the overall objectives, ambitions and expected outcomes of "Shunt-E 4.0—Autonomous Zero Emission Shunting Processes in Port and Hinterland Railway Operations": a practical research program focuses on improvements in last mile rail operation. It is conducted together with Bremen's port railway which is regarding the modal share of rail hinterland transport already as one of the leading European port railway systems.

At this point, a distinction must be underlined. In some ports, arriving and departing trains transport goods belonging to different maritime terminals, in other ports all cargo loaded on a train have a single maritime terminal as destination/origin. In the **first case**, trains arriving at the port have to be splitted into cars depending on the terminal of destination (note that the departing trains have to be composed by cars from different terminals). In the **second case**, the whole train arriving at the port area have to be transferred to the terminal of destination (and viceversa for departing trains). Depending on the type of the port the shunting operations to perform are different and consequently the arising problems too.

We start analyzing the **first case**. The main shunting activities of the export cycle consist on receiving freight trains from the hinterland and shunting the freight wagons depending to their final terminal of destination. Considering the import cycle, the activities are: receiving the wagons from maritime terminals, accumulating wagons according to their hinterland of destination and allowing the freight train to depart. Of course, besides these main activities, other activities concerning the preparation of customs documents, border police controls, technical check of wagons, etc., might be carried out within the shunting yard. The route of the wagons through the shunting yard completely depends on the station topology. The lines inside the station are grouped according to the main activities to perform. Therefore, the tracks in the shunting yard are organized into Receiving Tracks, Classification Tracks, and Departure Tracks. The transfer of wagons between Receiving Tracks and Classification Tracks is performed by crossing over the hump of the yard. The trains from land (also called the inbound trains) are waited in Receiving Tracks. The trains' engines are removed and another engine (belonging to the station) for maneuvers is attached. The technical and the commercial activities are made, and the train stops to exist as an entity. Wagons are separated according to the destination (in this case, maritime terminals) using the shunting installation. This latter is composed of one or two humps, a braking system, and an inclined plane Rusca *et al.* [2019].

As aleady said, the port shunting yards specificity raise several problems such as, for examples, limited access, the in-output of large transit flows of cargo relatively to the scarcity of the departure/arrival of a ship, limited land availability for implementing solutions to serve these flows. It is necessary to identify technological solutions that lead to an answer to these problems. Rusca *et al.* [2016] propose a simulation model developed with ARENA computer simulation software suitable for shunting yards which serve sea ports with access to the rail network. In this work, the principal aspects of shunting yards and adequate measures to increase their transit capacity are investigates. A solution to improve the yard capacity is to change a normal shunting process, which includes a shunting process with a single train, to simultaneous shunting processes, i.e. two train are shunting in same time.

The comparative study between the repeated and simultaneous shunting was based on two models: the analytical model, in which the operating conditions are deterministic, and the simulation model, which takes into account the variation and interaction in time of the considered variables and allows the reproduction in small details of the dynamics of system status.

In Rusca *et al.* [2018] the authors analyze the topological structure of port shunting yard and evaluate transit capacity through its various compartments. A discrete simulation model is developed with ARENA computer simulation software for wagons shunting process and various technologies for shunting process are tested to identify optimal solutions for increasing the transit capacity of the port shunting yard.

Again, Rusca *et al.* [2019] develop a discrete simulation model taking into consideration the input flow characteristics of freight trains from land network, the input flow characteristics of freight wagons from maritime terminals, the number of tracks in a port railway station, the technology used for separation of freight wagons on destinations (inside the port or from hinterland), and the duration of technological processes. From the results, they extract the values of trains and wagon sets that transit through the main areas of the railway station (Receiving Tracks, Hump(s) and Departure Tracks). The shunting yard works like a buffer between circulation on the railway network and activities inside the maritime port terminals. The simulation model for shunting yards considers the railway station a queuing system. Using a simulation model, comparisons between different shunting yard typologies can be made to hierarchize the investment in the short or long term.

More papers dealing with this kind of problems, i.e. management of the trains splitting in the rail yard, exist but they are focused on the operations in the hinterland terminals instead on the ports area, so we don't address them here. For whose interested, see the following reviews: Cordeau *et al.* [1998], Ahuja *et al.* [2005], Boysen *et al.* [2012] and Borndörfer *et al.* [2018].

Going ahead, in the **second case**, when cargo on a train has a single maritime terminal as destination (and origin), the process and the arising problems are lightly different. The process provides that a train arriving at the station can either wait until it is moved to a shunting park or reach immediately a park. Then, the train can have to wait in the park until its terminal of destination is ready to receive it. In some particular cases, the train can be positioned in more than one shunting park before reaching the terminal; otherwise, in the opposite case, it can be directly transferred from the station to the terminal.

Given the schedules of both the trains on the railway network, on one side, and the arriving and departing ships, on the other one, the port rail shunting optimization problems concern the scheduling of the shunting activities and their re-scheduling when delays or cancellations occur. To the authors knowledge there are really few works dealing with scheduling and re-scheduling of shunting operations. In Caballini *et al.* [2016] and Fioribello *et al.* [2016], the authors studied a discrete-time model in order to size port rail networks and planning shunting operations only for containers terminal and from the containers point of view. The difficulty is in scheduling all these activities and managing in the correct way the resources necessary to realize them in the shunting zone, in accordance with the terminals time requirements. It is convenience to distinguish all these activities in waiting operations, which are performed on the tracks of either the station or the parks, and shunting operations that comprehend the movement of the trains in the port area. Waiting operations starting and ending times depend on the times of the shunting ones.

Ambrosino & Asta [2019] is the first attemp to solve the port rail shunting scheduling problem introduced above. The authors present a discussion on possible approaches for solving this problem, together with a first approach based on an operations-time network.

To the authors knowledge, no more literature exists on these scheduling and re-scheduling for the port rail shunting activities. However a huge number of papers on general scheduling and re-scheduling problems in the rail field can be found. Among these we cite two works, Tomii *et al.* [1999] & Tomii & Zhou [2000], because the problem that the authors address is really similar to the port rail shunting optimization one.

They deal with the shunting scheduling problems set in a rail station. The problems include the schedule of both tasks, shunting and workforce assignment. In this problem, movement of trains between tracks is called shunting. Shunting is necessary in the following three cases: a train which arrives at a track and is scheduled to depart from a different track, a train which arrives and departs from the same track, but another train is planned to use the track in the same time interval, a train on a side track if other train uses it. Tasks might be various, in fact, there exist several types of inspections, maintenance, cleaning of rolling stock together with the shunting of rolling stock. In the schedule of tasks, the type, the track where the task is performed, start and end times are prescribed. In the scheduling of shunting, origin and destination tracks and their execution timings have to be decided. Workforce assignment includes assignment of workforce for each task together with the time for rest. Summarizing, the aim of depot shunting scheduling problem is to decide for trains which need shunting, side tracks to be assigned and shunting times. Arrival and departure times of trains and tracks are prescribed by train schedules. That is exactly as in port rail shunting optimization problems.

Various kinds of limits have to be considered in making depot shunting schedules. The more similar to the scheduling problem in port area are: i) constraints concerning train schedules: a solution can not be inconsistent with planned times prescribed by the train schedule; ii) constraints caused by facility conditions: for examples, existence of routes, length of tracks, i.e., there can not be another train on the route of a shunting;iii) temporal constraints: for examples, minimum dwell times on tracks to complete assigned tasks have to be retained, shunting running time, crossover conflicts; iv) constraints about tasks: all the necessary tasks must be performed.

Shunting operations schedule play an important role in railways connection. It has to be made respecting the train circulation schedule. A train schedule prescribes arrival/departure times and tracks for each train. It is not always possible to make a complete feasible and optimal schedule of trains and shunting because they are made indipendently. In such a case, it is desired to make a shunting schedule to satisfy the constraints of train schedule as much as possible. In Tomii *et al.* [1999], the authors believe that a computer system that automatically makes shunting schedules will be quite beneficial to reduce the time and labor and obtain schedules of much higher quality.

A difference between port rail shunting and depot shunting scheduling problems consists in the fact that in the first there aren't actual tasks to perform. Mainly, in port area the shunting activities are the ones to schedule. Moreover, the big space that is dedicated to the workforce assignment and management in the second is really smaller in the first one.

In Tomii *et al.* [1999], the authors propose an efficient algorithm for depot shunting scheduling problems combining genetic algorithm (GA) and PERT. GA and PERT are combined so that the candidates for answers in GA are evaluated by PERT. This enables them to reduce the search space of GA to a great extent. To this end, they focus on the following two points. One is to develop algorithms with higher functions such as to make transportation schedules automatically. Although little attention seems to have been paid to automatic scheduling at the moment, it is helpful to improve the efficiency of transportation scheduling. The other is to develop algorithms that are applicable in various kinds of circumstances irrespective of the differences caused by the specific conditions of the site of application.

Finally, Tomii & Zhou [2000] introduce an algorithm that makes depot shunting schedules automatically. One of the characteristics of this algorithm is that it is not dependent upon the specific conditions of depot such as the track layout and other factors.

2.3.1.2 Rail operations in maritime terminal

Problems related to the rail operations in the maritime terminal fit into the landside transport optimization and storage and stacking planning processes, as defined in Steenken *et al.* [2004] and Stahlbock & Voß [2008]. The main problems concern the definition of trains loading plans and management of the storage area.

The unloading and loading operations from/to a train in the maritime terminal must be performed within specific time windows. When the train (a set of wagons) arrives at the maritime terminal it must be unloaded: export containers are unloaded and transferred for their loading on the ships. More attention is required for the management of import containers, in order to efficient perform their loading on trains. Import containers can be stored either in the general terminal yard or in a dedicated area, closer to the tracks and used as a buffer.

The dedicated rail yard can be used either to store containers that have to leave the terminal by trains for example in few days, or simply as a buffer to prepare and prestow containers for their loading on the next leaving train. In this latter case, the terminal adopts a premarshalling strategy in such a way to have containers to laod ready near the tracks. This approach permits to evoid reshuflles and other movements in the storage area. Reshuffles are unproductive moves required to gain access to a desired container that is blocked with other containers.

Therefore, the storage space assignment deals with finding the best allocation of containers to storage spaces. A good storage space assignment is one that reduces the storage yard operations cycle time (i.e., the time to store, retrieve, and reshuffle). The fitness of a storage space assignment depends on the availability and quality of the arrival and departure time information for the import, export and transshipped containers handled.

A deep analysis on storage yard operational strategies in container terminals have been proposed in Carlo *et al.* [2014], even if these are stategies mainly devoted to export containers. The main decisions in the yard include storage and retrieval of containers, in particular, storage assignment of containers and dispatching and routing of material handling equipment. Rarely studies refer to the rail yards in maritime terminals.

In Gillen & Hasheminia [2018] has been evaluated the advantege of having a rail yard in the maritime terminal in terms of dwell time, in Schönemann [2010] the transhipment process between ships and railways in German seaports is analysed pointing out the importance of flow of information in terms of time reduction.

Flow rates are defined as container movements between the rail terminal and the storage yards at each period, which may change dynamically over the discharging and loading time windows. In Xie & Song [2018], the authors propose an integrated model to optimise the decisions of both container prestaging and container flow rates in the presence of uncertainties.

Caballini *et al.* [2016] model the movements of containers in the maritime rail terminal area in order to optimise the timings of the trains and the use of the handling resources devoted to rail port operations. In a previous work Caballini *et al.* [2012], the authors propose a discrete-time model for optimizing the rail port cycle from the point of view of the containers. In particular, they represents the transfer of import containers from their storage area until their exit from the terminal by train. The standing of containers and their movements inside the terminal are modelled by a set of queues.

The major part of literature focuses on the train load planning problem. Steenken *et al.* [2004] define the train loading plan as the problem of determining on which wagon a container has to be placed; this decision generally depends on the destination, type and weight of the container, the maximum load of the wagon, the train composition and the container location in the storage area.

A review of the models proposed for defining the train load planning problem is presentd in Heggen *et al.* [2016]. The first study on train loading problem in a maritime terminal extended the work of Bruns & Knust [2012] for including some aspects related to the storage of containers. In a sea terminal context is essential to combine the aim of maximising the train utilisation, minimising the unproductive operations in the terminal, with particular attention to the rail storage yard (minimising the reshuffles in the storage area (Ambrosino *et al.* [2011])), and minimizing the equipment utilization (unproductive movements of cranes) (Ambrosino & Siri [2015]). Anghinolfi & Paolucci [2014] considers a sequence of trains with different destinations with the aim to minimise the distances between the container locations in the storage area and the assigned wagons extending the model presented in Ambrosino *et al.* [2011].

Mantovani *et al.* [2018] studies the double-stack load planning problem, characteristic of North America intermodal railway terminals. The authors include in the analysis and in the proposed optimization model complex loading rules that the load plan must satisfy and that depend on specific container and railcar characteristics.

In the import process, the rail operator will produce a load list specifying the specific containers to be loaded on a particular train. The load list can be communicated to the terminal operator in a time windows between several hours and one day in advance of the train's arrival. However, the load list is subject to changes and it may happens that the rail operator requires loading of a certain number of containers with very short notice. Failing to load such specific containers might provide a heavy penalty for the terminal operator Xie & Song [2018].

2.4 Conclusion

Chapter 2 provides the review of the literature useful for approaching the Port Rail Shunting Optimization Problems (PRSSP & PRSRP).

The main aspects that these problems have in common with the general scheduling problems are the shunting operations that have to be schedule. We can say that operations can be represented as tasks to schedule on machines. In this particular context, the most similar problem is the job shop scheduling with dedicated and unrelated machines, because each machine, considered as set of resources for performing a specific task, are specialized for each task and the speed may change depeding on the task. Moreover, the job shop is the most applicable beacuse the number of tasks for each job might be different in PRSSP and PRSRP. Anyway, the port rail shunting optimization problems are characterized by two main features that bring the job shop scheduling problem difficult to be applied: the first concerns the resources needed to perform the shunting operations, which are usually shared among the machines, the second consists in the fact that also the spatial dimension has usually to be considered in the operations scheduling. In fact, both the problems, introducted in Section 1.2.3 of Chapter 1, include time and space decisions.

Then, passing to the scheduling problems in the rail network, they are in some way all involved with routing decisions because they works with the infrastraucture network itself. That is different from the PRSSP and the PRSRP, in which we already know the operations to perform, the sequence of the latter and the path of the trains in the infrastructure.

Note that, one important point in this study is to look for a structure approach that can be applied to different kind of port area layouts, keeping a macroscopic view without going inside the details of the infrastructure.

Among the few papers dealing with the rail process in the port area, the interesting papers in terms of closeness to the problems under inspection are Tomii *et al.* [1999] and Tomii & Zhou [2000]: they have in common the shunting operations timing decisions and in contrast the fact that in our problems there aren't tasks to perform on waiting trains and there isn't the workforce assignment and management.

For the reasons above mentioned, it has been decided to approach the port rail shunting optimization problems using the well known time-space networks in an innovative way in order to easily manage both the shunting operations, the time-space decisions and the capacity constraints.

Note that, some aspects of the network approach that we have used are contiguous to the approaches presented in Borndörfer *et al.* [2018] and Dal Sasso *et al.* [2021]. As in the first paper, our network has starting and ending nodes and each node is associated to an operation, which represents the occupancy of the resources. In the paper, they consider the little parts of the infrastructure as resources, while we have to consider more kind of resources then the infrastructure, such as the available shunting engines and the available number of teams for performing the operations. Moreover, another difference consists in time management: in the model presented in Borndörfer *et al.* [2018] there are continuous variables for defining the time in which each train starts occupying the resources, while in our model the time has been distretized using fixed time intervals, composed by two time istants (starting and ending) and the nodes, which represent the operations, are replicated for each time instants in order to be able to control and manage the availability of all the types of needed resources. An time approach that can be in some way considerated similar is the one used in Dal Sasso *et al.* [2021]. The similar aspect is that also here the time is managed using time intervals, called tick. The difference is that these tick are unspecified, continuous and can be of different lenght, while in our approach are known and the lenght is fixed.

The innovative way to use the time-space networks is deeply analysed in Chapter 3, which is completely dedicate to explain this new approach. Chapter 3 includes both an overview of the common use of time-space networks and the description of the innovative one.

Then, Chapters 4 and 5 are dedicated to PRSSP and PRSRP, respectively. These two Chapters provide the problems definition, the description of the innovative space-time networks application to each problem and the computational tests.

Chapter 3

Innovative use of time-space networks

Summary

The PRSSP and PRSRP introducted in Section 1.2.3 of Chapter 1 are the focus of the present thesis. The first is a scheduling problem of shunting operations based on a given infrastructure and the second is a real-time re-scheduling problem of the same shunting operations and based on the same infrastructure. The useful literature for these two particular problems have been reviewed in Chapter 2. As cited in the conclusion of the latter, these problems can be approached by using the time-space networks in an innovative way for representing also the operations. This Chapter is used to address this innovative use of a time-space network, in fact, it describes the operation-time-space network that can be easily adapted to model different logistics problems involving time and space decisions. In the recent literature, different time-space networks have been proposed to deal with specific requirements of problems arising in the logistics and transportation fields. Thanks to the proposed network we are able to deal with different types of capacity and

time constraints that characterized the most part of logistics problems. This innovative network has been submittes for publication on *Networks* (D. Ambrosino, V. Asta. An innovative operation-time-space network for solving different logistics problems with capacity and time constraints. Submitted for publication in *Networks* - July 2020. Review submitted - March 2021)

This Chapter briefly reviews the main uses of the time-space networks and explains how to construct the operation-time-space network and how to model time and capacity constraints of different types.

3.1 Introduction and literature review

Time-space networks are generally used for solving problems containing both time and space decisions. When temporal aspects are relevant, a physical network can be extended to a time-space network (TSN). TSN have been proposed for solving problems in different fields, in particular in the logistics and transportation systems fields. In the recent literature, even more structurated TSNs have been proposed for representing and solving complex problems, as briefly discussed in the following.

In the logistic field, the activities are related to plan, organise and control materials, goods and information flows, often in an intergrated way. At strategical level, some network design problems, including location, inventory and routing decisions, may arise, while at operative level operational problems emerging in a single node of the logistic network can be considered: from procurement and production to distribution and transportations problems. For what concerns the transportation systems, the activities are related to the design and the management of a network to facilitate the movements of passengers, freight and again information, thus involving both strategical and operational decisions. The mobility is supported by one or more transportation modes, such as road, rail, maritime and air transport.

TSN for logistics problems

In the logistic field, TSNs have been used for solving, just to cite some, service network design, crane scheduling and vehicle routing problems.

An example of service network design is in Agarwal & Ergun [2008]. They use a time-space network to represent a ship scheduling and containerized cargo routing problem formulating it as a multicommodity flow problem with side constraints. Each node of the network represents a port on a day of a week, that is the considered planning horizon. Different types of arcs are used to represent the ships in the ports, their movements and the movements of cargo of the ships.

At operational level, a service problem is presented in Archetti & Peirano [2020]; the authors use a time-space network for solving the air transportation freight forwarder service problem. A freight forwarder has to organize shipments, during a certain horizon, by choosing the best options among a wide offer of transportation services in such a way to minimize costs and respect delivery times. The possible sequences of transportation services available for a shipment, characterized by an origin and a destination, are the layers of the network used by the authors. A flow model is solved on the network.

Scheduling problems, which contain both time and space decisions, are often faced by TSNs. In Guan *et al.* [2013] the authors develop a time-space network flow formulation with non-crossing constraints for the crane scheduling problem for a vessel during terminal operations.

In Yuan & Tang [2017], the authors uses an event-based time-space network flow model to solve a crane scheduling problem in a coil warehouse. They determine, simultaneusly, the sequence of the crane operations (i.e., storage, retrieval and shuffling requests) and the positions to which the coils are moved. In the event-based time-space network, each node represents a location in the warehouse at a specific time of the planning horizon that is the end of a specific scheduling stage, and each edge indicates a move of a crane between two locations in a stage (a stage corresponds to a move of a crane).

The most time-space networks are classified as discrete-time approaches in which the planning horizon is divided into a number of time units of uniform duration. In Yuan & Tang [2017], the authors prefer a different apprach since the decision activities only happen at the start and end point of each movement of the crane, rather than every time point in the planning horizon. Thus, they use a an event-based time-space network flow formulation based on continuous time.

In a recent work, Murakami [2020], a time-space network is used to model the an AGV routing problem formulating it as a mixed-integer linear programming problem. The system is described in details, with its machines, its products and the tasks. Each node of the network represents a machine. A task is picked up, delivered, and then processed by a machine. Finally, the task changes into another one. Each product is finished via few tasks and then conveyed to the depository. The author considers capacity for the machines and time constraints on the pickup-and-delivery.

Mahmoudi & Zhou [2016] propose a three-dimensional state time-space hyper-network to formulate vehicle routing problem with pickup and delivery services with time windows (VRPPDTW) as a time-discretized, multi-dimensional, multi-commodity flow model with linear objective function and constraints. With this novel network, the autors are able to enumerate possible transportation states at any given time along vehicle space time paths and to forward a dynamic programming solution algorithm to solve the single VRPPDTW problem.

Lu *et al.* [2019] propose a resource time-space network to formulate a resource constrained location routing problem as a multi-commodity flow problem. The proposed network is a three dimensional network that combines the space dimension, the time dimension and the resource states. Thus, a node of the phisical network in new three-dimensional network indicates that a vehicle in that physical node maintains a certain resource level at a given time istant. The authors consider both transportation demands and resouces recharging requirements to satisfy. The arcs of the network correspond to a vehicle traveling from one physical location to another one, requiring a certain time and a certain resource.

TSN for transportation systems

The problems arising in the field of transportation planning and management have been faced by many researchers proposing specific time-space networks to respond to particular requirements of the different transportation systems; among others we can cite Li *et al.* [2015] for road network with signal settings, Liu & Zhou [2016] for urban transit network, Lu [2016] for bike-sharing network.

More innovative applications of TSNs are proposed in Tong *et al.* [2015] to maximize the individual accessibility within travel time budgets, in Liu & Zhou [2019] for studying the observability problem in public transportation systems, in Yang *et al.* [2020] to both determine the last-train timetable on each line of the urban rail transit network and evaluate the schedule performances in terms of quality of the different trains connection.

Often a physical network on a plane is transformed into a time-space network represented by three-dimensions (Yang *et al.* [2020]): two coordinates denote the space and the third denotes the discritized time horizon. In this way, each point of the network is used to represent the specific position of a mean of transport or a passenger in a particular time instant. Moreover, it is possible to represent the movements of both the means of transport and passengers thanks to different kind of arcs: time-space running arcs, time-space dwelling arcs, time-space transfer arcs and time-space waiting arcs.

Other applications of TSNs

Despite of the above mentioned uses of the time-space networks in their different forms, some authors adopt them also for facing other problems, related again to the logistics field. Among these innovative uses of networks we can mention Zehendner & Feillet [2014] and Ambrosino & Peirano [2016] that use a TSN for evaluating a truck appointment system for a maritime terminal. Zehendner & Feillet [2014] determine the number of truck appointments to offer while allocating straddle carriers to different transport modes by solving a mixed integer linear programming model based on the network; Ambrosino & Peirano [2016] investigate the management of truck arrivals by offering a non-mandatory truck appointment system; the authors propose a multi-commodity network flow model for representing a general terminal in order to manage the truck arrivals and to grant the trucks a certain service level.

From the above brief discussion, we can note that different TSNs have been proposed to deal with specific requirements of problems having in common spatial and time decisions. The aim of this work is to generalize a TSN in such a way to be able to use it for solving different problems in the logistics field, both at strategical and operational level. In particular, we try to generalize the network proposed to solve a scheduling problem arising in the maritime rail-sea exchange nodes (Ambrosino & Asta [2019]), that is the scheduling of port rail shunting operations (PRSSP). The first aim is to be able to solve the same optimization problem (PRSSP) in different port areas and for different operators that can have to manage different jobs, different resouces on different infrastructures and, then, to solve different flow optimization models for defining the scheduling of several activities sharing some limitated resources and having some time constraints.

3.2 Innovative operation-time-space network

In this section, the innovative operation-time-space network is presented. Let us introduce it by using an example.

Let's consider a network representing the physical layout of a plant where three activities (A,B and C) can be performed, i.e. the physical network reported in Figure 3.1a, in which two paths followed by two jobs are shown. Each path indicates the sequence of activities executed by each job. The weight of each node rapresents the duration of the activity.

Researchers often find more convenient to use a time-space network to follow the execution of activities during time, as reported in Figure 3.1b. The paths indicate the sequence of activities executed by each job, but the time dimension is here easier to understand. Time horizon T is discretized; T is split into equal time intervals $T = \{t_0, t_1, t_2, ...t_s\}$.

In the following, we propose an innovative operation-time-space network that can be easily derived by the time-space network of Figure 3.1b and that permits to know in each time period both the activities that the jobs are executing and the resources consumption.

In the proposed operation-time-space network there are:

- horizontal arcs: arcs that represent the execution, during the time horizon, of the activities;
- vertical arcs (also called transfer arcs): arcs that permit the jobs to pass from one actity to another; transfer arcs are present for each couple of compatible activities, i.e. couple of activities that can be realized in sequence.

This network is depicted in Figure 3.2a, where the two paths of the jobs corresponding to those reported in Figure 3.1 are shown.

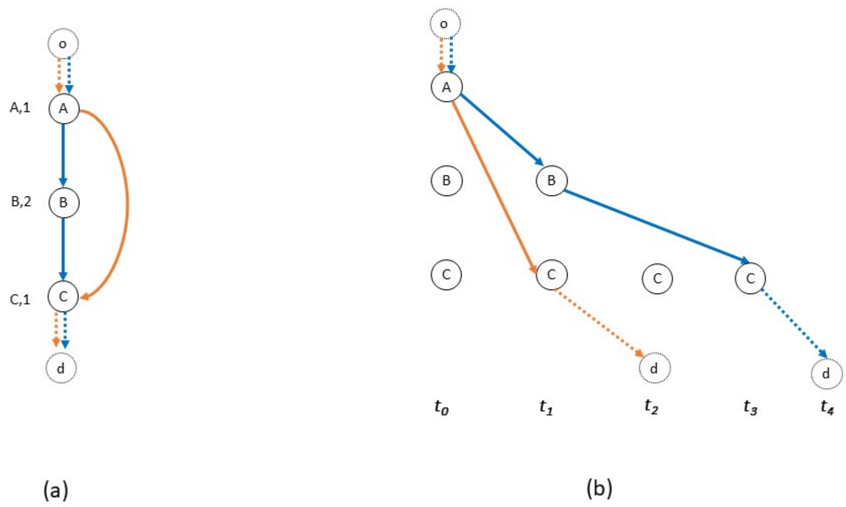

Figure 3.1: Physical network, time-space network

Each time interval t in T is here denoted $[t; t+1)$ and when an operation is related to time t (i.e., has index t) means that it happens in the interval $[t; t+1)$. The blue job in Figure 3.2a enters in the network in t_0 and immediately starts activity A. The blue job finishes activity A in t_1 and passes to activity B which starts in t_1. Activity B is performed until t_3; then, the job starts activity C. After having completed C, the job in t_4 leaves the network. The readers can note that the nodes of the network represent the activities but, thanks to the vertical arcs, nodes also indicate the ending and the starting time of the corresponding activities.

This operation-time-space network allows an easier implementation of capacity contraints that may regard either an activity, a set of activities or the whole network. Moreover, different levels of detail are allowed; for example we can

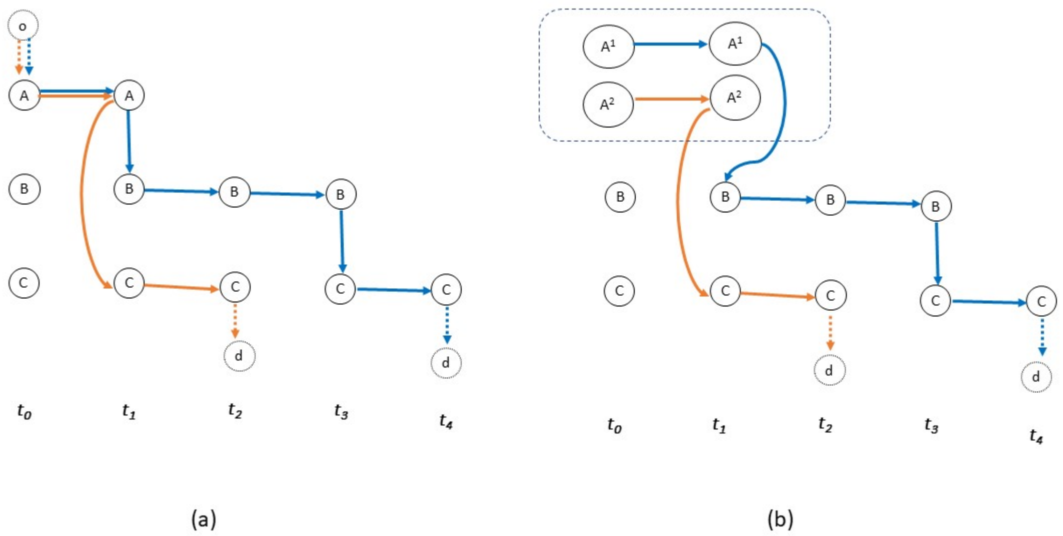

Figure 3.2: operation-time-space network

include details on the resources required to perform the activities.

Looking at Figure 3.2a, we can note that activity A has enough capacity to perform both the jobs in t_0. The resources needed to perform activity A are not explicity considered in the network. Now, suppose that it is required to distinguish among resources available for performing activity A. In this case we can duplicate the node A, one node A for each resource, as in Figure 3.2b (where two resources are considered).

Now, let us suppose to have a new job, the green one in Figure 3.3, that has to pass through B and C. We can distinguish different cases:

i) B can operate two items in each t (and the resources used in B are not distinguished); the paths on the network in Figure 3.3a represent this new situation.

ii) B has the capacity to execute only one job in each t; therefore, the blue job cannot start activity B before t_2. Two situations can arise:

ii.a) there is not the possibility for the job to wait in the network, that is each job has to execute its activities in sequence. In this case, the blue job has to perform activity B as soon as it finishes activity A, and for doing that the blue job has to enter in the network later, in t_1. Then, it has to execute A and, in t_2, it starts B, as depicted in Figure 3.3b.

ii.b) the job can wait in the network, which means that there is a waiting area/buffer (W). In this case, the blue job finishes activity A and has to wait before starting activity B since in t_1 there is the green job. The blue job enters in the buffer in t_1, remains in W and then in t_2 passes to activity B, as depicted in Figure 3.4.

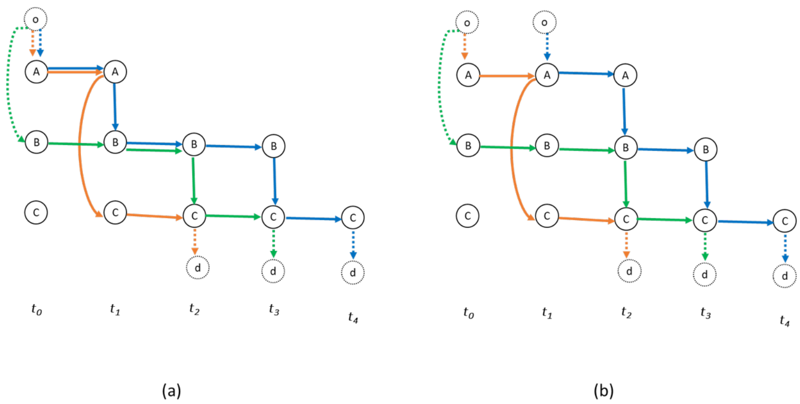

Figure 3.3: operation-time-space network without wait buffers

This operation-time-space network also allows an easier implementation of time constraints. In addition to precedence relationships that are represented in the network by vertical arcs, a job can have time constraints indicating the exact starting and/or ending time for an activity that it has to execute and either a deadline or a time window for starting and/or ending an activity. These constraints are

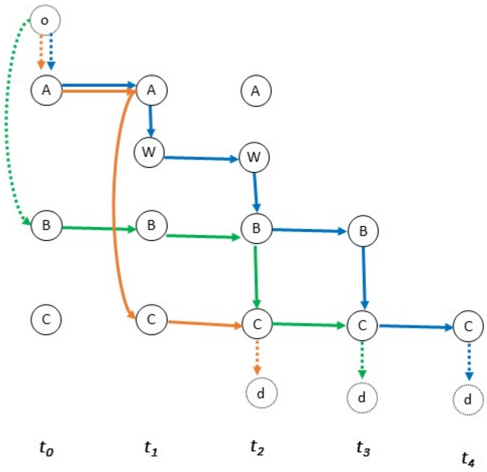

Figure 3.4: operation-time-space network with wait buffers

discussed in details in Guerriero & Talarico [2010] where the authors developed a method to find the critical path in a network with different types of time constraints on activities.

We will show how this operation-time-space network permits to consider all these time constraints in Section 3.2.1, where the flow model formulation is presented.

Note that, different optimization problems can be solved as 0/1 integer linear flow models based on the discretized operation-time-space network described above.

Let us introduce more formally this network.

Let $G = (N, A)$ be the operation-time-space network, where N is the set of nodes and A the set of arcs.

Note that, N is the union of different subsets that have to be better specified; their specification depends on the presence of one or more resources that we need to consider separately for each activity (as shown in Figure 3.2). Moreover, N includes also the source and the sink nodes of the network.

Let be:

O the set of activities that can be performed and that we want to model on G, included also the entry and the exit from the network;

J the set of jobs that need to perform either all activities in O or some of the activities in O;

O_j the set of activities that job j must perform;

O^+ the set of activities for which we have to distinguish the available resources in terms of different alternative tracks that can be used for the activity itself (e.g. the waiting operation to exploit in a park that is composed by several tracks would belong to this set);

$O^- = O - O^+$ the set of activities for which we have not to distinguish the available resources;

O^C the set of couple of activities that can be executed in sequence, necessary to define transfer arcs (that permit the jobs to pass from one activity to another and to enter and leave the network);

$\mathcal{R}_i$ the set of resources available for executing the activity i, $i \in O^+$;

H the set of groups of activities sharing a given resource;

O_h the set of activities of group h, $h \in H$.

Nodes of the network

$\mathcal{N} = \cup_{i \in O} \mathcal{N}^i$ where, depending on the necessity to distinguish the set of operations in O^- and O^+ :

$\mathcal{N}^i = \{n_t^i | t \in T\}$, $\forall i \in O^-$ the set of nodes representing the activity i, in each t of the time horizon;

$\mathcal{N}^i = \cup_{r \in \mathcal{R}_i} \mathcal{N}_r^i$, $\forall i \in O^+$ with

$\mathcal{N}_r^i = \{n_{r,t}^i | t \in T\}$ the set of nodes representing the resource r of activity i, $i \in O^+$, $r \in \mathcal{R}_i$, in each t of the time horizon.

Arcs of the network

$\mathcal{A} = (\cup_{i \in O} \mathcal{A}^{\mathcal{N}^i}) \cup (\cup_{i,j \in \mathcal{O}^{\mathcal{C}}} \mathcal{A}^{\mathcal{N}^i,\mathcal{N}^j})$ the set of arcs of the network, given by the union of horizontal arcs ($\mathcal{A}^{\mathcal{N}^i}$) and vertical ones ($\mathcal{A}^{\mathcal{N}^i,\mathcal{N}^j}$), defined as in the following.

As before, each subset of the horizontal arcs $\mathcal{A}^{\mathcal{N}^i}$ can be defined in one of the following ways, depending on the kind of activity it refers to:

$\mathcal{A}^{\mathcal{N}^i}$ the set of arcs $\{(n_t^i, n_{t+1}^i), \forall t \in T\}$, $\forall i \in O^-$

$\mathcal{A}^{\mathcal{N}^i} = \cup_{r \in \mathcal{R}_i} \mathcal{A}^{\mathcal{N}_r^i}$, $\forall i \in O^+$ with

$\mathcal{A}^{\mathcal{N}_r^i}$ the set of arcs $\{(n_{r,t}^i, n_{r,t+1}^i), \forall t \in T\}$, $\forall r \in \mathcal{R}_i, \forall i \in O^+$

For what concerns vertical arcs, they link couple of activities that can be executed in sequence. Some different definitions are required, depending on the kind of activity they refer to:

$\mathcal{A}^{\mathcal{N}^i,\mathcal{N}^l}$ the set of arcs $\{(n_t^i, n_t^l), \forall t \in T\}, \forall i, l \in O^C : i, l \in O^-$

$\mathcal{A}^{\mathcal{N}^i,\mathcal{N}^l}$ the set of arcs $\{(n_t^i, n_{r,t}^l), \forall t \in T, \forall r \in \mathcal{R}_l\}, \forall i, l \in O^C : i \in O^-, l \in O^+$

$\mathcal{A}^{\mathcal{N}^i,\mathcal{N}^l}$ the set of arcs $\{(n_{r,t}^i, n_t^l), \forall t \in T, \forall r \in \mathcal{R}_i\}, \forall i, l \in O^C : i \in O^+, l \in O^-$

$\mathcal{A}^{\mathcal{N}^i,\mathcal{N}^l}$ the set of arcs $\{(n_{r,t}^i, n_{r',t}^l), \forall t \in T, \forall r \in \mathcal{R}_i, \forall r' \in \mathcal{R}_l\}, \forall i, l \in O^C : i, l \in O^+$

Note that, we denote:

$\mathcal{A}_t$ the subset of arcs of the network related to time period t, $t \in T$;

$\mathcal{A}_t^+$ the set of outbound arcs in t, $t \in T$;

$\mathcal{A}_t^{\mathcal{N}^i,+}$ ($\mathcal{A}_t^{\mathcal{N}_r^i,+}$) the subset of outbound vertical arcs of node n_t^i ($n_{r,t}^i$), that depends on the compatible activities;

$\mathcal{A}_t^{-,\mathcal{N}^i}$ ($\mathcal{A}_t^{-,\mathcal{N}_r^i}$) the subset of inbound vertical arcs of node n_t^i ($n_{r,t}^i$), that depends on the compatible activities.

3.2.1 The network flow model

In this section, we introduce a flow formulation based on the network described above that can be adapted to solve many real applications.

The following additional notation useful for the flow formulation is introduced.

d_{ij} the duration of activity i for job j, $\forall j \in J, \forall i \in O_j$;

s_{ij} the eventual obliged starting time of activity i for job j, $\forall j \in J, \forall i \in O_j$;

e_{ij} the eventual obliged ending time of activity i for job j, $\forall j \in J, \forall i \in O_j$;

$[s_{ij}^{min}, s_{ij}^{max}]$ the time window for job j for starting the execution of activity i, $\forall j \in J, \forall i \in O_j$;

$[e_{ij}^{min}, e_{ij}^{max}]$ the time window for job j for completing activity i, $\forall j \in J, \forall i \in O_j$;

u_t^i the maximun number of jobs that can execute activity i in t, $\forall i \in O^-, \forall t \in T$;

k_t^h the maximun number of jobs that can execute the activities of group h in t, $\forall h \in H, \forall t \in T$;

q_t the maximun number of jobs that, in each t, can be present in the nework, $\forall t \in T$.

Decision variables

The model based on the network above descripted has the following flow variables indicating for each job, when it is performing the operations (horizontal arcs) and when it is passing from one operation to another (vertical arcs); thus, let be:

$x_{a,j} \in \{0,1\} \ \ \forall j \in J, \forall a \in \mathcal{A}$,

$x_{a,j} = 1$ if arc a is used for job j.

Constraints

The constraints of the flow model, from the classical flow conservation constraints to capacity and time constraints are presented here below. Note that, the presence of activities, and thus of nodes, that are split for specifing the available resources, affects the readibility of this notation, but permits a real representation of particular capacity constraints.

Flow conservation constraints

The flow conservation constraints assure, for each unit of flow on the network, i.e. for each job j, that the sum of all the variables related to the inbound arcs of a node in time instant t is equal to the sum of the variables related to the outbound arcs of the same node.

$$\sum_{a:a\in\mathcal{A}_{t-1}^{\mathcal{N}^i}\cup\mathcal{A}_t^{-,\mathcal{N}^i}} x_{a,j} = \sum_{a:a\in\mathcal{A}_t^{\mathcal{N}^i}\cup\mathcal{A}_t^{\mathcal{N}^i,+}} x_{a,j} \quad \forall j\in J, \forall i\in O_j, \forall t\in T$$

Time constraints

The constraints related to time requirements can be easily written specifying which are the time instants t to include in the sum of the constraints.

For example, if it is required a job j to start activity i within a time window $[s_{ij}^{min}, s_{ij}^{max}]$, we can simply require to have one arc entering in node n_t^i ($n_{r,t}^i$) with $s_{ij}^{min} \leq t \leq s_{ij}^{max}$. The resulting constraints are the following:

$$\sum_{a:a\in(\cup_{s_{ij}^{min}\leq t\leq s_{ij}^{max}}\mathcal{A}_t^{-,\mathcal{N}^i})} x_{a,j} = 1 \quad \forall j\in J, \forall i\in O_j$$

If the requirement for job j is to end activity i within a time window we can simply re-write the above constraints requiring to have one arc exiting from the node within the time window.

If the requirement for job j is to end activity i within a deadline e_{ij} we can simply require to have one arc exiting from node n_t^i ($n_{r,t}^i$) within the fixed time. The resulting constraints are the following:

$$\sum_{a:a\in(\cup_{t\leq e_{ij}}\mathcal{A}_t^{\mathcal{N}^i,+})} x_{a,j} = 1 \quad \forall j \in J, \forall i \in O_j$$

If job j must complete all its activities within a time limit (that is it has to leave the network within a time limit), having considered the exit of the job j as an operations in O_j, we can use the above constraints formulation.

If job j has to start the activity i in a precise instant s_{ij}, constraints are like the following:

$$\sum_{a:a\in\mathcal{A}_{t=s_{ij}}^{-,\mathcal{N}^i}} x_{a,j} = 1 \quad \forall j \in J, \forall i \in O_j$$

Capacity constraints for each activity/for a group of activities/for the whole network

If there is a limitation on the number of jobs that can execute simultaneusly an activity i, we need some constraints imposing that the sum of variables related to the outbound arcs of node n_t^i must be less or equal to this limit (i.e. u_t^i) for each t:

$$\sum_{j\in J}\sum_{a:a\in\mathcal{A}_t^{\mathcal{N}^i}\cup\mathcal{A}_t^{\mathcal{N}^i,+}} x_{a,j} \leq u_t^i \quad \forall t \in T, \forall i \in O^-$$

For activities with specific resources, i.e., $i \in O^+$, and thus for nodes $n_{r,t}^i$ we need the following constraints requiring the sum of variables related to the outbound arcs of node $n_{r,t}^i$ to be less or equal to one for each t:

$$\sum_{j\in J}\sum_{a:a\in\mathcal{A}_t^{\mathcal{N}_r^i}\cup\mathcal{A}_t^{\mathcal{N}_r^i,+}} x_{a,j} \leq 1 \quad \forall t \in T, \forall i \in O^+, \forall r \in R_i$$

At most k_t^h jobs can execute simultaeously activities of group h in each t.

Therefore, the sum of variables related to the outbound arcs of the nodes (either n_t^i or $n_{r,t}^i$) of the activities belonging to the group h (O_h) must be less or equal to k_t^h for each t:

$$\sum_{j \in J} \sum_{a: a \in \cup_{i \in O_h} (\mathcal{A}_t^{\mathbb{N}^i} \cup \mathcal{A}_t^{\mathbb{N}^i,+})} x_{a,j} \leq k_t^h \quad \forall h \in H, \forall t \in T$$

The maximum number of activities that can be executed simultaneously depends on the number of available resources shared among all the activities, in each t. Therefore, the formulation of the required constraints follows:

$$\sum_{j \in J} \sum_{a: a \in \mathcal{A}_t^+} x_{a,j} \leq q_t \quad \forall t \in T$$

Processing time of the activities

The time that job j has to spend for executing activity i is known (i.e. d_{ij}); therefore, the following constraints are defined:

$$\sum_{a \in \mathcal{A}^{\mathbb{N}^i}} x_{a,j} = d_{ij} \quad \forall j \in J, \forall i \in O_j$$

Objective function

The flow optimization models used for solving logistics problems present different objective functions, among others the maximisation of the flow, the minimization of the total costs of the flow passing through the network. In scheduling problems the minimization of the total time required to execute all the jobs is a common objective. In time constrained networks different aims can be persuived, like the minimization of the total waiting time spent by the jobs in the waiting areas (buffers).

Chapter 4

Port Rail Shunting Scheduling Problem (PRSSP)

Summary

Chapter 4 deals with the Port Rail Shunting Scheduling Problem introducted in the paper **D. Ambrosino, V. Asta, T.G. Crainic. Port Rail Shunting Scheduling Problem**, submitted for pubblication in 2021. It includes the problem definition (Section 4.1), the model approach (Section 4.2) with some possible improvements (Section 4.3) and the computational tests (Section 4.4).

4.1 Problem definition and basic notation

Since this moment, it is assumed a general layout of a port shunting area in order to define the PRSSP. The general layout includes one railway station, one shunting zone and several maritime terminals. Moreover, as assumption, the railway station has a given number of tracks connected with the shunting zone. The shunting zone is composed by one park with several tracks and other possible tracks. These latter are used to join the station to the shunting park, the shunting park to the terminals and the station directly to the terminals. The considered layout is shown in Fig. 4.1, where, just to let be the figure more readible, it is assumed that two tracks are present in the railway station and in the shunting park, and two maritime terminals can be reached.

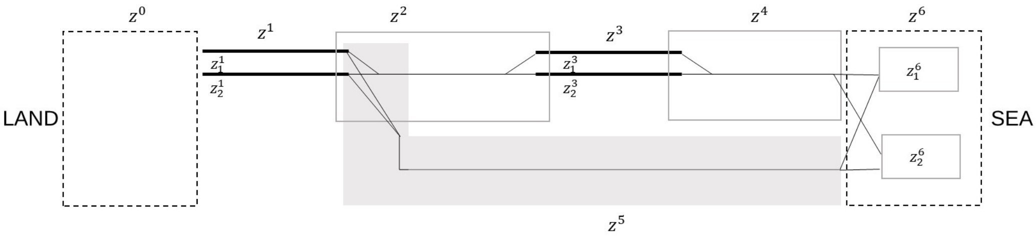

Figure 4.1: Port area as modal exchange node - Physical layout scheme

Let's describe the specific elements in more details.

- z^0 represents the whole railway network; it is included in this sketch just for completeness since it represents the rail transport system imposing some time constraints to the shunting manager;
- z^1 represents the railway station with its two tracks here denoted (z_1^1 and z_2^1) where the waiting operations for trains arriving from either the railway network or the shunting area can be performed;
- z^2 represents the connection between the station (z^1) and the park (z^3). This zone, inside the shunting area, is here called primary area. The connection track in the primary area is used to transfer trains from the railway station to the shunting park and viceversa;
- z^3 represents the shunting park, in the following just called park, with its two tracks here denoted (z_1^3 and z_2^3) where the waiting operations for trains arriving from either the railway station or the terminals can be performed;

- z^4 represents the connection between the park (z^3) and the terminals (z^6). This zone, inside the shunting area, is here called secondary area. The connection track in the secondary area is used to transfer trains from the shunting park to the terminals and viceversa;

- z^5 represents the connection between the station (z^1) and the terminals (z^6). This zone, inside the shunting area, is here called unique area. The connection track in the unique area is used for the direct transfer of trains from the railway station to the maritime terminals and viceversa;

- z^6 represents the seaside exchange node, here represented by two terminals (z^6_1, z^6_2), that are either the origin or the destination of trains passing through the port area.

Let us indicate as I the set of import trains, those having the terminals as origin, and E the set of export trains, those having the terminals as destination. Thus, J, the union of sets E and I, represents the trains that have to be managed during the time horizon T.

Time horizon T is discretized; T is split into τ equal time intervals $T = \{1, 2, ...\tau\}$. Each time interval t in T is here denoted $[t; t+1)$ and when an operation is related to time t (i.e., has index t) means that it happens in the interval $[t; t+1)$.

An export train j arrives at the railway station in a given time instant e_j (i.e., at the beginning of time interval $[e_j; e_j+1)$). It's here assumed that it has to be at destination, i.e., inside its destination terminal (p_j), within a given time window $[e_j^{min}; e_j^{max}]$ that is a subset of T.

On the opposite, let's assume that an import train j' is ready inside its origin terminal $p_{j'}$ for being picked up within a given time window $[e_{j'}^{min}; e_{j'}^{max}]$ and has to depart from the railway station in a given time instant $e_{j'}$.

Each train has to perform specific operations in order to be transferred within the port area, that is it has to pass through some of the different zones in the shunting area (see Fig.4.1).

Let us indicate Z the set of zones of the port area in which shunting and waiting operations are preformed and Z_j the set of zones that must be visited by train j.

An export train may have to realize one of the following paths:

- it arrives at the railway station z^1 in its fixed arrival time, enters the shunting zone for reaching the unique area z^5 and finally enters in its destination terminal in z^6;
- it arrives at the railway station z^1 in its fixed arrival time, enters the shunting zone for reaching the primary area z^2, enters the park z^3, enters the secondary area z^4 and finally enters in its destination terminal in z^6.

For an import train the possible paths are the following:

- it leaves its origin terminal in z^6, enters the shunting zone for reaching the unique area z^5 and finally arrives at the railway station z^1 from which it will enter in the national railway network respecting its fixed departure time;
- it leaves its origin terminal in z^6, enters the shunting zone for reaching the secondary area z^4, enters the park z^3, enters the primary area z^2 and finally arrives at the railway station z^1 from which it will enter in the national railway network respecting its fixed departure time.

The time necessary to execute the operations in each zone on each train is known too, and is here denoted $d_{i,j}$ $\forall i \in Z, \forall j \in J$.

The specific operations that we have considered through the identified zones allow us to avoid to consider eventual transfer times between operations because, given that some operations represent exactly the transfer of trains between zones, these times are included in the processing time. Let's see Fig.4.1, the operation through z^2 is the one for transferring a train from a track of the station (z^1) to a track of the shunting park (z^3). This operation doesn't need other transfer times to be considered because its starting time represents the train that starts the transfer from the track in z^1 and the ending time represents the train that end the transfer on one track of the z^3. At this point, the train is located on a track of the shunting park where it immediately starts the waiting operation. And so on with all the other operations. For what concerns the coupling and decoupling times for the shunting engines, they have been included in the processing times of the operations ($d_{i,j}$).

Note that, in each time interval, there is a limited number of teams (k_t) for realizing all the operations in the different zones. Moreover, there is also a capacity for the number of operations that can be realized in each time interval, in each zone, depending on the specific layout of these zones. Let u_t^i be the maximum number of operations that, in each time interval, can be realized in zone i, $\forall i \in Z$.

In the present problem, in each zone only one type of operation can be executed on the trains; thus, in the following, we will identify the operations with the zones and we will refer to them using the term "operation i".

The decisions to take are related to the time instant in which to perform the required operations on each train, in such a way to respect time constraints (the time windows for entering and leaving terminals, the arrival/departure time at the railway station) and capacity constraints related to the resources needed to perform the shunting operations.

Due to the above mentioned limited resources, a train may has to wait when passing from one zone to another in the port area. A train can wait in the railway station and in the park. In these cases, the train is performing a waiting operation. Thus, the time spent by a train to go trough its path depends on the duration of both the shunting operations and the waiting ones. The main aim is to perform all the required operations on trains minimizing their total waiting time.

4.2 The model approch used for PRSSP

The model developed for solving PRSSP is based on an operation-time-space network representing the operations that might be performed on the import and export trains in each zone of the port (see Fig. 4.1 in Section 4.1) and the terminals that are either the origin or the destination of the trains. This network has been derived from the one proposed in Ambrosino and Asta, 2020 and explained in Chapter 3.

The nodes of the network, representing the zones and, at the same time, the operations to execute on trains, are replicated for each time interval of the schedule horizon T. Arcs are events representing the starting time of each operation, more precisely, vertical arcs represent the end of a given operation in the time instant t and, therefore, the simultaneous begin of the following required operation, and, thus, the transfer of a train from one zone to another one. The horizontal arcs represent the temporal advancement of the operations, i.e. from t to $t+1$, that is the time spent by a train in a given zone. Note that, in some zones it is necessary to distinguish the available resources, such as the tracks, while in other zones this is not required.

The operation-time-space network and the flow network model developed for PRSSP are explained in the following.

4.2.1 The operation-time-space network for PRSSP

Starting from the layout shown in Fig.4.1 of Section 4.1, let us define the operations time space network $G = (N, A)$, where N is the set of the nodes and A the set of the arcs. In the next, we will use only the term network to refer to this operations time space network. Note that, the set of zones (operations) Z that we want to represent as nodes of G includes zones in which we have to distinguish the available resources used for performing operations on trains; for this purpose, let us define:

Z^+ the set of operations for which we have to distinguish the available resources, i.e., the available tracks (in the considered layout, see Fig. 4.1, $Z^- = \{z^1, z^3\}$);

$Z^- = Z - Z^+$ the set of operations for which we have not to distinguish the available resources (in the considered layout, see Fig.4.1, $Z^- = \{z^2, z^4, z^5\}$);

$Z^C = Z^C(E) \cup Z^C(I)$ the set of couple of operations that can be executed in sequence, both on export trains ($Z^C(E)$) and on import ones ($Z^C(I)$), necessary to define transfer arcs (that permit the trains to pass from one operation to another, i.e. from one zone to another one, and to enter and leave the network).

$\mathcal{R}_i$ the set of resources available for executing the operation i, $i \in Z^+$.

Nodes of the network

$\mathcal{N} = \cup_{i \in Z} \mathcal{N}^i$ where :

$\mathcal{N}^i = \{n_t^i | t \in T\}$, $\forall i \in Z^-$ the set of nodes representing the operation i, in each t of the time horizon;

$\mathcal{N}^i = \cup_{r \in \mathcal{R}_i} \mathcal{N}_r^i$, $\forall i \in Z^+$ with

$\mathcal{N}_r^i = \{n_{r,t}^i | t \in T\}$ the set of nodes representing the resource r available for operation i, $i \in Z^+$, $r \in \mathcal{R}_i$, in each t of the time horizon.

Arcs of the network

$\mathcal{A} = (\cup_{i \in Z} \mathcal{A}^{\mathcal{N}^i}) \cup (\cup_{i,j \in Z^{\mathcal{C}}} \mathcal{A}^{\mathcal{N}^i, \mathcal{N}^j})$ the set of arcs of the network, given by the union of horizontal arcs ($\mathcal{A}^{\mathcal{N}^i}$) and vertical ones ($\mathcal{A}^{\mathcal{N}^i, \mathcal{N}^j}$), defined as in the following:

$\mathcal{A}^{\mathcal{N}^i}$ the set of arcs $\{(n_t^i, n_{t+1}^i), \forall t \in T\}$, $\forall i \in Z^-$

$\mathcal{A}^{\mathcal{N}i} = \cup_{r \in \mathcal{R}_i} \mathcal{A}^{\mathcal{N}_r^i}$, $\forall i \in Z^+ - Z^6$ with

$\mathcal{A}^{\mathcal{N}_r^i}$ the set of arcs $\{(n_{r,t}^i, n_{r,t+1}^i), \forall t \in T\}$, $\forall r \in \mathcal{R}_i, \forall i \in Z^+ - Z^6$

Note that, terminals represent either origin or destination for the trains; we are not interested at representing the operations on trains inside the terminals, thus, in the network, horizontal arcs connecting the terminals are not necessary.

For what concerns vertical arcs, they link couple of operations that can be executed in sequence and different definitions are required, depending on the kind of operations they refer to. In general form, let us denote:

$\mathcal{A}^{\mathcal{N}^i,\mathcal{N}^l}$ the set of arcs $\forall i, l \in Z^C$

In detail:

$\mathcal{A}^{\mathcal{N}^i,\mathcal{N}^l}$ the set of arcs $\{(n_t^i, n_t^l), \forall t \in T\}, \forall i, l \in Z^C : i, l \in Z^-$

$\mathcal{A}^{\mathcal{N}^i,\mathcal{N}^l}$ the set of arcs $\{(n_t^i, n_{r,t}^l), \forall t \in T, \forall r \in \mathcal{R}_l\}, \forall i, l \in Z^C : i \in Z^-, l \in Z^+$

$\mathcal{A}^{\mathcal{N}^i,\mathcal{N}^l}$ the set of arcs $\{(n_{r,t}^i, n_t^l), \forall t \in T, \forall r \in \mathcal{R}_i\}, \forall i, l \in Z^C : i \in Z^+, l \in Z^-$

$\mathcal{A}^{\mathcal{N}^i,\mathcal{N}^l}$ the set of arcs $\{(n_{r,t}^i, n_{r',t}^l), \forall t \in T, \forall r \in \mathcal{R}_i, \forall r' \in \mathcal{R}_l\}, \forall i, l \in Z^C : i, l \in Z^+$

Moreover, let us denote:

$\mathcal{A}_t^{\mathcal{N}^i}$ ($\mathcal{A}_t^{\mathcal{N}_r^i}$) the horizontal outbound arc of node $n_{r,t}^i$, in t, i.e. $(n_t^i, n_{t+1}^i)(n_{r,t}^i, n_{r,t+1}^i)$;

$\mathcal{A}_t^{\mathcal{N}^i,\mathcal{N}^l}$ the vertical arc from node n_t^i to node n_t^l in t, with $i, l \in Z^C$ and $i, l \in Z^-$;

$\mathcal{A}_t^{\mathcal{N}^i,+}$ ($\mathcal{A}_t^{\mathcal{N}_r^i,+}$) the subset of vertical outbound arcs of node n_t^i ($n_{r,t}^i$), that depends on the compatible operations;

$\mathcal{A}_t^{-,\mathcal{N}^i}$ ($\mathcal{A}_t^{-,\mathcal{N}_r^i}$) the subset of vertical inbound arcs of node n_t^i ($n_{r,t}^i$), that depends on the compatible operations.

Fig.4.2 shows a portion of the network used in the present work for PRSSP in two cases: (a) for the export cycle and (b) for the import one. Note that, all the nodes are represented and only two tracks, for the station and the park, and two terminals are shown.

Concerning the arcs of the network, only the ones that can be travelled for going from z^0 to z^6 (export cycle) and viceversa (import cycle), which represent

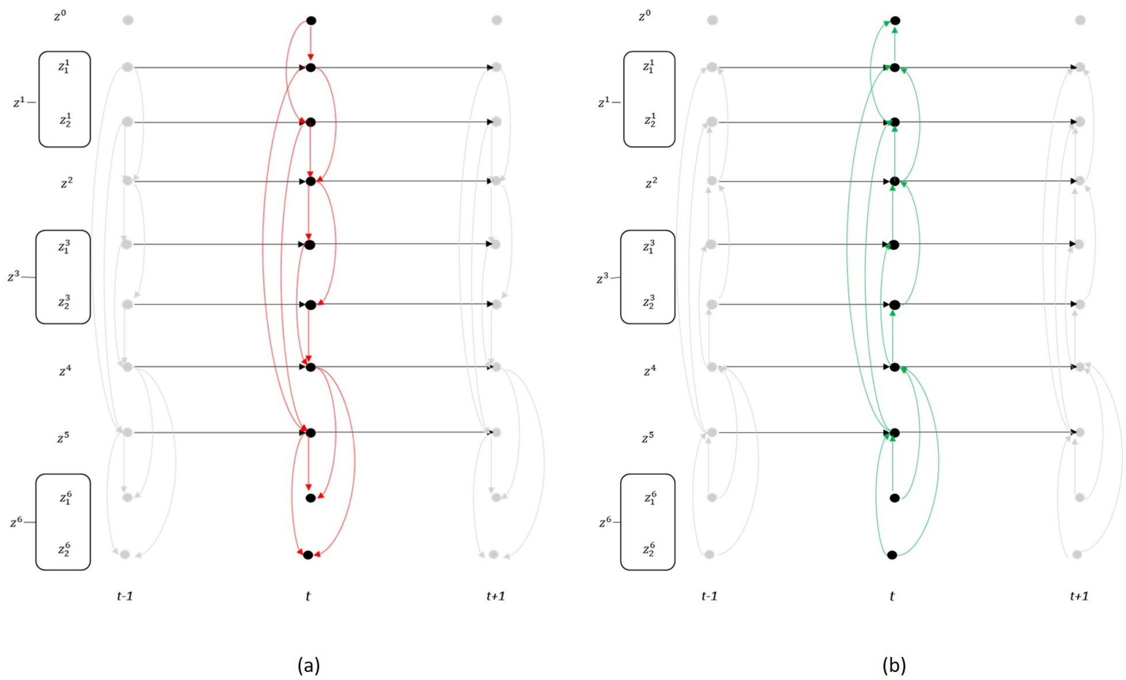

Figure 4.2: Portion of the network

the couples of compatible operations, are shown. In particular, the red arcs in (a) are for the export cycle and the gree arcs in (b) are for the import one.

In the next section, we introduce the flow formulation used to solve PRSSP.

4.2.2 The network flow model for PRSSP

Each export train that has to be transferred from the rail station to the maritime terminal represents a unit of flow that enter in a given time instant and has to reach its destination terminal within a given time window. The viceversa is for import trains. Note that, while managing trains some capacity constraints must be satisfied.

Let us now introduce the additional useful notation for the flow formulation.

The model based on the above described network has both the flow decision variables and the auxiliary variables for computing the train's waiting time in the network (to be minimized); let be:

$x_{a,j} \in \{0,1\}, \forall j \in J, \forall a \in A$, $x_{a,j} = 1$ if arc a is used for train j;

$y_j^i \geq 0, \forall j \in J, \forall i \in \{z^1, z^3\}$, define the total time spent by a train j in the rail station or in the shunting park, waiting either for its departure or for starting the shunting operations.

The resulting model is the following:

$$MIN \sum_{j \in J} \sum_{i \in \{z^1, z^3\}} y_j^i \tag{4.1}$$

subject to:

$$\sum_{a \in \mathcal{A}^{N^i}} x_{a,j} = y_j^i \quad \forall j \in J, \forall i \in \{z^1, z^3\} \tag{4.2}$$

$$\sum_{a \in \mathcal{A}_{e_j}^{N^{z^0}, N^{z^1}}} x_{a,j} = 1 \quad \forall j \in E \tag{4.3}$$

$$\sum_{a \in \mathcal{A}_{e_j}^{N^{z^1}, N^{z^0}}} x_{a,j} = 1 \quad \forall j \in I \tag{4.4}$$

$$\sum_{a \in (\cup_{e_j^{min} \leq t \leq e_j^{max}} \mathcal{A}_t^{\mathcal{N}^-, z_r^6})} x_{a,j} = 1 \quad \forall j \in E, r = p_j \tag{4.5}$$

$$\sum_{a \in (\cup_{a_j^{min} \leq t \leq a_j^{max}} \mathcal{A}_t^{\mathcal{N}^{z_r^6}, +})} x_{a,j} = 1 \quad \forall j \in I, r = p_j \tag{4.6}$$

$$\sum_{a \in \mathcal{A}_{t-1}^{\mathcal{N}^i} \cup \mathcal{A}_t^{-,\mathcal{N}^i}} x_{a,j} = \sum_{a \in \mathcal{A}_t^{\mathcal{N}^i} \cup \mathcal{A}_t^{\mathcal{N}^i,+}} x_{a,j} \quad \forall j \in J, \forall i \in Z_j, \forall t \in T \tag{4.7}$$

$$\sum_{j \in J} \sum_{a \in \mathcal{A}_t^{\mathcal{N}_r^i} \cup \mathcal{A}_t^{\mathcal{N}_r^i,+}} x_{a,j} \leq 1 \quad \forall t \in T, \forall i \in \{z^1, z^3\}, \forall r \in \mathcal{R}_i \tag{4.8}$$

$$\sum_{j \in J} \sum_{a \in \mathcal{A}_t^{\mathcal{N}^i} \cup \mathcal{A}_t^{\mathcal{N}^i,+}} x_{a,j} \leq u_t^i \quad \forall t \in T, \forall i \in Z^- \tag{4.9}$$

$$\sum_{j \in J} \sum_{a \in \cup_{i \in Z^-} (\mathcal{A}_t^{\mathcal{N}^i} \cup \mathcal{A}_t^{\mathcal{N}^i,+})} x_{a,j} \leq k_t \quad \forall t \in T \tag{4.10}$$

$$\sum_{a \in \mathcal{A}^{\mathcal{N}^i}} x_{a,j} = d_{ij} \quad \forall j \in J, \forall i \in Z_j \tag{4.11}$$

$$\sum_{j \in J} \sum_{a \in \mathcal{A}_t^{-,\mathcal{N}_r^{z^6}} \cup \mathcal{A}_t^{\mathcal{N}_r^{z^6},+}} x_{a,j} \leq 1 \quad \forall t \in T, \forall r \in R_{z^6} \tag{4.12}$$

The objective function (4.1) minimizes the sum of the trains' waiting time both in the shunting park and in the rail station. Constraints (4.2) define the auxiliary variables $y_j^{z_1}$ and $y_j^{z_3}$. Constraints (4.3) and (4.4) impose that for each export/import train there must be one variable representing the entrance/exit of the train in the network equal to 1 exactly at time $t = e_j$:

Constraints (4.5) and (4.6) are related to both the entrance and the exit of the trains in/from the their destination/origin terminals; they permit to satisfy the time windows for the entry into and the exit from the terminals.

The flow conservation constraints (4.7) assure, for each unit of flow on the network, i.e. for each train j, that the sum of all the variables related to the inbound arcs of a node in time instant t is equal to the sum of the variables related to the outbound arcs of the same node.

As far as the capacity of tracks is considered, thanks to constraints (4.8) it is imposed that at most one train can wait on each station's track and on each shunting park's tracks in each time instant t.

Given that we consider the set of tracks and junction needed for operations as a single track we have to include in the model the constraints (4.9) to limit the number of simultaneous shunting operations.

The maximum number of simultaneous shunting operations in the whole port area depends on the number of available shunting teams for each time instant t as imposed by constraints (4.10).

Constraints (4.11) fix the processing time of each shunting operation that each train has to execute to the required time d_{ij}.

Constraints (4.12) are used to impose that there must be at maximum one shunting operation for each time instant involving the same terminal.

Note that, in the following, we will refer to the model (4.1) - (4.12) using MOD-0.

Fig.4.3 shows an example of flows in the network. The paths of two trains, one export and one import, that have to be trasferred within the considered area are depicted. In detail, the two paths represent the sequence of operations to be performed by each train: the green for the export train and the red for the import one.

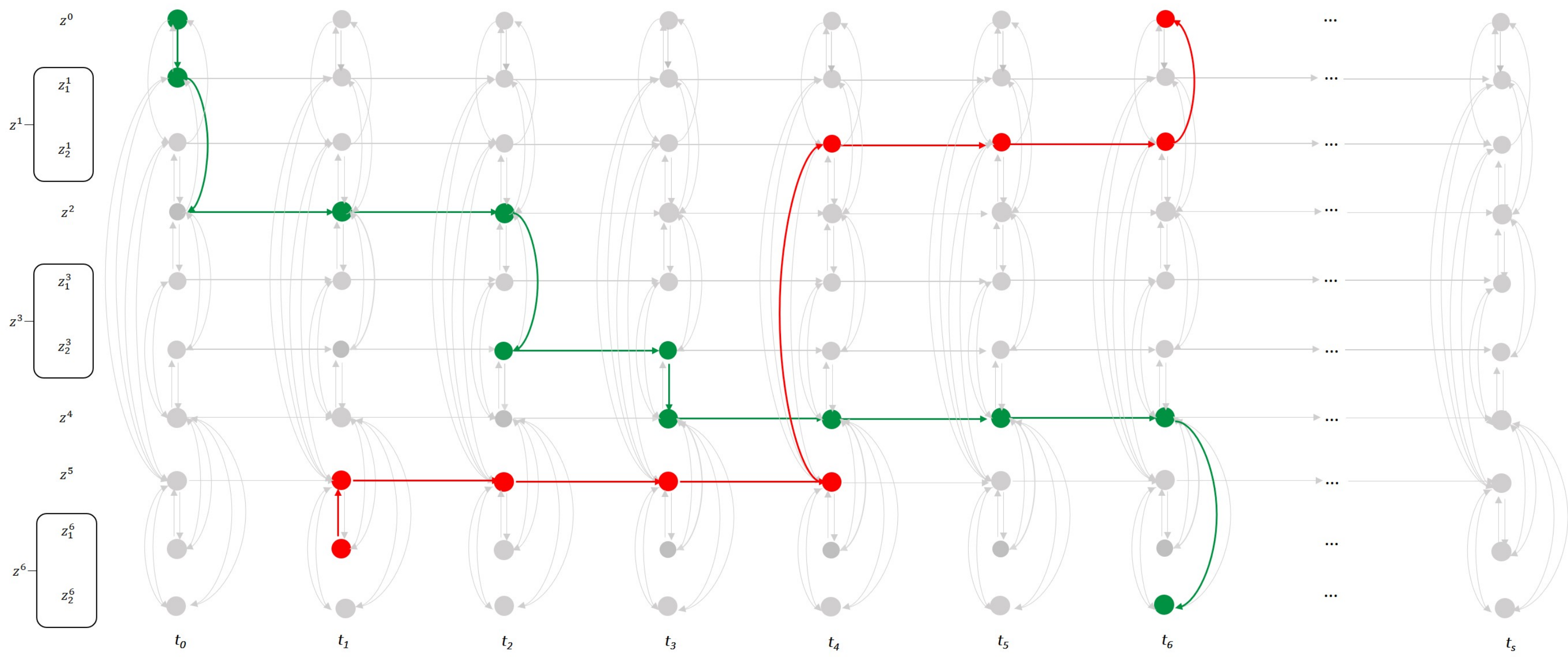

Figure 4.3: Example of paths on the operation-time-space network

Let's suppose that the import train (red path) must leave the terminal z_1^6 within the time window $[t_0; t_1]$, and its departure time from the rail station is t_6. The export train (green path) has the arrival time at the railway station t_0 and has to be at destination, i.e., inside the terminal z_2^6 within the time window $[t_6; t_8]$.

Looking at Fig.4.3, the export train (green) arrives from the rail network (z^o) in t_0 at the first track of the station (z_1^1) and it immediatly starts the shunting operation through primary zone (z^2). The train performs this operation for two time intervals (i.e., [t_0;t_1) and [t_1;t_2)). In time instant t_2 it finishes its primary operation and starts waiting operation in the second track of the shunting park (z_2^3). The train waits for one time interval and then, in time instant t_3 starts the secondary operation in z^4. It performs this latter operation for three time interval until t_6. Finally, in t_6, once finished its secondary operation, the train arrives at its destination (terminal z_2^6).

As far as the import train is considered (red path), it is ready at origin, i.e. terminal z_1^6, in time instant t_1. In this time instant, it starts the shunting operation through unique zone (z^5). It performs this operation for three time interval until t_4; in t_4 it starts to wait in the second track of the train station (z_2^1). The train waits for two time intervals and then, in time instant t_6, departs on the railway network (z^o).

In the next section, we explain a simple but complete example of PRSSP and its resolution in order to clarify the real application or the approach.

4.2.3 Example of problem and solution

This subsection reports a simple example of PRSSP with the solution obtained solving the model MOD-0 explained in the previous sections. MOD-0 has been implemented in Python 3.7, and solved by commercial solver Gurobi 8.1.0, on a machine intel (R), i5, 7200U CPU, 2.5 GHz, 8.00 GB RAM.

Note that, in this example, and in all the following tests, 10-minutes time unit has been adopted. The reason concerns the fact that the minimum time between the analysed processing times for the operations that has a fixed duration is 20 minutes (2 time units of 10 minutes). Of course, using 20-minute time unit we would have had fewer variables obtaining probably benefit in computation times, but this would have influenced too much the waiting operations that has a free duration. In fact, by using 20-minutes time units, the waiting operations would have been set for 0 minutes or for 20 minutes and multiples, even if the stopover could have been between 1 and 19 minutes. Therefore, time units of 10-minutes seemed to be the best trade-off between effectiveness and time granularity.

The simple example is a problem with 10 trains that must be managed on monday (3 import trains and 7 export ones). The trains data are shown in Fig. 4.4, which presents for each train the identification number, the arrival/departure time, the information if it belongs to the export or import cycle, the terminal of origin/destination, and the time window in which it has to arrive at destination/depart from origin (TWmin - TWmax).

Each train has to perform two operations: one through the primary zone and the other through the secondary zone. Both the operations have the same duration for each train. The operation through the primary zone has a duration of 20 minutes and the operation through the secondary zone has a duration of 1 hour.

Note that, we used the specific assumption of the same duration for each train depending on the operation because this represents the most of practical situations but, of course, the model is valid also if the same operation requires different times on different trains (see the definition of $d_{i,j}$ in the Section 4.1).

Concerning the resources, there are 2 shunting teams, which means that at maximum two operations through the secondary zone and the primary one can be performed simultaneously. Then, for each time instant, only one operation

Train number	A/D time	E/I	Maritime Terminal	TW min	TW max
1	18:00:00	E	1	19:00:00	22:00:00
2	15:27:00	E	1	17:00:00	19:00:00
3	07:20:00	I	2	04:00:00	07:00:00
4	11:57:00	I	1	09:00:00	10:30:00
5	23:00:00	I	2	18:00:00	21:00:00
6	13:38:00	I	1	08:00:00	10:00:00
7	22:21:00	I	1	20:00:00	21:20:00
8	14:32:00	I	2	10:00:00	11:00:00
9	00:54:00	I	2	23:00:00	23:50:00
10	14:48:00	E	1	16:00:00	17:00:00

Figure 4.4: Example - data

through the secondary zone related to the same terminal can be performed.

In Figures 4.5 and 4.6 the solution related to the scheduling of these 10 trains is reported. Fig.4.5 includes the details on the starting and ending times of the shunting operations for each train, while Fig.4.6 is a graphical representation of the different shunting operations of the trains during the time horizon. These to figures show the same informations in two different ways: one is a table representation while the second is a graphical representation.

We are going to detail data and solution for two of the ten involved trains, the export train number 2 and the import train number 7, in order to clarify the example. Looking Fig.4.4, the export train number 2 arrives at 15:27:00 on a track of the train station and must be transferred to Terminal 1. It has to reach the terminal between 17:00:00 and 19:00:00 that is the time window in which the terminal is ready to receive the train. The import train 7 has to depart from the rail station at 22:21:00, leaving Terminal 1 between 20:00:00 and 21:20:00, the time window in which the terminal assures the readiness of the train itself. Note that, train number 2, being an export train, has to perform firstly the operation in primary zone and then the operation in secondary zone. On the contrary, train

Train number	A/D time	MONDAY	Shunting Operation: Primary Zone Start	Primary Zone End	Secondary Zone Start	Secondary Zone End
1	18:00:00	x	18:00:00	18:20:00	18:20:00	19:20:00
2	15:27:00	x	15:30:00	15:50:00	16:30:00	17:30:00
3	07:20:00	x	07:00:00	07:20:00	04:00:00	05:00:00
4	11:57:00	x	10:40:00	11:00:00	08:10:00	09:10:00
5	23:00:00	x	20:30:00	20:50:00	19:30:00	20:30:00
6	13:38:00	x	13:10:00	13:30:00	09:20:00	10:20:00
7	22:21:00	x	22:10:00	22:30:00	21:10:00	22:10:00
8	14:32:00	x	14:10:00	14:30:00	10:30:00	11:30:00
9	00:54:00	x	00:30:00	00:50:00	23:30:00	00:30:00
10	14:48:00	x	14:50:00	15:10:00	15:10:00	16:10:00

Figure 4.5: Example solution - schedule

number 7, which is an import train, has to perform the operation in secondary zone before the shunting one in primary zone.

Fig.4.5 shows the solution obtained by solving MOD-0 for PRSSP: the last four columns report the starting and the ending time of the shunting operations for each train. In particular, concerning the two trains we are keeping as example, the number 2 starts the operation through primary zone at 15:30:00 and ends it at 15:50:00, while it starts the operation through secondary zone at 16:30:00 and ends it at 17:30:00 (respecting the time window of availability of the terminal). In this case, when it arrives at the rail station (15:27:00) has to start almost immediately the first operation, without waiting on a track of the station. Then, the ending time of the first operation and the initial time of the second one are different. It means that this train has to wait in the shunting park between the two operations. In particular, it has to wait 40 minutes, i.e. the difference between the initial time of the second operation and the ending time of the previous.

Going ahead with the import train number 7, it starts the operation through secondary zone at 21:00:00 (respecting the time window of availability of the terminal) and ends it at 22:00:00. Then, it performs the shunting operation through primary zone from 22:00:00 to 22:20:00, respecting its departing time (22:21:00). Note that, train 7 has not to wait neither in the shunting park between the two operation nor in the rail station before departing.

Finally, in the graphical representation of Fig.4.6, the schedule of the operations that the ten trains listed in Fig.4.4 have to perform are shown using different colours. In detail, we used yellow to represent shunting operations in primary area, orange for operations in secondary area, green for waiting operations on tracks of the rail station and grey for waiting operations on tracks of the shunting park.

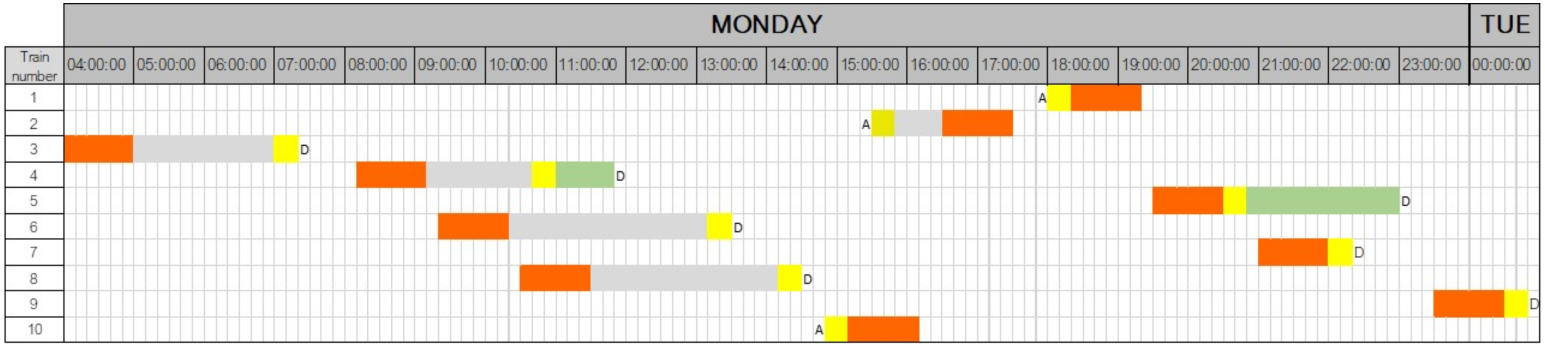

Figure 4.6: Example solution - graphical representation

Just for completing the description, letter "A" in Fig.4.6 means that a train arrives in that time instant, while letter "D" is used for the departure time.

We would like to highlight that the graphical representation of the solution is very helpful to visualize either the entire time horizon or part of it in order to quickly see the resources usage. Moreover, thanks to the representation, it is possible to understand both the eventual critical points and the available changes to do.

In fact, from Fig. 4.6, we can immediately note that the operation through the primary zone for train 4 has planned to be simultaneous to the operation through the secondary zone for train 8. The overlap slot for these two operations is between 10:40 and 11:00. Moreover, the operation through the primary zone for train 2 and the one through secondary zone for train 10 are simultaneous. The overlap slot is between 15:30 and 15:50. Anyway, that is allowed and doesn't create problems because in both cases the two operations can be performed simultaneously.

Then, no export trains have to wait in the station after their arrival, while two import trains (4 & 5) have to wait in the station before their departure. In particular, train 4 has to wait 50 minutes and train 5 has to wait 2 hours and 10 minutes.

Finally, we can see from the figure that there are at maximum two trains that have to wait in the shunting park simultaneously. The slots in which it happens are from 10:20 to 10:40 and between 11:30 and 13:10. The time slots in which the shunting park is empty are: 04:00-05:00, 07:00-09:00, 14:10-15:50, 16:30-23:59 for monday and 00:00-01:00 for tuesday.

4.3 Model improvements

This section addresses two of the possible improvements of the model MOD-0 presented in Section 4.2 for solving PRSSP. The first one (Subsection 4.3.1) can be used for decreasing the dimensions of the model, while the second (Subsection 4.3.2) for relaxing some capacity constraints.

4.3.1 How to decrease the model dimensions

The network flow model defined in Section 4.2 for solving PRSSP presents huge dimensions. This is mainly due to the high number of variables to generate, i.e. one variable for each train and each arc of the network. Remember that there is one type of arc for each time period considered. Obviously, these high dimensions cause an increase of the CPU for solving any instance of PRSSP.

Therefore, the arising question is:

Are all these variables essential for solving the problem?

This subsection is used to answer the question.

As explained in Section 4.1, each train to include in the schedule has some specific characteristics, those to use for reducing the variables number are reminded in the following.

For each export train, both the arrival time and the time window of availability for the maritime terminal of destination are known. Then, for each import train, both the departure time and the time window of availability for the maritime terminal of origin are known.

At this point, it is possible to assume that the shunting operations on each train have to be performed within a specific slot of the whole considered time horizon, called competence slot. For export trains, the competence slot starts when the train phisically arrives outside the port area (export train arrival time) and ends depending on the upper bound of the time window of availability of the terminal of destination. On the other side, for import trains, the competence slot starts depending on the lower bound of the time window of availability of the terminal of origin and ends when the train has to depart from the station outside the port area (import train departure time).

Therefore, instead of generating all the variables in the time horizon considered, is possible to generate only the variables inside the competence slots. In this way, the dimensions decrease.

In detail, it might be used:

$e_j \leq t < e_j^{max}$ to generate the related variables only between the time of arrival of train j and the maximum time in which it has to be at destination. In particular, $t \leq e_j^{max}$ only for variables representing the end of either the secondary or unique operations, bringing trains to their destination and $t < e_j^{max}$ for the variables representing the start of the operations, because there would be not enough time to perform the operations if they start at time period e_j^{max}.

$e_j^{min} \leq t \leq e_j^{max}$ to generate only the variables which bring the trains at destination with respect of their specific time window.

The same reasoning can be done for the import cycle. Therefore, $e_j^{min} \leq t < e_j$ to generate the variables between the minimum time in which the train j can be picked up from the terminal and its departure time and again $e_j^{min} \leq t \leq e_j^{max}$ for respecting the time window of availability.

For example, hust to clarify the concept, refering to the example of Section 4.2.3, the competence slot of the export train number 1 is between 18:00 (e_j) and 22:00 (e_j^{max}), while the competence slot of the import train number 3 is from 4:00 (e_j^{min}) to 7:20 (e_j).

Let's refer to Section 4.4.2.1 for a computational test on the possible reduction of the model dimensions.

4.3.2 Model improvements for relaxing some constraints

It might be possible that model MOD-0 results infeasible for an instance with certain data. This infeasibility means that one or more constraints cannot be respected for the instance under inspection. In this cases, would be useful to see which constraints are violated because in real cases it may happen that the feasibility can be reached with a little change in the instance data. In the real process, for the PRSSP, changing some instance data might means that the involved operators have to agree on the change: sometimes it is absolutely possible and convenience for the whole system, sometimes it is impossible.

The improvement of the model for avoiding infeasibility due to input data is useful also if some studies have to be done for testing the system capacity by using the model. These tests might be useful, for example, for evaluating investments in the infrastructure, for deciding the management of particular areas, for evaluating the maximum level of traffic that the system can manage, and so on.

The constraints that might be considered for this improvements are listed here below:

1. The constraints that limit the availability of the terminals within a time window (**Time Windows deviation**);

2. The constraints that limit the number of the same type of operations which can be performed simultaneously (**Maximum for zones**);

3. The constraints that limit the number of resources for each time period (**Maximum resources**);

4. The constraints imposing that all the trains provided as input have to be scheduled (**Exclusion of trains**);

In the following, each point is addressed.

1. Time Windows deviation

Constraints (4.5) and (4.6) are related to both the entrance and the exit of the trains in/from their terminals within the availability time windows. With the model improvement here reported, it is permitted to an export train j to enter the terminal either before e_j^{min} or later than e_j^{max}. When an import train is considered, it can leave the terminal before e_j^{min} or later than e_j^{max}. For doing that, we have to compute, and minimize in the objective function, the distance of the entry/exit time from its required time window for each train. Let be m_j the deviation of the train j with respect to its original time window.

When permitting this violations, the variables reduction described in Section 4.3.1 in no longer possible. Thus, we have to modify thw competence slot definition in such a way to generate only the variables which bring the trains at destination with respect of their specific time window enlarged of one day; let be (en_j^{min},en_j^{max}) the new time window of availability for a train j. Note that, constraints (4.5) and (4.6) are re-written summing up considering the modified time windows (the new constraints are (4.13) and (4.14)) and some new constraints are required for computing the deviation form the original time windows. Those latter are constraints (4.15) and (4.16).

$$\sum_{a \in (\cup_{en_j^{min} \leq t \leq en_j^{max}} \mathcal{A}_t^{-,\mathcal{N}_r^{z^6}})} tx_{a,j} - e_j^{max} \leq m_j \quad \forall j \in E, r = p_j \qquad (4.13)$$

$$e_j^{min} - \sum_{a \in (\cup_{en_j^{min} \le t \le en_j^{max}} \mathcal{A}_t^{-,\mathcal{N}_r^{z^6}})} tx_{a,j} \le m_j \quad \forall j \in E, r = p_j \tag{4.14}$$

$$\sum_{a \in (\cup_{en_j^{min} \le t \le en_j^{max}} \mathcal{A}_t^{\mathcal{N}_r^{z^6},+})} tx_{a,j} - e_j^{max} \le m_j \quad \forall j \in I, r = p_j \tag{4.15}$$

$$e_j^{min} - \sum_{a \in (\cup_{en_j^{min} \le t \le en_j^{max}} \mathcal{A}_t^{\mathcal{N}_r^{z^6},+})} tx_{a,j} \le m_j \quad \forall j \in I, r = p_j \tag{4.16}$$

2. Maximum for zones

Constraints (4.10) limit the number of simultaneous shunting operations in the port area depending on the number of available shunting teams for each time instant t. Thanks to the improvements here reported, it is permitted to use more than the number of available shunting teams. Note that, constraints (4.10) have to be re-written considering this possibility (see constraints (4.17)). We have to compute for each time instant the number of additional teams required; let be gk_t this deviation for time instant t. We have to minimise this difference for the whole time horizon T.

$$\sum_{j \in J} \sum_{a \in \cup_{i \in Z^-} (\mathcal{A}_t^{\mathcal{N}^i} \cup \mathcal{A}_t^{\mathcal{N}^i,+})} x_{a,j} \le k_t + gk_t \quad \forall t \in T \tag{4.17}$$

3. Maximum resources

Constraints (4.9) limit the number of the shunting operations that can be executed simultaneously each time instant t due to the limited available resources. Relaxing constraints (4.9), it is permitted to use more resources than the available.

Thus, constraints (4.9) have to be re-written considering this possibility (see constraints (4.18)). We have to compute for each time instant and for each operation the number of additional resources used; let be gu_t^i this deviation for time instant t and for operation i. We have to minimise this difference for the whole time horizon T.

$$\sum_{j \in J} \sum_{a \in \mathcal{A}_t^{\mathcal{N}^i} \cup \mathcal{A}_t^{\mathcal{N}^i,+}} x_{a,j} \leq u_t^i + gu_t^i \quad \forall t \in T, \forall i \in \ Z^- \tag{4.18}$$

4. Exclusion of trains

Thanks to constraints (4.3) and (4.4), every train (export/import) must enter/leave the network at time $t = e_j$. Including the present improvement, it is possible not to serve a train. For allowing that, we have modified constraints (4.3) and (4.4) adding a binary variable g_j for each train which will assume value 1 when a train is not served (see constraints (4.19) and (4.20)).

$$\sum_{a \in \mathcal{A}_{e_j}^{N^{z^0}, N^{z^1}}} x_{a,j} = 1 - g_j \quad \forall j \in E \tag{4.19}$$

$$\sum_{a \in \mathcal{A}_{e_j}^{N^{z^1}, N^{z^0}}} x_{a,j} = 1 - g_j \quad \forall j \in I \tag{4.20}$$

Note that, we also have to modify constraints (4.11) which, in their new form (4.21), fix the processing time of every shunting operation of each served train to the required time d_{ij}.

$$\sum_{a \in \mathcal{A}^{\mathcal{N}^i}} x_{a,j} = d_{ij}(1 - g_j) \quad \forall j \in J, \forall i \in Z_j \tag{4.21}$$

As already said, we have to minimise all the violations permitted and described in this section. The resulting objective function to minimize is the following:

$$MIN \sum_{j\in J}\sum_{i\in\{z^1,z^3\}} y_j^i + \alpha_1 \sum_{j\in J} m_j + \alpha_2 \sum_{t\in T} gk_t + \alpha_3 \sum_{t\in T}\sum_{i\in Z^-} gu_t^i + \alpha_4 \sum_{j\in J} g_j \quad (4.22)$$

where $\alpha_1, \alpha_2, ...\alpha_4$ are the penalties used for the deviations described above.

4.4 Computational tests on random generated instances

In the following, different experimental campaigns for testing the model and for evaluating the quality of the obtained solutions are described.

All the tests are based on a port system with a given layout, as described below, and the time horizon is fixed to 6 days, discretized into time intervals of ten minutes.

For these tests, we used the model (4.2)-(4.22), where (4.2)-(4.12) are the basic constraints, (4.13)-(4.21) are the new constraints for the model improvements introducted in Section 4.3 and (4.22) is the new objective function. Note that, for semplicity, we will refer to this model using MOD-1 in the following. MOD-1 has been implemented in Python 3.7, and solved by commercial solver Gurobi 8.1.0, on a machine intel (R), i5, 7200U CPU, 2.5 GHz, 8.00 GB RAM.

The port system under investigation has a rail station, a shunting park and 4 terminals. Two terminals are directly connected to the station, while the others are connected to the shunting park. It is assumed that the rail station and the shunting park have 10 tracks each.

The first experimental campaign is focused on instances with a particular number of trains to manage, and the aim is to evaluate the difficulty in solving these instances in accordance with both different arrival distribution of the trains during the time horizon, and different distribution of the trains with respect to the operations to perform and their origin/destination terminals.

The second experimental campaign is focused on testing the capability of the proposed model to solve bigger instances, in terms number of trains to manage.

4.4.1 Test 1

Instances used in this first experimental campaign have 50 trains, 50% are import trains and 50% export ones.

The random instances have been generated varying the arrival distribution for the trains along the considered time horizon. Four types of arrival distribution are analyzed; three different homogenous distributions and one compact distribution.

Homogeneous means that the number of trains is constant for specific time intervals within the time horizon. We considered the following time intervals: i) 2 days (i.e. 3 time intervals of 2 days in the 6 days time horizon, called **Homo2d**); ii) 1 day (i.e. 6 time intervals of 1 day in the 6 days time horizon, , called **Homo1d**); iii) a working shift (i.e. 3 shifts per day, thus 18 time intervals in the 6 days time horizon, called **Homo1shift**).

In the compact distribution (called **Compact**), we used two days time intervals and, in contrast with the homogeneous distribution, we imposed that the 50% of the total number of trains arrive within the first interval, the 25% in the second and the third. Note that, respecting the distributions here explained, the arrival and departure time for each train within the time intervals is randomly assigned.

Moreover, each train has a terminal as origin/destination and is characterized by the types of shunting operations that has to perform. We have decided to analyze different scenarios with respect to these elements characterizing the trains.

Thus, with respect to the shunting zones (i.e., the shunting operations) the distribution of trains can be balanced (B), unbalanced (U) or strongly unbalanced (S_U). A distribution is balanced (B) if the partition of trains between the operations ranges from 45-55 (unique operation)/55-45 (others operations); it is unbalanced (U) if the ranges are 25-35/75-65 and strongly unbalanced (S_U) with ranges as 15-20/85-80.

The trains distribution with respect to the terminals can be either balanced (B) or strongly unbalanced (S_U). Note that, the distribution of trains among the terminals regards the trains having the same types of operations to execute. For instance, the strongly unbalanced distribution among the terminals means that the distribution of the trains having to perform the unique operations is unbalanced between the two terminals that can be reached by the station (passing through the unique zone) and the distribution of the trains having to perform the primary and secondary operations is unbalanced between the two terminals that can be reached by the park. A distribution is balanced (B) if the split of trains between the terminals ranges from 40-60/60-40, while in case of strongly unbalanced (S_U) the range varies between 15-25/85-75.

Note that, the combination of these distributions inside the port area, i.e. three kind of distributions for operations (B, U, S_U) and two kind of distributions for terminals (B, S_U), provides six different sets of instances, as shown by Table 4.1.

The average number of trains in each day of the time horizon corresponding to the different arrival distributions are reported in Table 4.2, while Table 4.3 shows the distributions between both the operations and the terminals.

Then, Tables 4.4 and 4.5 summarize the characteristics of the generated instances: for each arrival distribution of trains the 6 above described set of instances are reported. Each row reports the average of 5 random generated instances.

Tables 4.6 and 4.7 reports the standard deviation in such a way to understand

Set name	Operations	Terminals
S_1	B	B
S_2	B	S_U
S_3	U	B
S_4	U	S_U
S_5	S_U	B
S_6	S_U	S_U

Table 4.1: Sets characteristics

	Mon	Tue	Wed	Thu	Fri	Sat
Homo2d	8.5	7.5	8.2	7.8	9.0	9.0
Homo1d	8.1	7.9	8.0	8.0	7.0	11.0
Homo1shift	12.4	11.7	6.4	5.6	7.4	6.4
Compact	10.5	8.4	8.1	7.8	8.3	6.8

Table 4.2: Trains distribution among the days of the planning horizon

the variability of the reported data.

In fact, the above mentioned Tables are formed by the same columns. The six columns of Tables 4.4 and 4.6 represent the days of the time horizon, from Monday to Saturday. Then, in Tables 4.5 and 4.7 two columns are dedicated to the split between the unique and the other operations and the last four represent the four terminals, one column for each one.

Instances of Tables 4.4 and 4.5 have been solved thanks to the proposed model MOD-1 by imposing a time limit of one hour, and the obtained results are shown in Tables 4.8, 4.9 and 4.10. Table 4.8 focuses on dimensions, CPU times, Objective values and Optimality GAP, while Tables 4.9 and 4.10 on the characteristics of the obtained solutions. All instances have been solved up to optimality, within the imposed time limit. The number of variables ranges between 306591 and 504285, while the number of constraints between 151358 and 229418. The CPU time is

	Unique	Others	Ter 1	Ter 2	Ter 3	Ter 4
S_1	48.00	52.00	50.00	50.00	46.15	53.85
S_2	48.00	52.00	20.83	79.17	18.85	81.15
S_3	30.20	69.80	47.01	52.99	48.42	51.58
S_4	29.40	70.60	21.88	78.12	20.96	79.04
S_5	15.40	84.60	41.69	58.31	49.18	50.82
S_6	15.00	85.00	24.08	75.92	23.05	76.95

Table 4.3: Trains distribution (in %) among operations & terminals

always smaller than half an hour. The objective function value varyies from around 370, for distribution Homo1shift, to about 3347, for distribution Homo2d.

Remember that, when solving model MOD-1 we do not consider the time windows, the teams and the maximum number of operations to execute in the primary zone as strong capacity constraints; in fact, we allow to propose a solution that does not respect one or more of these constraints. Obviously, we penalize both these over capacity requests and the violated time windows. After a tuning phase for choosing the parameters for the penalties, the following weights have been fixed: 10 for the time window deviation, 100 for the deviation from both the operations in the zone and the maximum number of teams and 500 for the trains exclusion.

Table 4.9 reports some data related to the usage of the port resources: the waiting time of the trains is generally spent for the most part in the track of the station. Anyway, the % of occupancy of the tracks in the rail station is slow, and range from 0.3 to 33.9%. The percentages of occupancy are lower for the tracks of the shunting park (from 0.1 to 17.9%). Then, as the reader can note from Table 4.10, only in few cases the obtained solutions present a deviation with respect to the time windows; anyway, the deviation is limited to few minutes. No other deviations have been obtained.

The arrival distributions have a great impact on the time spent by the trains

Distr	Set	Mon	Tue	Wed	Thu	Fri	Sat
Homo2d	S_1	7.80	8.40	7.20	8.60	11.60	6.40
	S_2	9.40	6.60	7.80	8.20	8.40	9.60
	S_3	8.20	7.80	8.20	7.80	8.20	9.80
	S_4	6.40	9.60	8.60	7.40	8.20	9.80
	S_5	10.20	5.80	8.40	7.60	9.00	9.00
	S_6	9.00	7.00	9.00	7.00	8.80	9.20
Homo1d	S_1	8.20	7.80	8.20	7.80	7.00	11.00
	S_2	8.00	8.00	8.00	8.00	7.00	11.00
	S_3	8.00	8.00	8.00	8.00	7.00	11.00
	S_4	8.00	8.00	8.00	8.20	6.80	11.00
	S_5	8.40	7.60	8.00	8.00	7.20	10.80
	S_6	8.00	8.00	8.00	8.00	7.00	11.00
Homo1shift	S_1	10.60	8.00	8.40	7.40	8.00	7.60
	S_2	10.20	8.60	8.20	8.40	8.00	6.60
	S_3	10.60	8.00	8.60	7.80	8.80	6.20
	S_4	10.20	9.00	7.40	8.40	8.60	6.40
	S_5	10.60	8.60	7.80	7.20	8.40	7.40
	S_6	10.60	8.40	8.20	7.80	8.20	6.80
Compact	S_1	14.00	10.20	5.80	6.40	7.20	6.40
	S_2	10.80	13.20	6.40	5.60	7.40	6.60
	S_3	13.20	10.80	6.60	5.60	8.20	5.60
	S_4	11.80	12.40	6.20	5.60	7.80	6.20
	S_5	12.00	12.00	6.40	5.80	6.60	7.20
	S_6	12.60	11.40	7.20	4.80	7.40	6.60

Table 4.4: (a) Trains distribution (average) in the different scenarios

in the port (see Tables 4.8, 4.9 and 4.10) and also on the model size and solution time. In case of instances of the set with Homo2day as distribution, we observed that scheduling all the activities is more complex: in fact, the total waiting time spent by the trains in the tracks of the port is the highest one, there is a little

Distr	Set	Unique	Others	Ter 1	Ter 2	Ter 3	Ter 4
Homo2d	S_1	24.00	26.00	12.00	12.00	12.00	14.00
	S_2	24.00	26.00	4.80	19.20	5.20	20.80
	S_3	14.80	35.20	6.80	8.00	17.20	18.00
	S_4	14.80	35.20	2.80	12.00	8.00	27.20
	S_5	8.40	41.60	3.20	5.20	20.80	20.80
	S_6	8.00	42.00	1.60	6.40	9.60	32.40
Homo1d	S_1	24.00	26.00	12.00	12.00	12.00	14.00
	S_2	24.00	26.00	4.80	19.20	4.40	21.60
	S_3	15.20	34.80	7.20	8.00	16.80	18.00
	S_4	14.00	36.00	4.00	10.00	7.60	28.40
	S_5	7.60	42.40	3.20	4.40	20.80	21.60
	S_6	6.80	43.20	1.60	5.20	10.40	32.80
Homo1shift	S_1	24.00	26.00	12.00	12.00	12.00	14.00
	S_2	24.00	26.00	5.20	18.80	4.80	21.20
	S_3	15.20	34.80	7.20	8.00	16.80	18.00
	S_4	14.80	35.20	3.20	11.60	6.40	28.80
	S_5	7.60	42.40	3.20	4.40	20.80	21.60
	S_6	7.20	42.80	2.00	5.20	9.60	33.20
Compact	S_1	24.00	26.00	12.00	12.00	12.00	14.00
	S_2	24.00	26.00	5.20	18.80	5.20	20.80
	S_3	15.20	34.80	7.20	8.00	16.80	18.00
	S_4	15.20	34.80	2.80	12.40	7.60	27.20
	S_5	7.20	42.80	3.20	4.00	20.80	22.00
	S_6	8.00	42.00	2.00	6.00	9.60	32.40

Table 4.5: (b) Trains distribution (average) in the different scenarios

violation of the time windows and also the corresponding CPU time is the highest one (1112 seconds on average). The opposite case is represented by the set with Homo1Shift as distribution, in which trains are homogenously distributed in each shift of each working day. The total waiting time is negligible. The optimal

Distr	Set	Mon	Tue	Wed	Thu	Fri	Sat
Homo2d	S_1	1.79	2.07	1.64	1.52	1.52	1.52
	S_2	1.14	1.14	0.84	0.84	1.67	1.67
	S_3	3.96	3.96	2.17	2.17	2.59	2.59
	S_4	1.14	1.14	2.70	2.70	2.17	2.17
	S_5	1.30	1.30	1.95	1.95	1.58	1.58
	S_6	2.24	2.24	0.71	0.71	2.17	2.17
Homo1d	S_1	0.45	0.45	0.45	0.45	0.00	0.00
	S_2	0.00	0.00	0.00	0.00	0.00	0.00
	S_3	0.00	0.00	0.00	0.00	0.00	0.00
	S_4	0.00	0.00	0.00	0.45	0.45	0.00
	S_5	0.55	0.55	0.00	0.00	0.45	0.45
	S_6	0.00	0.00	0.00	0.00	0.00	0.00
Homo1shift	S_1	0.55	1.00	0.55	0.55	1.00	0.55
	S_2	0.45	0.89	1.10	1.14	0.71	0.55
	S_3	0.55	0.71	1.52	0.84	1.10	0.45
	S_4	0.45	1.00	1.14	0.55	1.14	0.55
	S_5	0.55	1.34	0.84	0.45	1.14	0.89
	S_6	0.55	0.55	0.84	1.30	1.48	1.10
Compact	S_1	1.22	1.64	2.17	2.61	1.79	1.67
	S_2	1.10	1.10	0.89	0.89	1.14	1.14
	S_3	2.86	2.86	1.52	1.52	1.92	1.52
	S_4	2.95	3.05	1.30	1.34	1.48	1.48
	S_5	2.55	2.55	1.52	1.64	1.34	1.10
	S_6	3.13	3.13	2.77	2.77	1.14	1.14

Table 4.6: (a) Trains distribution in the different scenarios - standard deviation

solutions is obtained in 227 seconds on average.

The results of Tables 4.8, 4.9 and 4.10 are also reported in the following graphs in order to provide a graphical representation of the obtained results. We divided this analysis into two macro area: the impact of the arrival distributions and the

Distr	Set	Unique	Others	Ter 1	Ter 2	Ter 3	Ter 4
Homo2d	S_1	0.00	0.00	0.00	0.00	0.00	0.00
	S_2	0.00	0.00	1.10	1.10	1.10	1.10
	S_3	1.10	1.10	1.10	0.00	1.10	0.00
	S_4	1.10	1.10	1.10	2.00	2.00	2.28
	S_5	2.19	2.19	1.10	1.10	1.10	1.10
	S_6	2.00	2.00	0.89	2.19	1.67	2.61
Homo1d	S_1	0.00	0.00	0.00	0.00	0.00	0.00
	S_2	0.00	0.00	1.10	1.10	0.89	0.89
	S_3	1.10	1.10	1.10	0.00	1.10	0.00
	S_4	0.00	0.00	0.00	0.00	1.67	1.67
	S_5	1.67	1.67	1.10	0.89	1.10	0.89
	S_6	1.10	1.10	0.89	1.79	1.67	2.68
Homo1shift	S_1	0.00	0.00	0.00	0.00	0.00	0.00
	S_2	0.00	0.00	1.10	1.10	1.10	1.10
	S_3	1.10	1.10	1.10	0.00	1.10	0.00
	S_4	1.10	1.10	1.10	1.67	0.89	1.10
	S_5	1.67	1.67	1.10	0.89	1.10	0.89
	S_6	1.79	1.79	0.00	1.79	1.67	2.28
Compact	S_1	0.00	0.00	0.00	0.00	0.00	0.00
	S_2	0.00	0.00	1.10	1.10	1.10	1.10
	S_3	1.10	1.10	1.10	0.00	1.10	0.00
	S_4	1.10	1.10	1.10	1.67	2.19	2.28
	S_5	1.10	1.10	1.10	0.00	1.10	0.00
	S_6	1.41	1.41	0.00	1.41	0.89	0.89

Table 4.7: (b) Trains distribution in the different scenarios - standard deviation

impact of the distributions inside the port area.

The impact of the arrival distributions

Distr	Set	Vars	Constrs	CPU	OBJ	Gap %
Homo2d	S_1	390606	188024	789.92	2932.00	0.00
	S_2	387736	186646	846.15	3059.00	0.00
	S_3	446129	207423	1258.38	2861.00	0.00
	S_4	444271	206766	901.05	2954.20	0.00
	S_5	486868	222193	1412.93	3138.40	0.00
	S_6	504285	229418	1465.21	3347.80	0.00
Homo1d	S_1	354611	171955	804.08	1712.40	0.00
	S_2	359878	173882	789.77	1763.80	0.00
	S_3	403805	189302	844.68	1681.20	0.00
	S_4	405832	190112	795.91	1713.60	0.00
	S_5	447526	205152	1272.23	1738.80	0.00
	S_6	449839	205771	899.57	1709.20	0.00
Homo1shift	S_1	308125	151924	82.93	370.40	0.00
	S_2	306501	151358	139.15	378.00	0.00
	S_3	354071	168285	245.93	380.60	0.00
	S_4	356634	169203	188.39	377.60	0.00
	S_5	392839	182081	303.95	381.80	0.00
	S_6	395627	183082	402.69	383.20	0.00
Compact	S_1	358724	173781	873.89	2347.00	0.00
	S_2	361196	175154	867.58	2461.80	0.00
	S_3	410611	192784	760.10	2516.00	0.00
	S_4	414603	194173	1196.64	2463.00	0.00
	S_5	462528	211471	913.33	2535.80	0.00
	S_6	444398	204006	1263.60	2314.00	0.00

Table 4.8: Computational results (obtained by solving model MOD-1)

The impact of the arrival distributions on the solutions is shown by the following graphs.

All the graphs report the different arrival distributions in the horizontal axis. The graph in Fig.4.7 allows the analysis of the following features: number of

Distr	Set	Tot Wait	% Wait (station)	% Wait (park)	Avg % occ (station)	Avg % occ (park)
Homo2d	S_1	2585.40	76.03	23.97	33.27	10.91
	S_2	2710.60	72.15	27.85	33.19	12.86
	S_3	2516.60	63.93	36.07	27.38	15.20
	S_4	2612.40	65.68	34.32	29.20	15.43
	S_5	2797.80	63.44	36.56	30.49	17.88
	S_6	3006.00	65.48	34.52	33.89	17.73
Homo1d	S_1	1367.80	70.79	29.21	16.36	6.83
	S_2	1418.40	67.41	32.59	16.02	7.97
	S_3	1336.60	56.58	43.42	12.81	9.80
	S_4	1371.40	63.96	36.04	14.49	8.47
	S_5	1396.00	51.00	49.00	12.13	11.49
	S_6	1367.20	49.03	50.97	11.30	11.90
Homo1shift	S_1	26.80	70.90	29.10	0.32	0.12
	S_2	35.20	76.14	23.86	0.42	0.13
	S_3	38.00	86.32	13.68	0.54	0.08
	S_4	35.00	80.00	20.00	0.45	0.11
	S_5	40.80	85.78	14.22	0.56	0.10
	S_6	41.20	81.55	18.45	0.56	0.13
Compact	S_1	2002.00	69.91	30.09	23.89	10.20
	S_2	2116.80	74.57	25.43	26.15	8.86
	S_3	2170.80	69.52	30.48	25.66	11.48
	S_4	2119.20	61.63	38.37	22.17	13.81
	S_5	2193.00	65.18	34.82	23.90	12.77
	S_6	1969.80	62.60	37.40	21.06	12.43

Table 4.9: (a) Computational results – characteristics of the solutions obtained by solving model MOD-1

variables and constrains, CPU times and objective function values. As the reader can see, we obtained the highest values for Homo2d distribution and the lowest for Homo1shift distribution. The results for the other two distributions are in

Distr	Set	Avg/train (station)	Avg/train (park)	Teams dev	Zones dev	Excl trains	TW dev
Homo2d	S_1	393.16	123.92	0.00	0.00	0	2.00
	S_2	391.16	150.96	0.00	0.00	0	4.00
	S_3	321.76	181.56	0.00	0.00	0	2.00
	S_4	343.16	179.32	0.00	0.00	0	0.00
	S_5	355.00	204.56	0.00	0.00	0	0.00
	S_6	393.64	207.56	0.00	0.00	0	0.00
Homo1d	S_1	193.64	79.92	0.00	0.00	0	2.00
	S_2	191.24	92.44	0.00	0.00	0	2.00
	S_3	151.24	116.08	0.00	0.00	0	2.00
	S_4	175.44	98.84	0.00	0.00	0	0.00
	S_5	142.40	136.80	0.00	0.00	0	0.00
	S_6	134.08	139.36	0.00	0.00	0	0.00
Homo1shift	S_1	3.80	1.56	0.00	0.00	0	0.00
	S_2	5.36	1.68	0.00	0.00	0	0.00
	S_3	6.56	1.04	0.00	0.00	0	0.00
	S_4	5.60	1.40	0.00	0.00	0	0.00
	S_5	7.00	1.16	0.00	0.00	0	0.00
	S_6	6.72	1.52	0.00	0.00	0	0.00
Compact	S_1	279.92	120.48	0.00	0.00	0	0.00
	S_2	315.68	107.68	0.00	0.00	0	0.00
	S_3	301.84	132.32	0.00	0.00	0	2.00
	S_4	261.20	162.64	0.00	0.00	0	0.00
	S_5	285.88	152.72	0.00	0.00	0	0.00
	S_6	246.60	147.36	0.00	0.00	0	0.00

Table 4.10: (b) Computational results – characteristics of the solutions obtained by solving model MOD-1

the middle. The gap percentages between the highest and the lowest values of variables and constraints is around 20%, while for CPU and objective function is around 85%.

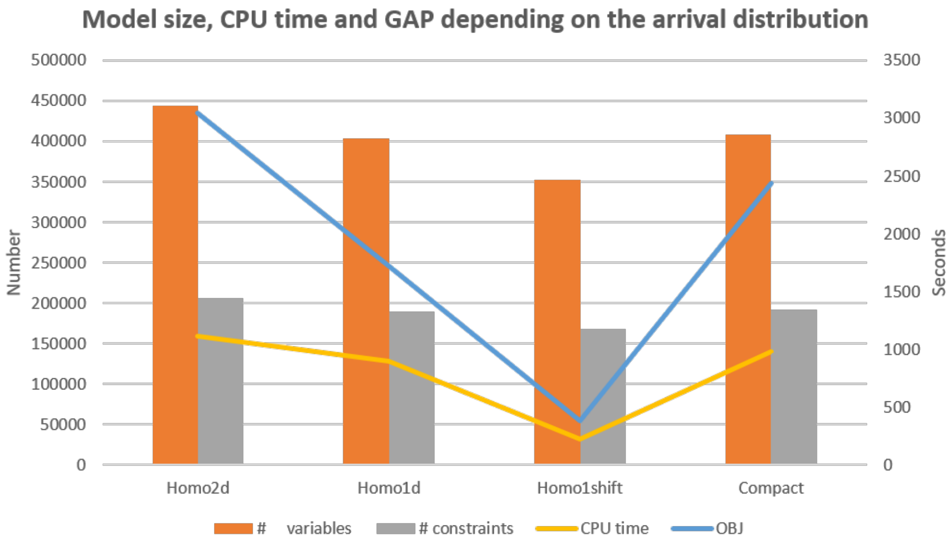

Figure 4.7: Variables, constrains, CPU time and objective function

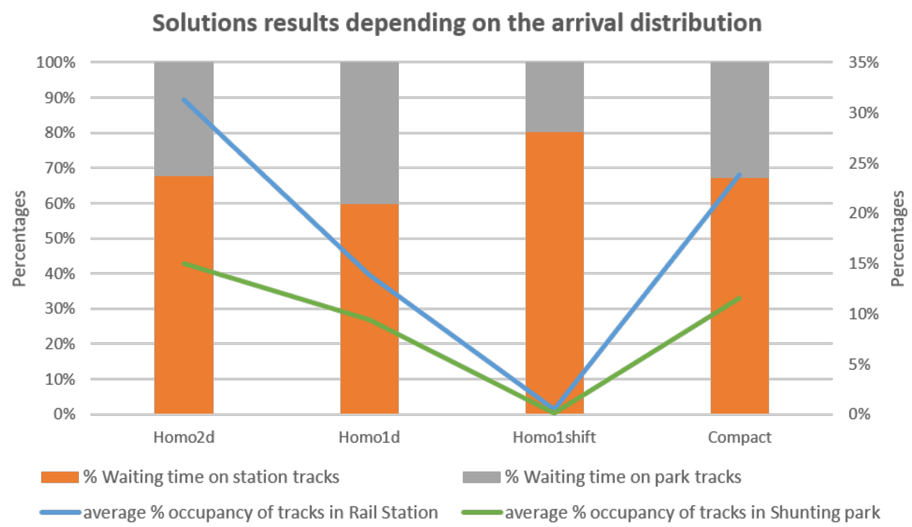

Figure 4.8: % of waiting times and tracks occupancy

The graph in Fig.4.8 reports the average percentages of both waiting times and tracks occupancy in the rail station and in the shunting park. Looking to the tracks occupancy, the worst case is again for distribution Homo2d, while the best

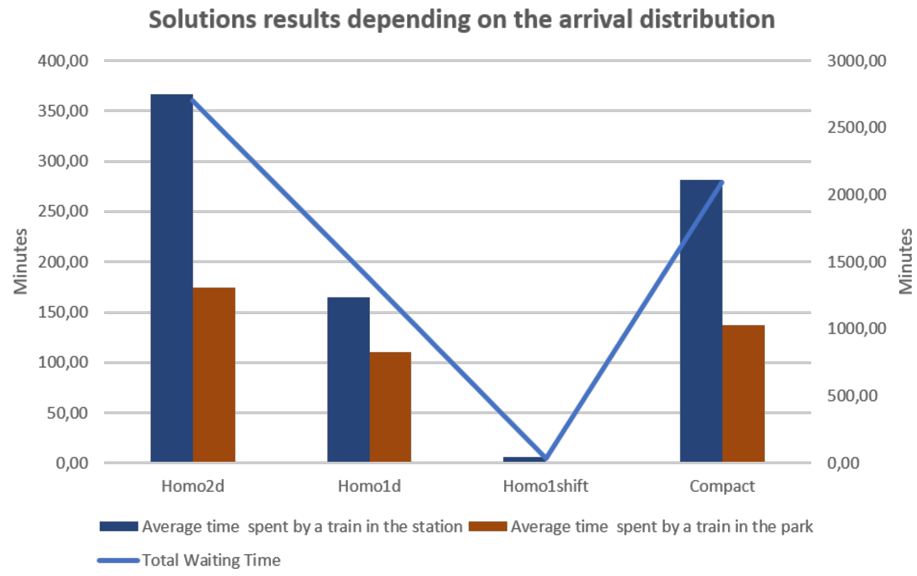

Figure 4.9: Waiting times analysis

results have been obtained for distribution Homo1shift.

If we focus on the total waiting time in the station and in the shunting park, we can note that the time passed in the rail station is higher than the time spent in the shunting park for all the distributions. The gap between the highest and lowest values is here 25% for the percentages of waiting time at the station, 51% for the percentages of waiting time at the shunting park and an average of 98% for the tracks occupancy.

Then, the total waiting time is shown in the graph of Fig.4.9, where, again, the worst case, i.e. when the waiting time is the highest, is for distribution Homo2d and the best one, i.e. when it is the lowest, for distribution Homo1shift. We obtained the same results also for the average time spent by a train in the rail station and in the shunting park (Fig.4.9). The gaps between the highest and the lowest values have here a mean of 98%.

Note that, the gaps have been computed by dividing the difference between the highest values and the lowest ones for the highest values.

Accordingly to these tests results and as expected, we can support that as more the trains are homogeneously distributed during the planning horizon as better they can be managed. In particular, they should be homogeneously distributed depending on the working shift of each day. Anyway, given that in reality this could be not always respected because trains paths have also commercial issues, also the other distributions can be managed using the model as support. In fact, we proved that the developed model have well managed the four different distributions for instances composed by 50 trains.

The impact of the distributions inside the port area

The impact of the distribution of the train inside the port area on the solutions is reported in the following graphs.

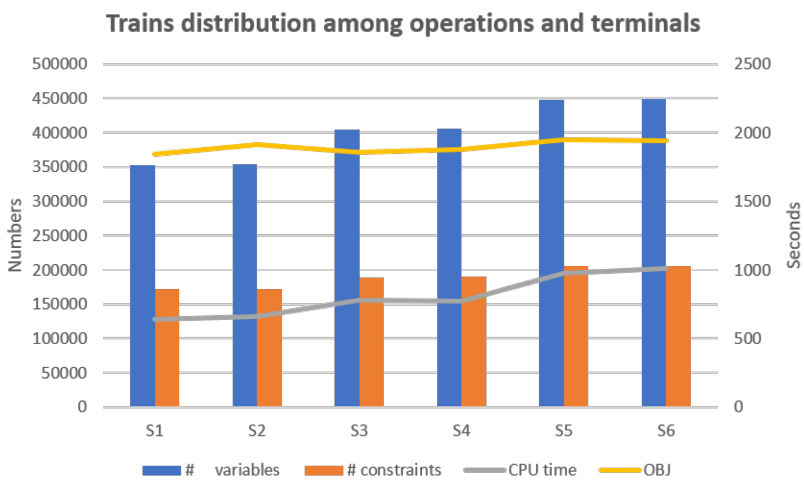

Figure 4.10: Variables, constrains, CPU time and objective function

The graph of Fig.4.10 shows the number of variables and constraints, the CPU

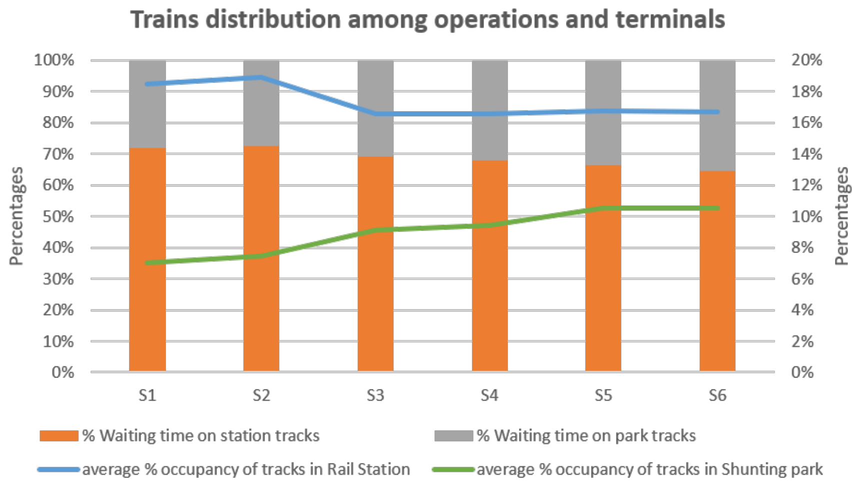

Figure 4.11: % of waiting times and tracks occupancy

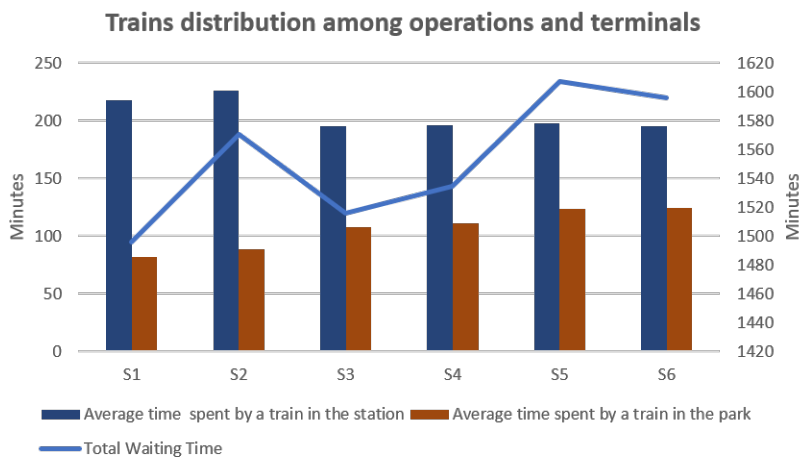

Figure 4.12: Waiting times analysis

times, and the objective function values for the different sets. As comments, while the values of the objective function are almost constant, the other values increase from S_1 (balanced) to S_6 (strongly unbalanced). The increase has an average gap of 25%. That is reasonable and represents the fact that as more the distributions of trains in a port rail system is unbalanced as more the trains management is difficult and the system is congestioned. Anyway, the developed model have been able to manage all the trains in these different cases.

Passing to the graph of Fig.4.11, we can say that, as before, also from the point of view of these six sets, the time passed in the railway station is higher than the time spent in the park. Then, from S_1 to S_6, the average percentages of tracks occupancy increase for the shunting park (33% between S_1 and S_6) and decrease for the rail station (12% between S_2 and S_3).

As last, in the graph of Fig.4.12, the reader can see that the highest total waiting time is for sets S_5 and S_6 (i.e. the strongly unbalanced sets), while the lowest is for the balanced set S_1. The gap between the highest and the lowest values is of 7%. If we focus on the average time spent by a train in the station, the trend is almost constant, while the trend of the same data in the shunting park increases from S_1 to S_6. The increasing gap is of 34%.

Note that, again, the gaps have been computed by dividing the difference between the highest values and the lowet ones for the highest values.

In general, even if the rail traffic in a port area is almost completely dependent to commercial logics, which might be difficult to direct, from these results, we can support that this kind of port rail system can be better managed, also in terms of waiting time for trains, if it is in some way balanced between operations and terminals.

4.4.2 Test 2

The second experimental campaign focuses on the evaluation of the capability of the proposed model MOD-1 to solve bigger instances, in terms number of trains to manage. The number of trains is increased from 50 to 60, 70, 80, 90 and 100.

For these tests we consider that the trains arrive with an homogenous distribution among each day of the time horizon (Homo1d). This distribution has been choosen because, between the considered distributions, it is realistic given also the related commercial issues. Then, inside the port area, we consider the following two scenarios. Trains may be: i) homogeneously distributed among both the shunting zones and the terminals (S_1); ii) unbalanced with respect to the shunting zones and strongly unbalanced with respect to the terminals (S_4). The average characteristics of these instances are reported in Table 4.11. Each row is the average of 30 instances (5 for each number of trains considered).

Set	Unique area % trains	Others % trains	Ter 1 % trains	Ter 2 % trains	Ter 3 % trains	Ter 4 % trains
S_1	49.24	50.76	23.94	25.30	24.42	26.34
S_4	31.29	68.71	7.64	23.65	15.75	52.96

Table 4.11: Characteristics of random generated instances of the two sets

The solution results are reported in Tables 4.12, 4.13 and 4.14. With a time limit of one hour we are able to solve up to optimality only instances with 50 trains. A small gap is obtained when solving instances with 60 trains, while growing to 70 trains the gap grows to more then 20%.

For what concerns the dimensions, the number of variables ranges between 354611 to 759987, while the number of constraints from 171955 to 357939. Moreover, the minimum CPU time is around 13 minutes and has been obtained for instances of 50 trains, while the maximum corresponds to the time limit of 1 hour. The objective function values vary from a minimum of around 1712 to a maximum of 12650.

The quality of the solutions can be analyzed by looking at Tables 4.13 and 4.14.

Table 4.13 includes the total waiting time, in terms of minutes, followed by the percentages of wait in the station rather than in the park. The train wait in

Set	trains ($\mid J \mid$)	# Vars	# Constrs	CPU time	OBJ	Opt Gap
S_1	50	354611	171955	804.08	1712.40	0.00
S_1	60	422632	202275	2114.09	2140.60	0.04
S_1	70	492659	233076	3600.57	3033.60	0.23
S_1	80	566292	265676	3601.07	4063.20	0.30
S_1	90	644110	299473	3600.56	6840.40	0.54
S_1	100	706089	326557	3600.81	11424.40	0.66
Average	75.00	531065	249835	2886.87	4869.10	0.30
S_4	50	405832	190112	795.91	1713.60	0.00
S_4	60	483612	223949	2278.98	2155.40	0.04
S_4	70	565643	259326	3601.15	3294.00	0.22
S_4	80	641940	292360	3600.96	4020.80	0.32
S_4	90	729953	329874	3600.37	6211.40	0.47
S_4	100	795987	357939	3600.27	12650.00	0.72
Average	75.00	603828	275593	2912.94	5007.53	0.29

Table 4.12: Obtained results

average the 62% of their waiting time in the station and the 38% in the shunting park. Moreover, there average percentages of tracks occupancy has reported. The average of the tracks occupancy is 20% for the station's tracks and 12% for the park's tracks. Another reported data is the average time spent both in the station and in the park for each train. For the station the average is around 161 minutes for each train and for the park it is 96 minutes for each one. Finally, the objective function terms deviation is reported for the teams, the zones, the exclusions of trains and the TW. We obtained little deviations for what concerns the number of needed shunting teams and no deviation for the zones. Then, we obtained an increasing number of excluded trains, from 0 to around 19, and an increasing TW deviation, from 0 to 344 minutes, with a mean of around 103 minutes.

Distr	Set	Tot Wait	% Wait (station)	% Wait (park)	Avg % occ (station)	Avg % occ (park)
S_1	50	1367.80	70.79	29.21	16.36	6.83
S_1	60	1599.40	74.87	25.13	19.98	6.59
S_1	70	1809.20	69.12	30.88	21.02	9.25
S_1	80	2148.60	66.35	33.65	24.42	12.08
S_1	90	2280.80	64.71	35.29	25.05	13.73
S_1	100	2355.20	63.13	36.87	25.36	14.89
Average	75.0	1926.83	68.16	31.84	22.03	10.56
S_4	50	1371.40	63.96	36.04	14.49	8.47
S_4	60	1603.60	60.16	39.84	15.97	11.03
S_4	70	1880.00	56.06	43.94	17.67	13.69
S_4	80	2102.80	53.57	46.43	18.86	16.27
S_4	90	2267.20	55.07	44.93	20.94	17.35
S_4	100	2107.60	50.68	49.32	18.24	17.23
Average	75.00	1888.77	56.59	43.41	17.70	14.01

Table 4.13: (a) Analysis of the quality of the solutions

We use the following graphs for understanding how the increased number of trains affects the difficulty in solving the problem and also the management of the scheduling of the trains.

From the graph in Fig.4.13, the reader can note that both the number of variables and constraints and the CPU times increase with the increases of the trains number. Passing from 50 to 70 trains the gap of the CPU time is of 78% and after the value is constant, representing the fact that for more than 70 trains the timelimit is always reached. The other values increases with an average percentage of 48% between the highest and the lowest.

From the graph in Fig.4.14, it is possible to note that the enormous growth

Distr	Set	Avg/train (station)	Avg/train (park)	Teams dev	Zones dev	Excl trains	TW dev
S_1	50	193.64	79.92	0.00	0.00	0.00	2.00
S_1	60	199.57	67.00	0.00	0.00	0.20	0.00
S_1	70	178.66	79.80	0.00	0.00	1.40	16.00
S_1	80	178.20	90.38	0.00	0.00	2.60	42.00
S_1	90	163.98	89.44	0.00	0.00	7.20	344.00
S_1	100	148.68	86.84	2.00	0.00	16.40	228.00
Average	75.0	177.12	82.23	0.33	0.00	4.63	105.33
S_4	50	175.44	98.84	0.00	0.00	0.00	0.00
S_4	60	160.80	106.47	0.00	0.00	0.20	0.00
S_4	70	150.57	118.00	0.00	0.00	1.60	98.00
S_4	80	140.80	122.05	0.00	0.00	2.40	132.00
S_4	90	138.73	113.18	0.00	0.00	6.20	210.00
S_4	100	106.82	103.94	6.00	0.00	19.40	174.00
Average	75.00	145.53	110.41	1.00	0.00	4.97	102.33

Table 4.14: (b) Analysis of the quality of the solutions

in the objective function value (86%) is due to the deviations that are weighted, while the total waiting time of served trains remain more or less the same. The time window deviations in minutes increases with a gap of 57% between 80 and 100 trains. It is possible to analyze the obtained solutions, from a management point of view. The layout structure and the available human resources seem to be not enough for serving more than 80 trains. In fact, when 90 trains are scheduled, on average more than 1 train is not served (see graph of Fig.4.15); the number of not served trains grows between 60 and 100 trains with a gap of 99%. In this latter case, note that there is also a major request in terms of human resources. The time windows deviation (same trend as in the previous graph) represents the

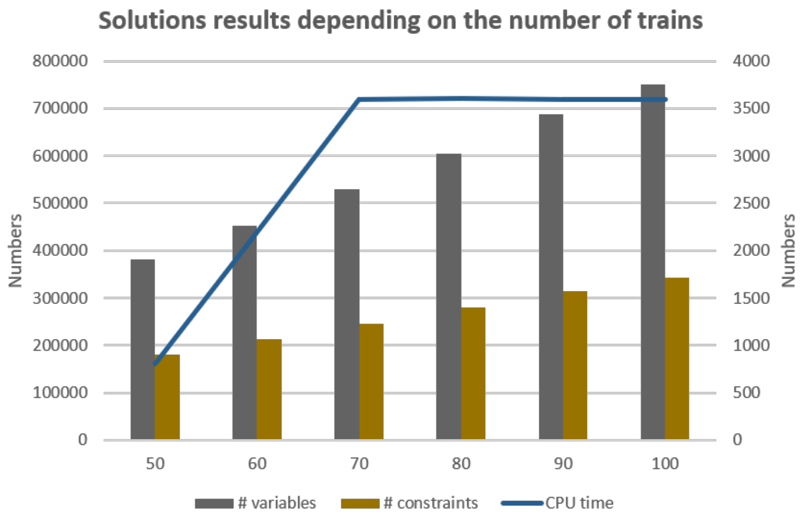

Figure 4.13: Dimensions and CPU times

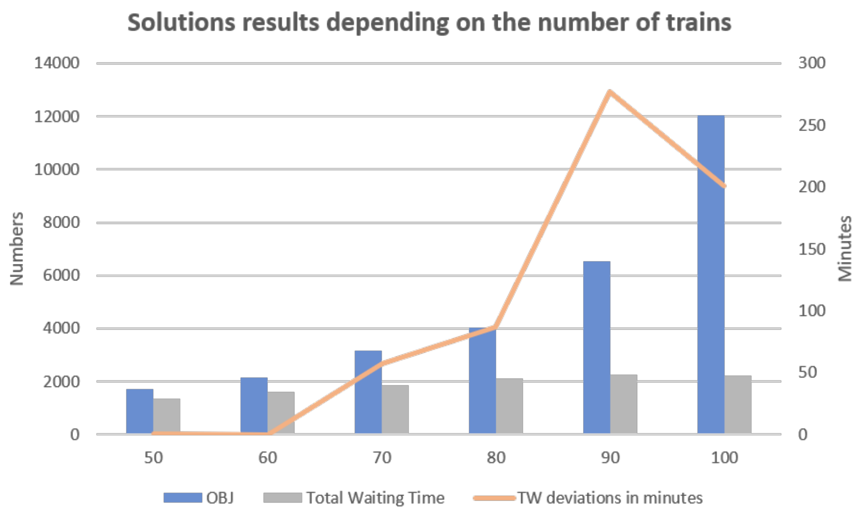

Figure 4.14: Objective function values, waiting times and TW deviations

less problem to highlight because they are random generated and it is possible that they have been not perfectly realistic generated. Therefore, an amount of TW deviations had been expected.

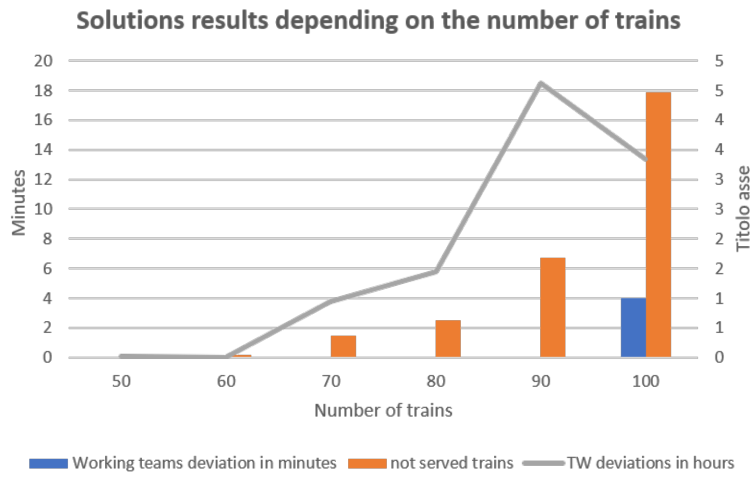

Figure 4.15: Working teams deviations, excluded trains and TW deviations (h)

However, thanks to the proposed model, this campaign has also revealed the capacity of the port to manage a higher number of trains, given the station and the shunting park capacity expressed in term of number of tracks (10 tracks each).

4.4.2.1 Test 2 - Focus: savings in the model dimensions

Given the model improvement explained in Subsection 4.3.1 for reducing the number of variables, we report in Table 4.15 an example of the dimensions in terms of variables and constraints number for both the two extremes cases, i.e. 50 and 100 trains, using three different criteria for defining the competence slots.

The used criteria are:

- Criteria 1: the competence slots for each train are the smallest; for export trains, they range from the arrival time to the TW max, while, for import trains, they range from the TW min to the departure time;

- Criteria 2: the competence slots are medium width; for export trains, they vary from the arrival time to the TW max plus a delta of 24 hours, while, for

	50 Trains		100 Trains	
	Vars	Constrs	Vars	Constrs
Criteria 1	144709	78813	298119	144501
Criteria 2	391876	184545	712234	328379
Criteria 3	745845	336631	1481539	654527

Table 4.15: Model dimensions using three different criteria

import trains, they vary from the TW min minus a delta of 24 hours until the departure time; the deltas are used to leave the possibility to deviate from the TW;

- Criteria 3: the competence slots are the biggest; for export trains, they range from the arrival time to the end of the planning horizon, while, for import trains, they are from the beginning of the planning horizon to the departure time.

Note that, for tests in Section 4.4, Criteria 2 has been used.

From the values of Table 4.15, we can say that, in the case of 50 trains, there is a variable saving of 81% and a constraints saving of 77% in criteria 1 with respect to criteria 3 and a variable saving of 47% and a constraints saving of 45% in criteria 2 with respect to criteria 3. Going ahead, in the case of 100 trains, we obtained a variables saving of 80% and a constraints saving of 70% using criteria 1 instead of criteria 3 and a variables saving of 52% and a constraints saving of 50% using criteria 2 rather than criteria 3.

In general, we obtained an average model dimensions reduction of 79% by using criteria 1 instead of criteria 3 and an average model dimensions reduction of 49% by using criteria 2.

The percentage of savings for a given criteria have been computed by diving the difference between the highest value and the value for the criteria by the highest value.

4.4.2.2 Test 2 - Focus: 80 trains

Note that, a deeper investigation on the model has been done considering the instances characterized by Homo1d-S1-80 trains in order to see the differents results while both imposing a timelimit and changing the objective function.

In particular, model MOD-1 has been solved with a time limit of 1 hour, 3 hours and also modifying the objective function in such a way to compare the obtained solutions in terms of CPU time and quality. The different cases are explained in the following.

- **TimeL 1 h**: the objective function is (4.22) and the TimeLimit has been fixed to 1 hour;
- **TimeL 3 h**: the objective function is (4.22) and the TimeLimit has been fixed to 3 hours;
- **Min prim**: minimizing only the waiting time in the station and in the park and the simultaneously operations through the primary zone. In this case, given the improvements of the model explained in Section 4.3, the number of the used shunting teams that exceed the number of the available and the time window deviation are not penalized;
- **Min prim & team**: minimizing only the waiting time in the station and in the park, the simultaneously operations through the primary zone and the number of the used shunting teams that exceed the number of the available. Again, given the model improvements explained in Section 4.3, time window deviation are not penalized;

- **Free**: minimizing only the waiting time in the station and in the park. Also here, given the model improvements, the simultaneously operations through the primary zone, the number of the used shunting teams that exceed the number of the available and the time window deviation are not penalized.

All the results have been reported in Tables 4.16, 4.17 and 4.18.

In particular, Table 4.16 includes the following information: test name, arrival distribution, set name, number of trains, CPU time, Objective function value, optimality GAP, total waiting time and percentages of waiting times in both the rail station and the shunting park. Then, in Table 4.17, the results in terms of average percentages of tracks occupancy (station and park) and average waiting time per train (station and park) are reported. Finally, Table 4.18 includes the deviations obtained in each test.

From Table 4.16 it is possible to note that the solution obtained by solving the model with a time limit of 1 hour has a gap of 30.5% but it is of good quality for the shuning manager. From a quality point of view, this solution is not so different from those obtained with a time limit of three hours. From Table 4.16 we can also note that the average percentage of waiting time at station is of 78% and at park is 22%.

Then, from Table 4.17, it is possible to see that both the tracks occupancy percentages and the average waiting time per train is very low for the last three cases (Min primary, Min primary & teams, Free). That is because in these last three cases we don't impose the respect of the time windows obtaining that trains are immediatly brought inside the terminals instead of left in the station or in the park waiting for the timw window of availability.

In fact, in Table 4.18, we can note that time window deviation for the last three cases is very high and is higher than the first two. Moreover, note that, in the last case, the free one, all the trains have been served with an high deviation for all the reported terms (Teams, Zones and Time windows).

The model has been solved with a time limit of one hour by modifying the objective function, that is without weighting the deviation of the Time windows, of the capacity of primary zone and of the number of teams. Some interesting results have been obtained by solving the model without the penalization of the time windows. In this case, all trains can be served and the unique deviation concernes time windows, even if these deviations are too high.

Note that, in case the manager is able to stipulate new agreements with the terminals, these solutions represent a good schedule planning.

Test	Distr	Set	$\mid J \mid$	CPU	OBJ	GAP %	Tot Wait	% Wait (S)	% Wait (P)
TimeL 1 h	Homo1d	S_1	80	3601.07	4063.20	0.30	2148.60	66.35	33.65
TimeL 3 h	Homo1d	S_1	80	10800.59	3300.40	0.13	2206.80	67.48	32.52
Min prim	Homo1d	S_1	80	539.97	30.00	0.00	30.00	95.33	4.67
Min prim & team	Homo1d	S_1	80	3259.97	677.40	0.04	87.00	74.48	25.52
Free	Homo1d	S_1	80	337.38	24.00	0.00	24.00	85.83	14.17

Table 4.16: (a) Analysis of the quality of the solutions

Test	Arrival distribution	Set	Avg % occ (Station)	Avg % occ (Park)	Avg/train (Station)	Avg/train (Park)
TimeL 1 h	Homo1d	S_1	24.42	12.08	178.20	90.38
TimeL 3 h	Homo1d	S_1	25.19	12.54	186.15	89.70
Min primary	Homo1d	S_1	0.50	0.02	3.58	0.18
Min primary & teams	Homo1d	S_1	1.08	0.37	8.10	2.78
Free	Homo1d	S_1	0.33	0.06	2.58	0.43

Table 4.17: (b) Analysis of the quality of the solutions

Test	Arrival distribution	Set	Teams dev	Zones dev	Excl trains	TW dev (min)	TW dev (h)	TW dev /train (min)	TW dev /train (h)
TimeL 1 h	Homo1d	S_1	0.00	0.00	2.60	42.00	0.70	0.53	0.01
TimeL 3 h	Homo1d	S_1	0.00	0.00	1.00	10.00	0.17	0.13	0.00
Min primary	Homo1d	S_1	528.00	0.00	0.00	20404.00	340.07	255.05	4.25
Min primary & teams	Homo1d	S_1	0.00	0.00	0.00	19778.00	329.63	247.23	4.12
Free	Homo1d	S_1	636.00	184.00	0.00	20596.00	343.27	257.45	4.29

Table 4.18: (c) Analysis of the quality of the solutions

Chapter 5

Port Rail Shunting Re-scheduling Problem (PRSRP)

Summary

This Chapter addresses the main aspects of Port Rail Shunting Re-Scheduling Problem (PRSRP) and the model developed to solve it, highlighting the main differences between PRSSP and PRSRP and between the flow models to solve them.

5.1 Problem description

Given the plan of the shunting operations built either automatically, by solving PRSSP, or manually, the shunting manager has the problem to re-schedule the activities when variations occur. Let's assume that these variations are known during the current day due to unpredictable events. The PRSRP aims at determining a new schedule for the shunting operations taking into account these variations.

Let's use the same assumptions used to describe the PRSSP in Chapter 4 for the general layout of a port shunting area. Thus, a port area that includes one railway station, one shunting zone and several maritime terminals. As assumption, the railway station has a given number of tracks connected with the shunting zone. The shunting zone is composed by one park with several tracks and some other tracks. These latter join the station to the shunting park, the shunting park to the terminals and the station directly to the terminals.

Let's remind to Fig.4.1 of Chapter 4 for both the considered layout and the description of both each specific zone, the discrete management of the time and the trains with their characteristics.

The differences, with respect to PRSSP, are explained in the following.

The main aspect concerns the fact that the schedule of the shunting operations is known. Therefore, new data have to be considered together with those for the PRSSP, which are explained in Section 4.1 of Chapter 4. The new data are $b_{i,j}$ and $f_{i,j}$, representing the starting and the ending times of operation i for job j, respectively. Moreover, there are other new input data that could have to be managed, those related to the eventual unpredictable events:

- e_j^* represents the new time instant of arrival/departure for the export/import train j when it delays;

- e_j^{*min} represents the new lower bound of the time window given by the terminal operator. If j is an export train, it means that the terminal changed its initial time of availability for receiving the train, while if j is an import train, it means that the terminal changed its initial time of availability for releasing the train;

- e_j^{*max} represents the new upper bound of the time window of availability given by the terminal. If j is an export train, it means that the terminal changed its final time of availability for receiving the train, while if j is an import train, it means that the terminal changed its final time of availability for releasing the train;

- $d^*_{i,j}$ represents the new duration needed by train j for executing operation i;
- $j^*, e_{j^*}, [e^{min}_{j^*}; e^{max}_{j^*}]$ represent the characteristics of the new extraordinary train $j*$ to schedule;
- if a train is suppressed, it is excluded from set J of the trains to schedule.

While in PRSSP the decisions concern the definition of the initial and final time instants for the shunting operations, now the scheduled initial and final time are known and new times have to be decided considering the new input data related to unpredicted events. We use the following example to clarify the problem.

Imagine to consider a certain time horizon (Fig.5.1), from t_{start} to t_{end}. Within this time horizon, the shunting and waiting operations to perform with their schedule are known.

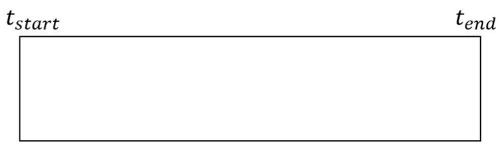

Figure 5.1: Time horizon considered

Then, imagine a time instant t^* belonging to the time horizon considered (Fig.5.2). It represents the time instant in which one or more unpredicted events become known.

Now, all the activities scheduled before t^* have been already performed. It is assumed that these latter have been performed exactly as they have been planned. Thanks to this assumption, it is possible to build a **system state** in t^*. The **system state** must represent the state of each operation in time t^*: some operations

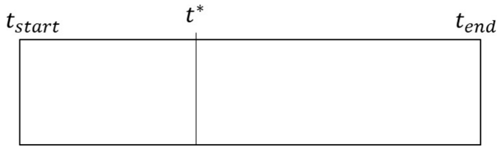

Figure 5.2: Time instant when unpredicted events become known

could be already finished, some could be in execution and some other could be still to start (Fig.5.3). These information are managed by modifying in the proper way the problem data. Subsection 5.1.1 provides the description on how to build the **system state** by modifying the problem data.

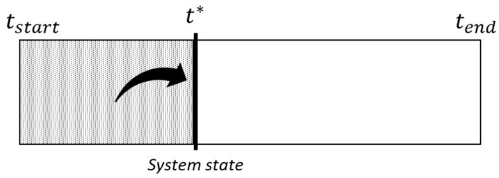

Figure 5.3: System state in t^*

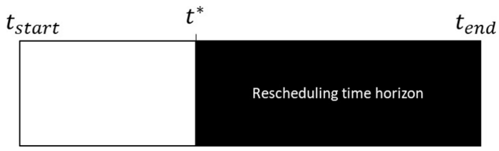

Figure 5.4: Re-scheduling time horizon

Given that the activities before t^* are considered finished, the re-scheduling time horizon become from t^* to t_{end} (Fig.5.4). The aim of the problem is to re-plan the schedule of the operations in the rescheduling time horizon minimizing the so called "disruption" between the predefined schedule (given by $b_{i,j}$ and $f_{i,j}$ data) and the new plan, i.e. trying to be as closed as possible to the predefined plan.

After the Subsection 5.1.1, which is dedicated to the system state building, Section 5.2 provides the explaination on how to update the model MOD-1 reported in Chapter 4 for solving PRSRP.

5.1.1 System state

This subsection includes the description on how to create the system state in t^*. Creating the system state means changing the operations schedule data in order to represent the state of each operation in time instant t^*. Note that, new data have to be compliant with respect to the used network flow structure. We use the following example for explaining the process, starting from a general and simple example to pass, then, to the specific PRSRP.

Let's consider a general operation with its related data: the blue rectangle in Fig.5.5, where $b_{i,j}$ and $f_{i,j}$ represent its initial and final time as in the schedule, while $d_{i,j}$ represents its duration.

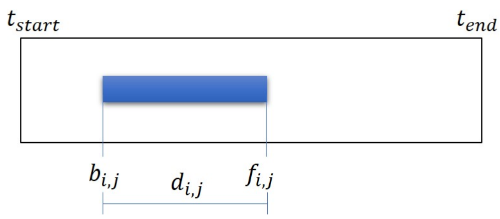

Figure 5.5: General operation with its data

Now, let's consider a time instant t^* within the time horizon; it might be in three positions with respect to the general operation under analysis:

1) Time instant t^* before $b_{i,j}$

$t*$ before $b_{i,j}$ (Fig.5.6) means that in t^* the operation hasn't start yet. The operation is completely inside the re-scheduling time horizon and no data have to

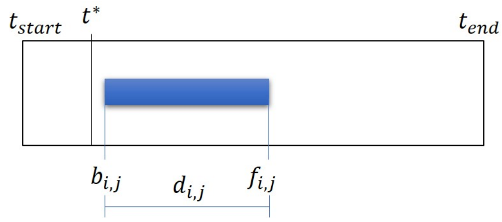

Figure 5.6: Time instant t^* before $b_{i,j}$

be changed. The operation will be taken into consideration in the re-scheduling problem as it is.

2) Time instant t^* between $b_{i,j}$ and $f_{i,j}$

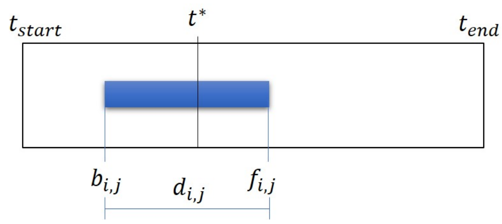

Figure 5.7: Time instant t^* is between $b_{i,j}$ and $f_{i,j}$

t^* between $b_{i,j}$ and $f_{i,j}$ (Fig.5.7) means that the operation is in execution. The part of the operation from $b_{i,j}$ to t^* has already done. The missing part is the one from t^* to $f_{i,j}$. In this case, we can forget the part of the operation already done changing two data: $b_{i,j}$ and $d_{i,j}$. Thus, in the re-scheduling problem, the initial time $b_{i,j}$ will be equal to t^* and $d_{i,j}$ will be equal to itself minus the time already passed from $b_{i,j}$ to t^*.

3) Time instant t^* after $f_{i,j}$

t^* after $f_{i,j}$ (Fig.5.8) means that the operation has already been done and, thus, we can forget about it.

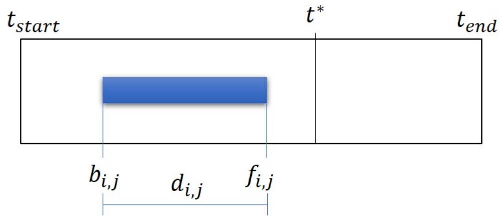

Figure 5.8: Time instant t^* is after $f_{i,j}$

Coming back to the PRSRP, the operations to perform are related to a single entity, which is the train, and might be both more than one and linked together, i.e. to be performed in sequence. Therefore, the question is *what happens if we have two operations in sequence?* Let's use again an example to explain that.

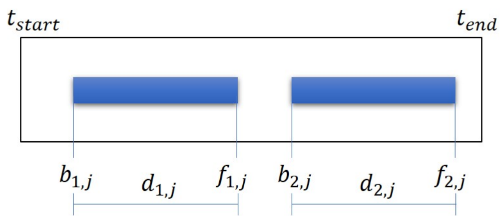

Figure 5.9: Two general operations with data

Imagine to have two general operations that have to be executed in sequence. The two operations are shown in Fig.5.9. Starting from the left, the first one is called here operation1 and the following operation2. In the relative data notation, there is 1 for operation1 and 2 for operation2.

Let's also represent operation1 and operation2 on the operation-time-space network proposed in the present work and explained in Chapter 3. As shown in Fig.5.10, let's imagine a general train that starts operation1 from the node origin o and takes two time interval to perform it. Then, there are two time intervals in which the train waits before starting operation2. Finally, the train starts operation2, it takes two time intervals to perform the operation and then it finishes its path in node d of destination.

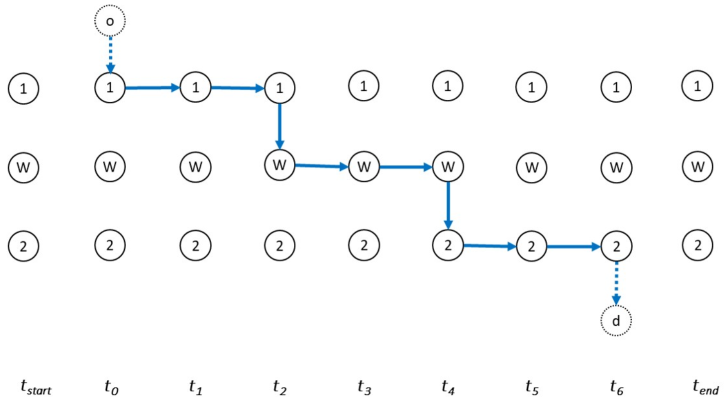

Figure 5.10: Network representation

The situation is clear when t^* is before $b_{1,j}$ or after $f_{2,j}$. In fact, if t^* is before $b_{1,j}$, both the two operations haven't start yet, while, if t^* is after $f_{2,j}$, both the operations have been already finished. Then, the situation is more complex when t^* is between $b_{1,j}$ and $f_{2,j}$. We explain the case when t^* is exactly between the two operations, i.e. after $f_{1,j}$ and before $b_{2,j}$ (Fig.5.11). Note that, when t^* is during the execution of either operation1 or operation2 we can refer to the above explained case 2 (**time instant t^* between $b_{i,j}$ and $f_{i,j}$**).

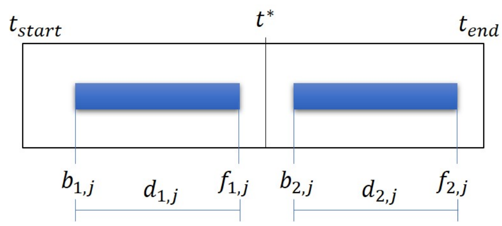

Figure 5.11: Time instant t^* is between $f_{1,j}$ and $b_{2,j}$

Let's see this situation on the operation-time-space network in order to understand how data have to change (Fig.5.12).

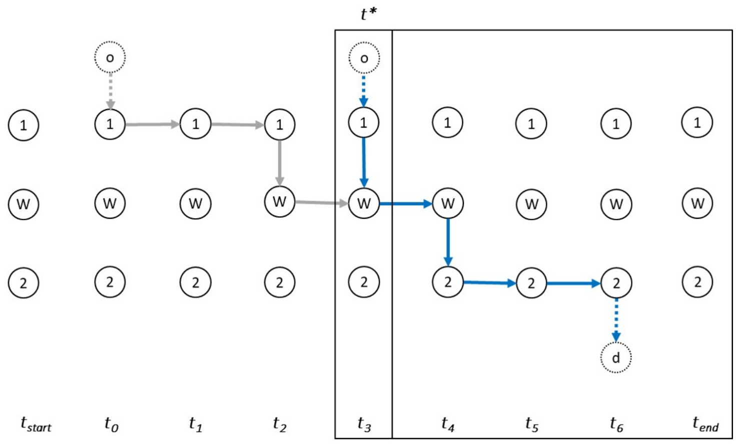

Figure 5.12: Network representation

Fig.5.12 shows the re-scheduling time horizon from t^* to t_{end}, the time instant t^* and the changes on data. Given that operation1 is completed before t^* but has to be considered because it is linked to operation2, we have to adjust its data in the following way. The starting and ending time of operation1, i.e. $b_{1,j}$ and $f_{1,j}$, have both to be equal to t^* and its processing time $d_{1,j}$ have to be equal to 0. In this way, the flow is allowed to go from origin to destination taking into account that operation1 is already done. No data related to operation2 has to be changed, given that it is already to perform.

In the case of PRSRP, depending on where t^* drops, several cases may happen. These cases depend also on the operations that the trains have to perform. In fact, different operations have to be performed if the train is export or import and if the train has to pass through the shunting park or not.

Let's consider the following different cases:

- **Case 1:** export train that have to pass through the shunting park (z^3)
- **Case 2:** import train that have to pass through the shunting park (z^3)
- **Case 3:** import train that have to pass through the unique zone (z^5)

- **Case 4:** export train that have to pass through the unique zone (z^5)

In the following, we report the passages for changing data in PRSRP in order to create the system state in time instant t^* in the listed four cases.

- **Case 1:** export train that have to pass through the shunting park (z^3)

Fig.5.13 shows the process for deciding which operations data have to be changed in **Case 1**. The process starts from the end time of the last operation and goes back until the first event in order to understand where is t^* with respect to the operations of train j.

If $f_{i \in z^4, j}$ is before t^*, we are in **Case 1.a** where the train has already been served because it already finished its last operation. If the train hasn't been already served, we have to check the starting time of the operation through z^4,b$_{i \in z^4, j}$. If it is before t^* we are in **Case 1.b** where the train is performing the operation through z^4. Otherwise, we have to check the final time of the operation through z^2, $f_{i \in z^2, j}$. If it is before t^*, it means that the train is waiting on the shunting park (z^3) to start the operation through z^4 (**Case 1.c**). Else, we have to check also the initial time of the operation through z^2, $b_{i \in z^2, j}$. If it is before t^*, we are in **Case 1.d**, where the train is performing the operation through z^2. Finally, if both the initial and final time of the operations of train j are before t^*, we have to check the arrival time e_j. If it is before t^*, we are in **Case 1.e**, where the train arrived and is waiting in the station (z^1) to start the operation through z^2. Otherwise, the train hasn't arrived yet.

For each undercase we report the data changes in the following:

Case 1.a

j excluded

The train that has be completely served have to be excluded from the rescheduling problem.

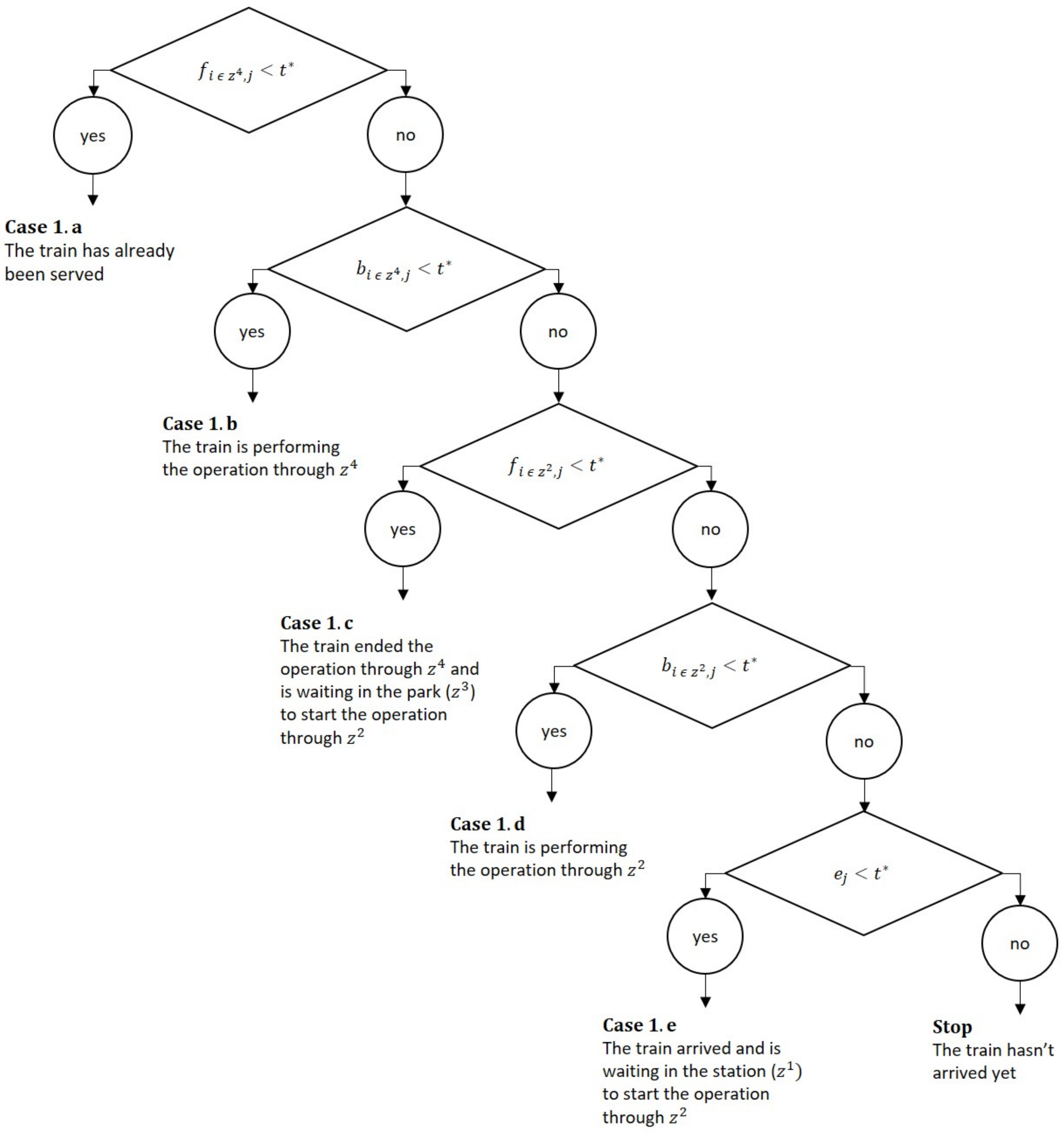

Figure 5.13: Diagram of **Case 1**

Case 1.b

$$
\begin{aligned}
&e_j = t^* \\
&e_j^{min} = t^* \quad if\ e_j^{min} < t^* \\
&d_{i \in z^2,j} = 0 \\
&d_{i \in z^4,j} = d_{i \in z^4,j} - (t^* - b_{i \in z^4,j}) \\
&b_{i \in z^2,j} = t^* \\
&f_{i \in z^2,j} = t^* \\
&b_{i \in z^4,j} = t^*
\end{aligned}
$$

The train has already arrived, performed the operation through z^2 and started the operation through z^4. Therefore, the arrival time, the initial and final time of operation through z^2, the starting time of the operation through z^4 have to be equal to t^*. Moreover, the minimum of the time windows of arrival in the terminal of destination has to be equal to t^*, if it is before. Finally, the processing time of the first operation has to be equal to 0 and the processing time of the operation in execution has to be equal to the remaining duration.

Case 1.c

$$
\begin{aligned}
&e_j = t^* \\
&e_j^{min} = t^* \quad if\ e_j^{min} < t^* \\
&d_{i \in z^2,j} = 0 \\
&b_{i \in z^2,j} = t^* \\
&f_{i \in z^2,j} = t^*
\end{aligned}
$$

The train has already arrived, performed the operation through z^2 and is waiting in the shunting park (z^3). Therefore, the arrival time and the initial and final time of operation through z^2 have to be equal to t^*. Moreover, the minimum of the time windows of arrival in the terminal of destination has to be equal to t^*, if it is before. Finally, the processing time of the first operation has to be equal to 0.

Case 1.d

$$
\begin{aligned}
&e_j = t^* \\
&e_j^{min} = t^* \quad if\ e_j^{min} < t^* \\
&d_{i \in z^2,j} = d_{i \in z^2,j} - (t^* - b_{i \in z^2,j}) \\
&b_{i \in z^2,j} = t^*
\end{aligned}
$$

The train has already arrived and started the operation through z^2. Therefore, the arrival time and the initial time of operation through z^2 have to be equal to t^*. Moreover, the minimum of the time window of arrival in the terminal of destination has to be equal to t^*, if it is before. Finally, the processing time of the operation in execution has to be equal to the remaining duration.

Case 1.e

$$\boxed{e_j = t^*}$$

The train has already arrived and is waiting in the station (z^1) to start the operation through z^2. Therefore, the arrival time have to be equal to t^*.

- **Case 2:** import train that have to pass through the shunting park (z^3)

Fig.5.14 shows the process for deciding which operations data have to be changed in **Case 2**. The process starts from the end time of the last operation and goes back until the first event in order to understand where is t^* with respect to operations of train j.

If the arrival time e_j is before t^* means that the train has already been completely served and it has also left the port area (**Case 2.a**). Otherwise, we have to check the final time of operation through z^2, $f_{i \in z^2, j}$. If it is before t^* we are in **Case 2.b** and it means that the train is waiting in the station (z^1) to depart because it already finished all the operations. Else, we have to check the initial time of the operation through z^2, $b_{i \in z^2, j}$. If it is before t^* (**Case 2.c**), the train is performing the operation through z^2. If not, we have to check if the final time of operation through z^4, $f_{i \in z^4, j}$, is before t^*. If it is, the train ended this operation and is waiting in the shunting park (z^3) to start the next (**Case 2.d**), while if it is not, we have to do the final check. The final check is on the starting time of the operation through z^4, $b_{i \in z^4, j}$. If it is before t^* (**Case 2.e**), the train is performing

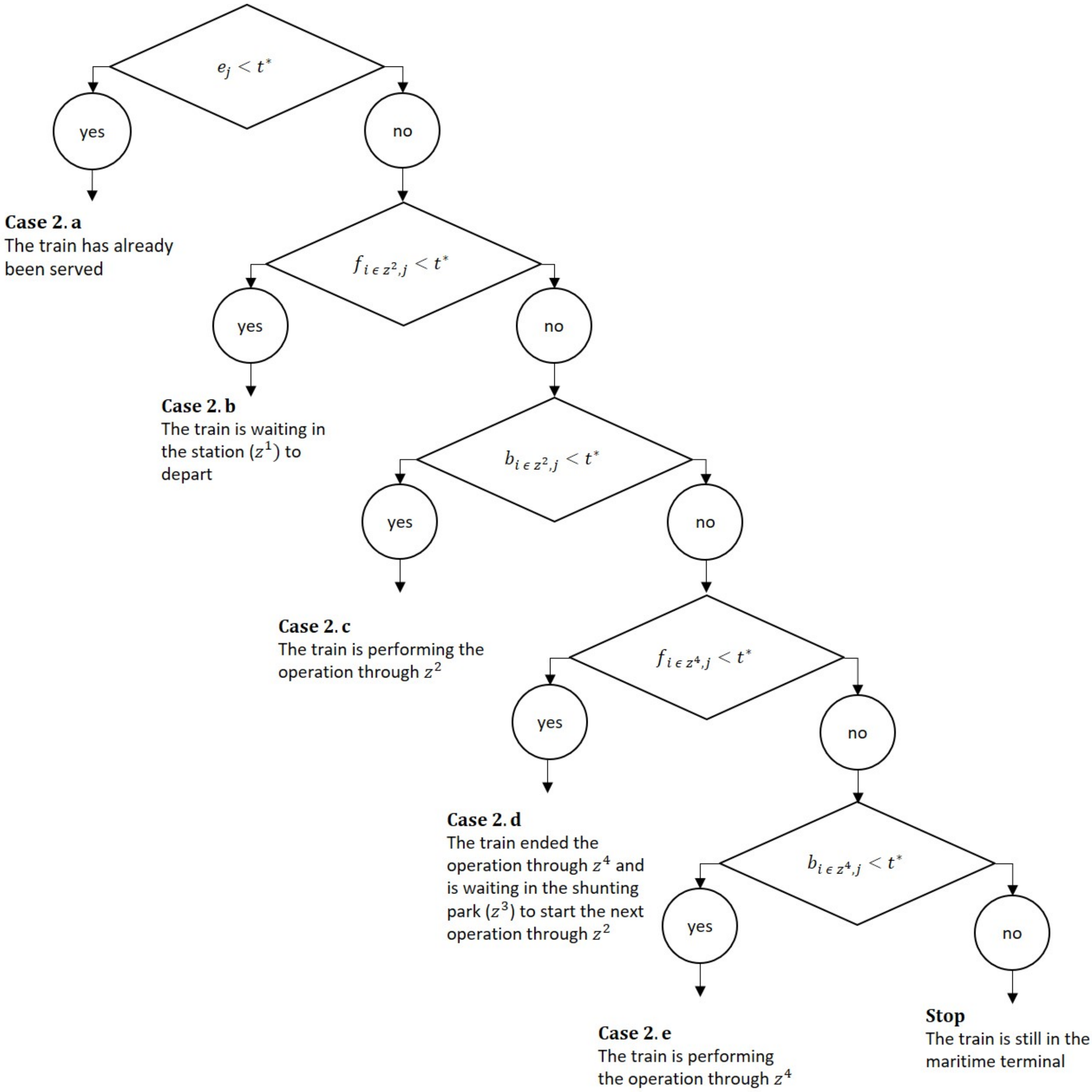

Figure 5.14: Diagram of **Case 2**

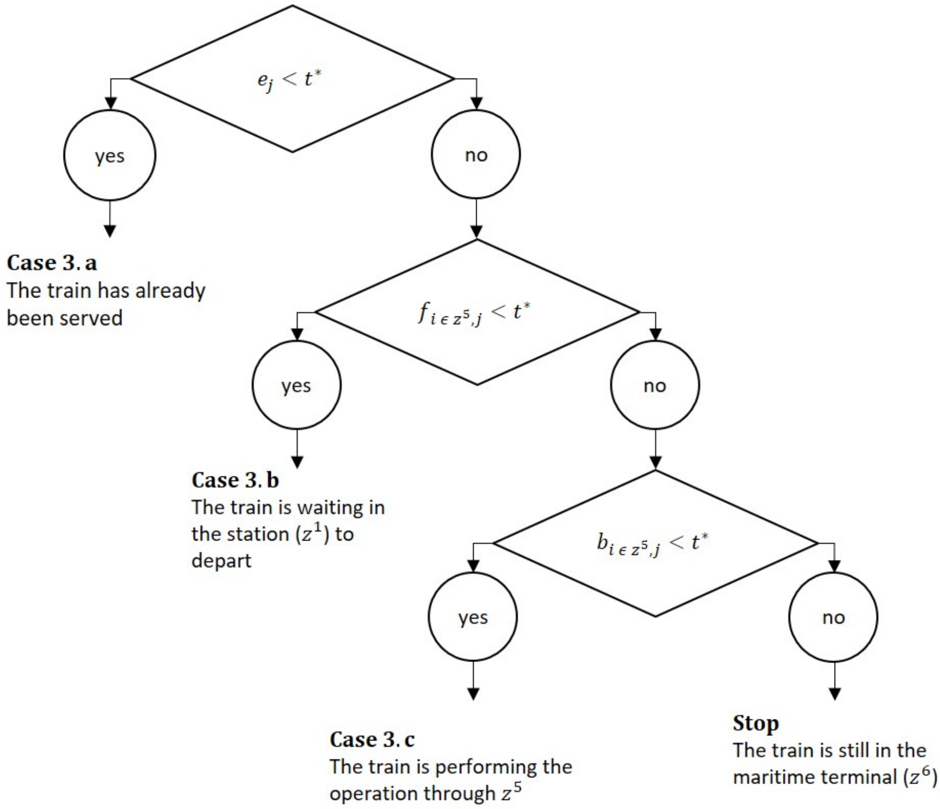

Figure 5.15: Diagram of **Case 3**

the operation through z^4, while, if not, the train is still in the maritime terminal of origin (z^6).

The data changes follow the same reasoning as in the undercases of **Case 1**.

- **Case 3:** import train that have to pass through the unique zone (z^5)

Fig.5.15 shows the process for deciding which operations data have to be changed in **Case 3**. The process starts from the end time of the last operation and goes back until the first event in order to understand where is t^* with respect to operations of train j.

If the arrival time e_j is before t^* means that the train has already been completely served and it has also left the port area (**Case 3.a**). Otherwise, we have to check the final time of the operation through z^5, $f_{i \in z^5, j}$. If it is before t^*, we are in **Case 3.b** and it means that the train is waiting in the station (z^1) to depart because it already finished the operation. Else, we have to check the initial time of the operation through z^5, $b_{i \in z^5, j}$. If it is before t^* (**Case 3.c**), the train is performing the operation through z^5. If not, the train is still in the maritime terminal of origin.

For each undercase we report the data change in the following:

Case 3.a

$$\boxed{j \; excluded}$$

The train that has been completely served has to be excluded from the rescheduling problem.

Case 3.b

$$\begin{aligned}
&e_j^{min} = t^* \quad if\ e_j^{min} < t^* \\
&e_j^{max} = t^* \quad if\ e_j^{max} < t^* \\
&d_{i \in z^5, j} = 0 \\
&b_{i \in z^5, j} = t^* \\
&f_{i \in z^5, j} = t^*
\end{aligned}$$

The train has already finished the operations and is waiting in the station (z^1) to left the port area. Given that the operation through z^5 has been already done, the initial and the final time have to be equal to t^*. Moreover, if the time windows of availability of the terminal is before t^* it has to be changed in order to be equal to t^* because it has been already respected. Finally, the processing time of the operations has to be equal to 0.

Case 3.c

$$e_j^{min} = t^* \quad if\ e_j^{min} < t^*$$
$$e_j^{max} = t^* \quad if\ e_j^{max} < t^*$$
$$d_{i \in z^5,j} = d_{i \in z^5,j} - (t^* - b_{i \in z^5,j})$$
$$b_{i \in z^5,j} = t^*$$

The train has already started the operation and it is executing it. Therefore, the initial time of the operation has to be equal to t^*. Moreover, if the time windows of availability of the terminal is before t^*, it has to be changed in order to be equal to t^* because it has been already respected. Finally, the processing time of the operations has to be equal to the remaining duration.

- **Case 4:** export train that have to pass through the unique zone (z^5)

Fig.5.16 shows the process for deciding which operations data have to be changed in **Case 4**. The process starts from the ending time of the last operation and goes back until the first event in order to understand where is t^* with respect to the operations of train j.

If $f_{i \in z^5,j}$ is before t^*, we are in **Case 4.a** where the train has already been served because it already finished its last operation. If the train hasn't been already served, we have to check the starting time of the operation through z^5, $b_{i \in z^5,j}$. If it is before t^*, we are in **Case 4.b** where the train is performing the operation through z^5. Otherwise, we have to check the final time of the operation through z^5, $f_{i \in z^5,j}$. If it is before t^*, it means that the train is waiting in the station (z^1) to start the operation throught z^5 (**Case 4.c**). Else, the train hasn't arrived yet.

The data changes follow the same reasoning as in the undercases of **Case 3**.

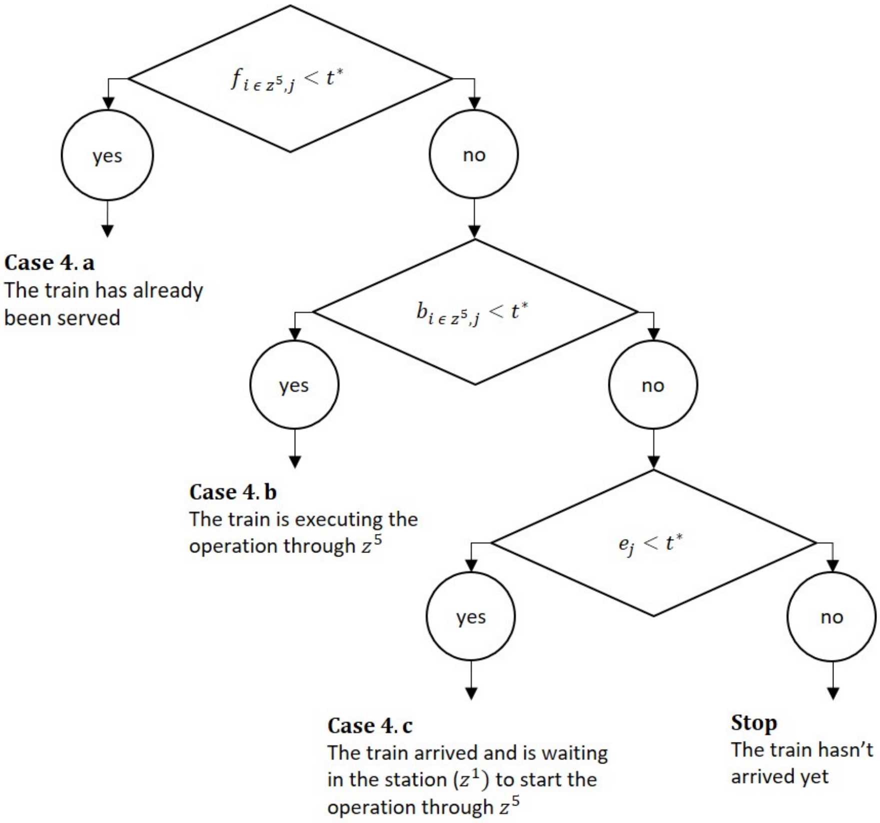

Figure 5.16: Diagram of **Case 4**

5.2 The flow model for solving PRSRP

The network flow model MOD-1 developed for solving PRSSP and explained in Chapter 4 can be adapted and used to solve also PRSRP. In fact, the framework is the same and only few elements have to be added. These elements, which concern data, variables, constrains and objective function, are explained in this section.

Data

In PRSRP, the following data are always known:

- $b_{i,j}$ that is the starting time of operation i of job j
- $f_{i,j}$ that is the ending time of operation i of job j

Variables

The new variables are the following:

- $v_{i,j} \geq 0$ defines the re-scheduled initial time of operation i of job j, $\forall j \in J, \forall i \in Z_j \cap Z^-$;
- $|\Delta_{i,j}| \geq 0$ defines the disruption, i.e. the difference between scheduled and re-scheduled initial time of operation i of job j, $\forall j \in J, \forall i \in Z_j \cap Z^-$;
- $\bar{v}_{i,j} \geq 0$ defines the re-scheduled final time of operation i on job j, $\forall j \in J, \forall i \in Z_j \cap Z^-$;
- $|\bar{\Delta}_{i,j}| \geq 0$ defines the disruption, i.e. the difference between scheduled and re-scheduled final time of operation i on job j, $\forall j \in J, \forall i \in Z_j \cap Z^-$;

Objective Function

In the objective function (4.22), we add the following terms (5.1) representing the difference between the scheduled and the re-scheduled times.

$$\sum_{i \in Z_j \cap Z^-} \sum_{j \in J} |\Delta_{i,j}| + \sum_{i \in Z_j \cap Z^-} \sum_{j \in J} |\bar{\Delta}_{i,j}| \tag{5.1}$$

The first term represents the total difference between the scheduled and the re-scheduled initial times of the operations that have to be perfomed on job j among the ones that might be executed, i.e. $Z_j \cap Z^-$. The second term represents the difference based on the ending times of the operations. Minimizing this "disruption", the new plan will be as closed to the predefined as possible.

Constraints

The following new constraints define the variables $v_{i,j}$ and $\bar{v}_{i,j}$.

$$v_{i,j} = \sum_{t \in T} \sum_{a \in \mathcal{A}_t^{-,\mathcal{N}^i}} t * x_{a,j} \ \ \forall j \in J, \forall i \in Z_j \cap Z^- \tag{5.2}$$

$$\bar{v}_{i,j} = \sum_{t \in T} \sum_{a \in \mathcal{A}_t^{\mathcal{N}^i,+}} t * x_{a,j} \ \ \forall j \in J, \forall i \in Z_j \cap Z^- \tag{5.3}$$

The variables $v_{i,j}$ are defined using the binary variables $x_{a,j}$ and the inbound arcs of nodes n^i, which represent the possible starting times of operation i. Note that, among the variables $x_{a,j}$, only one can assume value 1 and it will be the one related to the t representing the starting time of operation i.

In the same way, the variables $\bar{v}_{i,j}$ are defined using the binary variables $x_{a,j}$ and the outbound arcs of nodes n^i, which represent the possible ending time of the operation i.

Finally, new constrains have to be added in order to define $|\Delta_{i,j}|$, for the starting times, and $|\bar{\Delta}_{i,j}|$, for the ending times. These variables are defined as the difference between the new timing, given by $v_{i,j}$ and $\bar{v}_{i,j}$, and the planned ones, given as input data by $b_{i,j}$ and $f_{i,j}$.

$$-\Delta_{i,j} \leq (v_{i,j} - b_{i,j}) \leq \Delta_{i,j} \forall j \in J, i \in Z_j \cap Z^- \tag{5.4}$$

$$-\bar{\Delta}_{i,j} \leq (\bar{v}_{i,j} - f_{i,j}) \leq \bar{\Delta}_{i,j} \forall j \in J, i \in Z_j \cap Z^- \tag{5.5}$$

The next section includes the computational tests on the model developed for solving PRSRP, in which the objective function is (4.22) with the addition of (5.1), while the constraints are from (4.2) to (4.21) and from (5.2) to (5.5). In the following, we will refer to this model using MOD-2.

5.3 Computational tests

This section presents the computational tests compaign that has been performed by using model MOD-2.

The system layout is the same considered in the computational tests Section of Chapter 4 (Section 4.4). Let's see this Section for the characteristics of the system.

The re-scheduling time horizon includes the 24 hours of one day. The instance under investigation includes 8 trains and their characteristics. The train characteristics are reported in Table 5.1. In particular, from left side, the columns represent: the train identification number, the distinction between export (E) and import (I) cycle, the terminal of origin or destination (p_j), the arrival/departure time (e_j) and the time windows minimum (e_j^{min}) and maximum (e_j^{max}) of terminal availability.

Moreover, the instance includes also the operations schedule for each train,

Train ID	E/I	p_j	e_j	e_j^{min}	e_j^{max}
0	E	4	12:50	16:40	22:40
1	E	1	13:40	17:20	23:20
2	E	3	09:00	17:50	23:50
3	E	2	06:00	09:50	15:50
4	I	1	15:30	06:00	12:00
5	I	1	21:30	07:10	13:10
6	I	2	15:00	06:00	11:50
7	I	3	15:30	06:10	12:10

Table 5.1: Instance data

which is reported in Table 5.2.

	$i \in z^2$		$i \in z^4$		$i \in z^5$			
Train ID	$b_{i,j}$	$f_{i,j}$	$b_{i,j}$	$f_{i,j}$	$b_{i,j}$	$f_{i,j}$	Station track	Park track
0	14:00	14:20	16:20	17:00			bs5	bp6
1					16:20	17:20	bs10	-
2	11:10	11:30	17:30	18:10			bs8	bp4
3					08:50	09:50	bs1	-
4					12:00	13:00	bs2	-
5					13:10	14:10	bs4	-
6					11:00	12:00	bs3	-
7	13:10	13:30	12:10	12:50			bs1	bp1

Table 5.2: Instance schedule

In fact, it shows the initial ($b_{i,j}$) and the final ($f_{i,j}$) times for the three possible shunting operations: through the primary zone ($i \in z^2$), the secondary zone ($i \in z^4$) and the unique zone ($i \in z^5$). The last two columns of this table show the planned tracks for waiting operations on the rail station and in the shunting park, respectively. Figures 5.17 and 5.18 show the graphical representation of the operations schedule data reported in the previous tables. In detail, Fig.5.17 represents the operations during the day, train by train. As explained in the legend, we used the yellow colour for the operations through the primary zone z^2, the green colour for the operations through the secondary zone z^4 and the orange colour for operations through the unique zone z^5. Then, Fig.5.18 shows the tracks and zones occupancy.

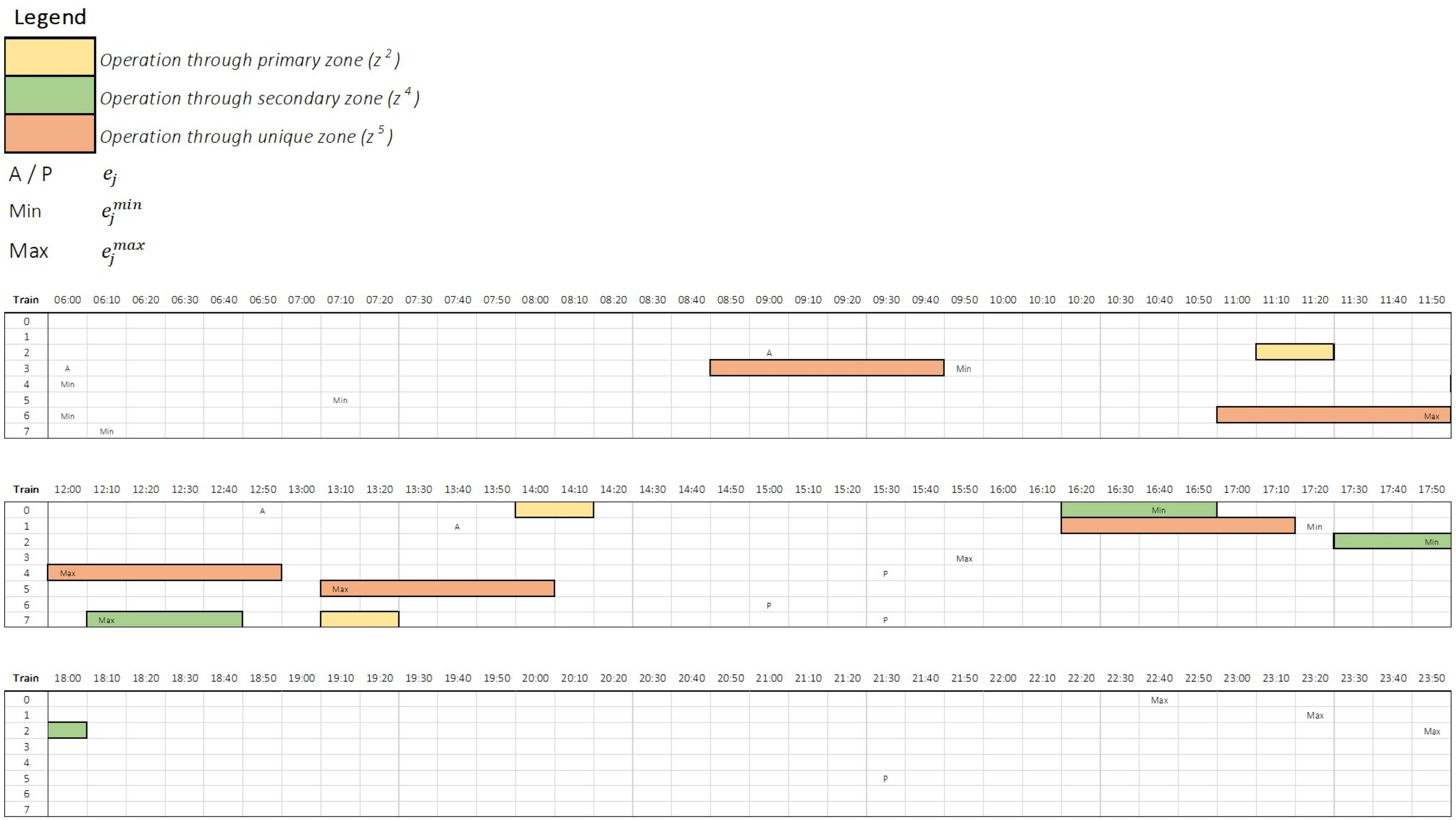

Figure 5.17: Instance schedule representation train by train

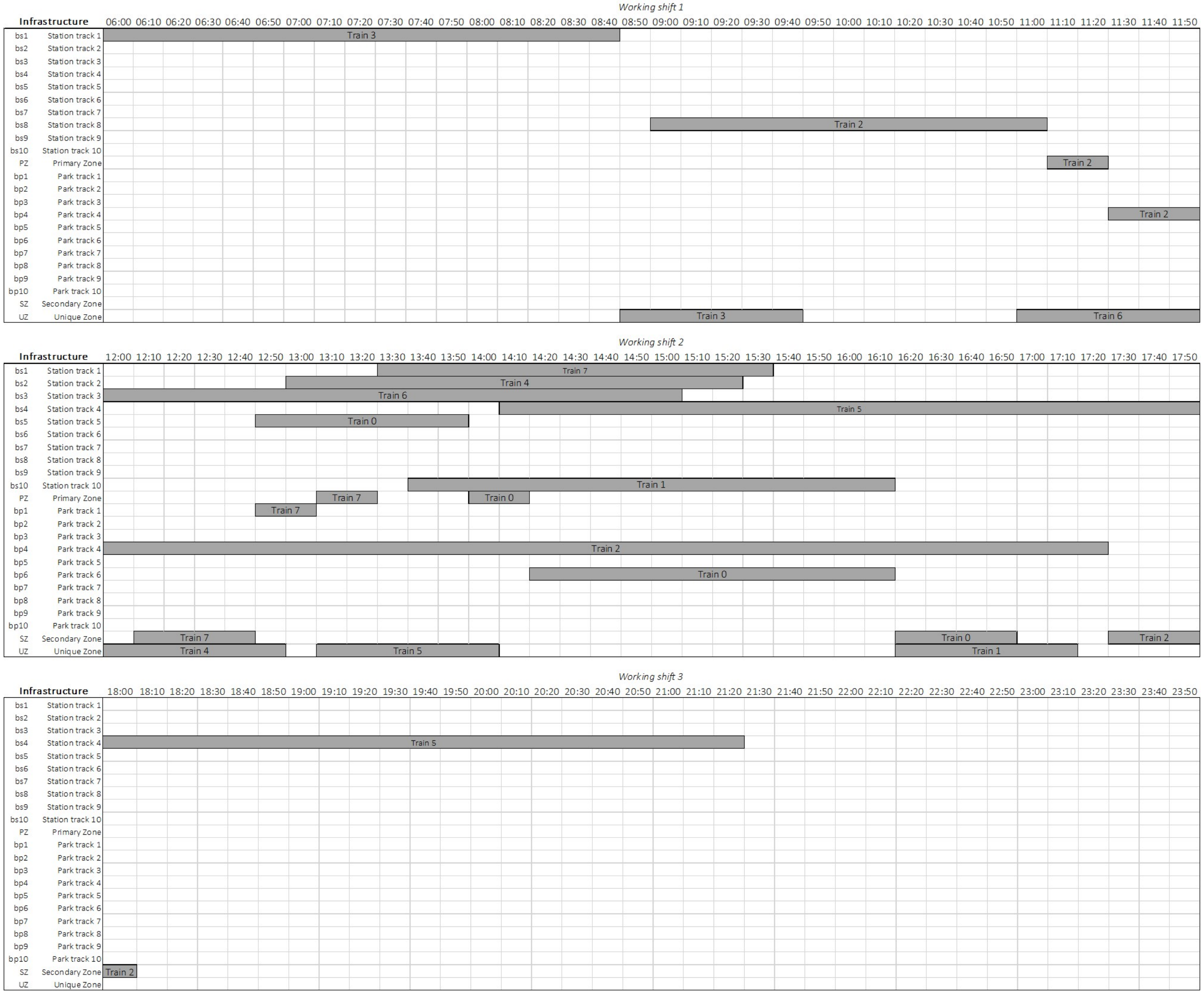

Figure 5.18: Instance tracks and zones occupancy representation

Given the test instance, each one of the following subsections presents some tests on a specific kind of unpredictable event that may happens with a brief description of the obtained results with respect to the plan just descripted. In detail, Subsection 5.3.1 is for trains delays, Subsection 5.3.2 is for delays of the time window minimum, Subsection 5.3.3 for the changes in the operation duration, Subsection 5.3.4 for suppressions, Subsection 5.3.5 for extraordinary trains and Subsection 5.3.6 for simultaneous events.

For this tests campaign, the model MOD-2 has been implemented using Python language (version 3.7) and solved by the commercial solver Gurobi 8.1.0 on a PC Intel Core i3, 2.00 G Hz; 4 G RAM.

5.3.1 Delays

This subsection deals with the trains delays. Some delays have been simulated in order to test the re-scheduling model.

Table 5.3 shows the delay data of the test instances: t^*, i.e. the time instant in which the event become known, Train ID, i.e. the train involved, the distinction between export (E) and import (I) cycle, the delay, e_j, i.e. the planned arrival/departing time, and e_j^*, i.e. the new arrival/departing time. The instances are grouped into classes because cases with different characteristics have been tested.

Note that, all the following data tables have the same first four columns (name of the Class and/or Instance, t^*, Train ID and the distinction between export (E) and import (I) cycle). The last columns will change depeding on the type of event that will be considered and will be explained each time.

Fow what concerns the tests here reported, classes **a**, **b** and **c** include instances where there is only one train on delay for each instance. The characteristics of each class are explained in the following.

In the instances of class **a**, the time instant in which the delay event become known is during the morning, i.e. t^* is between 6:00 and 12:00, and, obviously, the delay concerns a train planned to arrive/depart after t^*. This class includes two instances with an export train on delay (instances a.1, a.2) and two istances with an import train on delay (instances a.3, a.4). In particular, in instance a.1 the export train delays of 1 hour, while in instance a.2 the same export train delays of 4 hours. These tests are used to understand the solution changes in the two cases: a relative small delay (1 hour) and a bigger delay (4 hours). The same scheme, with a small and a big delay, is followed in instances a.3 & a.4 for an import train.

In the instances of class **b**, the delay event become known during the afternoon, i.e. t^* is between 12:00 and 18:00, and the involved train is planned to arrive/depart after t^*. Again, this class includes two instances with an export train on delay (instances b.1, b.2) and two istances with an import one on delay (instances b.3, b.4). The same scheme with a small and a big delay for the same train is followed in instances b.1 & b.2 (involving an export train) and instances b.3 & b.4 (involving an import train).

Then, in the instances of class **c**, t^* is during the evening, i.e. between 18:00 and 23:50, and again concerns a train planned to arrive/depart after t^*. Note that, class **c** includes a single instance because only one import train is planned during the evening in the considered day. Moreover, for train 5 in this class, only the delay of 1 hour has been tested because its planned departure time is 21:30 and, with a delay of 4 hours, it would have been departed out of the time horizon considered.

Finally, instances of classes **d** and **e** are different from those descripted above. Both these classes include only one instance in which two trains are on delay. The two delaying trains are one export and one import. The two delays become known at the same moment (t^*). The difference between the instance of class **d** and the one of class **e** concerns the moment of the day in which the delays become known. In the instance of class **d**, the delays become known during the morning, i.e. t^* is between 8:00 and 12:00. In the instance of class **e**, the delays become known during the afternoon, i.e. t^* is between 12:00 and 18:00.

Class	Instance	t^*	Train ID	E/I	Delay	a_j	a_j^*
a	a.1	08:20	2	E	01:00	09:00	10:00
	a.2	08:20	2	E	04:00	09:00	13:00
	a.3	08:20	4	I	01:00	15:30	16:30
	a.4	08:20	4	I	04:00	15:30	19:30
b	b.1	12:40	0	E	01:00	12:50	13:50
	b.2	12:40	0	E	04:00	12:50	16:50
	b.3	12:40	6	I	01:00	15:00	16:00
	b.4	12:40	6	I	04:00	15:00	19:00
c		18:20	5	I	01:00	21:30	22:30
d		08:20	2	E	04:00	09:00	13:00
		08:20	4	I	04:00	15:30	19:30
e		12:40	0	E	04:00	12:50	16:50
		12:40	6	I	04:00	15:00	19:00

Table 5.3: Delays - test data

The solutions values are reported in Table 5.4.

Apart from the first two columns of Table 5.4 that report the classes and the instances, the other columns show: the deviation from the time windows of terminal availability, the so colled distruption between the new schedule with respect to the planned one, the number of variables and constraints, the CPU and the GAP. Note that, all the following solutions tables will have the same characteristics.

As shown in Table 5.4, each instance has been optimally solved with a CPU

Class	Instance	TW dev	Disruption	Vars	Constrs	CPU (s)	GAP
a	a.1	00:00	00:00	17961	10573	5.07	0%
	a.2	00:00	04:20	17007	10177	5.76	0%
	a.3	00:00	00:00	18411	10771	6.62	0%
	a.4	00:00	00:00	18807	10969	5.59	0%
b	b.1	00:00	00:00	14009	8112	2.45	0%
	b.2	00:00	07:40	13055	7716	2.69	0%
	b.3	00:00	00:00	14459	8310	3.52	0%
	b.4	00:00	00:00	14855	8508	3.50	0%
c		00:00	00:00	13197	6742	0.07	0%
d		00:00	04:20	17535	10441	6.92	0%
e		00:00	07:40	13583	7980	2.99	0%

Table 5.4: Delays - results

time always less than 10 seconds. No instance has a TW deviation and few have a disruption. In particular, in both a.2 and the instance of class **d** the disruption is about 4 hours and 20 minutes. This obtained disruption is due to the fact that in both instances the delay of train 2 is considered. Then, in both b.2 and the instance of class **e** the disruption is about 7 hours and 40 minutes. The obtained disruption is because in both instances train 0 with a delay of 4 hours is considered and it creates some changes in the new plan.

We report, as example, the graphical representation of Fig.5.19. It shows the comparison between the old and the new schedule in the istance a.2 results. The upper part represents the plan and the lower one the new plan. In the new plan the red limit is t^* and only the rescheduled operations have been yellow coloured. The disruption time between the plan and the new plan is, in fact, of 4 hours and 20 minutes. In particular, Fig.5.19 shows that two operations have been changed.

The first is the operation through the primary zone ($i \in z^2$) of train number 2, which is the one on delay. The second is the operation through the primary zone ($i \in z^2$) of train number 7 that has been shifted forward in order to be able to manage the delay event, respecting all the constraints of the system.

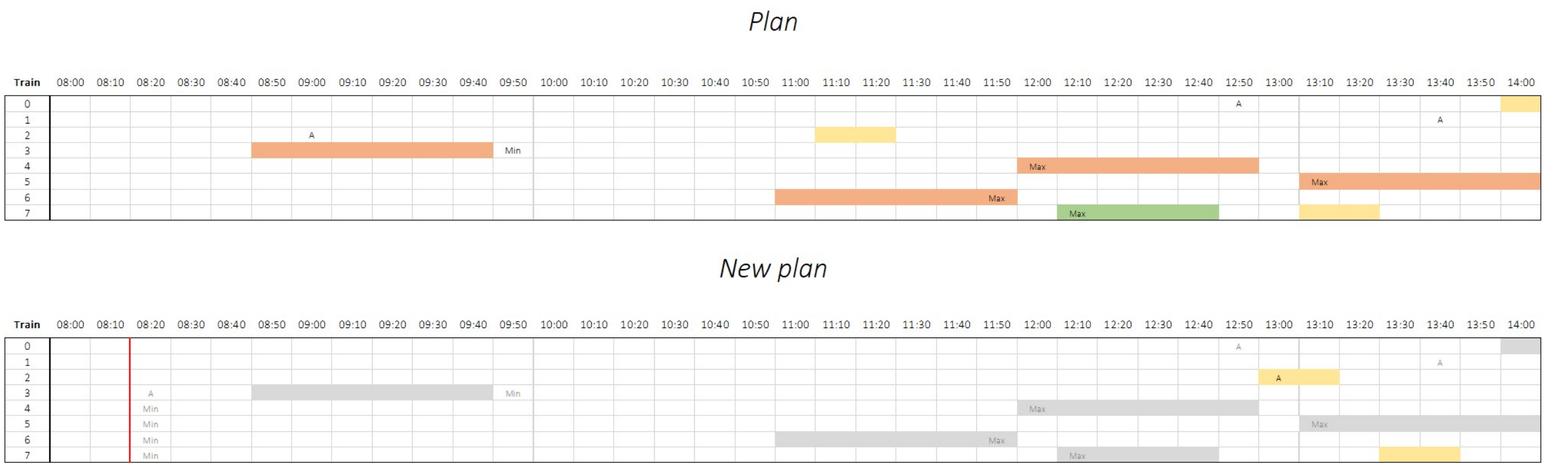

Figure 5.19: Instance a.2 result scheme

5.3.2 Delay of the time window minimum

This subsection describes the model tests in case of a change of the time window minimum. In the process, this delay in this data represents the maritime terminal that is unable to receive or release the train during the predefined time window of availability. This is more common for import trains due to loading operations delays in the terminal area, which depend also on the ships schedule. For this reason, we propose here some tests where only import trains are involved.

Table 5.5 shows the instances data. In this case, for each instance, the last columns show: the delay of the time window minimum, the planned time window minimum (e_j^{min}) and the new time (e_j^{*min}).

Class/Instance	t^*	Train ID	E/I	Delay	e_j^{min}	e_j^{*min}
a	10:40	6	I	05:10	06:00	11:10
b	10:40	6	I	05:50	06:00	11:50
c	12:00	7	I	06:20	06:10	12:30
d	12:00	7	I	07:30	06:10	13:40
e	13:50	5	I	11:30	07:10	18:40

Table 5.5: Delay of the time window minimum - test data

Table 5.6 shows the results.

Class/Istance	TW deviation	Disruption	Vars	Constrs	CPU (s)	GAP
a	00:10	00:00	16391	9491	6.99	0%
b	00:00	03:40	16391	9491	11.40	0%
c	00:10	00:20	15015	8660	2.91	0%
d	00:00	00:00	15015	8660	2.08	0%
e	00:00	00:00	12783	7373	1.42	0%

Table 5.6: Delay of the time window minimum - results

All the instances have been solved up to optimality in less than 12 seconds.

Note that, the time window deviation is admitted but it has to respect a specific limit because in the real process a relative small deviation from the time window of availability can be manually managed between the operators. In particular, it has to be within 30 minutes. In the tests under analysis, this limit has been respected in instances **a** and **c**, for which, in fact, the table report a time window deviation of 10 minutes in each instance.

Instances **b**, **d** and **e** exceeded the limit of 30 minutes. Given that the focus of the test is on the time window, the relative penalty in the objective function of the model has been increased in order to obtain a solution with a smaller time window deviation. Let's see Section 5.2 for what concernes the objective function of model MOD-2. The new results are those reported in Table 5.6. No one of instances **b**, **d** and **e** have a time window deviation.

Then, istances **b** and **c** present a disruption of 3 hours and 40 minutes and 20 minutes, respectively. This result is due to the changes that had to be applied in the schedule in order to manage the delay.

We report in the following, as example, Fig.5.20, which shows the comparison between the old and the new schedule in the instance **b** results. As in the previous example, the upper part represents the plan and the lower represents the new plan. In the new plan, the red limit is t^* and the re-scheduled operations have been orange coloured. The disruption between the plan and the new plan times is of 3 hours and 40 minutes. Fig.5.20 shows also that two operations through the unique zone z^5 have been changed. In particular, the operation of train 6 has been modified because of its time window minimum delay and the operation of train 4 has been anticipated in order to allow the management of the unpredictable event.

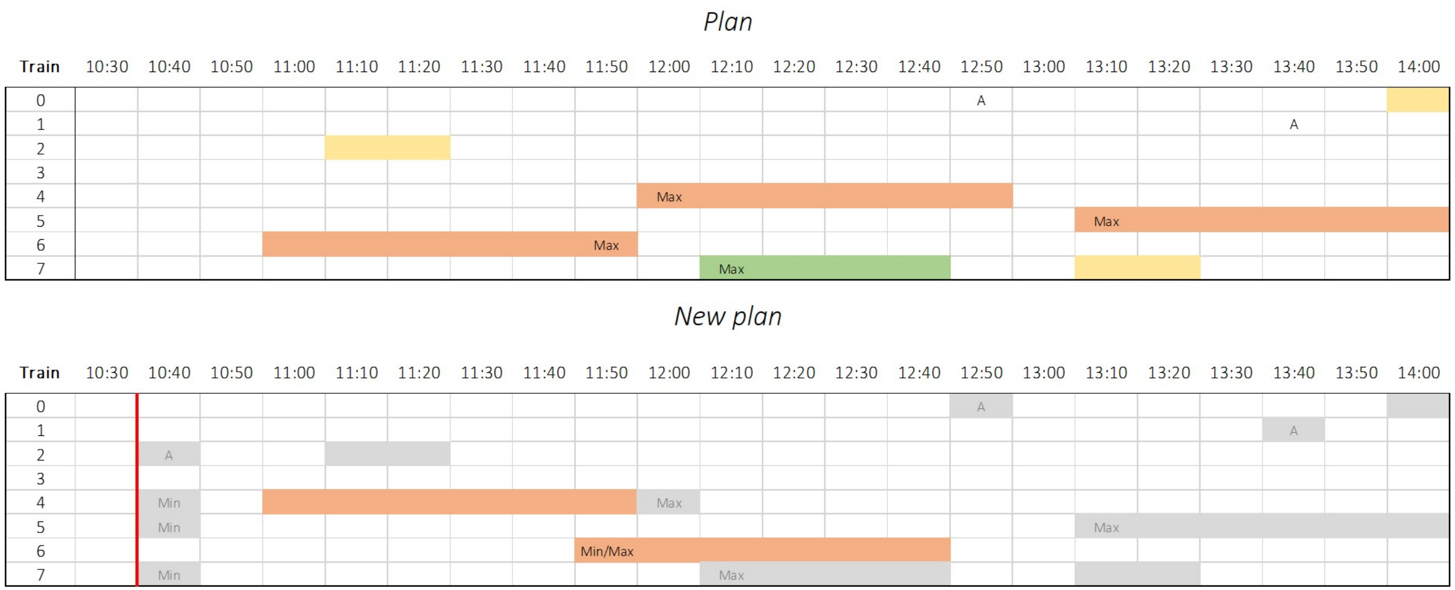

Figure 5.20: Instance b result scheme

5.3.3 Changes in the duration of an operation

This subsection includes the model tests in case of a change in the duration of a shunting operation. This change may be necessary if an operation need more time to be performed. This might be due to different problems: technical verifications on an import train before departure, problems on the infrastructures, problems for the resources, and so on.

Table 5.7 shows the instances data. There are two classes of instances, class **a** and class **b**. Class **a** includes the instances in which t^* is before the starting time of the involved operation, while class **b** includes the instances in which t^* is during the execution of the involved operation. That is realistic because sometimes the change could be known in advance, while other times this necessity can arise during the execution of the operation itself. The involved trains are the same in classes **a** and **b**. In this case, for each instance, the last columns represent: the involved operation (i), the planned and the new duration of operation i ($d_{i,j}$ and $d^*_{i,j}$).

Class	Instance	t^*	Train ID	E/I	i	$d_{i,j}$	$d^*_{i,j}$
a	a.1	11:40	0	E	z^4	00:40	01:20
	a.2	11:40	1	E	z^5	01:00	01:40
	a.3	11:40	7	I	z^4	00:40	01:00
	a.4	11:40	4	I	z^5	01:00	02:00
b	b.1	16:30	0	E	z^4	00:30	00:50
	b.2	16:30	1	E	z^5	00:50	02:00
	b.3	12:30	7	I	z^4	00:20	00:40
	b.4	12:30	4	I	z^5	00:30	00:40

Table 5.7: Processing time changes - test data

Table 5.7 shows the obtained results.

Class	Instance	TW deviation	Disruption	Vars	Constrs	CPU (s)	GAP
a	a.1	00:00	00:40	15359	8868	2.37	0%
	a.2	00:00	00:40	15359	8868	3.08	0%
	a.3	00:00	00:40	15359	8868	4.62	0%
	a.4	00:30	02:00	15359	8868	3.57	0%
b	b.1	00:00	00:20	9382	5434	0.35	0%
	b.2	00:00	01:10	9382	5434	0.40	0%
	b.3	00:00	00:40	14499	8348	3.00	0%
	b.4	00:00	00:30	14499	8348	2.35	0%

Table 5.8: Processing time changes - results

All the instances have been solved up to optimality in less than 5 seconds. Analysing the results, only instance a.4 has a time window deviation of 30 minutes and all instances have disruptions. The disruptions are unvoidable due to the fact that we are modifying the input data related to the operations duration, which means that the final times of the involved operations have to change.

We used here the instance a.4 as example because it is the one with the maximum disruption that we obtained. Fig.5.21 shows the comparison between the old and the new schedule. Again as before, the upper part represents the plan and the lower represents the new one. In the new plan, the red limit is t^* and only the rescheduled operations have been orange coloured. The disruption is of 2 hours. Fig.5.21 shows two changes. The first concerns the operation through the unique zone z^5 of train number 4. This change is due to the unpredictable event of this test, which consists in the increase of the processing time of this operation for train 4 itself. The second change is about the operation through the unique zone z^5 of train 5. The latter has been shifted forward in order to allow the managmement of the unpredictable event respecting all the constraints of the problem.

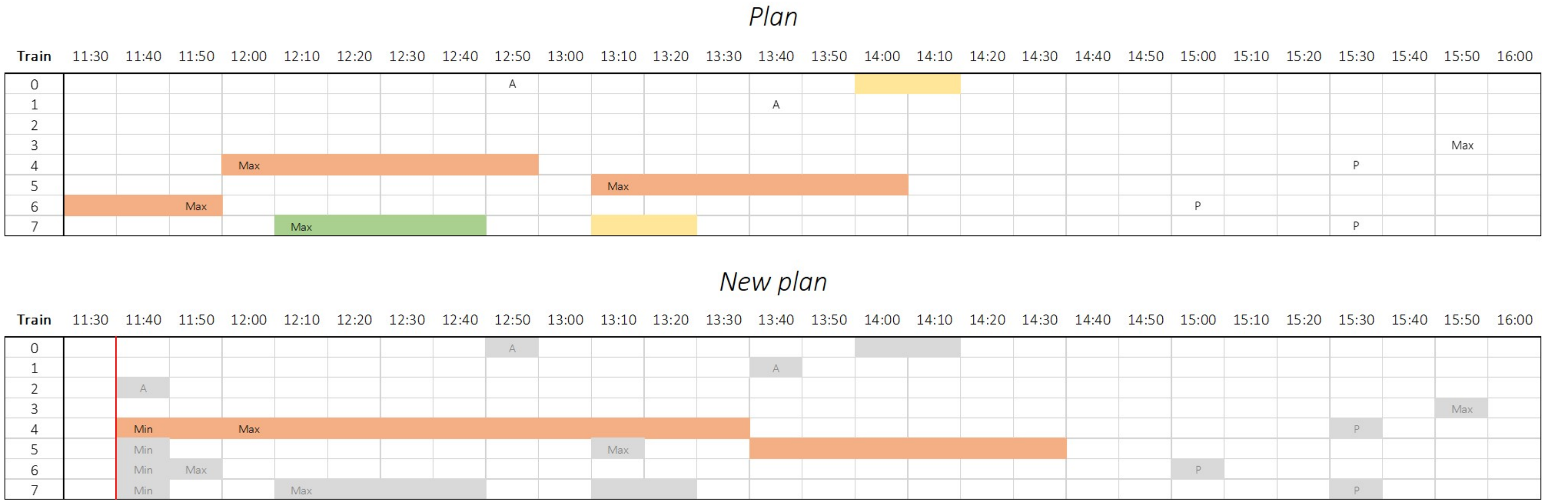

Figure 5.21: Instance a.4 result scheme

5.3.4 Suppressions

This subsection includes the model tests in case of a train suppression. The train suppression might become known either much time before the train arrival/departure time or relatively closed to that time. The two classes of instances reported in Table 5.9, which shows the test data, represent exactly this difference. Class **a** includes instances where t^* is some time before the arrival/departure time of the suppressed train, while class **b** includes instances where t^* is closed to the arrival/departure time of the suppressed train. Note that, for import trains, t^* closer to the departure time means, however, before the starting of the first operation. In Table 5.9, for each instance, the last columns represent: the ID of the suppressed train and its planned time of arrival/departure. The involved trains are the same in classes **a** and **b**. In each one, there are two instances, one with a suppression of an export train and the other with a suppression of an import train.

Class	Instance	t*	Train ID	E/I	e_j
a	a.1	09:10	1	E	13:40
	a.2	09:10	5	I	21:30
b	b.1	13:20	1	E	13:40
	b.2	12:50	5	I	21:30

Table 5.9: Suppression - test data

Table 5.10 shows the obtained results.

All the instances have been solved up to optimality in less than 7 seconds. Analysing the results, no instances have neither a time window deviation or a disruption. This means that all the other operations have to be performed as planned and it is due to the fact that the objective of the model is to minimize the disruption. Note that, as explained in Section 5.2, the objective function comprehends

Class	Instance	TW deviation	Disruption	Vars	Constrs	CPU (s)	GAP
a	a.1	00:00	00:00	17521	10218	4.35	0%
	a.2	00:00	00:00	17521	10218	6.25	0%
b	b.1	00:00	00:00	13480	7762	2.30	0%
	b.2	00:00	00:00	14155	8140	1.80	0%

Table 5.10: Suppression - results

the minimization of the disruption, of the waiting times and of the deviations. Therefore, even if we didn't obtain disruptions, they could have presented positive values. However, given that no disruptions or deviations has been obtained in these tests, the obtained new schedule are equal to the planned and no example will be graphically represented.

5.3.5 Extraordinary trains

This subsection includes the model tests in case of extraordinary train/s. One or more extraordinary trains might have to be included in the daily schedule.

Table 5.11 shows the test data. The test instances have been grouped into two classes. Class **a** includes instances with only one extraordinary train, while class **b** includes instances with two extraordinary trains.

In this case, the columns show the data related to the new train adding also the terminal of origin/destination (p_j), the planned arrival/departure time (e_j) and the time window minimum and maximum of terminal availability (e_j^{min}, e_j^{max}) of the new train/s.

The results are reported in Table 5.12.

All the instances have been solved up to optimality in less than 25 seconds. Some instances has an higher CPU time with respect to the previous tests. In

Class	Instance	t^*	Train ID	E/I	p_j	e_j	e_j^{min}	e_j^{max}
a	a.1	12:50	8	E	1	18:40	19:10	23:40
	a.2	17:50	8	E	1	18:40	19:10	23:40
	a.3	06:00	8	E	3	10:00	10:50	15:50
	a.4	09:50	8	E	3	10:00	10:50	15:50
	a.5	12:50	8	I	2	23:20	18:40	22:30
	a.6	17:50	8	I	2	23:20	18:40	22:30
	a.7	06:00	8	I	4	15:50	10:00	14:10
	a.8	04:00	8	I	4	15:50	10:00	14:10
b	b.1	09:50	8	E	1	18:40	19:10	23:40
		09:50	9	E	3	10:00	10:50	15:50
	b.2	09:10	8	I	3	23:20	18:40	22:30
		09:10	9	I	4	15:50	10:00	14:10
	b.3	09:50	8	E	3	10:00	10:50	15:50
		09:50	9	I	4	15:50	10:00	14:10
	b.4	18:10	8	E	1	18:40	19:10	23:40
		18:10	9	I	2	23:20	18:40	22:30

Table 5.11: Extraordinary trains - test data

general, this means that the problem of including extraordinary trains is more difficult to solve in terms of more operations to re-organize in order to be able to serve also the new train/s. That is also confirmed by the resulting time window deviations and disruptions values, which are higher when the CPU is higher as well. Analysing the results, the major time window deviation are in instances a.1, a.5 and b.1. Moreover, there are disruptions in instances a.1, a.5, b.2 and b.4. The disruptions mean that some planned operations have to be re-scheduled in order to serve also the extraordinary train/s.

We use instance b.3, in which there are two extraordinary trains as input, as example for the graphical representation. The obtained solution is shown in

Class	Instance	TW deviation	Disruption	Vars	Constrs	CPU (s)	GAP
a	a.1	02:40	18:00	14887	8505	23.51	0%
	a.2	00:00	00:00	11054	5989	0.35	0%
	a.3	00:00	00:00	24735	13872	4.26	0%
	a.4	00:00	00:00	21733	11876	3.35	0%
	a.5	02:40	18:00	15591	8857	11.77	0%
	a.6	00:00	00:00	11098	6011	0.30	0%
	a.7	00:10	00:00	23463	13344	5.55	0%
	a.8	00:10	00:00	19242	10842	4.94	0%
b	b.1	01:30	00:00	22949	12482	2.96	0%
	b.2	00:10	00:40	21644	12095	6.30	0%
	b.3	00:10	00:00	23724	12707	11.26	0%
	b.4	00:00	10:00	14551	7443	0.15	0%

Table 5.12: Extraordinary trains - results

Fig.5.22. In this particular case, there ins't the disruption because the two extraordinary trains could be included in the daily schedule with no changes to the other operations. This means that there were two free slots, in terms of available resources, in which it has been possible to insert these two trains. The upper part represents the plan and the lower represents the new plan. In the new plan, the red limit is t^* and only the new operations have been orange coloured. The disruption between the plan and the new plan times is, in fact, null.

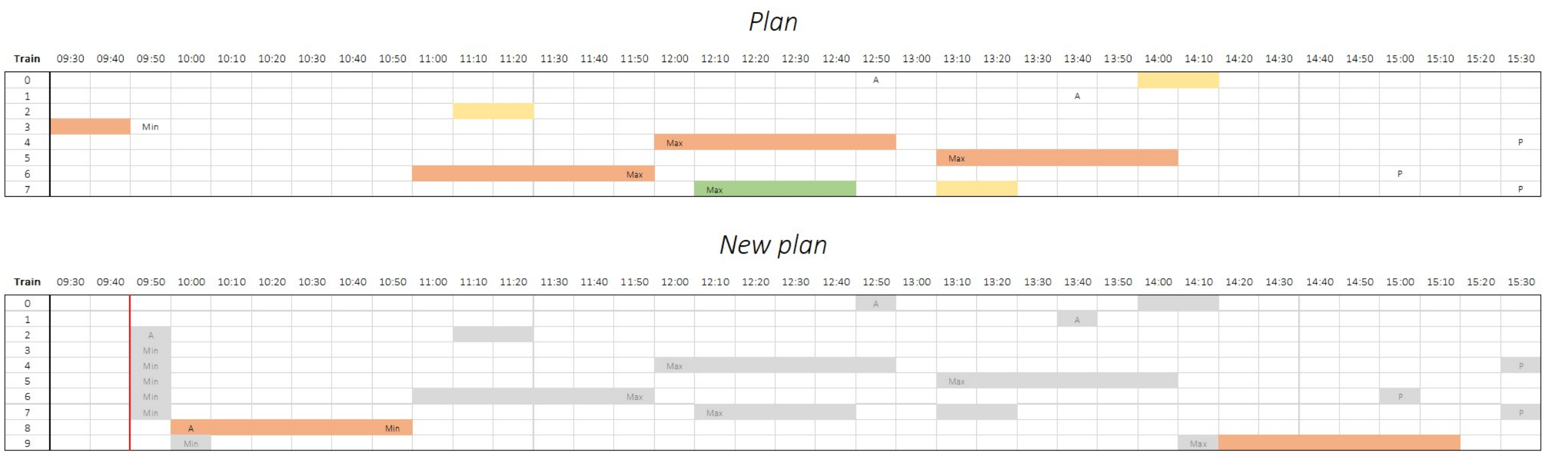

Figure 5.22: Instance b.3 result scheme

5.3.6 Simultaneous events

In this subsection, particular tests are reported. The test instances, in fact, includes each one two unpredictable events to manage together in the same time instant t^*, where both the events become known.

The considered events are the following:

I. **Export train delay** with the data reported in Table 5.13, where the specific columns show the delay, the planned arrival time e_j and the new arrival time e^*_j.

t^*	Train ID	E/I	Delay	e_j	e^*_j
12:10	0	E	01:00	12:50	13:50

Table 5.13: Export train delay

II. **Processing time change of an import train operation** with the data reported in Table 5.14, where the specific columns include the involved operation (i) and its planned and new duration ($d_{i,j}$ and $d^*_{i,j}$).

t^*	Train ID	E/I	i	$d_{i,j}$	$d^*_{i,j}$
12:10	7	I	z^4	00:40	01:00

Table 5.14: Processing time change of an import train operation

III. **Time window minimum change for an import train** with the data reported in Table 5.15, where the dedicated columns show the delay of the time window minimum, the planned time e_j^{min} and the new time e_j^{*min}.

t*	Train ID	E/I	Delay	e_j^{min}	e_j^{*min}
10:40	6	I	05:10	06:00	11:10

Table 5.15: Time window minimum change for an import train

IV. **New extraordinary train** with the data reported in Table 5.16, where the specific columns report the terminal of the new train and its arrival time (e_j), time window minimum (e_j^{min}) and maximum (e_j^{*max}).

t^*	Train ID	E/I	p_j	e_j	e_j^{min}	e_j^{max}
10:40	8	E	1	18:40	19:10	23:40

Table 5.16: New extraordinary train

V. **Train suppression** with the data reported in Table 5.17.

t^*	Train ID	E/I
10:40	1	E

Table 5.17: Train suppression

The just explained events have been combined in order to be tested simultaneously. Four instances have been created, each one with two of the above mentioned events, as explained in the following.

Instance **a** includes the events I and II, while instance **b** considers III and IV. Then, instance **c** includes the events III and IV as the previous and the difference between the two instances is the penalty for the time window deviation in the objective function of instance **c**. In the latter, the penalty for the time window deviation in the objective function has been increased because otherwise the results exceeded the limit of 30 minutes. Finally, instance **d** includes the events III and V. Also in this case, the penalty for the time window deviation has been increased in the objective function because otherwise the results exceeded the limit of 30 minutes.

The obtained results are shown in Table 5.18. All the instances have been solved up to optimality in less than 10 seconds. Analysing the results, instance **b** is the only one with a time window deviation. Moreover, each instance has a

Instance	TW deviation	Disruption	Vars	Constrs	CPU (s)	GAP
a	00:00	00:10	14525	8424	1.26	0%
b	00:10	00:00	17123	9855	2.28	0%
c	00:00	03:20	17123	9855	9.75	0%
d	00:00	03:20	16391	9491	5.06	0%

Table 5.18: Simultaneous events - results

disruption apart for instance **b**.

As example, Fig.5.23 shows the comparison between the old and the new schedule in the result of instance **a**. In the new plan, in the lower part of the schedule, the red limit is t^* and only the new operations have been green coloured. The disruption between the plan and the new plan times is of 20 minutes. Fig.5.23 shows also that the first event, which is the delay of train number 0, has been managed without disruption because the planned time for its operations were however after the new arrival time. The only disruption concerns the second event, i.e. the change in the processing time of operation through the secondary zone z^4. In fact, the end time of the latter has been changed of 20 minutes, which is exactly the introduced increase of time.

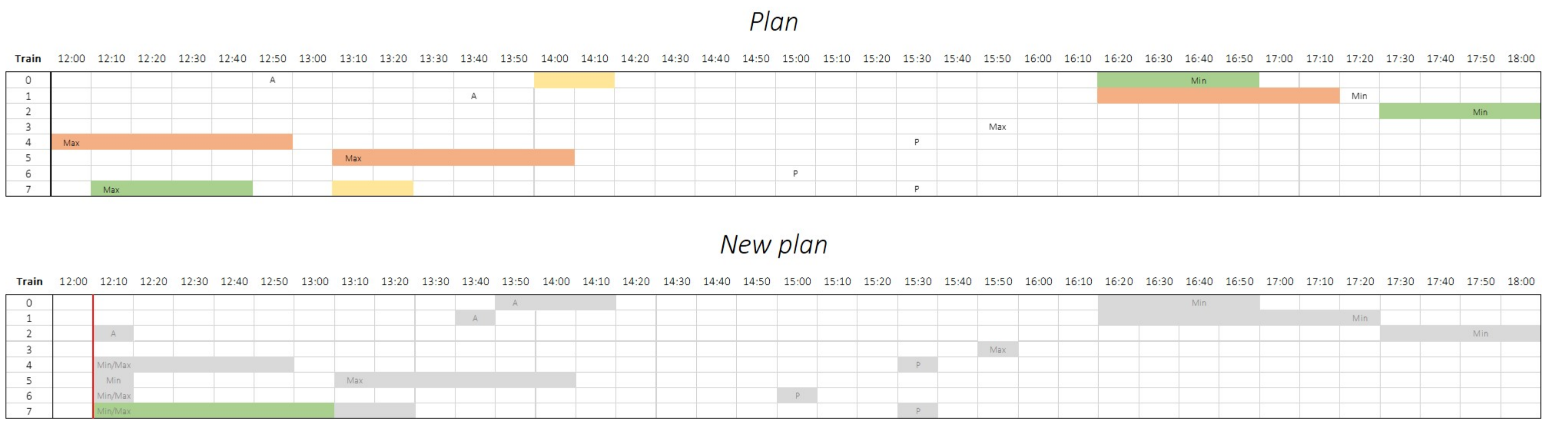

Figure 5.23: Instance a result scheme

Chapter 6

Case study: the shunting process of a port area in Italy

Summary

Analysing the layout of some port areas in Italy, where the rail shunting operations are managed, it is easy to note that they present different layouts, different capacities and different time constraints. The operation-time-space network proposed in this thesis can be easily used to model and solve the port rail shunting optimization problems in different port areas, i.e. different layouts.

This Chapter presents the discussion about a real case where the models for solving PRSSP and PRSRP has been tested. The real case is a port area located in Italy. For privacy issues, we won't refer to the specific name of it. The Chapter includes a description of the considered system (Section 6.1) and an overview of the main tests and results based on this case study (Section 6.2).

6.1 Case study specificities

The rail-sea modality exchange node configuration of the case study port area includes a railway station considered as a buffer, a shunting park and five maritime terminals. The system is depicted in Fig.6.1 and each zone is described in the following.

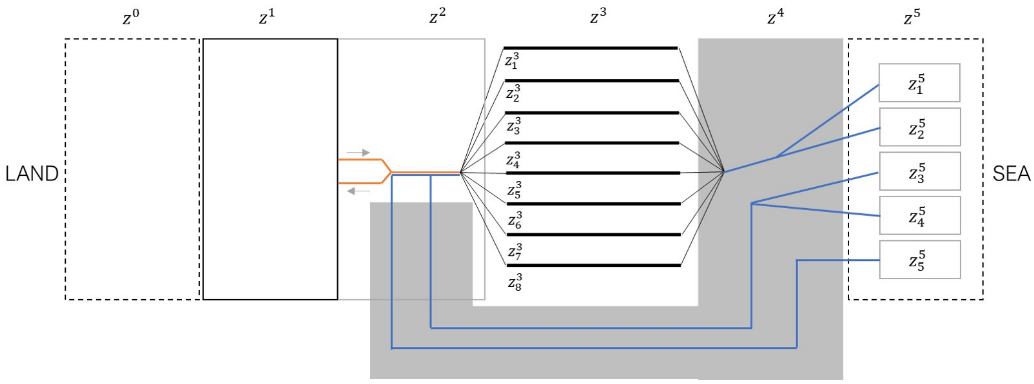

Figure 6.1: Case study port area configuration

- z^0 represents the whole railway network; it is included in this sketch just for completeness since it represents the rail transport system imposing some time constraints to the shunting manager;
- z^1 represents the railway station considered here as buffer where trains start and end their shunting operations and can wait after their arrival or before their departure. Note that, even if this station has some tracks we consider it as a general buffer because in reality these tracks are managed by another operator;
- z^2 represents the connection between the station area (z^1) and the park (z^3). This zone, inside the shunting area, is here called primary area. The connection track in the primary area is used to transfer trains from the railway

station area to the shunting park and viceversa. This area is composed by two tracks, one for the export cycle and one for the import, which come together before reaching the park;

- z^3 represents the shunting park, in the following called just "park", with its eight tracks here denoted z_1^3, z_2^3, z_3^3, z_4^3, z_5^3, z_6^3, z_7^3, z_8^3 where the waiting operations for trains arriving from either the railway station or the terminals can be performed;

- z^4 represents the connection between the park (z^3) and the terminals (z^6). This zone, inside the shunting area, is here called secondary area. The connection tracks in the secondary area are used to transfer trains from the shunting park to the terminals and viceversa. Note that, in this particular layout, all the maritime terminals have to be reached passing through the shunting park. Looking to Fig.6.1, from the up, the first two terminals can be reached by passing completely through the shunting park and, then, using the blu link on the right side of the park itself; the other terminals (3, 4 and 5) have to be reached again from the shunting park by passing on the left side of it. This means that the trains for these terminals have to arrive on a track of the shunting park from the previous operation and then they have to leave the park using the tracks on the left side. Note that, doing that, these trains share a portion of the infrastructure with the trains passing through zone z^2;

- z^5 represents the terminals z_1^5, z_2^5, z_3^5, z_4^5, z_5^5, that are either the origin or the destination of trains passing through the port.

Note that, differently from the general case explained in Chapter 4, in this case, there isn't an operation/zone for the direct link between the railway station and the terminals. As already said, all the trains have to pass through the shunting park in their paths.

In this system, both import and export trains have to be managed.

An *export* train arrives in the buffer station accordingly with its arrival time and has to be inside the terminal of destination after the complection of both the shunting operation through the primary zone (z^2) and the shunting operation through the secondary zone (z^4). Note that, in some cases, a train has to wait on a track of the park (z^3) between the two shunting opeations depending on the system congestion.

An *import* train has to perform the shunting operation through the secondary zone (z^4) and the shunting operation through the primary zone (z^2), respectively. At the end of the latter it arrives in the station area on time to leave the port accordingly with its departure time. Note that, as before, a train can wait on a track of the park (z^3) between the two shunting operations.

The duration of the shunting operations is known. The duration of the waiting operations derives from the initial and final times of the shunting ones.

Given that the paths change, depending on the terminal of origin/destination, all the possible paths of an export train in the system are described in the following (see Fig.6.1).

For what concerns the export trains:

- *Terminals z^5_1 or z^5_2 as destination*

From the station (z^1), an export train need to pass through the primary zone (z^2) until a track of the park (z^3). Then, it has to pass through the secondary zone (z^4) in order to reach the terminal of destination. In the latter passage, trains going to terminals z^5_1 and z^5_2 share a portion of the infrastructure.

- *Terminals z^5_3, z^5_4 and z^5_5 as destination*

The trains with these terminals as destination share some portions of the same infrastructure and, thus, the paths are explained together. From the buffer station (z^1), an export train need to pass through the primary zone (z^2) until the park (z^3). Then, the train has to pass through the secondary zone (z^4), sharing a portion of the infrastructure also with the trains passing through the primary zone (z^2), in order to reach one terminal among z_3^5, z_4^5 and z_5^5.

For what concerns the import trains:

- *Terminals z_1^5 or z_2^5 as origin*

An import train is picked up from terminal and transferred through the secondary zone (z^4) until a track of the park (z^3) where it can wait or not. During the operation through the secondary zone (z^4), trains arriving from terminals z_1^5 and z_2^5 share a portion of the same infrastructure. Then, the train have to be transferred through the primary zone (z^2) until the station buffer, where it will depart leaving the port area.

- *Terminals z_3^5, z_4^5 and z_5^5 as origin*

Given that the trains with these terminals as origin share some portions of the same infrastructure, the paths are explained together. An import train is picked up from the terminal and transferred through the secondary zone (z^4) until a track of the park (z^3). During the operation through the secondary zone (z^4), the train has to share a portion of the infrastructure also with the trains passing through the primary zone z^2. Then, the train has to be transferred through the primary zone (z^2) until the station buffer for leaving the port area.

Depending on the own rail infrastructure, maritime terminals has a different capacity in terms of receiving trains. Terminals z_1^5 and z_5^5 can host two trains simultaneously while terminals z_2^5, z_3^5 and z_4^5 only one.

Note that, given the complex situation of the system in terms of shared infrastructure, we have identified two generic zones where only one operation of the several possible can be executed in each time instant. These generic zones are here called $Zona1$ and $Zona2$. $Zona1$ includes the operation through the primary zone and the operations through the secondary zone linked to terminals z_3^5, z_4^5 and z_5^5, i.e. the red lines in Fig.6.2. $Zona2$ includes the operations through the secondary zone linked to terminals z_1^5 and z_2^5, i.e. the green lines in Fig.6.2.

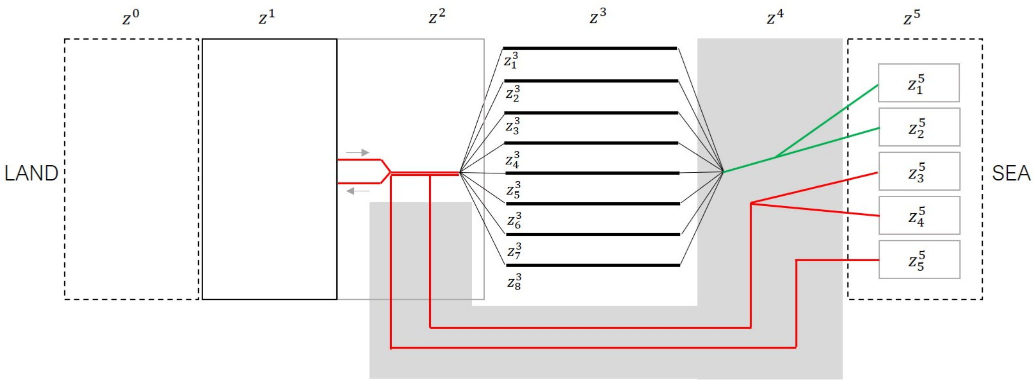

Figure 6.2: Graphical representation of $Zona1$ and $Zona2$

Other resources needed to perform the shunting operations have to be taken into account. This set of resources includes the shunting teams depending on each working shift, the engines and all the other mandatory figures to perform the activities. In particular, we consider a specific figure that is mandatory for the shunting operation through the primary zone (z^2), which is here called pilot, and both a shunting team and an engine for performing the shunting operations through secondary zone z^4.

Fig.6.3 shows a portion of the used operation-time-space network: (a) for the export cycle and (b) for the import one. Concerning the arcs of the network, only the ones that can be travelled for going from z^0 to z^5 (export cycle) and viceversa (import cycle) are shown, representing the couples of compatible operations. In particular, the red arcs in (a) are for the export cycle and the green arcs in (b) are for the import one.

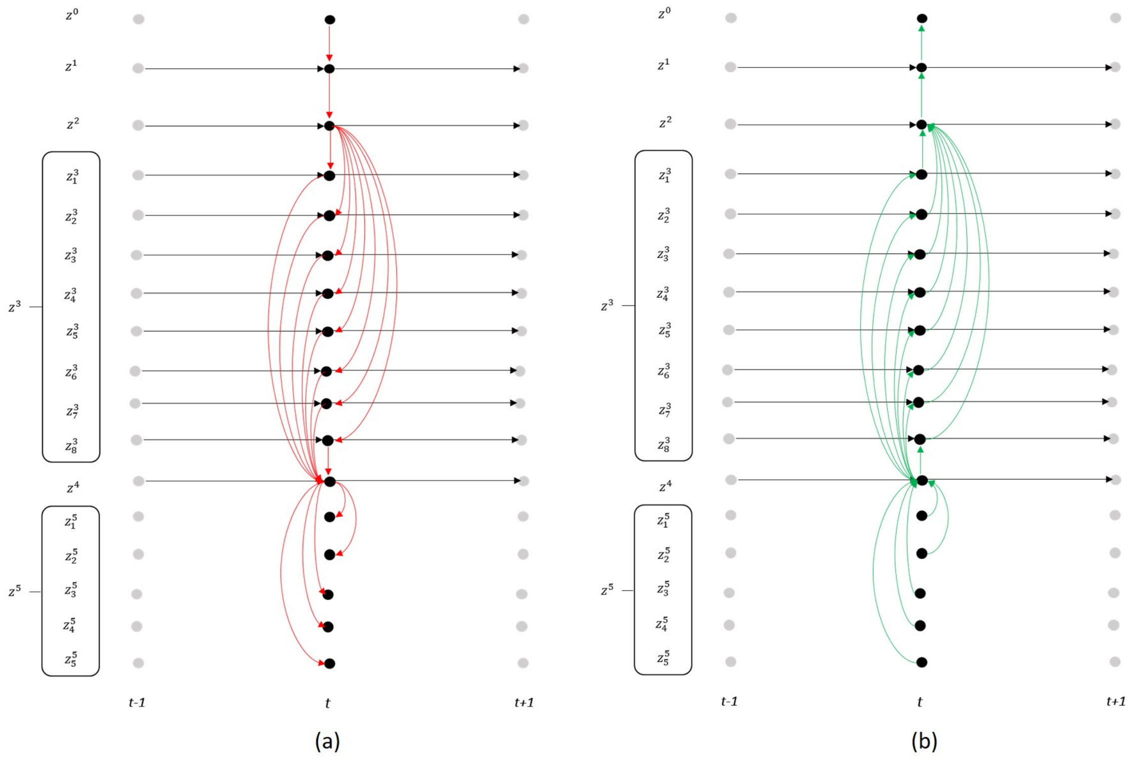

Figure 6.3: Portion of the used operation-time-space network

Then, Fig.6.4 shows two examples of flows in the network. The paths of two trains, one export and one import, that have to be trasferred within the considered area are depicted. In detail, the two paths represent the sequence of operations to be performed by each train: the red for the export train and the green for the import one.

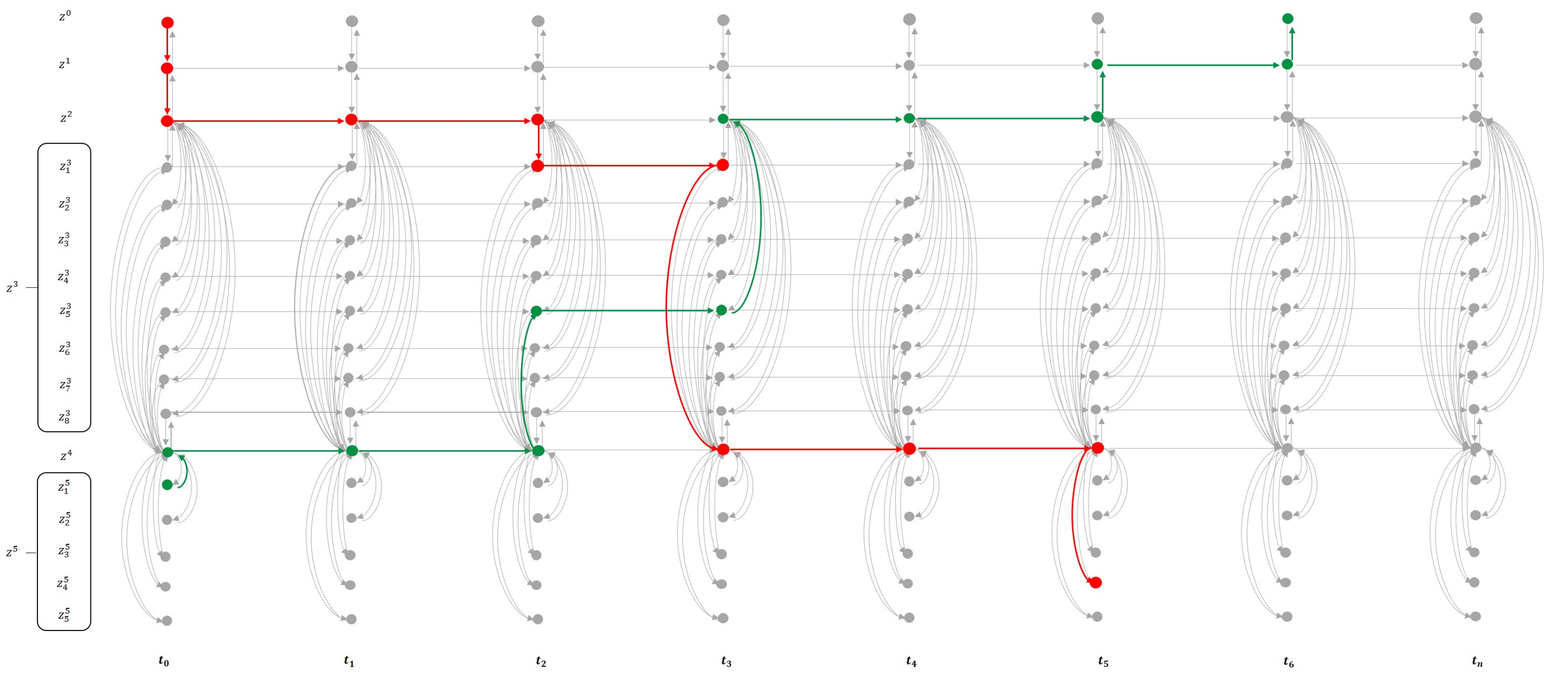

Figure 6.4: Example of paths on the operation-time-space-network

Let's suppose that the import train (green path) must leave the terminal z_1^5 within the time window $[t_0; t_1]$, and its departure time from the rail station is t_6. Then, the export train (red path) has the arrival time at the railway station in t_0 and has to be at destination, i.e., inside the terminal z_4^5 within the time window $[t_4; t_6]$.

Looking at Fig.6.4, the export train (the red one) arrives from the rail network (z^0) in t_0 and it starts immediatly the shunting operation through primary zone (z^2). The train performs this operation for two time intervals, i.e. $[t_0;t_1)$ and $[t_1;t_2)$. In time instant t_2 it finishes its primary operation and starts the waiting operation in the first track of the shunting park (z_1^3). The train waits for one time interval and then, in time instant t_3 starts the secondary operation in z^4. It performs this latter operation for two time intervals until t_5. Finally, in t_5, once finished its secondary operation, the train arrives at destination (terminal z_4^5).

Then, considering the import train (the green path), it is ready at origin, i.e. terminal z_1^5, in time instant t_0. In this time instant, it starts the shunting operation through the secondary zone (z^4). It performs this operation for two time intervals until t_2; in t_2 it starts to wait in the fifth track of the shunting park (z_5^3). The train waits for one time interval and then, in time instant t_3, starts the operation through the primary zone (z^2). It performs the operation for two time intervals and, then, in time instant t_5, it arrives in the buffer station. It waits one time interval here and, then, departs in time instant t_6.

The next section includes the description of the test campaign performed on the real system just explained.

6.2 Tests on the case study system

The present section proposes specific tests on the case study system. In particular, real data related to the trains characteristics have been used (arrival/departure times, terminal of origin/destinations, directions, and so on). The specific data won't be reported for privacy issues, but the main characteristics of the instances will be explained.

The tests on the generic models developed for PRSSP and PRSRP have been deeply analysed in Section 4.4 of Chapter 4 and Section 5.3 of Chapter 5, respectively. The main aim of the test campaign reported in this section is to verify that the general models are applicable in a real context, both in terms of solution time and goodness of the results.

In the following, subsection 6.2.1 concerns the performed tests for the Port Rail Shunting Scheduling Problem, while subsection 6.2.2 concerns the tests for the Port Rail Shunting Re-scheduling Problem.

6.2.1 PRSSP

This subsection includes the tests on the model developed for PRSSP and adapted to the case study system.

The model has been implemented using Python language (version 3.7) and solved by the commercial solver Gurobi 8.1.0 on a PC Intel Core i3, 2.00 G Hz; 4 G RAM.

Two instances have been tested: the first is characterized by a certain real dimension always manually managed in the system, the second has bigger dimensions in terms of number of trains to schedule. Even if the second instance has bigger dimensions, it is however realistic and concern a case under inspection.

Tests on a real instance with a certain number of trains are reported in the subsection 6.2.1.1, while subsection 6.2.1.2 includes the tests perfomed by using an instance with bigger dimensions. In fact, the main difference between the two concerns the number of trains to schedule: in the first they are 64 while in the second 86.

6.2.1.1 Instance with real dimensions

Firstly, some tests using real data related to a given week (from monday to saturday) have been performed. Each day is composed by 4 working shifts of 6 hours each. These tests includes 64 trains as input, 33 export trains and 31 import trains. The distribution of these 64 trains during the week is shown in Table 6.1.

Mon	Tue	Wed	Thu	Fri	Sat
8	11	13	14	11	7

Table 6.1: Weekly distribution of the trains

The highest number of trains per day is on Thursday and similar numbers have been registered for Tuesday, Wednesday and Friday. Monday and Saturday have less trains than the other days.

Then, the distribution of these 64 trains among the 5 maritime terminals of the system is shown in Table 6.2. The test week has the highest number of trains for terminal z_3^5, a similar number of trains with origin/destination the terminals z_1^5 and z_5^5 and less trains for terminal z_2^5. Terminal z_4^5 has no train in the testing week.

z_1^5	z_2^5	z_3^5	z_4^5	z_5^5
12	6	29	0	17

Table 6.2: Trains distribution among the terminals

Starting from the just explained input data, seven different tests have been executed in order to evaluate the results in terms of needed time for a good and applicable solution in a real context.

Note that, the maximum number of pilots for each working shift is 1 for all the seven tests, while the other data are going to be explained in the following.

The used flow model has been derived from MOD-1 proposed in Chapter 4 and adapted in accordance with the real network decribed in Section 6.1. The adapted objective function has the following terms (out of order):

1. **Waiting time in the rail station:** is the total number of time instants passed by trains in the rail station;

2. **Waiting time in the shunting park:** is the total number of time instants passed by trains in the shunting park;

3. **Time window deviation:** is the total deviation from the time windows of availability of the maritime terminals;

4. **Number of shunting teams deviation:** is the total deviation between the maximum number of available shunting teams and the number of shunting teams needed for the resulting plan;

5. **Number of pilots deviation:** is the total deviation of the number of needed pilot for the resulting plan with respect to the maximum number of available pilots in the system;

6. **Number of** $Zona1$ **deviation:** is the total difference between the simultaneous operations belonging to $Zona1$ in the resulting plan with respect to the maximum possible;

7. **Number of** $Zona2$ **deviation:** is the total deviation between the simultaneous operations of $Zona2$ in the resulting plan with respect to the maximum possible;

8. **Train not served:** are the trains that haven't been served in the resulting plan.

Some weights have been attributed to each term of the objective function in order to give a preference order. In this experimental campaign, we solved the same instance varying the weights that have been attributed to the several terms of the objective function, in order to give more or less importance to some possibilities and see the corresponding results. Seven different tests, named **a**, **a'**, **b**, **c**, **c'**, **d**, **d'**, are considered, as report in Table 6.3. The preference order of the objective function terms in these tests are explained in the following.

In tests **a**, **a'**, **b** the preference order is (the first is the less penalized and the last is the most penalized):

$$2, 1/3, 4/5, 6/7, 8$$

Thus, in this cases, the less penalized is the waiting time in the park (2), then waiting time in the rail station (1) and time window deviation (3) have the same weight. Again, number of shunting teams deviation (4) and number of pilots deviation (5) have the same weight, bigger than the previous three terms. After that, there are the number of deviation for $Zona1$ (6) and for $Zona2$ (7), which have the same weight. Finally, the most penalized is the train not served (8).

In tests **c** and **c'** the preference order has been slightly changed; in particular, the order is:

$$2, 1/3, 4, 5, 6/7, 8$$

The difference from the previous is that the number of shunting teams deviation (4) and number of pilots deviation (5) have here different weights and the first is less penalized than the second.

Finally, in tests **d** and **d'** the preference order has been changed again in order to give less importance to the time windows deviation:

$$2/3, 1, 4, 5, 6/7, 8$$

In the preference order of tests **d** and **d'**, in fact, the time window deviation is in the first position together with the waiting time in the shunting park. The other weights are in the same order as in tests **c** and **c'**.

All the other tests characteristics are shown in Table 6.3. Note that, we vary also the maximum number of shunting teams, which is 2 for test **b** and 1 for the others, and the time limit stopping criteria, which is fixed either to 1 hour (3600 seconds) or 7 hours (25200 seconds).

Test	Teams_{Max}	TimeLimit (s)	MIP Gap
a	1	3600	/
a'	1	25200	/
b	2	3600	/
c	1	3600	/
c'	1	25200	/
d	1	3600	/
d'	1	25200	/

Table 6.3: Tests characteristics

As shown in Fig.6.3, only the TimeLimit parameter has been set. The TimeLimit for tests **a**, **b**, **c**, **d** has been fixed to 1 hour, while for tests **a'**, **c'**, **d'** has been fixed to 7 hours. Note that, the tests with the same letter are the same apart for the value of the TimeLimit parameters. For test **b**, there wasn't the need to increase the TimeLimit parameter because it has been optimally solved in 1 hour.

The tests results are reported in Tables 6.4, 6.5 and 6.6.

The model has 376157 variables and 165628 constraints. Table 6.4 shows the CPU time and the GAP. All the tests exept for test **b** reached the TimeLimit parameter (see Table 6.3) with the GAP reported in the last column. However, they found good solutions as explained in the following. Test **b** reached the optimal solution in less than one hour.

Test	CPU (s)	GAP
a	3600	100%
a'	25200	11%
b	2830	0%
c	3600	100%
c'	25200	12%
d	3600	100%
d'	25200	31%

Table 6.4: Tests results - dimensions, CPU and GAP

Table 6.5 shows the solutions quality in terms of waiting operations of the considered trains. The first column is the test name. The second column represents the time, expressed in hours, spent by the trains waiting at the station after arrival/before departing. Note that, this result is the sum of the time spent in station by the 64 trains during the entire week. In all the tests this time is less than 3 hours in total, which is a great result given that in the real case this waiting time is about 7 hours. The third colums represents the average time per train spent by waiting in the rail station. These values vary from less than 1 minute to a maximum of 2 minutes and half in average and in the worst case is 50 minutes. The fourth column represents the hours spent by the trains waiting at the shunting park. Again, these results are the sum of the time spent in the park by the 64 trains during the entire week. In all the tests this time is less than 155 hours in total. This result is also good given that the same waiting time in the real case is about 500 hours. The fifth column shows the average time per train spent in the shunting park. These values vary in a range between 1 hour and half and 2 hours and half in average and in the worst case is 8 hours. The last column represents the park tracks utilization in % during the week. This result shows that the park tracks are used less than the 20% of their capacity and, therefore, they don't seem to constitute a bottleneck of the system.

Test	Station (h)	Avg per Train (h)	Park (h)	Avg per Train (h)	Park utilization (%)
a	0.83	0.01	104.17	1.63	11%
a'	2.67	0.04	141.83	2.22	17%
b	2.17	0.03	112.17	1.75	13%
c	0.83	0.01	128.33	2.01	15%
c'	2.33	0.04	144.00	2.25	15%
d	0.67	0.01	129.33	2.02	15%
d'	2.67	0.04	154.33	2.41	17%

Table 6.5: Tests results - waiting operations

Table 6.6 shows the results in terms of deviations, with respect to the resources capacity of the system, and not served trains. All the trains have been served in tests **a'**, **b**, **c'** and **d'**, while 8 trains haven't been served in test **a** and 5 in tests **c** and **d**. Note that, tests **a'**, **c'** and **d'**, for which the TimeLimit is fixed to 7 hours, reached better MIP Gaps and results (values of the objective function) than tests **a**, **c** and **d**, respectively, for which the TimeLimit is fixed to 1 hour. In fact, among the other results, in tests **a'**, **c'** and **d'** all the trains have been served.

Moreover, there aren't deviations except in tests **c**, **d** and **d'**. Test **c** presents a deviation from the capacity of *Zona1* for less than half an hour. In the reality this can be manually adjusted working with, for example, the time window of terminal availability. Going ahead, tests **d** and **d'** present both a time window deviation of about 3 hours and half each. These last results are due to the fact that the penalization of the time window deviation in these tests, as explain before, is smaller than in the other tests. In complex, Table 6.6 shows that all the results are good because not huge deviations have been obtained.

The penalties in the objective function have an influence in the behaviour of the solutions and can be used to slightly direct the model in a way rather than in an other, which might be interesting in the reality to perform scenario analysis and evaluate different possibilities. Finally, the needed time to obtain best solutions seems to be around 7 hours, which doesn't contitute a problem given that this kind of standard weekly plan is organized around two times a year and some months in advance.

Test	Teams_{Dev} (h)	Pil_{Dev} (h)	$\textit{Zona1}_{Dev}$ (h)	$\textit{Zona2}_{Dev}$ (h)	Trains NotServed (#)	TW_{Dev} (h)
a	0	0	0	0	8	0
a'	0	0	0	0	0	0
b	0	0	0	0	0	0
c	0	0	0.33	0	5	0
c'	0	0	0	0	0	0
d	0	0	0	0	5	3.17
d'	0	0	0	0	0	3.83

Table 6.6: Tests results - deviations

The last analysis concerns the number of shunting teams and pilots used in each working shift, compared with with the same elements of the real case. This comparison is due to the fact that the obtained results in terms of waiting time are better than the real case, but in many tests this is a consequence of the more resources that have been used than in the real case.

In the following, tests **a'** and **b** will be analyzed more in depth in order to inspect their results in terms of used shunting teams and pilots. We have selected these two tests as example because they have the best results in terms of MIP Gap reached between the tests with 1 and 2 shunting teams as maximum, respectively.

Tables 6.7, 6.8 and 6.9 shows the number of used shunting teams for each day and working shift in the three cases: real case, test **a'** and **b**.

	Mon	Tue	Wed	Thu	Fri	Sat
I	0	0	1	1	1	0
II	1	2	1	2	1	1
III	1	1	1	1	1	1
IV	1	1	1	1	1	1

Table 6.7: Shunting teams - real case

	Mon	Tue	Wed	Thu	Fri	Sat
I	1	1	1	1	1	1
II	1	1	1	1	1	1
III	1	1	1	1	1	1
IV	1	1	1	1	1	0

Table 6.8: Shunting teams - test **a'**

	Mon	Tue	Wed	Thu	Fri	Sat
I	1	1	2	2	1	1
II	1	2	2	2	2	2
III	1	2	1	1	1	1
IV	2	1	1	2	1	0

Table 6.9: Shunting teams - test **b**

Summing the used shunting teams for each case we obtain three comparable numbers representing the necessity of shunting teams during the week in the three cases. As shown in Table 6.10, in test **a'** is used the same number of shunting teams as in the real case, while in test **b** there is a delta of 10.

	Real case	Test **a'**	Test **b**
Shunting teams	23	23	33

Table 6.10: Shunting teams - necessity

Tables 6.11, 6.12 and 6.13 shows the number of used pilots for each day and working shifts in the three cases: real case, test **a'** and **b**.

	Mon	Tue	Wed	Thu	Fri	Sat
I	0	1	1	1	1	1
II	1	1	1	1	1	1
III	2	1	1	1	1	1
IV	1	1	1	1	1	0

Table 6.11: Pilots - real case

	Mon	Tue	Wed	Thu	Fri	Sat
I	0	1	1	1	1	1
II	1	1	1	1	1	1
III	1	1	1	1	1	1
IV	1	1	1	1	1	0

Table 6.12: Pilots - test **a'**

Summing the used pilots for each case we obtain three comparable numbers representing the necessity of pilots during the week in the three cases. As shown in Table 6.14, in test **a'** and **b** is used the same number of pilots that is almost the same as in the real case.

	Mon	Tue	Wed	Thu	Fri	Sat
I	0	1	1	1	1	1
II	1	1	1	1	1	1
III	1	1	1	1	1	1
IV	1	1	1	1	1	0

Table 6.13: Pilots - test **b**

	Real case	Test **a'**	Test **b**
Shunting teams	23	22	22

Table 6.14: Pilots - necessity

6.2.1.2 Instance with bigger dimensions

In this second part, the number of trains has been increased of about the 35% (86 trains in total). The purpose of this second part of tests was to understand if the model is able to manage a bigger number of trains in the system under inspection.

The distribution of the 86 trains during the week is shown in Table 6.15.

Mon	Tue	Wed	Thu	Fri	Sat
11	17	17	19	13	9

Table 6.15: Weekly distribution of the trains

The highest number of trains per day is on Thursday and similar numbers have been registered for Tuesday, Wednesday and Friday. Monday and Saturday have less trains than the other days.

The distribution of these 86 trains among the 5 maritime terminals of the system is shown in Table 6.16.

The tested week has the highest numer of trains for terminal z_3^5, almost the same number of trains with origin/destination the terminals z_1^5 and z_5^5 and less trains for terminal z_2^5. Terminal z_4^5 has 2 trains in the tested week.

z_1^5	z_2^5	z_3^5	z_4^5	z_5^5
19	10	34	2	21

Table 6.16: Trains distribution among the terminals

Starting from the just explained input data, five different tests have been done varying the maximum number of shunting teams and the termination criteria based on the TimeLimit and MIP Gap % as explained in the following.

The maximum number of pilots for each working shift is 1 for all the five tests. For these tests, we used the same model as in previous test (Section 6.2.1.1). As already descripted, some weights are attributed to each term of the objective function, in order to give a preference order. Considering the terms of the objective function listed in the previous section, here the preference order for all the five tests is (the first is the less penalized and the last is the most penalized):

$$2, 1/3, 4/5, 6/7, 8$$

The other characteristics of each test are shown in Table 6.17.

Test	Teams$_{Max}$	TimeLimit (s)	MIP Gap
a	1	3600	/
a'	1	25200	/
b	2	3600	/
b'	2	25200	/
b"	2	/	20%

Table 6.17: Tests characteristics

The maximum number of shunting teams is 1 in tests **a** and **a'** and 2 in tests **b**, **b'** and **b"**. Then, the differences concerns the termination criteria TimeLimit and

MIP Gap %. Starting with the tests called with letter a, the TimeLimit parameter is 1 hour in test **a** and 7 hours in test **a'**. Then, concerning the tests called with letter b, the TimeLimit parameter is 1 hour in test **b** and 7 hours in test **b'**. In test **b"** the TimeLimit hasn't been change and the MIP Gap % has been set to 20%.

The tests results are reported in Tables 6.18, 6.19 and 6.20.

The solved model has 512351 variables and 218157 constraints. Table 6.18 shows the CPU time and the obtained GAP. All the tests reached the TimeLimit except for test **b"** that reached a GAP of 20% in around 4 hours. All the GAPs are reported in the last column. All the tests found good quality solutions as explained in the following.

Test	CPU (s)	GAP
a	3600	100%
a'	25200	100%
b	3600	100%
b'	25200	8%
b"	14902	20%

Table 6.18: Tests results - dimensions, CPU and GAP

Table 6.19 includes the results in terms of waiting operations of the considered trains. The structure of the Table is the same as Table 6.5 of the previous Section 6.2.1.1. The result is the sum of the time spent either in the station or in the shunting park by the 86 trains during the week. In all the tests, the time spent in the station is less than 12 hours in total. That is a good result, given that, in the previous real case that has the 35% of trains less, the waiting time is about 7 hours. Then, the average time per train to pass in the rail station, is always less than half an hour. The worst case is a train that has to wait around two hours and half in the rail station.

For what concerns the hours spent by the trains in the shunting park, as for the rail station, this result is the sum of the time spent in the park by the 86 trains during the week. In all the tests, this time is less than 210 hours in total. Also this result is good given that the same waiting time in the previous real case, with the 35% of trains less, is about 500 hours. Moreover, looking to the average time per train spent in the shunting park, the values vary in a range between little more of 1 hour and 2 hours and half and the worst case is a train that has to stay in the shunting park for 11 hours and half.

Finally, the park tracks utilization in % during the week is 25% of their capacity and, therefore, also in this situation with 35% of trains more to manage, we can say that they don't constitute a bottleneck of the system.

Test	Station (h)	Avg per train (h)	Park (h)	Avg per train (h)	Park utilization (%)
a	29.00	0.34	103.33	1.20	13%
a'	10.50	0.12	205.67	2.39	23%
b	2.33	0.03	122.17	1.42	13%
b'	9.50	0.11	163.50	1.90	19%
b"	11.50	0.13	160.50	1.87	20%

Table 6.19: Tests results - waiting operations

Table 6.20 shows the results in terms of deviations, with respect to the capacity of the system, and not served trains.

The worst result has been obtained in test **a**, which presents 29 not served trains and around 10 hours of time windows deviation. The following, result of test **a'**, has been obtained in 7 hours and is definitely better even if the GAP is still 100%. The high GAP is due to the huge penalization imposed for each train that has been not served. Test **a'** result presents 5 not served trains and no time windows deviation. Going ahead with the tests called with letter b, increasing the

Test	Teams_{Dev} (h)	Pil_{Dev} (h)	$\mathit{Zona1}_{Dev}$ (h)	$\mathit{Zone2}_{Dev}$ (h)	NotServedTrains (#)	TW_{Dev} (h)
a	0	0	0	0	29	10.5
a'	0	0	0	0	5	0
b	0	0	0	0	15	0
b'	0	0	0	0	0	0
b"	0	0	0	0	0	0

Table 6.20: Tests results - deviations

TimeLimit stopping criteria from 1 to 7 hours, the solution improves considerably passing from 15 not served trains to all the trains served. Note that, we arrived at the same solution serving all the trains with a shorter CPU time by imposing another termination criteria based on the % of reached MIP Gap (fixed to 20%). Finally, apart in the first test, which is the worst, there aren't deviations in all the other solutions.

The last analysis concerns the number of shunting teams and pilots used in each working shift with respect to the real case. This comparison is used to understand the necessity of shunting teams and pilots considering the increase in the number of trains. The tests that will be analyzed in detail are tests **a'** and **b'**. The latter have been choosen because test **a'** has the best results with 1 shunting team as maximum and test **b'** has the best results with 2 shunting team as maximum. Note that, tests **b'** and **b"** has the equal solution in terms of deviations but test **b'** results slightly better in terms of waiting times.

Tables 6.21, 6.22 and 6.23 shows the number of used shunting teams for each day and working shift in the three cases: real case, test **a'** and **b'**.

Summing the used shunting teams for each case we obtain three comparable numbers representing the necessity of shunting teams during the week in the three cases. As shown in Table 6.24, in test **a'** is used almost the same number of

	Mon	Tue	Wed	Thu	Fri	Sat
I	0	0	1	1	1	0
II	1	2	1	2	1	1
III	1	1	1	1	1	1
IV	1	1	1	1	1	1

Table 6.21: Shunting teams - real case

	Mon	Tue	Wed	Thu	Fri	Sat
I	1	1	1	1	1	1
II	1	1	1	1	1	1
III	1	1	1	1	0	1
IV	1	1	1	1	1	0

Table 6.22: Shunting teams - test **a'**

	Mon	Tue	Wed	Thu	Fri	Sat
I	1	2	2	1	2	1
II	2	2	2	2	2	1
III	1	2	1	2	0	1
IV	2	2	1	2	1	0

Table 6.23: Shunting teams - test **b'**

shunting teams as in the real case, while in test **b'** there is a delta of 12 shunting teams.

	Real case	Test **a'**	Test **b'**
Shunting teams	23	22	35

Table 6.24: Shunting teams - necessity

Tables 6.25, 6.26 and 6.27 shows the numbers of used pilots for each day and working shift in the three cases: real case, test **a'** and **b'**.

	Mon	Tue	Wed	Thu	Fri	Sat
I	0	1	1	1	1	1
II	1	1	1	1	1	1
III	2	1	1	1	1	1
IV	1	1	1	1	1	0

Table 6.25: Pilots - real case

	Mon	Tue	Wed	Thu	Fri	Sat
I	0	1	1	1	1	0
II	1	1	1	1	1	1
III	1	1	1	1	1	1
IV	1	1	1	1	1	0

Table 6.26: Pilots - test **a'**

	Mon	Tue	Wed	Thu	Fri	Sat
I	0	1	1	1	1	0
II	1	1	1	1	1	1
III	1	1	1	1	1	1
IV	1	1	1	1	1	0

Table 6.27: Pilots - test **b'**

Summing the used pilots for each case we obtain three comparable numbers representing the necessity of pilots during the week in the three cases. As shown in Table 6.28, in test **a'** and **b'** is used the same number of pilots that is little less than the one used in the real case.

	Real case	Test **a'**	Test **b'**
Shunting teams	23	21	21

Table 6.28: Pilots - necessity

From the results analysis exploited in this section some considerations can be done. Firstly, the model developed can be used to solve also the analyzed bigger instance, which was the tests purpose. The performances are different and are affected by the maximum number of the available shunting teams. That is comprehensible, given that this kind of resource is crucial for the shunting operations execution. Therefore, tests with letter a, where the maximum number of shunting team is 1, need more time to find a good solution. In fact, test **a'**, where the TimeLimit parameter has been set to 7 hours, reaches definitely a better solution. Thus, in this case, give more time permits to obtain the better solutions. Note that, we didn't test only the MIP Gap because in 7 hours, which is the limit for a reasonable CPU time in the reality for this kind of problem, we found a solution with still a MIP Gap of 100%.

Then, among the tests with letter b, which have 2 available shunting teams as maximum, we can say that the parameter setting in test **b''**, i.e. imposing the MIP Gap at 20%, permits to obtain the best solutions. That method is better than the TimeLimit because it performs equal in terms of deviations (see Table 6.20) and similar in terms of waiting times (see Table 6.19) using less CPU time (4 hours).

In the proposed tests, we noted that, when the number of available resources is bigger, its better to impose a MIP Gap, instead of trying to reach 0% of MIP Gap, in order to obtain a good solution in less time.

6.2.2 PRSRP

This section proposes some tests on the model developed for PRSRP and adapted to the real case under inspection, descripted in Section 6.1.

As already explained, the problem consists of re-scheduling an already scheduled plan. The plan used as input in these tests has been obtained as result of

PRSSP. In detail, the plan is the result of test **c** of Section 6.2.1.1, which has been optimally solved. Even if the plan is a six days schedule, the PRSRP focuses on one day, thus we used one day of that plan. We limit the time horizon to 1 day because the re-scheduling problem comprehends one day in which unpredictable events may happen. In particular, has been selected a day with 14 trains, 7 export and 7 import.

The distribution of the 14 trains among the maritime terminals, distinguishing between export and import cycle, is shown in Table 6.29. As shown in the table, terminals z_1^5 and z_2^5 have 2 trains each, terminal z_5^5 has 4 trains, 2 export and 2 import, terminal z_3^5 has 6 trains, 3 export and 3 import, which is the maximum number of trains per terminal in this day, and terminal z_4^5 has no trains.

	z_1^5	z_2^5	z_3^5	z_4^5	z_5^5	TOT
E	1	1	3	0	2	7
I	1	1	3	0	2	7
TOT	2	2	6	0	4	**14**

Table 6.29: PRSRP tests - Train distribution

Given that real data has been used, for privacy reasons the specific data related to the characteristics of each train won't be shown. General data useful to understand the instance characteristics, such as the dimensions and the distributions of trains among the terminals, will be shown and explained.

Using the real data of the above descripted day as input, 4 tests have been executed. These 4 tests, which will be explained in the following, are related to real events that may happen during a day in the port area under inspection.

The first test, called in the following **Test 1**, concerns a delay of the export train of terminal z_2^5. The delay of this train becomes known one hour before its arrival time. The delay is half an hour. In the schedule of the activities this train should have immediately started the first operation after arrival, therefore, at least its operations will have to be rescheduled.

The second, called in the following **Test 2**, concerns a change in the processing time of the second shunting operation for the export train of terminal z_1^5. The duration of this operation has to be increased of half an hour due to a scrap of a wagon that broke down on the way. This change becomes known together with the initial time of the shunting operation itself, due to an inspection.

The third, called **Test 3**, concerns a train suppression due to commercial issues. The suppressed train is one import train of terminal z_3^5. The suppression becomes known ten minutes before the planned time in which the first shunting operation of the train itself had to start.

Finally, **Test 4** concerns an extraordinary train for the day under analysis. This new train is an export one and it has to reach terminal z_4^5. The event becomes known three hours before the arrival time of the new train.

Tables 6.30, 6.31 and 6.32 shows the obtained results, which are explained in detail in the following.

Test	Vars	Constrs	CPU (s)	GAP
Test 1	22563	11466	19.81	0%
Test 2	14347	5898	1.88	0%
Test 3	19651	9590	1.40	0%
Test 4	24504	12027	8.99	0%

Table 6.30: Dimensions, CPU and GAP

Table 6.30 includes dimensions, CPUs and GAPs. The dimensions changes depends on both the lenght of the time horizon of rescheduling and the number of trains already done before the time instant in which the new event becomes known. The latter because, as already explained, the trains that are already served before are excluded from the re-scheduling problem. As shown in the table, all the tests have been optimally solved in less than 20 seconds.

Test	Station	Park
Test 1	00:30	21:20
Test 2	00:30	02:10
Test 3	00:20	17:50
Test 4	00:30	09:50

Table 6.31: Waiting operations

Going ahead in the results analysis, Table 6.31 shows the waiting operations duration. The first column denotes the test name and, then, the second and the third columns represent the sum of the time that the trains have to wait either in the station or in the park, respectively. The time to pass in the station is few and it means that all the trains in the different tests wait really few time after their arrival and before their departure. The trains spend more time in the shunting park and this depends on both the state of the infrastructure and the availability of the terminals.

Test	Teams_{Dev}	Pil_{Dev}	$\mathit{Zona1}_{Dev}$	$\mathit{Zona2}_{Dev}$	TW_{Dev}	Disruption
Test 1	0	0	0	0	0	06:20
Test 2	0	0	0	0	00:10	03:00
Test 3	0	0	0	0	0	01:00
Test 4	0	0	0	0	0	01:20

Table 6.32: Deviations

Finally, Table 6.32 reports the results in terms of deviations and disruptions. From left, after the names, the columns represent the deviation from the maximum number of shunting teams, pilots, operations for *Zona1*, operations for *Zona2*, time windows of availability and the disruption. As shown in the table, all the capacity constraints have been satisfied. Then, **Test 2** has a really small time windows deviation of 10 minutes. Given that the disruption is the focus of these tests, each tests result in terms of disruption will be explained in details.

Test 1 has a disruption of 6 hours and 20 minutes due to the management of the train in delay which brought to change the planned times of the operations of other three trains. Anyway, the new plan respects both all the trains arrival/departure times and the time windows of availability of the terminals.

Test 2 has a disruption of 3 hours due to the management of the unpredicted event which brought to change the planned times of the operations of other two trains. Here again, the new plan respects all the trains arrival/departure times. The new plan has a little time window deviation of 10 minutes for only one train.

Test 3 has a disruption of 1 hour due to the train suppression which brought an improvement in the schedule in terms of waiting time in the park.

Test 4 has a disruption of 1 hour and 20 minutes due to the management of the new train which brought to change the planned times of the operations of other three trains. As in **Test 1**, the new plan respects both all the trains arrival/departure times and the time windows of availability of the terminals.

Chapter 7

Conclusions

The present Chapter is used to retrace the main steps of the thesis work and to summarize conclusions and future steps.

The present work arises within a collaboration between the Italian Center of Excellence on Logistics, Transports and Infrastructure (CIELI) and the company Circle SpA. The latter, which provides process and management consulting services, innovative technological solutions and digital marketing solutions having specific vertical expertise in ports, maritime and intermodal logistics, is interested in the optimization applications for solving real problems in the intermodal supply chain. In this context, the problems of scheduling and re-scheduling the rail shunting operations in a port area in order to reach good levels of organization and efficiency has been a really interesting challenge.

The starting idea was to find the best way to model the port rail shunting process in such a way to be able to develop the optimization models for scheduling and re-scheduling the main operations. The schedule of the main operations is useful for the involved stakeholders for organizing all the linked activities, while the re-scheduling model is for a better real time management of unpredictable events.

The scheduling problem, deeply analyzed in Chapter 4, consists in defining, for a given time horizon, the starting and the ending time of all the operations necessary for transferring trains within the considered area, respecting the time limits imposed by both the railway network and maritime terminal (influenced by the ships schedule), and the limits due to the finite number of available resources.

Given the plan of the shunting operations build either automatically by solving PRSSP or manually, the re-scheduling problem, addressed in Chapter 5, aims at determining a new schedule for the shunting operations taking into consideration the unpredictable events that may occur. The aim is to build a new plan as closed to the previous as possible in order to perform all the operations without changing too much the already planned activities. As in PRSSP, all the limitations have to be respected.

Note that, being each port area very different from each other, one of the main point to consider was to build a general structure of the problem in order to be able to apply it in the most possible cases. For this reason, after the analysis of the concerning literature, has been decided to develop an approach based on a flow model on an innovative time-space network.

The analyzed literature concerns both the freight trains circulation on the railway network, which includes works on planning problems and works on real time management problems, the management of freight trains in rail terminals, which might be either rail, rail-road or rail-sea yards, and the well known time-space networks.

An innovative network, here called operation-time-space network has been build and deeply descripted in the dedicated Chapter 3. This network has the advantage to be able to easily manage both shared pieces of infrastructure, shared resources and time constrains.

Thanks to this passage, a network flow model based on an operation-time-space network for solving PRSSP has been developed. This model has been tested using random generated instances providing good results. The same network flow model has been extended in order to solve PRSRP and it has been tested using, again, random generated instances providing good results as well.

Finally, both the network and the flow models have been adapted for solving the case study of a port area located in Italy in order to test the applicability of the developed models to a real system. The tests have been performed by using real data and the real infrastructure system as input. These tests provided good results confirming the possibility to apply the proposed approach for solving these problems.

Note that, some systems in the Italian context have been analysed and one of the strenght points of the developed models is that they might be easily extended and adapted to a great number of different layouts thanks to their structure, which is based on the operations to perform instead of on a graph representing their specific physical layout, which would have made the models too rigid to be extended in an easy way.

Currently, pilot tests are in progress. The direction is to build a digital optimization tool for the rail shunting operations scheduling and re-scheduling as decision support system for helping the involved managers and stakeholders to organize this complex and particular process. This tool and its results might help to increase the efficiency of the process by suggesting the optimal operations plan respecting the main constraints of the real process.

This tool could also be used to implement interesting and realistic scenarios analysis in order to understand the variations and the consequences in several cases. These cases may vary from problems or changes in the physical infrastructure, passing through the availability of the resources until the changing in the number of manageble trains and in the rail network/ships schedule.

For what concerns the developed model for solving PRSSP, the computational tests using both random generated instances and real data show that for instances of around 80 trains really good solutions can be found in a range between the 1 and 7 hours depending on the complexity brought by the trains distributions among both the time horizon and the maritime terminals. These models, which are applicable thanks to the fact that the trains schedule is usually done two times per year and some months before, might be improved in terms of computational time by developing a time rolling heuristic approach. Given that, the interval in which both an export and an import train have to be scheduled should be 48 hours at maximum. Developing a time rolling heuristic approach based on the network flow model means to solve the model more times, i.e. one time every 48 hours, keeping in consideration only the trains to schedule in that smaller time horizon. In this way, it should be possible to obtain a good and realistic solution by decreasing the computational time. Currently, this idea is under analysis and the development of the heuristic procedure is in progress.

Another interesting idea is to use the network flow model to solve a more operative problem. It is possible that the trains planned for the a generic day have some changes which are communicated the day before. In this case, being in the day x it is possible to modify the values of the input data accordingly to the changes and solve the scheduling problem with one day as time horizon, i.e. day $x+1$. Thus, in this case, we are in day x and we are planning the day $x+1$. This bring that, instead of having an empty initial state of the system, some trains might be located in the maritime terminals and some other in the shunting park. These trains could have to perform only part of the operations because they have already executed the other parts. For this application, other than reducing the time horizon to one day it is necessary to modify the model allowing to include and consider data related to the system state in addition to the standard data of PRSSP.

Passing to the developed model for solving PRSRP, as showed in the dedicated Chapters, it has really good performance both in terms of applicable solutions in real contexts and computational time. Therefore, no huge improvements are needed. Anyway, given that the objective function is composed by several terms managed with different weights in order to direct the model with a preference order, the analysis already performed on these weights might be either addressed more or extended in order to understand, for example, the eventual correlations among the several terms brought by the model constraints.

References

ABBINK, E.J., ALBINO, L., DOLLEVOET, T., HUISMAN, D., ROUSSADO, J. & SALDANHA, R.L. (2011). Solving large scale crew scheduling problems in practice. *Public Transport*, **3**, 149–164. 22

ACUNA-AGOST, R., MICHELON, P., FEILLET, D. & GUEYE, S. (2011). A mip-based local search method for the railway rescheduling problem. *Networks*, **57**, 69–86. 25

AGARWAL, R. & ERGUN, Ö. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, **42**, 175–196. 46

AHUJA, R.K., CUNHA, C.B. & ŞAHIN, G. (2005). Network models in railroad planning and scheduling. In *Emerging Theory, Methods, and Applications*, 54–101, INFORMS. 34

AMBROSINO, D. & ASTA, V. (2019). Intermodality and rail transport: Focus on port rail shunting operations. In *Advances in Optimization and Decision Science for Society, Services and Enterprises*, 351–366, Springer. 3, 35, 49

AMBROSINO, D. & PEIRANO, L. (2016). Truck arrival management at maritime container terminals. In *ECMS*, 114–120. 49

Ambrosino, D. & Siri, S. (2015). Comparison of solution approaches for the train load planning problem in seaport terminals. *Transportation Research Part E: Logistics and Transportation Review*, **79**, 65–82. 40

Ambrosino, D., Bramardi, A., Pucciano, M., Sacone, S. & Siri, S. (2011). Modeling and solving the train load planning problem in seaport container terminals. In *2011 IEEE International Conference on Automation Science and Engineering*, 208–213, IEEE. 40

Anghinolfi, D. & Paolucci, M. (2014). A general purpose lagrangian heuristic applied to the train loading problem. *Procedia-Social and Behavioral Sciences*, **108**, 37–46. 40

Archetti, C. & Peirano, L. (2020). Air intermodal freight transportation: the freight forwarder service problem. *Omega*, **94**, 102040. 46

Bektas, T. & Crainic, T. (2007). *A brief overview of intermodal transportation*. CIRRELT. 12

Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G. & Weglarz, J. (2019). *Handbook on scheduling*. Springer. 17

Boccia, M., Mannino, C. & Vasilyev, I. (2013). The dispatching problem on multitrack territories: Heuristic approaches based on mixed integer linear programming. *Networks*, **62**, 315–326. 24

Borndörfer, R., Klug, T., Lamorgese, L., Mannino, C., Reuther, M. & Schlechte, T. (2018). *Handbook of Optimization in the Railway Industry*, vol. 268. Springer. 19, 27, 34, 42

Boysen, N., Fliedner, M., Jaehn, F. & Pesch, E. (2012). Shunting yard operations: Theoretical aspects and applications. *European Journal of Operational Research*, **220**, 1–14. 29, 34

Bruns, F. & Knust, S. (2012). Optimized load planning of trains in intermodal transportation. *OR spectrum*, **34**, 511–533. 39

Caballini, C., Pasquale, C., Sacone, S. & Siri, S. (2012). A discrete-time model for optimizing the rail port cycle. In *Proc. of 13th IFAC Symposium on Control in Transportation Systems*, 341–346. 39

Caballini, C., Fioribello, S., Sacone, S. & Siri, S. (2016). An milp optimization problem for sizing port rail networks and planning shunting operations in container terminals. *IEEE Transactions on Automation Science and Engineering*, **13**, 1492–1503. 35, 39

Cacchiani, V. & Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research*, **219**, 727–737. 19

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L. & Wagenaar, J. (2014). An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, **63**, 15–37. 23

Caprara, A., Fischetti, M., Guida, P.L., Toth, P. & Vigo, D. (1999). Solution of large-scale railway crew planning problems: The italian experience. In *Computer-aided transit scheduling*, 1–18, Springer. 22

Caprara, A., Fischetti, M. & Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations research*, **50**, 851–861. 18

CARBONI, A. & DEFLORIO, F. (2020). Simulation of railroad terminal operations and traffic control strategies in critical scenarios. *Transportation Research Procedia*, **45**, 325–332. 30

CARLO, H.J., VIS, I.F. & ROODBERGEN, K.J. (2014). Storage yard operations in container terminals: Literature overview, trends, and research directions. *European journal of operational research*, **235**, 412–430. 38

CHENG, Y.H. & YANG, L.A. (2009). A fuzzy petri nets approach for railway traffic control in case of abnormality: Evidence from taiwan railway system. *Expert systems with applications*, **36**, 8040–8048. 26

CORDEAU, J.F., TOTH, P. & VIGO, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation science*, **32**, 380–404. 34

CORMAN, F., D'ARIANO, A., PACCIARELLI, D. & PRANZO, M. (2009). Evaluation of green wave policy in real-time railway traffic management. *Transportation Research Part C: Emerging Technologies*, **17**, 607–616.

CORMAN, F., D'ARIANO, A. & HANSEN, I.A. (2010a). Disruption handling in large railway networks. *WIT Transactions on the built environment*, **114**, 629–640. 24

CORMAN, F., D'ARIANO, A., PACCIARELLI, D. & PRANZO, M. (2010b). A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological*, **44**, 175–192. 23

CORMAN, F., D'ARIANO, A., HANSEN, I.A., PACCIARELLI, D. & PRANZO, M. (2011). Dispatching trains during seriously disrupted traffic situations. In *2011 International Conference on Networking, Sensing and Control*, 323–328, IEEE. 27

Corman, F., D'Ariano, A., Pacciarelli, D. & Pranzo, M. (2012). Bi-objective conflict detection and resolution in railway traffic management. *Transportation Research Part C: Emerging Technologies*, **20**, 79–94. 26

Dal Sasso, V., Lamorgese, L., Mannino, C., Onofri, A. & Ventura, P. (2021). The tick formulation for deadlock detection and avoidance in railways traffic control. *Journal of Rail Transport Planning & Management*, **17**, 100239. 27, 42

D'ariano, A., Pacciarelli, D. & Pranzo, M. (2007). A branch and bound algorithm for scheduling trains in a railway network. *European journal of operational research*, **183**, 643–657. 17

D'Ariano, A., Corman, F., Pacciarelli, D. & Pranzo, M. (2008a). Reordering and local rerouting strategies to manage train traffic in real time. *Transportation science*, **42**, 405–419. 24

D'Ariano, A., Pacciarelli, D. & Pranzo, M. (2008b). Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research Part C: Emerging Technologies*, **16**, 232–245. 24

Dotoli, M., Epicoco, N., Falagario, M., Seatzu, C. & Turchiano, B. (2016). A decision support system for optimizing operations at intermodal railroad terminals. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **47**, 487–501. 30

Fioribello, S., Caballini, C., Sacone, S. & Siri, S. (2016). A planning approach for sizing the capacity of a port rail system: scenario analysis applied to la spezia port network. *IFAC-PapersOnLine*, **49**, 371–376. 35

Gestrelius, S., Aronsson, M. & Peterson, A. (2017). A milp-based heuristic for a commercial train timetabling problem. *Transportation Research Procedia*, **27**, 569–576. 20

Gillen, D. & Hasheminia, H. (2018). Empirical analysis and simulation modeling of a canadian seaport transportation network. *Journal of Supply Chain and Operations Management*, **16**, 17. 39

Guan, Y., Yang, K.H. & Zhou, Z. (2013). The crane scheduling problem: models and solution approaches. *Annals of Operations Research*, **203**, 119–139. 46

Guerriero, F. & Talarico, L. (2010). A solution approach to find the critical path in a time-constrained activity network. *Computers & operations research*, **37**, 1557–1569. 54

Hansen, I.A. & Pachl, J. (2014). Railway timetabling & operations. *Eurailpress, Hamburg*. 19

Heggen, H., Braekers, K. & Caris, A. (2016). Optimizing train load planning: review and decision support for train planners. In *International Conference on Computational Logistics*, 193–208, Springer. 39

Hoogervorst, R., Dollevoet, T., Maróti, G. & Huisman, D. (2020). Reducing passenger delays by rolling stock rescheduling. *Transportation Science*, **54**, 762–784. 21

Huisman, D. (2007). A column generation approach for the rail crew re-scheduling problem. *European Journal of Operational Research*, **180**, 163–173. 22

IANNONE, F. (2012). A model optimizing the port-hinterland logistics of containers: The case of the campania region in southern italy. *Maritime Economics & Logistics*, **14**, 33–72. 14

KECMAN, P., CORMAN, F., D'ARIANO, A. & GOVERDE, R.M. (2013). Rescheduling models for railway traffic management in large-scale networks. *Public Transport*, **5**, 95–123. 25

KRÄMER, I. (2019). Shunt-e 4.0—autonomous zero emission shunting processes in port and hinterland railway operations. *Journal of Traffic and Transportation Engineering*, **7**, 157–164. 30, 31

KRASEMANN, J.T. (2012). Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. *Transportation Research Part C: Emerging Technologies*, **20**, 62–78. 25

KROON, L., MARÓTI, G. & NIELSEN, L. (2015). Rescheduling of railway rolling stock with dynamic passenger flows. *Transportation Science*, **49**, 165–184. 21

LI, P., MIRCHANDANI, P. & ZHOU, X. (2015). Solving simultaneous route guidance and traffic signal optimization problem using space-phase-time hypernetwork. *Transportation Research Part B: Methodological*, **81**, 103–130. 48

LIU, J. & ZHOU, X. (2016). Capacitated transit service network design with boundedly rational agents. *Transportation Research Part B: Methodological*, **93**, 225–250. 48

LIU, J. & ZHOU, X. (2019). Observability quantification of public transportation systems with heterogeneous data sources: An information-space projection approach based on discretized space-time network flow models. *Transportation Research Part B: Methodological*, **128**, 302–323. 48

Lu, C.C. (2016). Robust multi-period fleet allocation models for bike-sharing systems. *Networks and Spatial Economics*, **16**, 61–82. 48

Lu, G., Zhou, X., Mahmoudi, M., Shi, T. & Peng, Q. (2019). Optimizing resource recharging location-routing plans: A resource-space-time network modeling framework for railway locomotive refueling applications. *Computers & Industrial Engineering*, **127**, 1241–1258. 47

Lusby, R.M., Haahr, J.T., Larsen, J. & Pisinger, D. (2017). A branch-and-price algorithm for railway rolling stock rescheduling. *Transportation Research Part B: Methodological*, **99**, 228–250. 21

Mahmoudi, M. & Zhou, X. (2016). Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. *Transportation Research Part B: Methodological*, **89**, 19–42. 47

Mantovani, S., Morganti, G., Umang, N., Crainic, T.G., Frejinger, E. & Larsen, E. (2018). The load planning problem for double-stack intermodal trains. *European Journal of Operational Research*, **267**, 107–119. 40

Murakami, K. (2020). Time-space network model and milp formulation of the conflict-free routing problem of a capacitated agv system. *Computers & Industrial Engineering*, **141**, 106270. 47

Narayanaswami, S. & Rangaraj, N. (2013). Modelling disruptions and resolving conflicts optimally in a railway schedule. *Computers & Industrial Engineering*, **64**, 469–481. 27

Nielsen, L.K. (2011). *Rolling Stock Rescheduling in Passenger Railways: Applications in short-term planning and in disruption management*. EPS-2011-224-LIS. 21

POTTHOFF, D., HUISMAN, D. & DESAULNIERS, G. (2010). Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science*, **44**, 493–505. 22

RODRIGUEZ, J. & KERMAD, L. (1998). Constraint programming for real-time train circulation management problems in railway nodes. *WIT Transactions on The Built Environment*, **37**. 26

RUSCA, A., POPA, M., ROSCA, E., ROSCA, M., DRAGU, V. & RUSCA, F. (2016). Simulation model for port shunting yards. In *IOP Conference Series: Materials Science and Engineering*, vol. 145, 082003, IOP Publishing. 33

RUSCA, A., RUSCA, F., ROSCA, E., DRAGU, V. & ROSCA, M. (2018). Improving capacity of port shunting yard. In *Progress in Maritime Technology and Engineering*, 35–42, CRC Press. 33

RUSCA, F., POPA, M., ROSCA, E., RUSCA, A., ROSCA, M. & DINU, O. (2019). Assessing the transit capacity of port shunting yards through discrete simulation. *Transport Problems: an International Scientific Journal*, **14**. 33

SCHÖNEMANN, R. (2010). Integrating railway services into the supply chain at the last mile of the transshipment interface seaport-rail. *Logistyka*, CD–CD. 39

STAHLBOCK, R. & VOSS, S. (2008). Operations research at container terminals: a literature update. *OR spectrum*, **30**, 1–52. 37

STEENKEN, D., VOSS, S. & STAHLBOCK, R. (2004). Container terminal operation and operations research-a classification and literature review. *OR spectrum*, **26**, 3–49. 37, 39

Tomii, N. & Zhou, L.J. (2000). Depot shunting scheduling using combined genetic algorithm and pert. *WIT Transactions on The Built Environment*, **50**. 35, 37, 41

Tomii, N., Zhou, L.J. & Fukumura, N. (1999). An algorithm for station shunting scheduling problems combining probabilistic local search and pert. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 788–797, Springer. 35, 36, 37, 41

Tong, L., Zhou, X. & Miller, H.J. (2015). Transportation network design for maximizing space–time accessibility. *Transportation Research Part B: Methodological*, **81**, 555–576. 48

Veelenturf, L.P., Potthoff, D., Huisman, D. & Kroon, L.G. (2012). Railway crew rescheduling with retiming. *Transportation research part C: emerging technologies*, **20**, 95–110. 22, 28

Walker, C.G., Snowdon, J.N. & Ryan, D.M. (2005). Simultaneous disruption recovery of a train timetable and crew roster in real time. *Computers & Operations Research*, **32**, 2077–2094. 28

Xie, Y. & Song, D.P. (2018). Optimal planning for container prestaging, discharging, and loading processes at seaport rail terminals with uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, **119**, 88–109. 39, 40

Yang, L., Di, Z., Dessouky, M.M., Gao, Z. & Shi, J. (2020). Collaborative optimization of last-train timetables with accessibility: A space-time network design based approach. *Transportation Research Part C: Emerging Technologies*, **114**, 572–597. 48

Yuan, Y. & Tang, L. (2017). Novel time-space network flow formulation and approximate dynamic programming approach for the crane scheduling in a coil warehouse. *European Journal of Operational Research*, **262**, 424–437. 46, 47

Zehendner, E. & Feillet, D. (2014). Benefits of a truck appointment system on the service quality of inland transport modes at a multimodal container terminal. *European Journal of Operational Research*, **235**, 461–469. 49