Doctor of Philosophy in
Computer Science and Automation
Roma Tre University
Department of Engineering

XXV Cycle

Rolling stock rostering and maintenance scheduling optimization

Giovanni Luca Giacco
PhD student

____
Signature

Prof. Dario Pacciarelli
Tutor

____
Signature

I would like to thank my advisor, Prof. Dario Pacciarelli, for his inspirational and timely advice and constant encouragement over the last several years. I have learned a great deal from his unique perspective on research.

I would like to express my special appreciation and thanks to Ing. Donato Carillo, Dr. Ruggero Gianfaldoni and Prof. Andrea D'Ariano who have supported my work and encouraged me through the writing of this dissertation. Without their time, attention, encouragement, thoughtful feedback, I would not have been able to grow as a research scientist.

Last but not least, I would like to thank my wife Manuela and my sons Simone and Giorgia for their love and constant support, for all the late nights and early mornings. Thank you for being my main editors and proofreaders. But most of all, thank you for being my best friends. I owe you everything.

Abstract

This thesis addresses identification and analysis of frameworks for optimizing medium-term maintenance planning and rolling stock rostering.

Rolling Stock Management (RSM) is the main cost factor for Rail Undertakings. For example, for high-speed trains, more than 30% of the lifecycle costs is spent for maintenance operations. In order to reduce the costs due to railway operations, every company should address the joint problem of rolling stock rostering and maintenance scheduling since they are strongly related parts of the same problem. Maintenance optimization can be a key factor to increase the productivity of railway companies. At the same time, in a competitive globalized and multimodal market, RSM is one of the competitiveness key factors because services quality level depends on it. The strategic relevance of RSM, in particular of maintenance scheduling, is thus due to the reduction of needs (such as platforms and human resources) and to the enhancement of quality standards (such as vehicle reliability and cleaning). From our point of view the literature is focused on manufacturing setting in order to reduce the occurrence of a failure while unfortunately the coordination of maintenance and rolling stock scheduling is still underinvestigated.

A key problem in railway planning process requires to cover a given set of services and maintenance works with a minimum amount of rolling stock units. Additional objectives are to minimize the number of empty runs and to maximize the kilometres travelled by each train between two maintenance operations of the same type. First,

the rostering and maintenance optimization problems are formulated by graph theoretical approaches that involve medium-term maintenance operations, the scheduling tasks related to train services and empty rides. The constraints of the maintenance optimization problem require that the different types of maintenance operations must be carried out for each train periodically. The various maintenance tasks can only be done at a limited number of dedicated sites. Starting from the solutions of the rostering and maintenance optimization problems, we developed another graph theoretical approach to optimize workshop management and in particular to minimize the number of drivers involved and to verify the feasibility of the maintenance plan at each site. For a set of timetables and rolling stock categories, we compare flexible versus rigid plans regarding the number of empty rides and maintenance kilometres.

For different feasible frameworks and different kinds of timetables, we provide new mixed-integer linear-programming formulations for train rostering and maintenance scheduling problems and we also show how the proposed scheduling formulations could be used as effective tools to absorb real-time timetable perturbations while respecting the agreed level of service.

The specific objective of our research is to related to the following questions: “How can the timetable be executed by an efficient use of resources such that the overall railway company costs are reduced? Which is the maximal improvement that can be achieved? At which cost?".

In this thesis, we give an answer to these questions by performing an assessment of key performance indicators.

The computational evaluation presents the efficiency of the new solutions compared to the practical solutions. Experimental results on real-world scenarios from Trenitalia show that these integrated approaches can reduce significantly the number of trains and empty rides when compared with the current plan. We use a commercial MIP solver for developing a decision support tool that computes efficient solutions in a short time.

Contents

Introduction

In the second half of twentieth century, the role played by railway in the transport market has been in constant decline (see Figure 1 taken from [36]). The foremost reason is dissatisfaction with the price and quality of rail transport. In other words, railroad does not respond to market changes or customers' needs such as other modes. The rise of other more flexible and less expensive transport (such as buses, lorries and the private cars) has motivated increasingly people and businesses to road transport.
The railways did not find new freight and passengers markets to compensate for this loss even in the sectors where they could be competitive. Moreover railroads costs are often too high and therefore the conditions of competition favor road and other more environmentally friendly modes [73]. These changes urge the railway undertakings to attract their customers by raising their service level and to cut their costs by improving their planning process.
A main challenge of railway undertakings is to reduce the overall cost of operations by means of a more efficient use of rolling stock and crew resources. The basic wishes of the modern railway customer are fairly simple: he wants to travel fast and comfortably for a reasonable price. Often the rail networks are not very adapted to new patterns of economic activity and urbanization and to the consequent changes in traffic flows. To turn the decline, the European commission increased focus on the general area of European railway transportation in the middle of the nineties. Exactly in 1996 the "Strategy for Revitalizing the Community's Railways" was composed with the aim to encourage development of railway as much

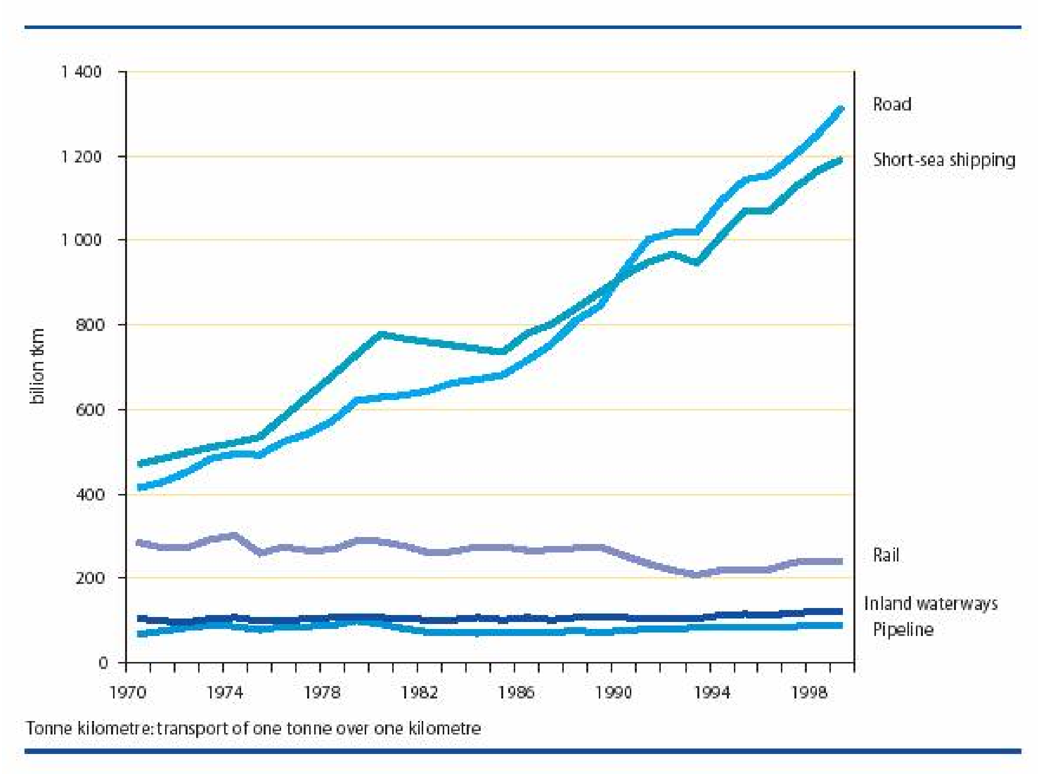

Figure 1: Railway market trend

as possible. The goal was to establish the concept of free competition and of free access to railway infrastructure.
In Italy the process of improving of free competition began in 1998 with specification that the infrastructure management should be a responsibility of the governments, but operating trains should be carried out by independent Train Operators on a commercial basis. In particular this break has lead some interesting advantages but on the other hand has lead a lot of losses too. The railway planning process should be managed together by Manager of Infrastructure and Train Operators with the aim of a global optimum. In fact Train Timetabling, Train Platforming, Rolling Stock Circulation, Train Maintenance Planning and Rolling Stock Scheduling are just some examples of activities that should be managed by both jointly.
Many European Projects have focused on modelling and solving problems arising in this field to reduce losses, to lead to an im-

provement of performance and to increase the flexibility of the railway system. The concept is that OR can help Railway industry to adapt in a faster way to changes in the environment.
The operating design process of a national railroad is an enormously complicated task. Often a huge set of variables must be taken into account to determine the lines to serve, to generate a timetable, to plan the movements of the rolling stock, to manage the crew, etc. The use of Operations Research models and solution techniques allows one to obtain fast, close to optimality and effective solutions to the problems with a consequent better use of the Railway resources.

Chapter 1

Operations Research in Railway Planning

The railway industry is a huge source of problems that can be modelled and solved by using Operations Research techniques (see [76] and [94]).
Many of these are still handled without automation and optimisation. Such problems exist in several forms and arise at different levels in the planning process for a railway company. The complete railway system managing is highly complex and it is often divided into a lot of sub-problems which are interconnected. Given an objective to achieve, a very difficult task is to understand what problems are involved and how they are related.
Very interesting surveys on Operations Research and Railway planning are by Cordeau et al. [29], Assad [9], Huisman et al. [52], Oum et al. [70] and Zhou [92].
We will report interesting publications found in literature addressing the main transportation planning problems.

1.1 Train Timetabling

One of the most important process planning steps for a train operator company is called *Network Planning*. The aim of this step is to coordinate different kinds of transports to achieve a network effect in order to maximize efficiency of services provided. Planning multimodal transport systems as seamless integrated networks rather than as a series of individual routes is a very critical task. *Network Planning*, that could be classified as long-term task, is followed by a mid-term task called *Train Timetabling Problem* (TTP) that consists in finding a train schedule on a given railway network maximizing company goals. Train Timetabling is often a manual process based on experienced planners. The general aim of the Train Timetabling Problem is to determine an optimal timetable satisfying a set of customers' needs and a set of operational constraints due firstly to safety rules. To increase services quality, timetables are also required to be robust against delays/disturbances along the network. In the literature appear a cyclic version of the problem and a non-cyclic version too. In a cyclic timetable, train arrival and departure at station are operated regularly with respect to a cycle time. In 1931, the Dutch company NS presented the cyclic Railway timetable concept, with a cycle time of one hour. Since the introduction of cyclic timetables in the Netherlands, many other European countries have adopted this concept in long distance trains too. So, each train leaves at the same time every cycle time (e.g. one day or one hour). Cyclic timetables are mainly used for passenger Railways (in fact cargo rail schedules are often non-cyclic). This is a great advantage for the passengers (in particular for commuters) who have not to remember difficult timetables because the cyclic timetables can be represented compactly. From a planning point of view, cyclic timetables have the advantage that we have to focus on only one cycle period. However, it produces higher costs and wastes. Moreover in a cyclic timetable, a Railway operator does not fine-tune the timetable to the demand for transportation. Serafini and Ukovich in [77] introduced in 1989 the Periodic Event Scheduling Problem (PESP). The PESP considers the problem of scheduling a set of periodically recurring events under periodic time window constraints. More specifically, the PESP aims at determining the time instants at which the periodic events have to take place. Unfortunately, the basic PESP only finds a feasible schedule. In fact, it does not use an objective function. Nachtigall [68] also used PESP constraints to model the cyclic behaviour of Railway timetables. He introduced an objective function based on

- cost of infrastructure;
- benefit of improving the synchronization in the timetable;
- passenger waiting time.

Nachtigall and Voget [68] also considered the problem of minimizing passenger waiting times in a cyclic Railway timetable but they started from an initial timetable that improved using a genetic algorithm. In 1997 they also used a bi-criteria approach taking into account the infrastructure investments and passenger waiting time. Weigand [90] did not use a PESP model. His solution method picks a spanning tree of the graph induced by the arrival and departure times of the trains, and computes an optimal timetable for that tree. Then, the algorithm iteratively moves to a new spanning tree in order to improve the best found timetable. Lichtenegger [59] considered the problem of the integrated fixed interval timetable as a special type of cyclic timetable. The idea was that at these time instants passengers can change trains, since most trains are present at a station. Lichtenegger would to minimize the infrastructure investments. The final result is a mixed integer program. Sherali [79] presented a non proportional assignment linear programming model for estimating origin-destination (O-D) trip tables from available data. The formulation is made to determine a traffic equilibrium network flow solution.
Goverde and Koelemeijer [47] considered the problem of evaluating the performance of a cyclic Railway timetable. They made some

performance indicators such as the critical circuits in the Railway network, the stability margin of the timetable, and the propagation of delays in case of a disruption. Kroon, Maróti et al. [56] have described a Stochastic Optimization Model that can be used to allocate the time supplements and the buffer times in a given timetable in such a way that the timetable becomes maximally robust against stochastic disturbances of cyclic Railway timetables. The non-cyclic timetabling is relevant for scenarios with a competitive environment among Train Operators. Wendler presented [91] an approach predicting the scheduled waiting time by means of a semi-Markovian queueing model. The process of timetable compilation in a Railway network with open access is shortly explained and described by means of queueing theory. The arrival process is determined by the requested train-paths. The description of the service process is based on an application of the theory of blocking times and minimum headway times. The approach is useful for predicting a quality measure for bottlenecks with mainly non-cyclic timetable structures. Ralf Borndörfer [13] discussed an auctioning approach to establish a rail track market, in which different train operating companies compete for tracks and timeslots on the same network. Cacchiani, Caprara and Toth [18] considered a new formulation of non-cyclic train timetabling with the goal of maximization of the profit of a collection of compatible paths in a suitable graph. Khan and Zhou [54] studied possible decompositions for robust timetabling problem into a series of sub problems that optimize the slack-time allocation for individual trains.

1.2 Train Platforming

The Train Platforming deals with the decision of the routes that each scheduled train must follow when entering, stopping and then leaving a station. In particular, each train has to perform a path from the point where it enters the station to the point where it leaves the station. This path passes through a platform. Given the rail network, a set of trains, with the corresponding timetable, and a set of platforms where each train can stop, the problem determines the optimal routing of the trains, avoiding incompatible routes (overlaps). This train platforming problem is a key problem in Railway station operations, and for a large station, many working days are required for an expert planner to construct the train-platforming.

Carey and Carville [25] considered this problem for complex train stations. They developed the constraints and objectives for platforming problem without finding a way to solve it by standard combinatorial search or integer programming methods. They also developed a scheduling heuristics analogous to those successfully adopted by train planners.

Caprara, Galli and Toth [21] have studied a general formulation of the train platforming problem. They consider a general quadratic objective function, and propose a new way to linearize it by using a small number of new variables along with a set of constraints that can be separated efficiently by solving an appropriate linear program. The resulting integer linear programming formulation has a continuous relaxation that leads to strong bounds on the optimal value. De Luca Cardillo and Mione [33] formulated a graph-coloring problem. These authors proposed to solve it by an efficient heuristic algorithm combined with reduction techniques. Billionnet [11] proposed an integer programming technique to solve the coloring problem. This technique can provide an exact solution of the problem and it is easy to implement and to adapt to take into account additional constraints and different measure criteria.

1.3 Rolling Stock Rostering

Given the departure and arrival times as well as the expected numbers of passengers, Rolling Stock Circulation deals with the assignment of locomotives and carriages to the timetable services. We could consider several objective criteria that are related to operational costs, service quality and reliability of the Railway system. The extended model can also handle combining and splitting of trains. The problem calls for determining for each trip the locomotive types and their number, and the carriage types and their number. These numbers depend on the locomotives power. This problem is related with the maintenance problem. In fact in a sparse network it is necessary to take into account the maintenance operations and build a schedule for them; on the contrary, in a dense network, the maintenance operations can be handled easily. Ramani and Mandal [74] developed an optimization-based decision support system that aims at minimizing fleet size. Locomotives and cars are treated separately and equipment switching is not considered. Bussieck et al. (1997) [17] surveyed mathematical programming methods for public rail transport planning. The authors state that the problem of assigning rolling stock to a set of scheduled trains can be formulated as a multi-depot vehicle scheduling problem and present a review of several papers on that subject. Cordeau et al. [29] developed an optimization system to solve the equipment cycling problem. This model is solved by a column generation approach embedded in a branch-and-bound search. Schrijver et al. [39] focused on to determine the rolling stock circulation for a generic week. They deal with an extension of the problem described by Peeters and Kroon [71] who used a branch-and-price approach. Fu et al.[41] proposed some models and algorithms to optimize passenger train departure and arrival time windows in scheduling. The main objectives to be optimized include minimizing the total passenger inconvenience and the number of passenger cars needed to operate trains.

1.4 Train Maintenance Planning

Train Maintenance Planning is the biggest cost factor for Rail Undertakings. For example, for high-speed trains, more than 30% of the lifecycle costs is spent for maintenance operations. To increase the productivity, railway companies are putting a great deal of effort into maintenance optimization.

The literature is too focused on manufacturing setting in order to reduce the occurrence of a failure while unfortunately the coordination of maintenance and rolling stock scheduling is still under investigated.

In view to the literature reviews by Ahuja et al. [2], Cordeau et al. [29], Garg et al. [42], Higgins [50], Sriskandarajah et al. [83], Waeyenbergh [87], Anily et al. [6], Anily et al. [7] and Silver et al. [80], we limit ourselves to review the following recent papers.

Penicka et al. [72] introduced a formal model of the train maintenance routing problem. However, the rostering problem with maintenance constraints and the maintenance routing are not addressed in this paper. Furthermore, in their case study rostering and rolling stock maintenance are considered as an appropriate method for long-distance trains only. Mároti et al. [65] presented a mixed integer formulation for the maintenance routing problem in which the shunting process is considered the process bottleneck.

Budai et al. [16] discussed the preventive maintenance sche-duling problem and the minimization of the time required for performing maintenance operations. The heuristic algorithms compute nearly optimal solutions by combining maintenance tasks on each track.

Wang [88] proposed a multiple criteria decision-making problem and evaluated maintenance strategies for different equipments.

The Maintenance Management is a very important function of industries and service organizations. Preventive maintenance is undertaken to keep equipment in a specified condition. Therefore, the scheduling of maintenance activities is an important topic since smart scheduling will reduce the overall maintenance budget. Maintenance is the activities required to conserve as nearly and as long as possible the original condition or resource while compensating for normal wear and tear of the rolling stock asset or of the railroad infrastructure. Maintenance is necessary in order to assure Safety Comfort Cleaning

Management optimizing could be helpful to increase productivity and product or service quality to be competitive in the global market place. The problem is formulated as an integer program and a branch and bound algorithm is used for its resolution. Sriskandarajah et al. [83] developed a Genetic algorithm (GA) for the optimization of maintenance overhaul scheduling of rolling stock.

1.5 Train Unit Shunting

Outside the rush hours there is a surplus of rolling stock. This idle rolling stock is parked at a shunt yard (for example during the night). Shunting of passenger train units is a highly combinatorial optimization problem, in which the resources time, infrastructure, and crews have to be utilized as efficiently as possible. So besides the rolling stock circulation, which determines the operating phase during the rush hours, the train unit shunting determines the parking phase of the rolling stock during the night hours. Train Unit Shunting Problem (TUSP) concerns the routing of the rolling stock between the station area and the shunting area, the short term maintenance and the inside and outside cleaning. It determines a matching of the arriving train units that have to be parked on a shunt yard with the departing train units, such that the overall cost is minimized. Generally we have some foremost subproblems:

- Routing: train units have to be routed from their arrival track to a shunt track and vice versa.
- Parking: train units have to be parked at a shunt track in such a way that they do not block each other's arrival or departure.
- Crew Scheduling: each shunting movement must be carried out by a train driver.

The objective is to make sure that the Railway processes can start up while the total number of shunting movements is as small as possible. Schrijver [57] described a new model for the Train Unit Shunting Problem. This model is capable of solving the matching and parking subproblems in an integrated manner, requiring a reasonable amount of computation time. Nonner and Souza [69], derived polynomial time algorithms for Train Shunting by reducing the problem to finding independent sets in bipartite graph but the computation time was not very good. Cordeau et al. [29] provided a recent overview of the use of Operations Research in Railway systems focusing on train routing and scheduling problems. Freling et al. [40] solved separately the matching and parking subproblems, nevertheless resulting in solutions with high quality. Some special cases of the TUSP were dealt by Blasum et al. [12] for dispatching trams in a depot. Other authors theoretically extended this approach with length restrictions and mixed arrivals and departures. Moreover, he also discussed an application to a bus depot, including computational results. Di Stefano [28] studied the computational complexity of several variants of sub-problems of the TUSP. Furthermore, he also presented algorithms for solving some of these subproblems, including bounds on the objectives and on the complexity of the algorithms. Hamdouni et al. [34] described a solution for buses TUSP. They had little different types of buses as possible in one lane, and within one lane they grouped together the buses of the same type as much as possible. Tomii et al. [85] and Tomii and Zhou [86] proposed a genetic algorithm that takes into account some related processes of TUSP. However, their parking problem was of a less complex nature, since in their context at most one train unit could be parked on a shunt track at the same time. Lubbecke and Zimmermann [62] discussed a related problem that arises at an in-plant private freight railroad. In this problem, one assigns transportation requests to certain regions of the in-plant railroad and selects cars of specific types from a shunt track in this region for servicing a specific request. He et al. [78] discussed the problem TUSP with the separation of train units from arriving freight trains, sorting these according to their destination and finally combining them to form new departing trains, which resembles the matching of arriving and departing shunt units without parking. Dahlhaus et al. [31] showed that this problem is NP-hard. However, the authors only discussed LIFO tracks and assumed that there was no prescribed order of different types of cars in a train and assumed that there were no limitations for the temporary parking of cars, when these were not servicing a request.

1.6 Integrated approaches

The tasks discussed above are mainly solved separately. So all relations and integrations between them are often ignored. We would like to focus on this integration. In particular we would like to optimize rail resource in according with train overhauls, timetables, station layout and Railway network. Assigning locomotives and cars to a set of scheduled trains is a complex but important problem for passenger Railways. But we have to take into account that each asset unit requires a preventive maintenance check after a certain number of kilometers or hour. This limit depends on rolling stock type. Lingaya and Cordeau [60] described a model and solution methodology for the car assignment problem that satisfy all operational constraints. Ziarati et al. [93] set up a large-scale integer programming model for locomotive assignment where maintenance routing plays an important role. Lingaya et al. described a model for supporting the operational management of locomotive-hauled Railway cars. They seeked for a maximum expected profit schedule that satisfies various constraints, among them also maintenance requirements. Anderegg et al. [5] addressed a realistic rolling stock rostering problem with maintenance constraints based on cyclic timetable assumption with the goal to minimize the general costs. These models consider maintenance routing as a part of the medium and long term vehicle scheduling problem. Small safety inspections form an important part of the problem specification and also the larger-scale maintenance checks were taken into account. These models compute the vehicle circulation for the forthcoming month such that each vehicle is scheduled for maintenance exactly once. Also, the vehicle schedules were created without taking shunting into account. Therefore, in a Railway application there is just a small probability that the output of the models in the literature can be carried out in practice.

Chapter 2

Railway in Europe

The majority of Europeans (71%) support opening the national rail to competition. The gradual liberalization of the railway market in the 1990s led a huge change in the organizational structure of some railways. Separation of railway infrastructure and operations laid the foundations for introduction of competition to railways as well as for their economic rationalization. European Directives 91/440/EEC, 95/18/EC, 95/19/EC specify the necessity for separate accounting of infrastructure and operations. The legislation is focused on the distinction between infrastructure managers who run the network and the railway companies that use it for transporting passengers or goods.
These directives do not specify how the infrastructure and operations must be separated. For this reason, Member States have adopted two different methods to achieve the required separation:

- Institutional separation
- Organizational separation

Institutional separation is the split of infrastructure manager and railway undertaking into autonomous entities (capitalization, staff and asset are separated). The infrastructure owner can be publicly owned as in Portugal (Portuguese Rail Infrastructure Authority (REFER)) and Sweden (Banverket (BV)) or privately owned as in the UK (Railtrack). However, a series of fatal accidents and serious infrastructure problems in the UK suggest that private ownership of railway infrastructure could be not a good idea. In France the infrastructure manager (RFF) and operator (SNCF) are completely separate legal entities with separate staff, but the relationship is closest because SNCF is a train operator and maintains the infrastructure based on contracts awarded from RFF. In Finland, the infrastructure manager is the Finnish Rail Administration (RHK), a department of the Ministry of Transport and Communications.

Organizational separation splits business units with a large degree of operational freedom. There are two basic patterns:

- Business units operating as part of railway operator:
  This method is used by Belgian National Railways (SNCB/ NMBS). The units have an independent management and a separate balance sheet but no legal autonomy.
- Autonomous business units organized within framework of holding firm:
  This method is used in Italy (Trenitalia, RFI, Italferr, . . .) under the holding company FS Holding and in Germany (DB Reise & Touristik, DB Regio, DB Cargo, DB Netz, DB Station & Service) under the holding company DB AG.

2.1 Ferrovie dello Stato Italiane

A very important milestone in Italy unification process was the establishment of “Ferrovie dello Stato" on April, 21, 1905. This company has contributed to Italy’s social and cultural development.
Due to the changing economic and political conditions of the last twenty years, there has been a real revolution in the organizational structure of Railways. Companies must meet the customers’ needs (such as safe, punctuality and price) but on the other hand have to follow the directives.
In 2000, in accordance with the dictates of the Community Directives on liberalization of rail transport is kicked off a new profound change.
On June, 7, 2000 was formed Trenitalia SpA, which was entrusted with the transport of goods and passengers. On April, 9, 2001 "Rete Ferroviaria Italiana" was established with the task of manage the main italian railway network and stations.
Today FS Group is the biggest company in Italy. About 72.000 people work hardly to operate more than 8.000 trains every day, to manage a network of 16.700 kilometres, to transport about 600 million passengers and 50 million tons of freight every year.
The current organisation is that of an industrial Group with an holding company, “Ferrovie dello Stato Italiane SpA", which heads the Operating Companies having their own specific corporate character and benefit from managerial independence in achieving business objectives.
Trenitalia is the main italian train operator that manages railway transport of passengers and freight in Italy and abroad. This company manages mainly railway transport services for its customers while contributing to the development of a great project of mobility and logistics for Europe.
“Ferrovie dello Stato Italiane SpA" creates and manages for its own clients works and services in railway transport, helping to develop a great project for Italy mobility and logistics. At the same time,

technological innovation and safety are this group's distinctive characters, attested by two excellent international awards.
These companies are not resting on the italian edge. Mediterranean, Middle East, Eastern Europe, the Balkans, Latin America, United States, India and Australia are the main areas in which they operate, exporting the excellence of Made in Italy for the development of rail networks in those countries. The main services offered on the international market comprises engineering, management of passenger transportation and logistics, infrastructure and stations.

Chapter 3

The Problem

In this thesis we describe how we have solved the Rolling Stock Management problem (RSM) by using Operations Research techniques. RSM is the main cost factor for Rail Undertakings. For example, for high-speed trains, more than 30% of the lifecycle costs is spent for maintenance operations (a very important part of RSM). In order to reduce the costs due to railway operations, every company should address the joint problem of rolling stock rostering and maintenance scheduling since they are strongly related parts of the same problem. Rolling Stock Management, and in particular maintenance optimization, is a very important key factor to increase the productivity of railway companies. At the same time, in a competitive globalized and multimodal market, RSM is one of the competitiveness key factors because services quality level depends on it. The strategic relevance of RSM is thus due to the reduction of needs (such as platforms and human resources) and to the enhancement of quality standards (such as vehicle reliability and cleaning).

The problem addressed in this thesis can be defined as follows: given timetables, rolling stock assets, maintenance workshops and maintenance operations, a rolling stock circulation solution has to be computed with minimum cost, that is expressed in terms of the number of used rolling stock units, the number of used empty runs, the number of train movements between platforms at sta-

tions and within workshops.
This problem involves generally different organizational units and different companies that try to solve their own problem partly manually and using the experience of experts. Very often the solutions carried out are not completely feasible because the task to find a feasible solution could be very hard (there are too many constraints to take into account). In practice, to control the huge complexity of the problem, even if fragmented, a resources priority list is made. This list is used to divide further the problem in a lot of smaller problems. Rail companies tend to solve firstly sub-problems affecting resources with a long time of acquisition (e.g. rail tracks or rolling stock) and secondly sub-problems involving resources with higher degree of flexibility (e.g. human resources). This decomposition on one hand can help to solve a part of problem but on the other hand leads to big wastes.

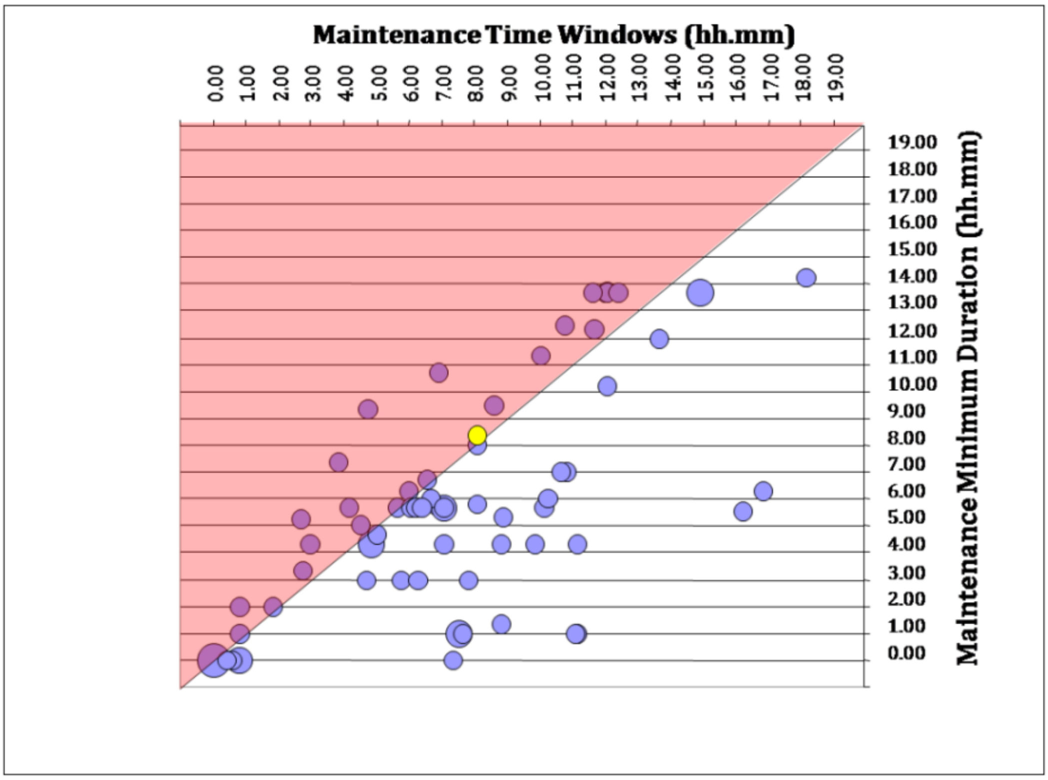

Figure 3.1: Maintenance Time Window vs Maintenance Minimum Duration

For example, Figure 3.1 compares the results of rolling stock rostering with the result of maintenance management for a given workday.
Above the diagonal (red triangle) there are trains, represented by circles, with maintenance minimum duration greater than allocated (by roster) maintenance time window while below the diagonal (white triangle) there are trains with maintenance minimum duration shorter than maintenance time window. In the first case workshop can not provide all maintenance activities because there is no enough time.
Can we say that the roster is not feasible?
Let's decompose, as shown in the Figure 3.2, the time given by roster between two train services in

- waiting time at the passenger station before the train is brought to workshop
- time to provide empty run to workshop
- maintenance processing time window (including recovery time)
- time to provide empty run to passenger station
- waiting time at the passenger station after the train leaves the workshop

Figure 3.2: RSM process

Therefore, the maintenance time window could not be enough for instance because the *Infrastructure Manager* (IM) takes the train long time in the station (IM could have an important role in the rolling stock management) and a short time is consequentially allowed to maintenance tasks. In this case we have a feasible roster, a

feasible station capacity allocation but an unfeasible maintenance plan.
These problems are often overcome enforcing some element of solution but the resulting wastes are not neglectable. Integrated approaches do not have problems like this.
Figures 3.3 and 3.4 represent waiting time in the passenger station before/after the train is moved to/from workshop (waiting times on x-axis and number of trains with that time leg on the y-axis).
Let' s see the case of the yellow train in the Figure 3.1. In this case it is not possible to complete the maintenance process but the Figures 3.3 and 3.4 (see yellow bars) suggest we could try to reduce the waiting times in the station and increase maintenance time. This is a typical kind of waste due to overall problem fragmentation.

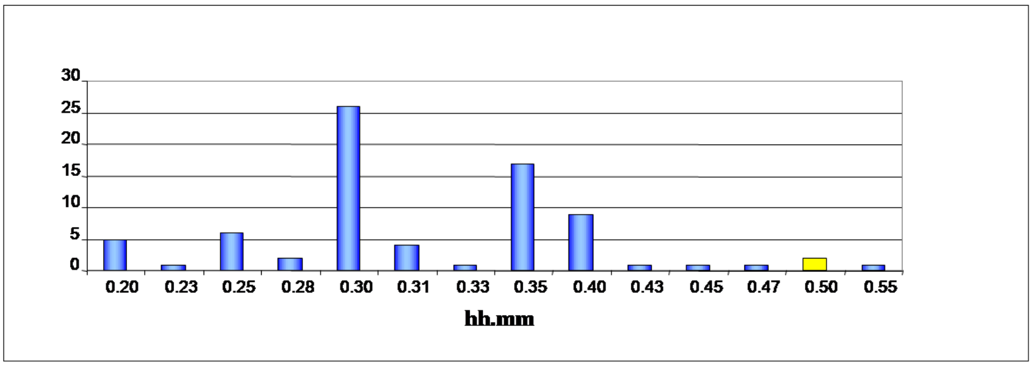

Figure 3.3: Waiting Time in the passenger station before the train is moved to workshop

Another typical example of waste happens when a passenger station can not recover a rolling stock and the time windows is not enough to move it to the workshop. In this case a big amount of money is spent to take the train running on the network until its departure time. This case is very frequent when the time leg in the passenger station is not very long (see Figure 3.5 representing waiting time on passenger station track and the number of train involved) and if there is no coordination between roster and station plans.

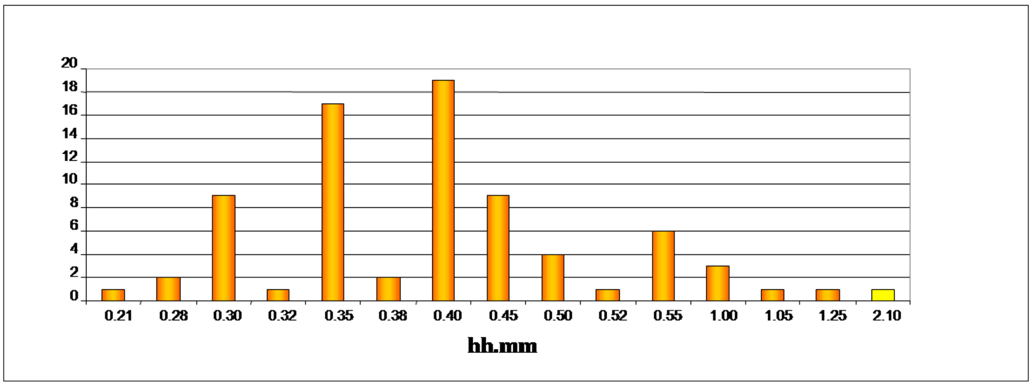

Figure 3.4: Waiting Time in the passenger station after the train is moved from workshop

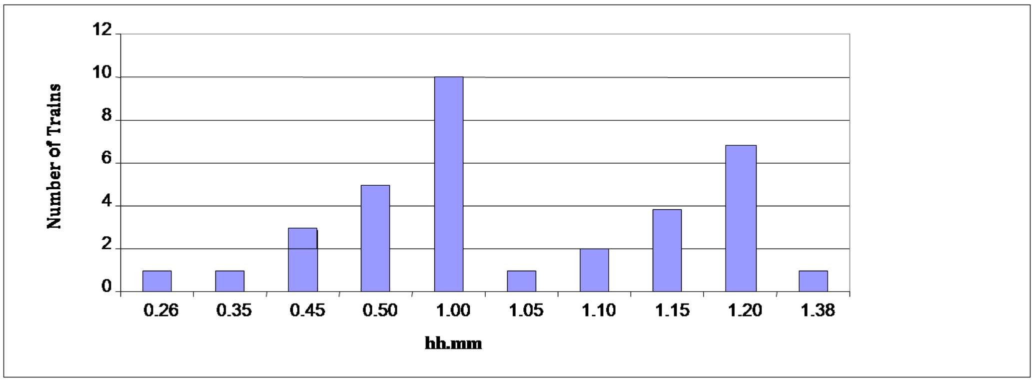

Figure 3.5: Waiting Time in the passenger station without maintenance

The importance of integrated approaches is shown in the Figure 3.7 in which is represented on the x-axis the difference between minimum time required to perform maintenance operations and the time available to the maintenance while on the y-axis is represented the waiting time in the passenger station. In the Figure 3.7 are plotted two red lines that identify four quadrants. The vertical line is on the 0 of x-axis while the horizontal line is close to 70 minutes (mean time spent at the station before and after maintenance operations).

- the area 1 contains cases in which it would be possible to reduce the time spent at the station to allow more time in maintenance

- the area 2 contains cases in which time is not a critical factor
- the area 3 contains cases in which it would be possible to reduce the time at workshop to increase the time spent at the station. Generally the cases of this area improvable are above the dotted red curve
- the area 4 contains cases in which time is a very critical factor and it is not possible to improve the solution quality

From our point of view, to minimize wastes, RSM should include:

- *Rolling Stock Rostering*
- *Maintenance Optimization*
- *Passenger Station Optimization*

and therefore the documents carry out by these planning steps must be coherent and integrated each other.

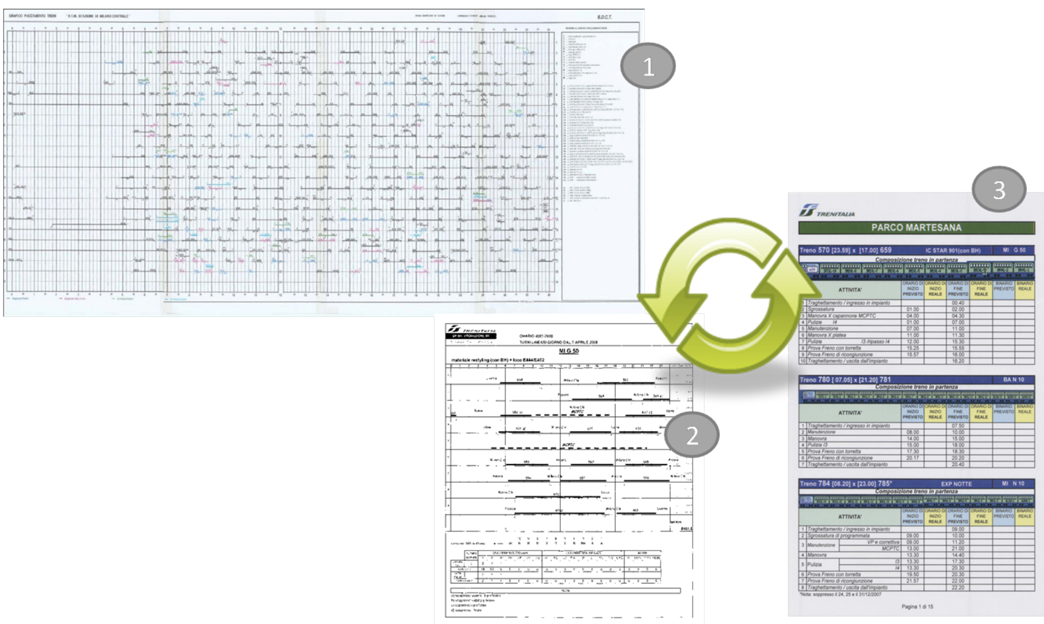

Figure 3.6: Documents involved

Figure 3.6 represents the main documents involved in our process:

1. *Station Capacity Management Plan*
2. *Train Roster*
3. *Maintenance Programs.*

3.1 Rolling Stock Rostering

The current practice of rolling stock rostering is focused on implementing slight modifications to the previous plan in presence of new requests. This is mainly due to the difficulty faced by manual planners when computing network-wide solutions. The computation of a globally feasible solution is already very complex task and the planners know the gap between their solutions and the optimal solutions related to specific performance indicators. Scientific research in this context is therefore worth to search for better quality solutions of practical interest.

Most of the scientific research considers the management of train rostering, empty rides balancing and cycles of rolling stock maintenance separately, even if these are parts of the same problem. A number of recent literature surveys are given in [1, 2, 22, 49, 63].

In view of their extensive reviews, we limit our discussion of the literature to recent analytical approaches quite related to this work.

Mároti and Kroon [65] developed a multicommodity flow model for preventive maintenance routing. They tried to improve the practical solutions by implementing a limited number of changes to a macroscopic rolling stock plan in which rolling stock units move on lines between aggregate stations. The objective function was related to the minimization of shunting plan deviations. Alfieri et al. [4] also proposed a multicommodity flow model for efficient rolling stock circulation on a single line of the Dutch railway network. Their objective was to minimize the kilometers travelled by train units of various types. Maintenance requirements were not considered in their formulation.

Budai et al. [16] provided a mathematical formulation for the long-term planning of railway maintenance works. The objective function was the minimization of the time required for maintenance, expressed as a cost function. Heuristic algorithms compute nearly optimal solutions by combining maintenance activities on each possible track.

Caprara et al. [23] studied the train timetabling problem from an infrastructure manager point of view: the objective is to improve the use of infrastructure resources. Maintenance operations were modelled as fixed constraints. An integer linear programming formulation was proposed and solved by a Lagrangian heuristic. Tests on a Italian test bed with different train types showed that maintenance constraint may seriously effect the quality of the overall timetabling process.
Recently, Borndörfer et al. [14] studied the problem to assign rolling stock to timetable services. A hypergraph based integer programming formulation was proposed for a cyclic planning horizon of one week.
In Cadarso and Marín [20], a more general rolling stock and train routing problem was addressed. The rolling stock subtask was to assign material to satisfy the timetable of a railway network, while the train routing subtask was to determine the best sequence for each material. Since the combined problem was not solved by commercial solvers, they proposed a new heuristic based on Benders decomposition. The objective function of this approach was to minimize a cost-based function related to commercial train services, empty movements, shunting and passengers in excess. Both the latter analytical approaches did not model medium-term maintenance operations and do not evaluate their cost impact.

3.2 Maintenance and Passenger Station Optimization

This thesis presents also some integer programming formulations for optimizing maintenance management at workshop. This is a key problem in railway industry that requires to provide maintenance with a minimum number of movement within workshop. The objective to achieve is a formulation able to schedule maintenance operations in order to find a minimum number of train movements in the workshop with constraints on the maintenance activities to be performed by each train and on the time windows defined by the rolling stock rostering and passenger station solutions. A workshop must manage some rolling stock units simultaneously and must be able to absorb small perturbations of the circulation. A time window for the maintenance operations between two commercial services is thus considered. Generally, the time windows are bigger than the sum of all activities and the workshop has some recovery time. Also, the workshop could be considered as a space buffer to store trains and to avoid unnecessary movements due to too busy traffic in passenger station areas.
We think that literature is too focused on manufacturing setting in order to reduce the occurrence of a failure (e.g. [89]) while unfortunately the coordination of maintenance and rolling stock scheduling is still under investigated.
In view to the literature reviews in [2], [29], [42], [50], [83], [87], we limit ourselves to review the following recent papers.
Penicka et al. in [72] introduced a formal model of the train maintenance routing problem. However, the rostering problem with maintenance constraints and the maintenance routing are not addressed in this paper. Furthermore, in their case study rostering and rolling stock maintenance are considered as an appropriate method for long distance trains only.
Mároti in [65] presented a mixed integer formulation for the maintenance routing problem in which the shunting process is consi-

dered the process bottleneck.
Budai in [16] discussed the preventive maintenance scheduling problem and the minimization of the time required for performing maintenance operations. Heuristic algorithms compute nearly optimal solutions by combining maintenance activities on each track.
Wang in [88] proposed a multiple criteria decision-making problem and evaluated maintenance strategies for different equipments.
Our results are based on real instances taken from a Trenitalia's maintenance site located in Naples (see Figure 5.1). We analyze 100 working days and we compare the programs used in real life with our model solutions. We consider all train units types and all maintenance types. Waiting time longer than one day is considered not as a variable but as a constraint on resources availability. A preprocessing was also performed in order to reduce the problem size to a sequence of one day problems, so we generated 100 one-day cases. A robustness analysis is provided in order to evaluate the schedule quality in case of medium-term traffic disturbances that alter the off-line plan of operations. The objective is to re-balance the use of workshop resources to compute of a feasible schedule, i.e. a schedule of all workshop operations within the time windows given by the rolling stock rostering planner and perturbed by traffic disturbances

3.3 Preliminary Notions

Let's define

- a *train route* as a path between two given stations, with a given travel time;
- a *train service* i as a route from a departure station d_i at departure time t_i^d to an arrival station a_i at arrival time t_i^a that must be covered by a specific train;
- a *roster* as a cycle spanning over several working days that covers all the services and the required maintenance tasks.

Train Timetabling Problem (TTP) consists in finding a train schedule on a given railway network maximizing train operator company goals. This process is made by two steps:

- generation of various alternative network plans based on different infrastructure's scenarios and on passengers' demand;
- choice of the network model after evaluation of the effectiveness and efficiency of each alternative.

The network plan is a document that describes the main train services to be provided. This document is made by a set of superposed layers. Generally, the network plan (see Figure 3.8) describes the main stations to serve (gray boxes), frequencies of trains (type of line), number of couples (labels) and type of rolling stock to be used (color of line).
A timetable is a detailed network plan showing also information on departure and arrival times and the days in which trains will be provided.
Timetable is made by some overcast layers. These are:

- *full patterns* that are sets of transports with the same characteristics (stops and travel distances) and repeated daily with a fixed frequency for a given time window;
- *partial patterns* that are patterns with little exceptions;
- *spot train* that are trains not belonging to any pattern (full or partial).

Figure 3.9 shows three different patterns highlighted with red, blue and green. These patterns have a frequency of two hours. The timetable has also a "spot train" highlighted in black around the middle of the second hour.
Cadences are generally the basic elements for a network plan and are often very important for railway company for industrial aims (cadences can help to reduce maintenance costs and improve operations management) and for commercial purpose (for example is simpler remember a cadence than a lot spot trains timetables). A very important aspect of the timetable is the *periodicity*. The commercial services (and therefore cadences) may take place only on certain days or at particular times of the year. We will denote by the term *Cyclic* the timetable with the same train services every day. Often timetables are not cyclic because the demand seasonality.
In general, the process of timetabling can not be separated from feasibility studies related to reliability and maintenance of rolling stock and infrastructure.
Reliability is the main key factor to running a successful railway. If the rolling stock, is not reliable, the railway is not workable. The performance of rolling stock's maintenance has a very big influence on passengers' safety and comfort. We understand maintenance as all activities which must be done with rolling stock, according to law, aiming to maintain it in good working order, prevent operational disturbance and/or uphold a given technical standard. We also consider belonging to maintenance tasks outside and inside cleaning of carriages, refuelling diesel engines, refilling supplies into restaurant carriages, water and oil refilling.
Rolling stock is the most maintenance intensive part of the railway system and is the most vulnerable if maintenance is neglected.
We could categorize maintenance of rolling stock in two types:

- failure based maintenance (often named *Corrective Maintenance*) that can not be avoided when a random failure occors;

- life based maintenance (often named *Preventive Maintenance*) that is provided at given time interval;
- life based on conditioning monitoring achieved by checking rolling stock units equipment.

Corrective Maintenance can be conveniently adopted if:

- possible preventive measures are too expensive, so it becomes more economical to repair the component when it breaks;
- the number of faults which may occur is so low that it is preferable to establish other priorities;
- the fault does not significantly affect production.

With reference to Preventive Maintenance, asset units need a regular preventive maintenance checks after a certain number of kilometres or working days. This limit depends on the rolling stock type and on its on board equipment. Train units must undergo different types of maintenance tasks. These tasks can take place at many workshops even if maintenance station are specialized on a subset of maintenance operations and of rolling stock types.
The layout of a workshop (see Figure 3.10) or depot consists of terminal tracks, inspection and light maintenance sheds (MAV1/2), wheel lathe, washing and cleaning areas and heavy maintenance shops (MAV1/2).

A very important aspect of maintenance is the kind of approach used in practise. In some cases (see Figure 3.11), a workshop must provide maintenance on a specific train unit in a given time window $[Twi, Twf]$ defined by operating unit. In this case, workshop can not switch rolling stock units if accidents or failures did not happen. We will name this approach *Non-buffered Maintenance.*

In other cases (see Figure 3.12), the workshop receives a list of operating unit needs in terms only of number of rolling stock units to recover and carry out day by day. In this case, workshop can manage rolling stock asset as it prefers but assuring respect of maintenance expiry dates. We will name this approach *Buffered Maintenance*.

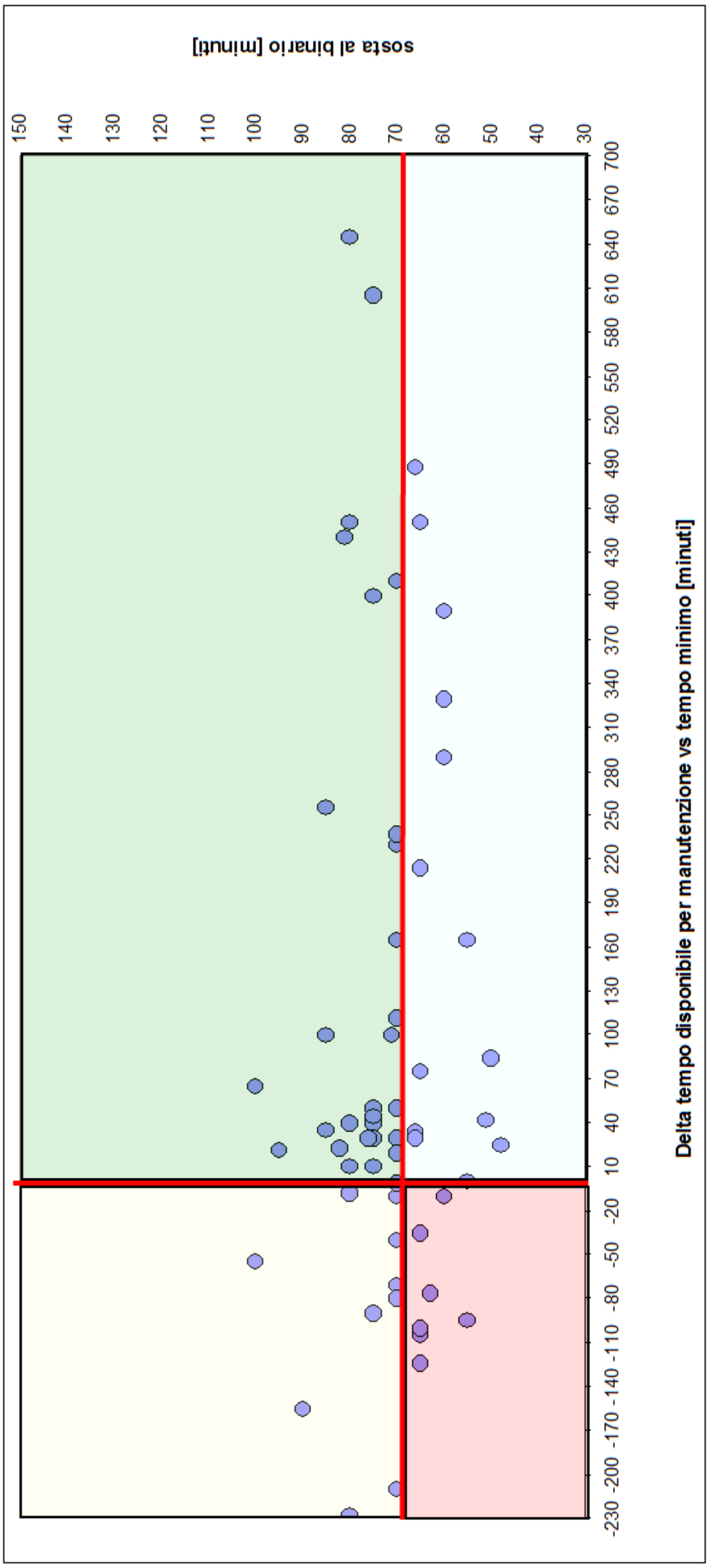

Figure 3.7: Integrated Approach

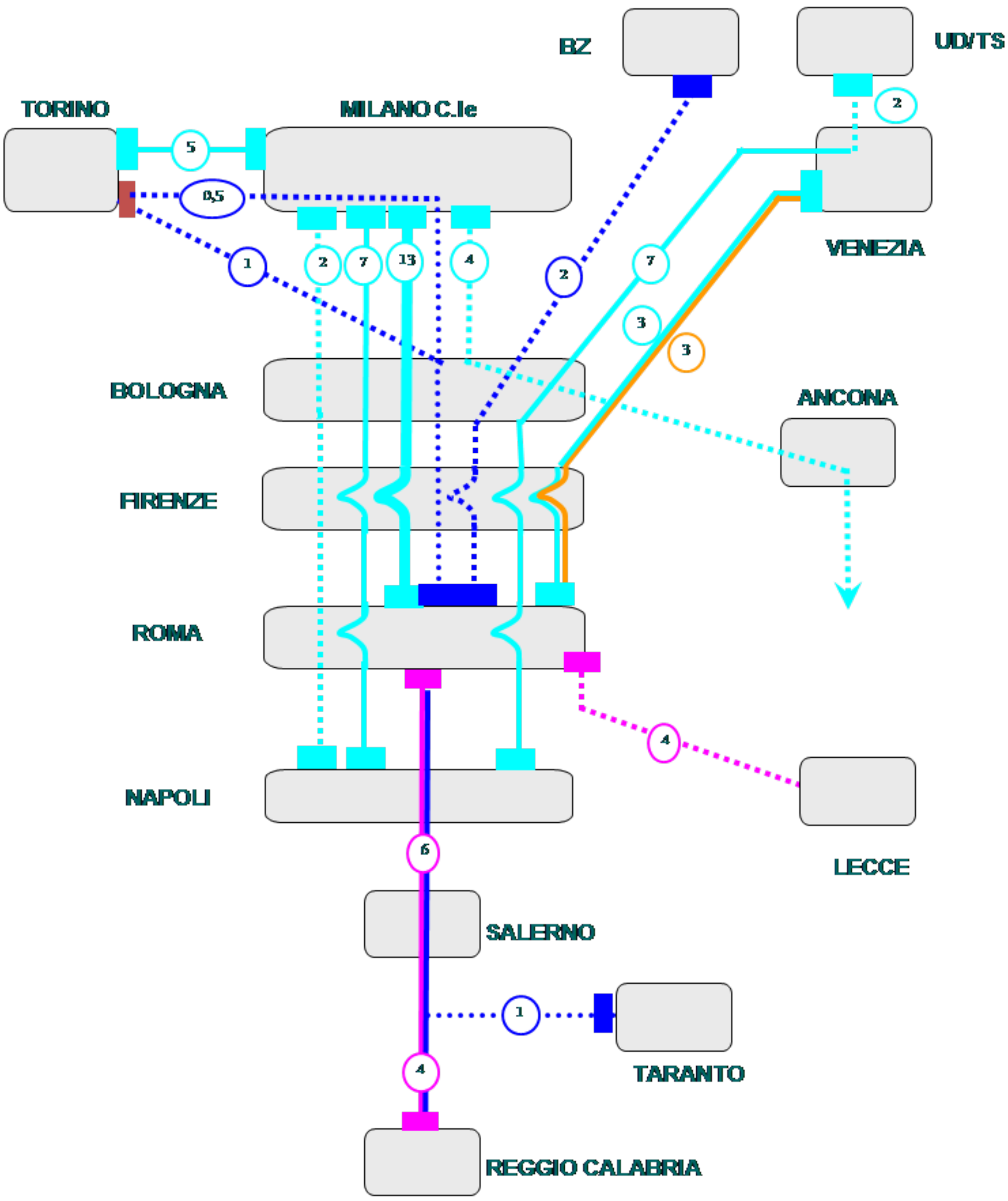

Figure 3.8: Network Plan Sample

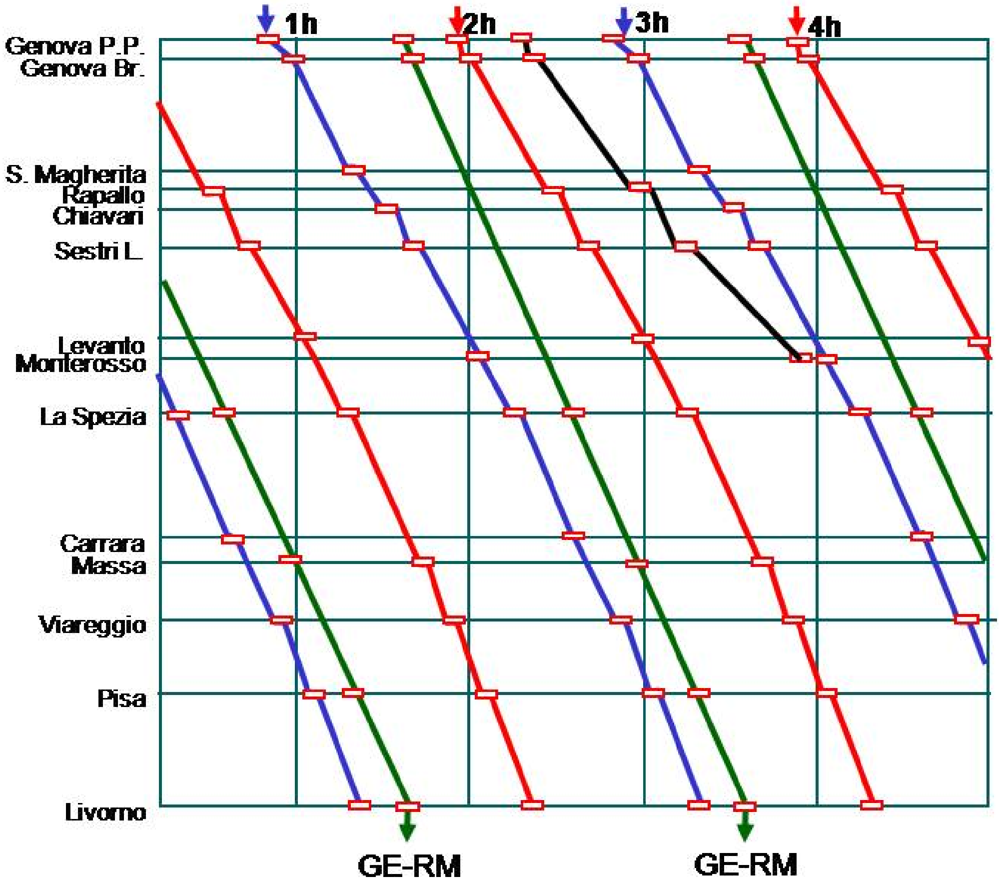

Figure 3.9: Pattern Sample

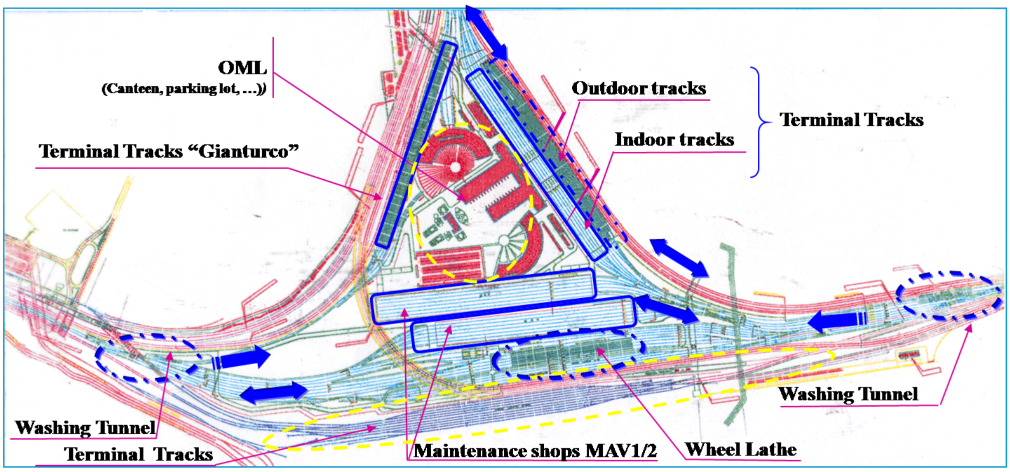

Figure 3.10: IDP Naples Layout

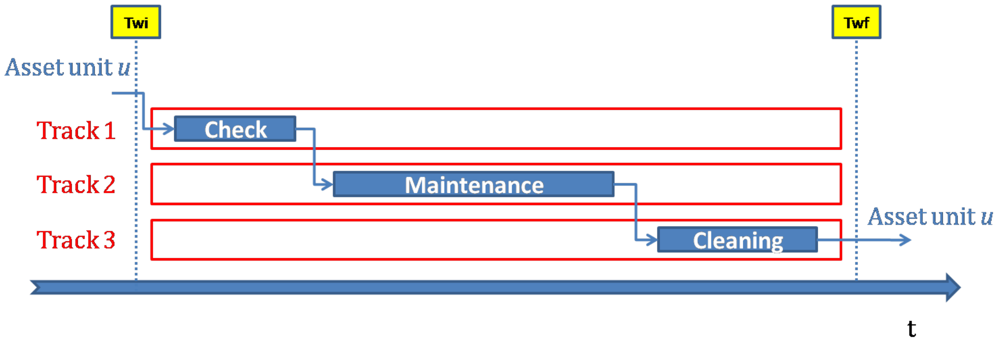

Figure 3.11: Non-buffered Maintenance

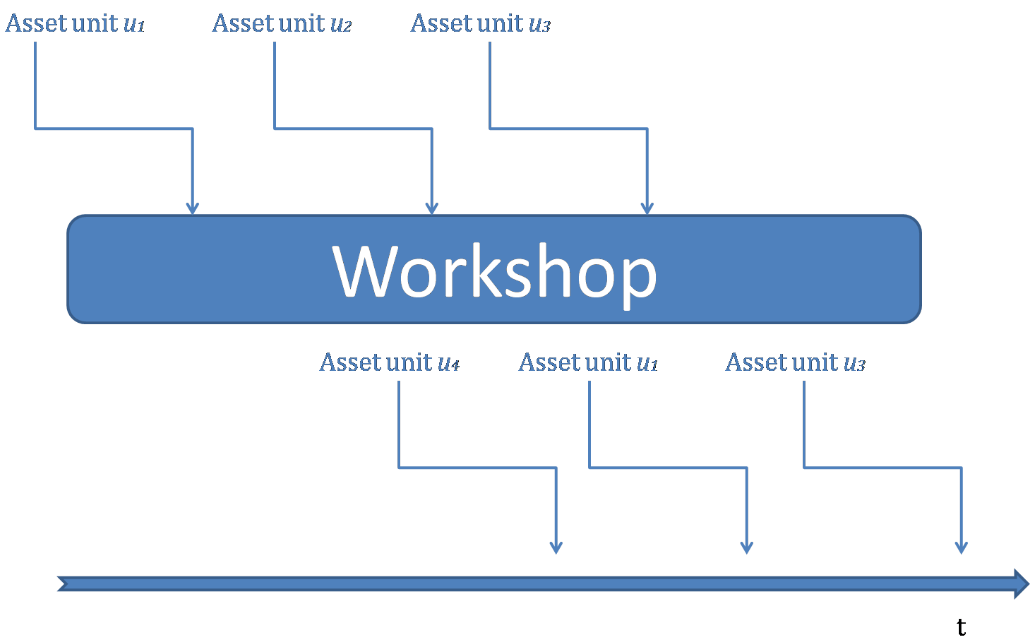

Figure 3.12: Buffered Maintenance

Chapter 4

Main Framework

The problem addressed in this work can be defined as follows: Given timetables, rolling stock assets, maintenance workshops and maintenance operations, a rolling stock circulation solution has to be computed with minimum costs, that are expressed in terms of the number of used rolling stock units, the number of used empty runs, the number of train movements between platforms at stations and within workshops. Figure 4.1 represents the framework proposed to solve the overall problem. Each coloured module (i.e. rostering, station and maintenance station) represents a specific sub-problem, and includes a set of timing constraints with the other modules. Each module is divided in inputs and outputs elements that are featured with a label representing if the element depends on external factors or on the solution provided by other modules. For each module, we intend to use a mathematical model that takes into account the solution provided by the other models. A sequential approach is proposed in order to integrate the solutions provided by each block. Our approach is to solve firstly the rostering problem for each asset unit type and then to solve the other two modules involving passenger stations and workshop operations. An automatic procedure is under development to manage the interaction between the modules and a feedback information is returned in case an infeasibility is provided by some modules. In the following, we briefly describe each module and the models we developed to solve it.

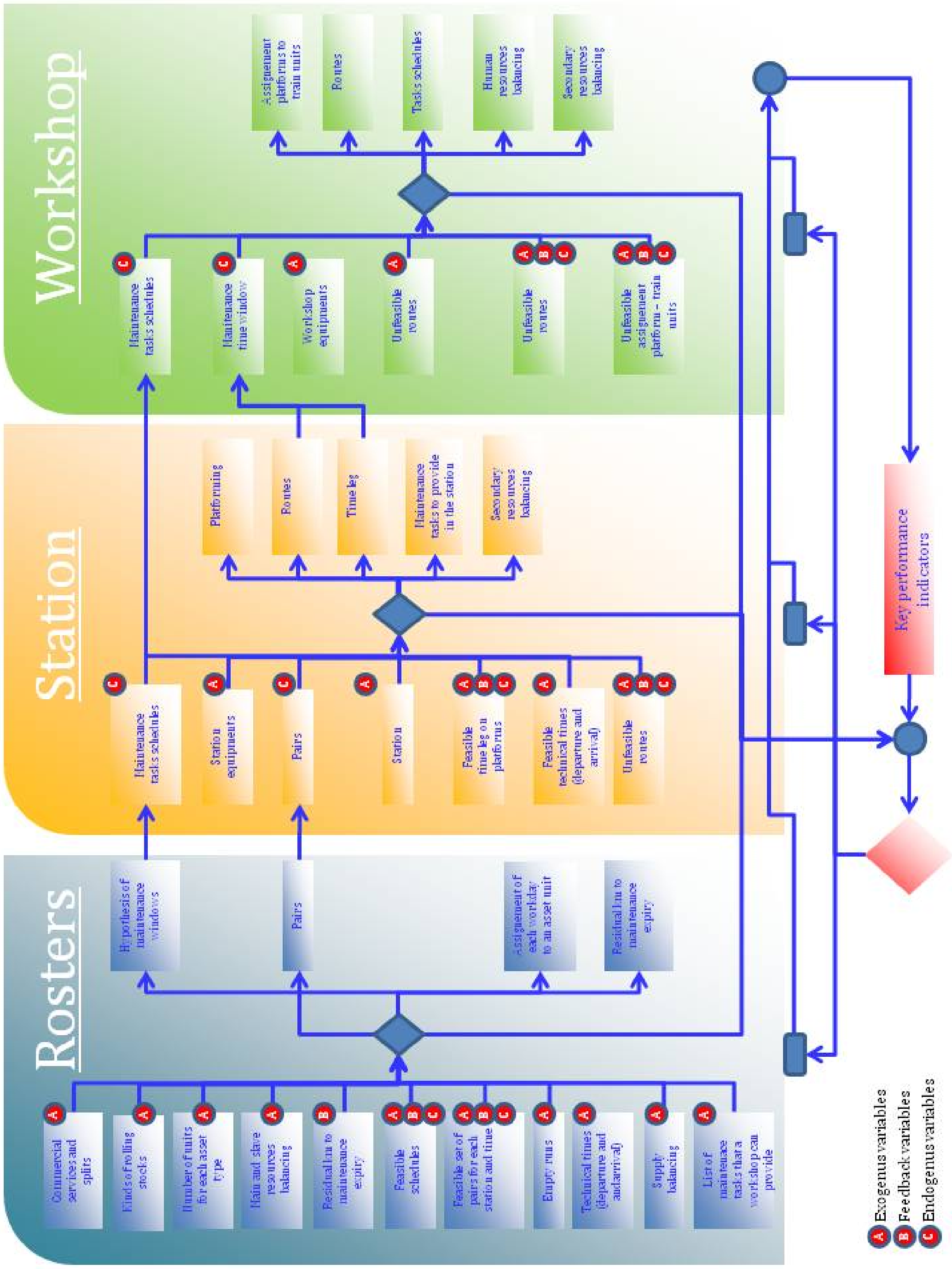

Figure 4.1: Interaction between the rostering, station and workshop modules

Rolling stock rostering The rostering module of our framework is to define a rolling stock roster that covers a set of commercial services and minimizes the costs related to number of asset units, the number of empty runs and the number of kilometers performed by the rolling stock between consecutive maintenance operations. We suppose the timetable is cyclic and the assignment of commercial services per asset kind follows brand promise principles. As described in [43], this problem corresponds to find an Hamiltonian path on a graph made by commercial services (nodes) and by feasible pairs (edges), representing time legs or maintenance activities to be provided. The basic idea is to treat this problem as a kind of travelling salesman problem with additional constraints and variables to guarantee the respect of maintenance expiry and a minimal maintenance efficiency. The output is a cyclic roster including the schedule of maintenance activities as time windows [twi, twf] to provide the required maintenance and an assigned workshop location.

Workshop scheduling The purpose of this module is to find a minimum number of train movements in the workshop with constraints on the maintenance activities to be performed by each train and on the time windows defined by the rolling stock rostering and passenger station solutions. A workshop must manage some rolling stock units simultaneously and must be able to absorb small perturbations of the circulation. A time window for the maintenance operations between two commercial services is thus considered. Generally, the time windows are bigger than the sum of all activities and the workshop has some recovery time. So, the workshop could also be considered as a space buffer to store trains and to avoid unnecessary movements due to too busy traffic in passenger station areas. In the following section, we will present a new formulation for the workshop maintenance problem with minimum number of train movements within a workshop. This new model outperforms the one we proposed in [44], both in terms of solution quality and computing speed. We will also study the system robustness, i.e. the ability of the workshop schedule to remain feasible in the presence of disturbances.

Passenger station scheduling This module of the framework is still under investigation. In principle, a modelling approach similar to the workshop scheduling can be suitable, even if different services are provided. In fact, a passenger station could be viewed as a workshop where the trains must be routed and wait in order to perform passenger related operations.

4.1 Rostering

This work is the first attempt to combine rolling stock and maintenance operational aspects of the rostering problem in an efficient way. We present a new mixed integer linear programming formulation that models in detail short-term maintenance operations. The general goal is to execute a given timetable and to minimize the use of rolling stock units and their maintenance. Given the departure and arrival times of each scheduled train service, the rostering problem is composed by three main tasks: (a) assign rolling stock units (i.e., trains) to the services, (b) schedule the maintenance tasks, (c) limit the number of empty rides.
The specific objective of our research is related to the following questions:

- "How can the timetable be executed by an efficient use of resources such that the overall railway company costs are reduced?"
- "Which is the maximal improvement that can be achieved? At which cost?".

In this work, we give an answer to these questions by performing an assessment of the main key performance indicators. The

computation experiments are obtained by implementing and solving the proposed model with a commercial MIP solver. We tested on a number of practical instances based on timetable examples from Trenitalia (the main Italian passenger company for passenger services) for year 2011. The real-life solutions are compared with those obtained by our solution method, in terms of the number of trains and empty rides needed to realize the given timetables.
The next sections define the integrated problem of train rostering and maintenance scheduling, describe the mathematical formulation of this problem, present computational experiments on real-world scenarios, discuss the obtained results and provide a description of further research directions. We consider a macroscopic description of the railway traffic flow. The network is composed by a number of tracks and stations.

4.1.1 Cyclic timetable

In this section we assume that the same timetable is repeated every day. In other words, we consider a *cyclic* timetable and do not study its variability e.g. in case of high/low demand days. With this assumption, finding a roster spanning over k days allows to cover all services in a day with k trains.

Problem description

A *maintenance site* is where maintenance work is performed and can coincide with a passenger station or not. Each maintenance site is dedicated to specific types of maintenance work, such as: interior or exterior cleaning, refuel (only for diesel units), regular inspection, repair (scheduled or not) and technical check-up. Each type of maintenance task must be performed regularly, i.e. within a maximum time limit or a maximum number of kilometres from the last maintenance of the same type. Since performing some maintenance task too often would cause an unnecessary cost for the company, each type of maintenance task should be performed in the

proximity of its maximum limit. In fact, maintenance tasks impose severe constraints and their effect on the line capacity is difficult to analyze [23].
Figure 4.2 presents a roster for a cyclic timetable with 14 train services (S1–S14). Each row shows the daily route to be covered by a specific train. The different colors identify the train services (red), the maintenance tasks (blue) and the empty rides (black). This roster can be considered as an example of daily cycle with 6 trains. Specifically, Figure 4.2 (up) shows a roster without empty rides while Figure 4.2 (down) shows a roster with two empty rides (E1–E2). The two schedules also differ in the number of maintenance tasks (3 in the first case and 2 in the second one).
More specifically this comparison shows how it is possible to reduce the number of maintenance tasks by using empty rides. It follows that we have reduced the maintenance costs but, on the other hand, we have increased the cost of empty rides too. The problem is to find the minimum global cost solution including the balancing of empty rides.
The problem addressed in this paper consists of finding a shortest roster, i.e. a sequence of all services spanning over the minimum number of days, such that all required maintenance tasks are inserted in the roster. Empty rides can be added to the roster in order to connect train services and/or to visit maintenance sites. Although empty rides cause a relevant cost (e.g. related to additional energy consumption, rolling stock and crew resources) for the company and increase the traffic in the network, their inclusion may help to reduce the maintenance cost and the roster length. For the above reasons, optimizing the scheduling of maintenance tasks, trains services and empty rides is an important contribution to reduce the overall company costs.
Figure 4.3 shows a kilometers cumulative line which is 0 when a maintenance activity is provided and increases when the rolling unit runs. The vehicle must comes back to workshop before the maintenance expiry. Very often vehicles come back in the work-

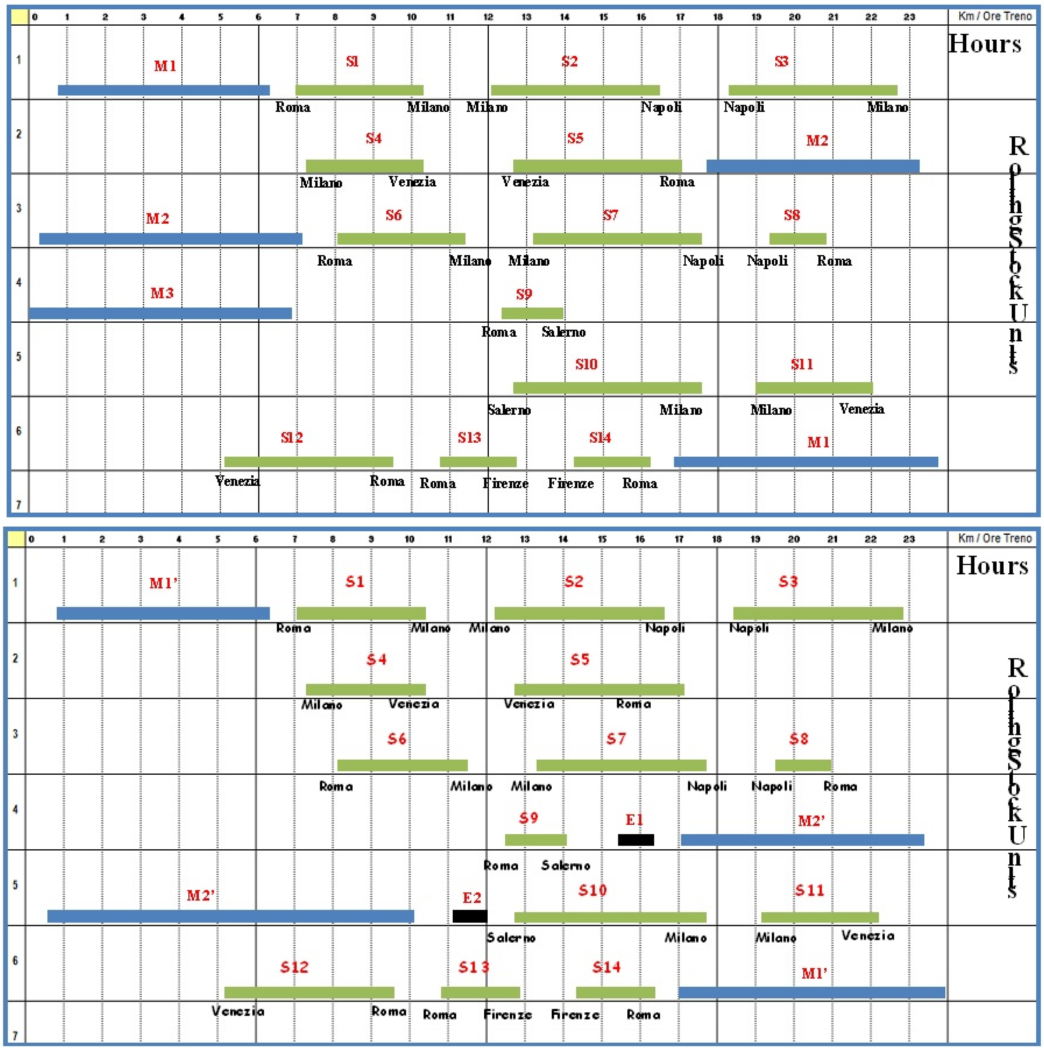

Figure 4.2: Rolling stock roster with and without empty rides

shop too early. The difference between kilometers run and maintenance expiry (horizontal blue line) is a waste. Our roster must respect a constraint of kilometers covered at moment of entrance in the workshop for a maintenance activity. Figure 4.4 shows how

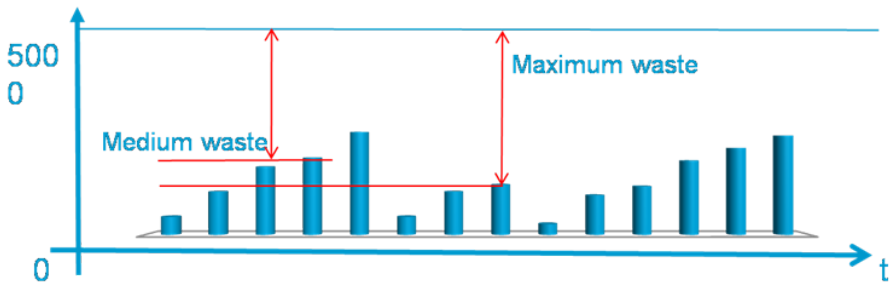

Figure 4.3: Rostering maintenance waste

empty runs can help to optimize maintenance management. In fact an empty run can help to find new opportunities to come back to workshop.

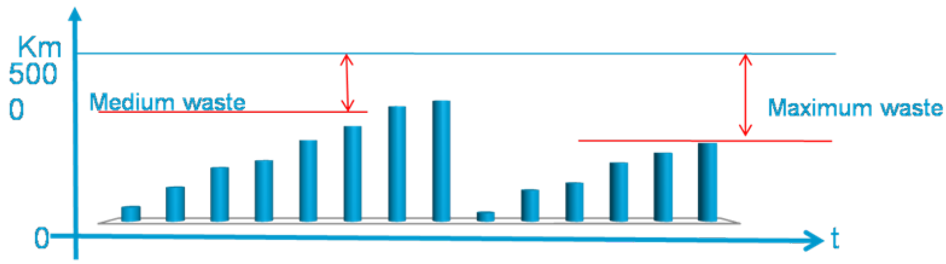

Figure 4.4: Maintenance Costs Reduction

In our approach, the input data are: rolling stock asset, timetable, infrastructure and train services. Alternative solutions are analyzed in terms of the number of required trains, empty rides and maintenance kilometres. The objective function is the minimization of the number of trains needed to perform all services while the latter two factors will be considered both as fixed or flexible values. Strict values are replaced by time windows of [minimum, maximum] values. A large time window corresponds to having more

flexibility during railway planning that would allow a greater possibility to design cost effective solutions [23, 30, 32].
We use the following data as problem input: the timetable with the scheduled train services; the maintenance sites; the maintenance tasks; the maximum number of kilometres and the planned time of an empty ride; the minimum and maximum number of kilometres for each type of maintenance task.

Problem formulation

The RSR problem is represented by a graph $\mathcal{G} = (V, A)$ in which the set of *nodes* V contains all the train services to be included in the roster, while the set of *arcs* A is associated to a feasible sequencing of train services in a roster, plus the possible inclusions of empty rides and maintenance tasks. There can be several types of arcs between any two nodes i and j, and we denote by z the type of arc (i, j, z) and by Z the set of arc types.
If the arrival station a_i of service i is equal to the departure station d_j of service j, we add to A a first arc of type $z =$*waiting* between i and j plus an arc of type $z =$*maintenance* for each type of maintenance task m that can be executed in the proximity of station a_i, i.e., such that the distance between a_i and the closest maintenance site enabled to perform m is smaller than a pre-defined value.
If $a_i \neq d_j$, we add to A a first arc (i, j, z) of type $z =$*empty ride* plus an arc for each maintenance task that can be processed in a maintenance site close to a_i, d_j or along the route from a_i to d_j.
Each arc (i, j, z) has a *cost* c_{ijz} that is the number of days required to process j after the completion of i, i.e., zero if i and j can be performed consecutively in the same day. The rostering problem can be viewed as the problem of finding a minimum cost Hamiltonian cycle in $\mathcal{G}$ with additional constraints related to the implementation of maintenance tasks.
For each arc (i, j, z), we consider two types of service pairings: the departure station of the first service is equal or not to the arrival station of the second service. In case of equality, the time lag bet-

ween the two services can be spent in one of the following cases: (i) a slack time is considered between the two services or (ii) a maintenance window is added in order to execute maintenance works for the rolling stock involved, eventually a slack time can still be added to the schedule. The decision depends on a number of factors: the maintenance status of the rolling stock involved in the two services, the distance from the station and the maintenance location, the size of the time lag available to perform the works. In case departure and arrival stations are different, the waiting time between the two services must be spent in one of the following cases: (i) an empty ride is added between the two services, (ii) in addition to the empty ride of case (i) a maintenance window is added in order to execute maintenance works for the rolling stock involved. The maintenance works can be either executed at a workshop nearby the arrival station of the first service or nearby the departure station of the second service.
To formulate the maintenance constraints, for each arc (i, j, z) and for each maintenance type m, we introduce a real variable g^m_{ijz} that counts the kilometres covered by each train since the last maintenance task of type m was performed. A lower bound β_m and an upper bound γ_m on the kilometres are specified for each maintenance task m and constraints $\beta_m \geq g^m_{ijz} \geq \gamma_m$ are added to the formulation to force each train to visit a maintenance site between β_m and γ_m kilometres.

Illustrative example

Figure 4.5 shows a small graph to illustrate the problem formulation.
For each train service, there is a (red) node with labels indicating departure and arrival stations, plus the associated times. The green arcs indicate paired services, the (solid) black arcs the empty rides without maintenance, the (dotted) black arcs the empty rides with maintenance tasks, the blue arcs the maintenance tasks without empty rides. The numerical labels show arc costs, while non-

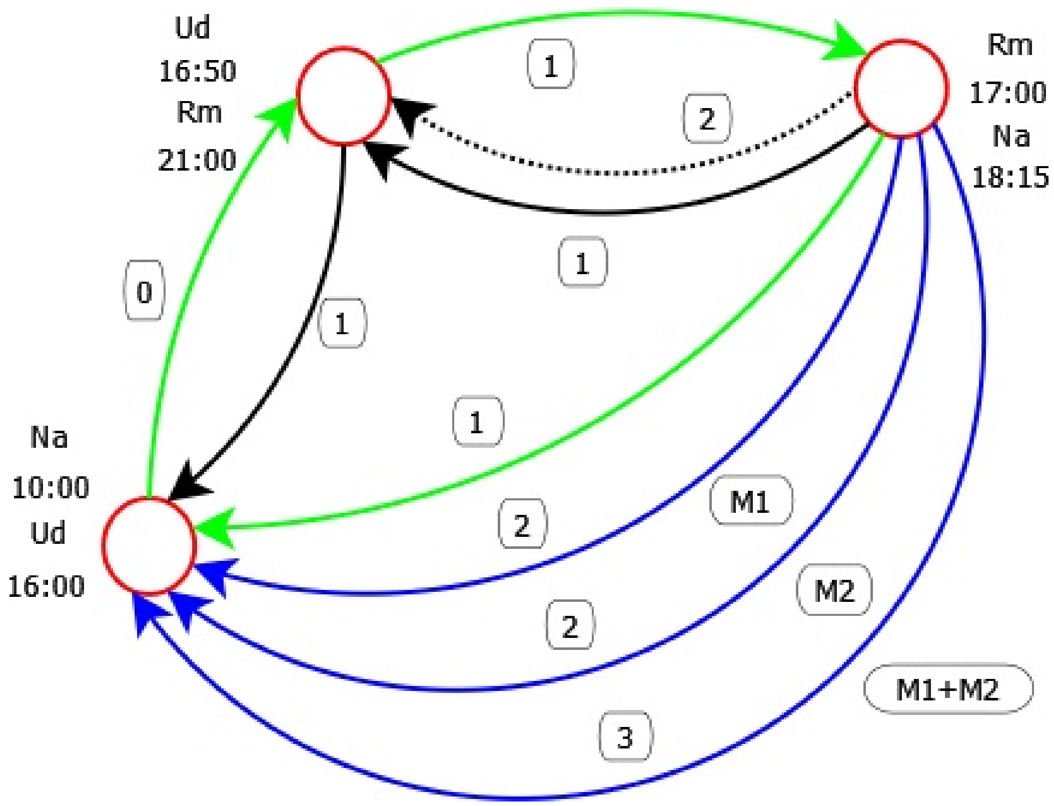

Figure 4.5: A graph formulation for three train services

numerical labels indicate maintenance types (M1, M2 and their combination). For simplicity, the maintenance costs are not shown in the graph. In Figure 4.5, we have three services (Napoli-Udine, Udine-Roma and Roma-Napoli) that require a number of trains, maintenance works and empty rides. A solution is an Hamiltonian path with one or more maintenance arcs to guarantee the maintenance expiry. In the solution the empty rides (black arcs) are optional.

List of notations

We now list the notation used in this section.

- V is the set of train services (i.e., the set of nodes)
- n is the cardinality of the set V
- A_1 is the set of empty ride arcs without maintenance tasks (i.e., the set of solid black arcs)
- A_2 is the set of empty ride arcs with maintenance tasks (i.e., the set of dotted black arcs)
- A_3 is the set of maintenance arcs without empty rides (i.e., the set of blue arcs)
- A_4 is the set of service pairings (i.e., the set of green arcs)
- A is the set of all arcs: service pairings, empty rides and maintenance tasks ($A = A_1 \bigcup A_2 \bigcup A_3 \bigcup A_4$)
- $A^{\bar{m}}$ is the set of service pairings, empty rides and maintenance tasks that do not include maintenance task m
- A_I^m ($A_I^{\bar{m}}$) is the set of empty ride arcs with maintenance tasks that (do not) include task m in a maintenance site at the beginning of their route
- A_{II}^m ($A_{II}^{\bar{m}}$) is the set of empty ride arcs with maintenance tasks that (do not) include task m in a maintenance site at the end of their route
- A_{III}^m ($A_{III}^{\bar{m}}$) is the set of empty ride arcs with maintenance tasks that (do not) include task m in a maintenance site in the middle of their route

- $Z_{i,j}$ is the set of arc types between nodes i and j
- (i, j, z) is an arc between start node i and end node j of type $z \in Z$
- K_i are the kilometres of train service i
- K^1_{ijz} are the kilometres to be performed by a train (associated to arc (i, j, z)) from a_i to a maintenance site in case of empty ride
- K^2_{ijz} are the kilometres to be performed by a train (associated to arc (i, j, z)) from a maintenance site to d_j in case of empty ride
- K^3_{ijz} are the kilometres to be performed by a train (associated to arc (i, j, z)) from a_i to d_j in case of empty ride
- α is a bound related to the maximum number of empty rides allowed in a solution
- β_m is a lower bound on the kilometres travelled by a train between consecutive executions of task m
- γ_m is an upper bound on the kilometres travelled by a train between consecutive executions of task m

Problem variables

The proposed formulation considers three types of variables: X is a set of binary variables such that $x_{ijz} \in X$ is equal to 1 if arc (i, j, z) belongs to the Hamiltonian cycle and zero otherwise, Y is a set of integer variables that are used for sub-tour elimination, G is a set of real variables. Variable $g^m_{ijz} \in G$ is used to force that if $x_{ijz} = 1$ then the kilometres travelled by a train between two consecutive executions of task m is always between β_m and γ_m. In a solution, the variables in Y and G can be derived from the variables in X.

Objective function

The objective function of the problem is the minimization of the number of days included in the roster, i.e., the number of trains required to perform all services in a day:

$$\sum_{(i,j,z)\in A} c_{ijz}x_{ijz}$$

where c_{ijz} is the cost of arc $(i,j,z) \in A$.

Path constraints

The first set of constraints is:

$$\begin{aligned} \sum_{i\in V} \sum_{z:(i,h,z)\in A} x_{ihz} = 1 \\ \sum_{j\in V} \sum_{z:(h,j,z)\in A} x_{hjz} = 1 \end{aligned} \tag{4.1}$$

The satisfaction of Equation (4.1) forces exactly a predecessor and a successor for each node $h \in V$.

Sub-tour elimination constraints

This set of constraints are introduced for modelling the roster solution as an Hamiltonian cycle.
Figure 4.6 shows an infeasible situation in which there is subtour in the graph. This situation is infeasible since the rolling stock would not return to their original location at the end of the day.
The basic idea to avoid sub-tours in the roster is to use node labels that count the order of nodes in the solution, beginning from a first node n_0 randomly chosen.
Along the path the label of each visited node is increased of one unit compared with the previous node (but not for the first node). So we have that the value of labels is from 1 to n and two nodes cannot have the same label.
In the example of Figure 4.6 we can observe that if we have a sub-loop we can't respect the previous conditions. In fact in case of a sub-loop, we can't have a progressing counting of the nodes(see for

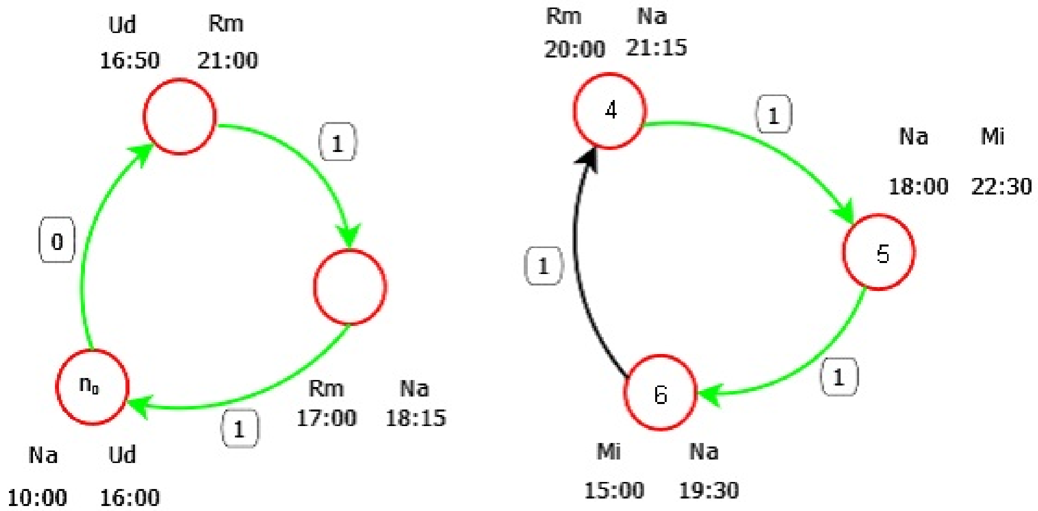

Figure 4.6: Example of sub-tour situation

instance the interior labels of the node [Rm 20:00 - Na 21:15] and of the node [Mi 15:00 - Na 19:30]).
An integer variable $y_{kj} \in Y$ is associated to each pair of nodes $k, j \in V$, with $k \neq j$, such that:

$$\sum_{i \in V} y_{ji} = \sum_{k \in V} y_{kj} + 1 \quad \forall j \in V \setminus \{n_0\} \tag{4.2}$$

$$0 \leq y_{ij} \leq n \sum_{(ijz) \in A} x_{ijz} \quad \forall y_{ij} \in Y \tag{4.3}$$

$$\sum_{i \in V} y_{n_0 i} = 1 \tag{4.4}$$

Equation (4.2) constrains the sum of the arcs entering each node, but n_0, to be equal to the sum of the arcs leaving the same node plus 1. Inequality (4.3) constrains the arc label values to be greater

than 0 if and only if a variable $x_{ijz} \in X$ of type z exists between nodes i and j with value greater than 0. With these equations, there is just one arc leaving and one arc entering each node with $y_{ij} > 0$. If two services i and j are executed consecutively (i.e., if there is a variable $x_{ijz} = 1$), the label of j is equal to the one of i plus 1. Equation (4.4) forces node n_0 to be numbered 1.
Sub-tour elimination constraints should be used when maintenance could be provided in workshops with different features. Generally one workshop is able to provide maintenance and to check rolling stock status. So, to avoid very long empty runs, we choose an Hamiltonian path to bring each rolling stock unit in the main workshop. In this case a new equation must be added to assure that the main workshop is involved in the roster.

$$\sum_{(i,j,m) \in A} x_{i,j,k} = 1 \quad \textit{where m is provided in the main workshop} \tag{4.5}$$

Obviously, in case of one workshop we don't use this onerous set of constraints. We have also to consider that sometimes rolling stock is used to cover an entire roster in order to spread evenly the stress related to covered services (different lines have different stress) and in order to assure the same service level to passengers. Even in case we need to use sub-tour elimination constraints. Other possible formulations are shown in [82],[81] and [24].

Maintenance constraints

Maintenance tasks need to be performed within a given time window of maintenance. However, the intention is to prevent the execution of an excessive number of maintenance tasks. The less are the maintenance tasks the more cost effective is the overall solution.
Figure 4.7 shows a simple example with two services, a maintenance site and two empty ride possibilities: including (see dotted

black arcs K^1 and K^2) or not including the maintenance work (see black arc K^3). In this case, travelling from origin station (Naples) to destination station (Udine) would be more costly if passing through a workshop. However, maintenance tasks must be taken if the kilometres travelled by the current train would exceed γ_m for at least a task m at the destination station.

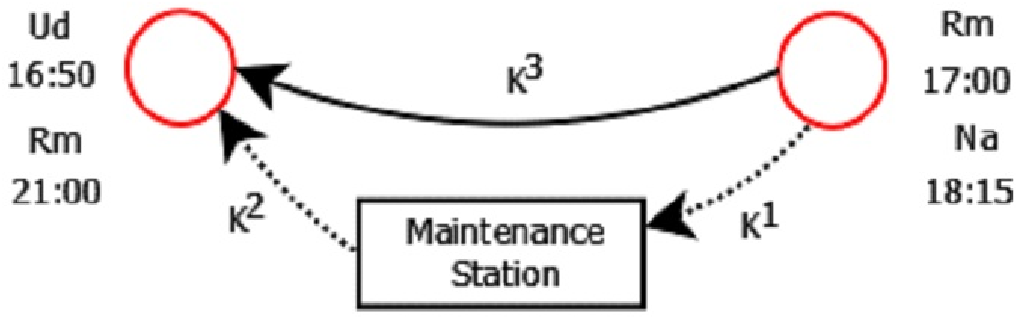

Figure 4.7: Example of empty ride and maintenance tasks

The formulation of maintenance tasks requires to the introduction of a new variable g^m_{ijz} for each maintenance task m and for each arc $(i, j, z) \in A$. This variable is increased at each train service and at each empty ride, and is set to 0 when the maintenance task m is performed. The set of maintenance constraints is as follows:

$$
\begin{aligned}
\sum_{l \in V} \sum_{z \in Z_{j,l}} g^m_{jlz} = K_j + \sum_{i \in V} \sum_{z \in Z_{i,j}:(i,j,z) \in A^{\bar{m}}} g^m_{ijz} + \\
\sum_{(i,j,z) \in A^m_{III}} K^2_{ijz} x_{ijz} + \sum_{(i,j,z) \in A^m_{I}} K^3_{ijz} x_{ijz} + \\
\sum_{(j,l,z) \in A^m_{III}} K^1_{jlz} x_{jlz} + \sum_{(j,l,z) \in A^{\bar{m}}_{I} \cup A^{\bar{m}}_{II} \cup A^{\bar{m}}_{III} \cup A^m_{II} \cup A_1} K^3_{jlz} x_{jlz} \\
\forall j \in V, \forall m
\end{aligned}
\tag{4.6}
$$

$$
g^m_{ijz} \leq \gamma_m x_{ijz} \quad \forall (i, j, z) \in A, \forall m \tag{4.7}
$$

$$g^m_{ijz} \geq \beta_m x_{ijz} \quad \forall (i,j,z) \in A_2 \cup A_3 : m \in Z_{i,j} \tag{4.8}$$

Equation (4.6) counts the kilometres performed by each train, including the empty rides. The terms represent the kilometres run by a train for a commercial service and empty ride. In case of a maintenance task along an empty ride, the kilometres run must be considered partially (before or after a maintenance task). Inequality (4.7) constrains the kilometres to be performed after a task of type m to be smaller than the upper bound, while inequality (4.8) constrains the kilometres to be performed before a task of type m to be at least equal to the lower bound. The real important reference is the expiry of each basic maintenance task even if it is possible to perform a combination of more than one basic maintenance task in a specific site.

Empty ride bound constraints

This type of constraints defines the maximum number of empty rides permitted in a solution:

$$\sum_{(i,j,z) \in A_1 \cup A_2} x_{ijz} \leq \alpha \tag{4.9}$$

where the bound α is an input parameter.

Computational experiments

This section presents a set of computational experiments on real-world cases from the Trenitalia timetable of year 2011. We consider practical rosters and solve the proposed model with CPLEX MIP solver 12.0 (see [66] and [53]).

Description of the instances

Table 5.3 presents five practical timetables (Column 1). The first four timetables (T1–T4) are based on real cases, and will be compared with the practical solutions, while the fifth(T5) is hypothetical, and has been created to test the computational limits of our approach.

Table 4.1: Description of the train services for each timetable

Timetable Scenario	Rolling Stock Categories	Num Train Services	Railway Cars	Total Length [m]	Maintenance Deadline [km]
T1	Loco E444	46	1	17	>>1500
T2	ETR 485	20	9	237	1500
T3	ETR 600	26	7	237	1500
T4	ETR 500	78	11	328	1500
T5	ETR 600+500	104	18	237/328	1500

For each timetable scenario in Table 5.3, Column 2 shows the rolling stock categories. Specifically, T1 uses locomotives only while the other four (T2-T5) use different types of high speed trains. For each category, Columns 3 shows the number of train services scheduled in the timetable, Column 4 the number of railway cars, Column 5 the total length of each car (in meters), Column 6 the deadline of its maintenance works (in kilometres).

Comparison of CPLEX versus real-life rosters

Table 4.2 presents results on the five timetables (Column 1). Column 2–3 describe the practical solutions in terms of the number of empty rides and train services defined in each timetable. To compare our solutions with the practical ones, Column 4 shows the results when the maximum number of empty rides α is fixed to the same value used in practice. Differently, Column 5 shows our solutions when the empty rides are additional variables that can be selected in a range of [0, 10] values.
Computational results in Table 4.2 show relevant potential application of the proposed formulation for improving the current prac-

Table 4.2: Assessment of practical and CPLEX solutions

Timetable Scenario	Practical Solution		CPLEX Solution: Fixed Empty Rides	CPLEX Solution: Flexible Empty Rides
	Empty Rides	Trains	Trains	Trains
T1	0	26	25	20
T2	0	9	8	8
T3	0	10	10	10
T4	8	40	38	35
T5	8	-	48	42

tical solutions. For timetables T1, T2 and T4, our model solution also compares favorably with the practical roster. a total reduction of up to 4 trains needed to cover all services.
Another observation from Table 4.2 is that the model with a fixed number of empty rides performs worst than the model with a window of min-max values for the empty rides. This is due to the additional flexibility added in the latter model that is able to reduce the necessary rolling stock. When relaxing the constraint on the empty rides, the maximum gain is obtained for timetable T1 for which our model solution with flexible empty rides presents a 23% reduction in the number of trains needed to cover all services.
For the set of experiments with fixed empty rides, the average computation time of CPLEX is around 10 seconds. In case of flexible empty rides, the average computation time of CPLEX solutions is around 1 minute for T1, T2, T3 and T4, while T5 requires around 7.5 minutes.

Maintenance optimization

We can compare the real life solution with our model solution around the maintenance efficiency. The real life solution presents a least efficiency value that we can increase of 53%. In other words, if we consider the timetable T4, we can reduce not only the number of rolling stocks needed to cover the timetable but we can have a real reduction of the maintenance cost. This result could be achieved using the minimum efficiency admitted as a parameter. We have tested the instance, called T4, to find the maximum feasible value for the minimum efficiency (β_m) with a fixed rolling stock units available. The minimum value of efficiency was increased of 1000 km for each step till we found an infeasible solution. The results are shown in the Table 4.3.

Table 4.3: Minimization of maintenance cost

Instance	Min Efficiency [Km]	Max Efficiency [Km]	Improvement [%]	Maintenance tasks
Real	5688	13040	-	6
Mod2	6688	13040	18	5
Mod3	7688	13040	35	5
Mod4	8688	13040	53	5

Evaluating the impact of empty ride flexibility

Figure 4.8 shows a second set of experiments in which we consider the empty rides as additional problem variables that can be selected in a range of min-max values. The experiments are based on the five timetables and use different settings of the maximum number of empty rides. We show on y-axis the number of trains needed for the roster and on the x-axis the maximum number of empty rides.

From the results of Figure 4.8, we have the following observations. For T2 and T3, increasing the empty rides has no effect on the rolling stock required to run all services, while for T1, T4 and T5 the rolling stock used is progressively reduced.

Considering T4, the practical solution (with 40 trains and 8 empty rides) can be improved by two actions: reducing the number of trains and/or reducing the number of empty rides. When comparing the practical solution versus the optimal solution of our model,

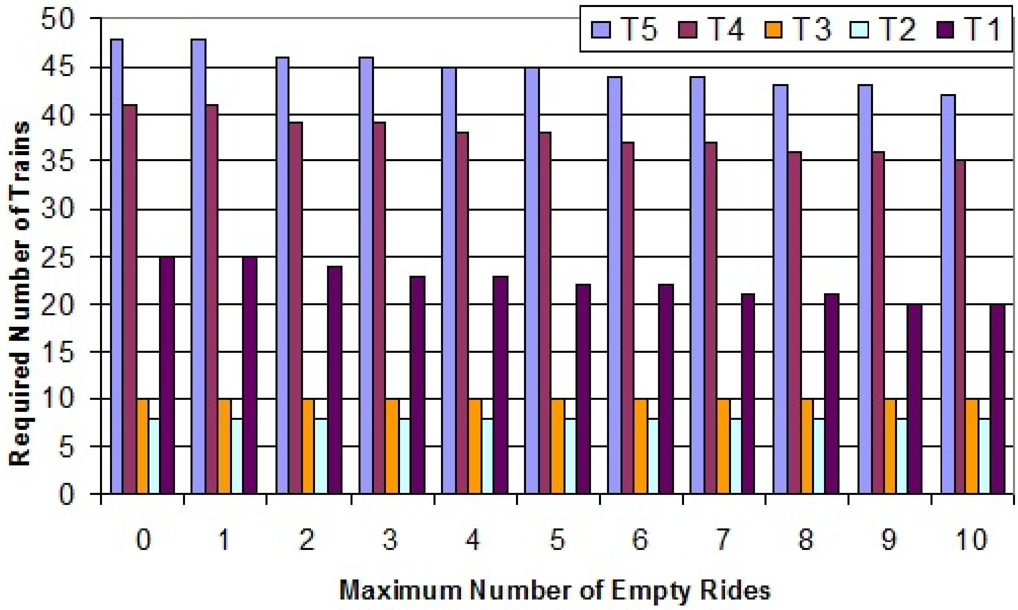

Figure 4.8: Measuring the compromise between rolling stock units and empty rides

there is a trade-off between the two actions for the two cases with 9 and 10 empty rides. For smaller values of empty rides, our model gives always better solutions than the practical one for both objectives. In the solution with 8 empty rides, the number of trains can be reduced up to 10%.

Furthermore, with reference on the results of the maintenance costs minimization, we can find a special solution with just 6 empty rides versus 8 of the real life solution. Is it possible to reduce the train units needed, to increase the maintenance efficiency and to reduce the number of empty rides? If we use an integrated approach our tests proof that is possible. In other words our integrated model can reduce the operational costs reducing one by one the terms of the overall cost equation.

Numerical example

Let's consider the timetable shown in the Table 4.4 and suppose it is cyclic.

Table 4.4: Example Timetable

Train	From	To	Departure DateTime	Arrival DateTime	Km
A	Milan	Naples	08:00	12:00	600
B	Milan	Naples	19:00	23:00	600
C	Naples	Milan	07:00	11:00	600
D	Rome	Milan	09:00	12:00	380

For empty runs let's suppose that

- the maximum number of empty runs in the roster is 1;
- the maximum empty run kilometres are 300 (so only between Rome and Naples);
- empty runs speed average 50 Km/h.

For maintenance let's suppose that

- only one workshop (located in Naples);
- only one kind of maintenance (named MC);
- the MC duration is 600 minutes;
- minimum number of km to cover between two maintenance operations = 1200;
- maximum number of km to cover between two maintenance operations = 3000 (maintenance expiry).

Furthermore, providing maintenance after/before/in the middle of an empty run is allowed.
For pairing let's suppose that

- the minimum pairing time is 120 minutes;
- no maximum pairing time.

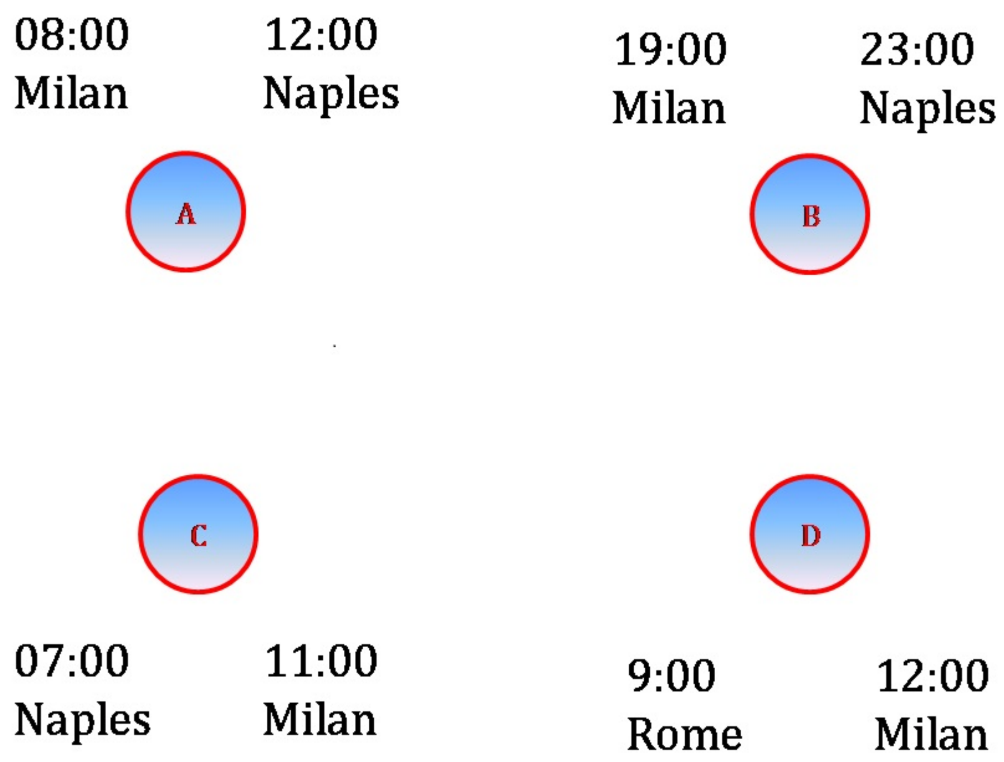

Figure 4.9: Initial example graph

Firstly we have to generate the graph using train services to cover (nodes of Figure 4.9).

Now we have to complete the graph adding all feasible pairs (edges). Table 4.10 shows the feasible pairing and the different types of edges for each couple of trains. The number of edges, given by the number of green checks, will be 12.
The final graph is shown in the Figure 4.11.
Now we can convert the graph in a Cplex linear programming file. Cplex response is shown in the Figure 4.12.
The solution could be represented in form of a graph shown in the Figure 4.13.
Note that the variable G between node D and B includes also km of an empty run from Naples to Rome.

Train "From"			Train "To"							
Train	**To**	**Arrival DateTime**	**Train**	**From**	**Departure DateTime**	**Time Leg**	**Waiting on Platform**	**Empty Ride**	**Empty Ride and Maintenance Task**	**Maintenance Tasks**
A	Naples	12:00	B	Milan	19:00	420				
A	Naples	12:00	C	Naples	7:00	1140	✓			✓
A	Naples	12:00	D	Rome	9:00	1260		✓	✓	
B	Naples	23:00	A	Milan	8:00	540				
B	Naples	23:00	C	Naples	7:00	480	✓			✓
B	Naples	23:00	D	Rome	9:00	600		✓	✓	
C	Milan	11:00	A	Milan	8:00	1260	✓			
C	Milan	11:00	B	Milan	19:00	480	✓			
C	Milan	11:00	D	Rome	9:00	1320				
D	Milan	12:00	A	Milan	8:00	1200	✓			
D	Milan	12:00	B	Milan	19:00	420	✓			
D	Milan	12:00	C	Naples	7:00	1140				

Figure 4.10: Feasible pairs

Further possible decomposition

Sometimes, for computational troubles, it could be better treat the problem using a decomposition. To simplify the problem we will suppose we have only one maintenance level. In a first problem we neglect maintenance constraints and we find a lower bound LW on original objective function (Figure 4.14 - check A). Note that if this reduced problem, a kind Traveller Salesman Problem TSP, is unfeasible then our rostering problem is unfeasible. After this step we can try to solve a new problem with a constraint on the number of train asset units (that must be equal to the lower bound LW) and a new objective function given by counting the number of couples of services (i, j) with possible maintenance tasks in the path (Figure 4.14 - check B). Given a solution for this problem, we can formulate a short model to find the final solution (Figure 4.14 - check C). For each edge of the solution with a possible maintenance task we generate a node of a new graph. We call V the set of these nodes. For each couple of nodes of this new graph we calculate the km between them (this is possible because we have the path) and if these km are greater than β_{MC} and lower than γ_{MC} we will add to the graph a new edge (i, j).
We will call A the set of these edges. The final problem will be to

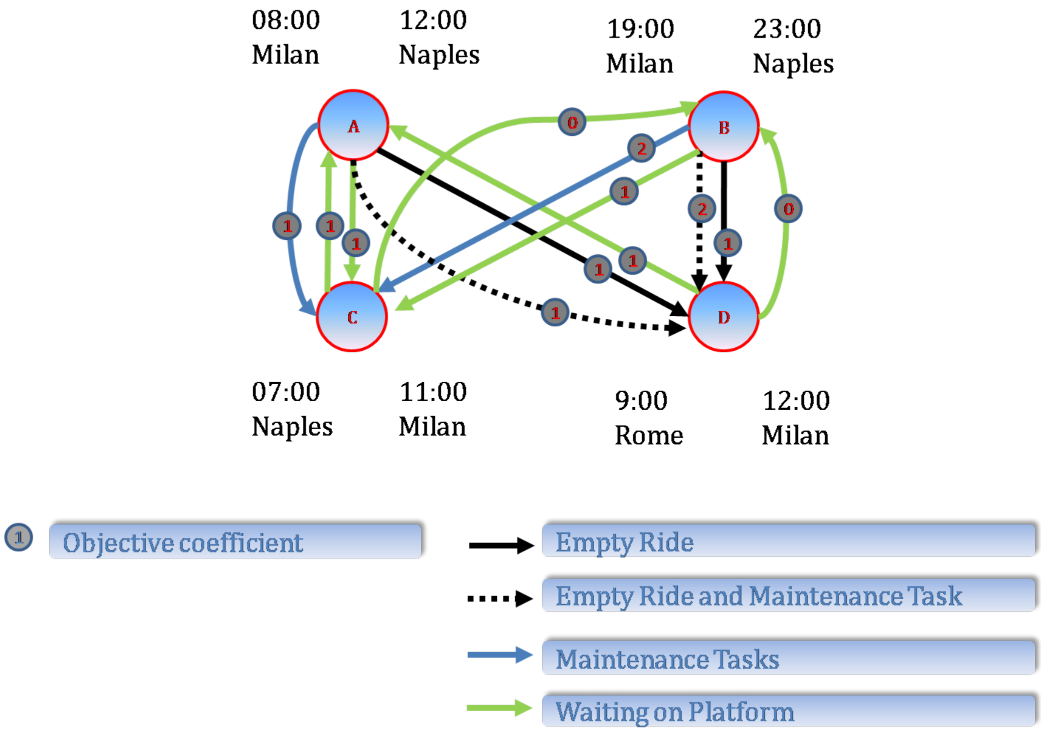

Figure 4.11: Final graph

find a loop on the graph. Therefore we will have a set of variables x_{ij} indicating the edges belonging to the solution. The formulation of this problem will be

$$\sum_{(i,j)\in A} x_{ij} = \sum_{(j,z)\in A} x_{jz} \leq 1 \quad \forall\ j \in V \tag{4.10}$$

Equation (4.10) is useful for the path construction.

$$\sum_{(i,j)\in A} c_{ij} x_{ij} = TOTKM \tag{4.11}$$

Equation (4.11) avoids solutions without nodes or with sub-loops (in this case we will have more maintenance tasks than we need).

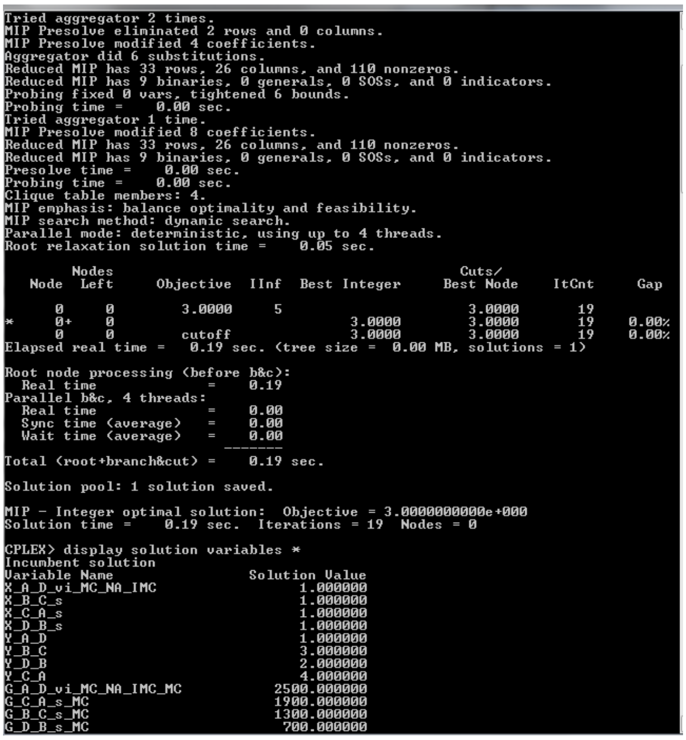

Figure 4.12: solution example

Decomposition numerical example

Let's consider the previous example problem with some little changes

- the MC duration is 600 minutes;
- we have a new maintenance edge between nodes D and A;
- minimum number of km to cover between two maintenance operations = 700.

Our graph is shown in the Figure 4.15.
Solving this problem we will have a LW = 3. Let's use this input data to solve the original problem

- without maintenance constraints;
- with the number of asset units = LW;
- with maximum number of pairs in the solution with maintenance tasks.

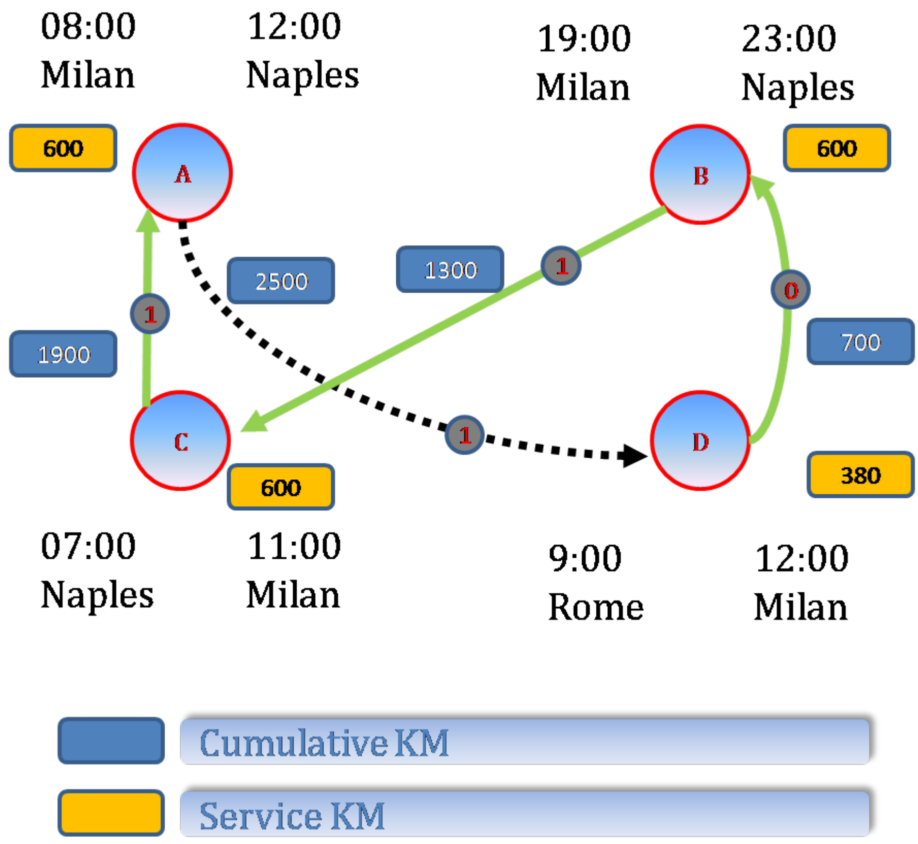

Figure 4.13: Solution Graph

We have a new graph with 3 nodes. For each couple of nodes we can calculate their distance (see Table 4.5).
Erasing edges with km not in the range [700,3000] we will have the new graph shown in the Figure 4.17.
The maintenance solution is represented in the Figure 4.18.
Note that we have to verify the correct position of empty runs to guarantee the respect of initial solution.

Further research

Future research will be dedicated to develop even more sophisticated formulations. We are studying how to extend our model by including scheduling maintenance and platforming operations.

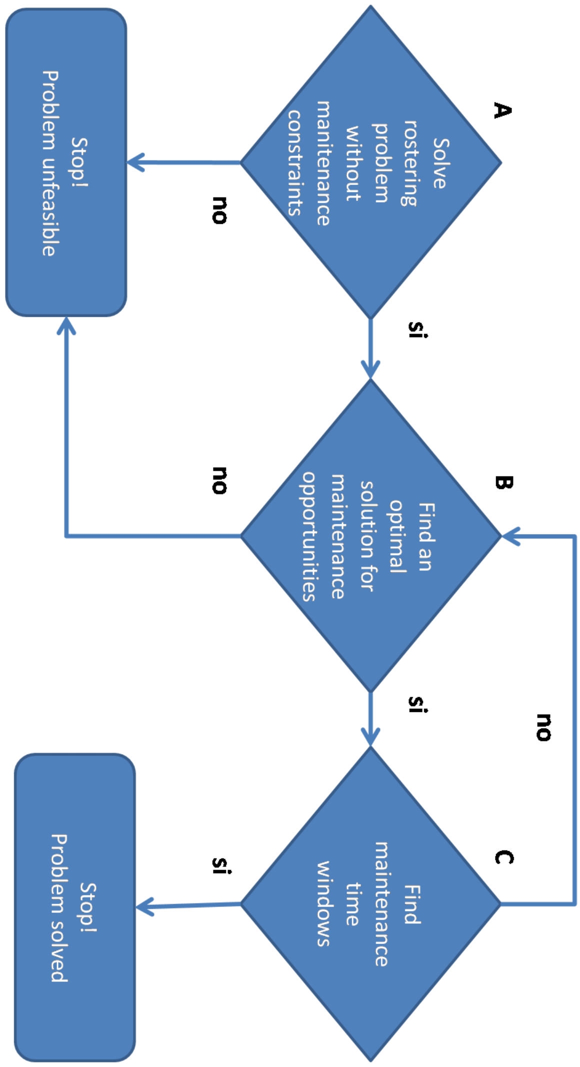

Figure 4.14: Decomposition process

Another issue is the definition of objective functions directly related to the monetary costs of circulation, maintenance and empty rides. Open issues are related to balance the use of resources when routing trains in wide-networks and to the limit the workload for the maintenance operators. Additional research directions should also be focused on developing methods for acyclic timetables and advanced algorithms for complex and large instances. For instance, relaxing the sub-tour elimination constraints could be used to compute lower bounds to the optimal solution of the rostering problem.

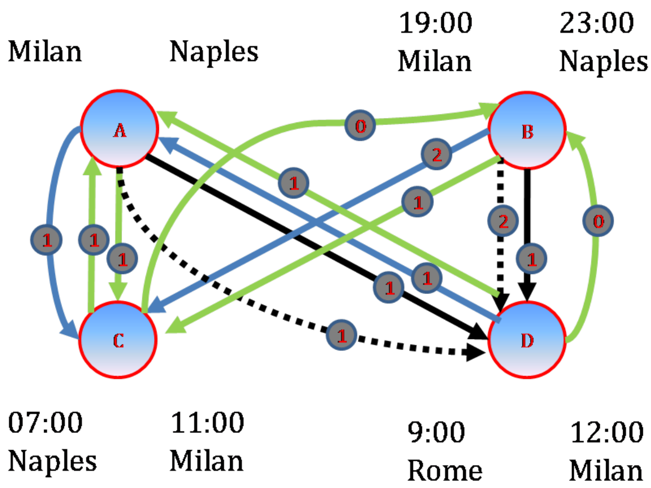

Figure 4.15: Initial graph

	Train From		
Train To	-	380	980
	1800	-	600
	1200	1580	-

Table 4.5: Nodes distances

4.1.2 Non-cyclic timetable

Hypothesis of cyclic timetable is often not verified in the real world. For this reason it is necessary to find different and efficient formulations. For non-cyclic timetables it is necessary to use formulations based on objects “train-day" (a single train service running in a day) which implies obvious troubles of complexity. In other words in case of cyclic timetable a train service has the same characteristics every day. Therefore we could solve the problem by working only on one virtual day. Unfortunately, in the case of non-cyclic timetable, we can not work on smaller problems because we have to take into account the different versioning of each train service. A cyclic timetable solution is useful up to the occurrence of significant changes of timetable which make it unfeasible, while in the case of non-cyclic timetable the solution is feasible only for a range of days. This is not a real problem because train operators

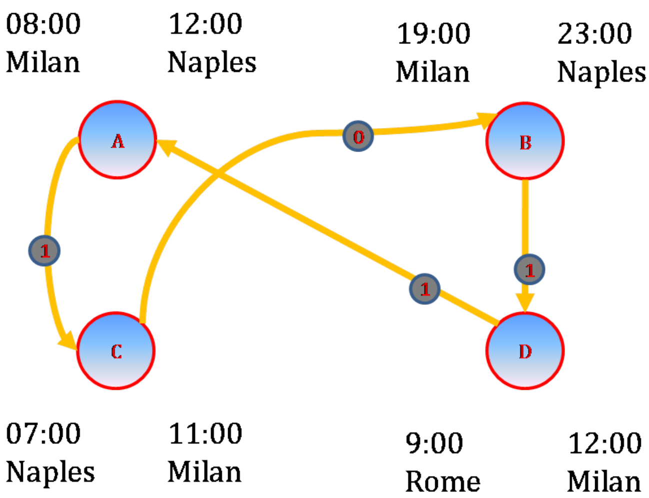

Figure 4.16: Second problem solution

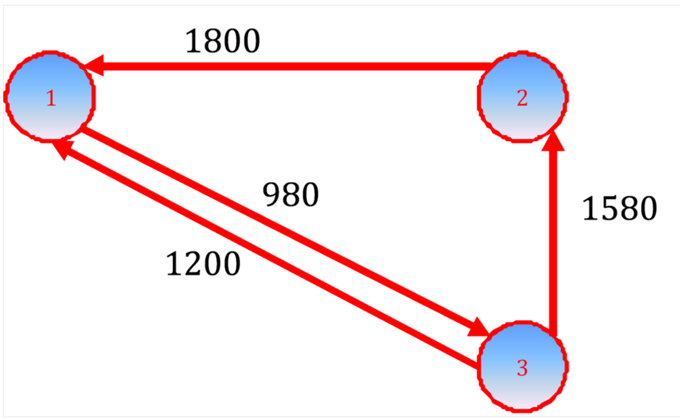

Figure 4.17: third problem graph

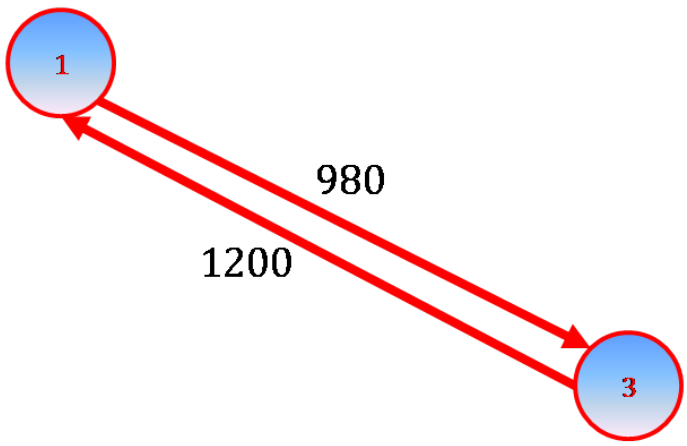

Figure 4.18: Solution

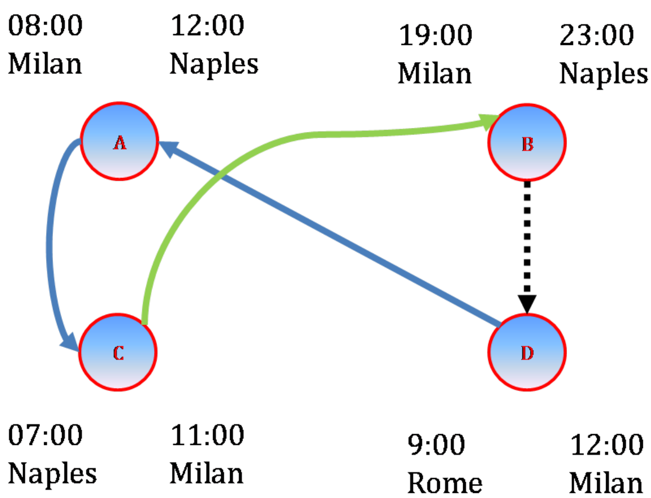

Figure 4.19: Global solution

change their timetables very frequently (in Italy for instance twice in a month). The basic logic of the “Hamiltonian path" used for cyclic timetable formulation is not acceptable and at same time the aspect of initial conditions for each rolling stock asset unit (kilometres run and history of maintenance provided) becomes crucial. The study of this problem has been addressed mainly looking for efficiency in order to carry out a tool to use in practice. A great effort has been made in the search for formulations gradually more efficient. The result is certainly very interesting in relation to the complexity of the problem.
We suppose we have a different timetable for each day in a given time period. The number of objects train-day therefore depends on the mean number of trains in the timetable and on the extension of the reference period. For instance, with reference to high speed trains named “FRECCIAROSSA" provided by Trenitalia, the mean number of services is more than 100 units. In this case, the number of objects train-days to be taken into account will be given approximately by the number of days of the period of validity of the roster multiplied by 100. Assuming we have a reference period of three months we will have more than 9000 objects to be analyzed.
We will try to adapt the model used to solve cyclic timetable to non-cyclic schedules. To understand the problem, let’s construct a graph by using the reference train-days objects as nodes. We will also insert in the graph a new node for each asset unit (a virtual source) and at last a sink node *sn*. Therefore, the number of nodes nn will be given using the following equation.

$$nn = number\ of\ train - day\ objects + number\ of\ asset\ units + 1$$

To complete the graph we need to identify possible links between nodes (graph edges). In order to find all the directed edges of the graph we have firstly to find all possible pairs between train-days. An ordered couple of objects train-days *(i,j)* is connectible if and only if the following conditions are all true

1. the departure time of *j* is greater than arrival time of *i*;
2. the difference of arrival time of *i* and departure time of *j* is included in the interval [*Minimum connection time* , *Maximum connection time*];
3. the arrival station of *i* is the same of departure station of *j*.

Another set of edges is made connecting each source node with all train-day nodes.
We should also add other edges (to take into account empty runs) between couple of nodes that respect the following conditions

1. the departure time of *j* is greater than arrival time of *i*;
2. the difference of arrival time of *i* and departure time of *j* is included in the interval [*Minimum time* to cover the path from arrival station of *i* and departure station of *j* by an empty run , *Maximum time* to cover the path from arrival station of *i* and departure station of *j* by an empty run];
3. The arrival station of *i* is not the same of departure station of *j* and the distance between them is less than *maximum empty runs kilometres allowed.*

A typical two days example of this kind of graph is shown in the Figure 4.20.
To solve our problem of minimization of the number of train units needed, we need to find the minimum number of paths, starting from a source node, such that all nodes are visited once and only once. Given a path, the initial node indicates which unit was chosen to cover the services represented by the other nodes in the path. In this kind of graph it is not possible to create subloops during the search of paths.
Other constraints to take into account are:

- maintenance deadlines;
- minimum maintenance efficiency;
- maximum number of empty runs allowed.

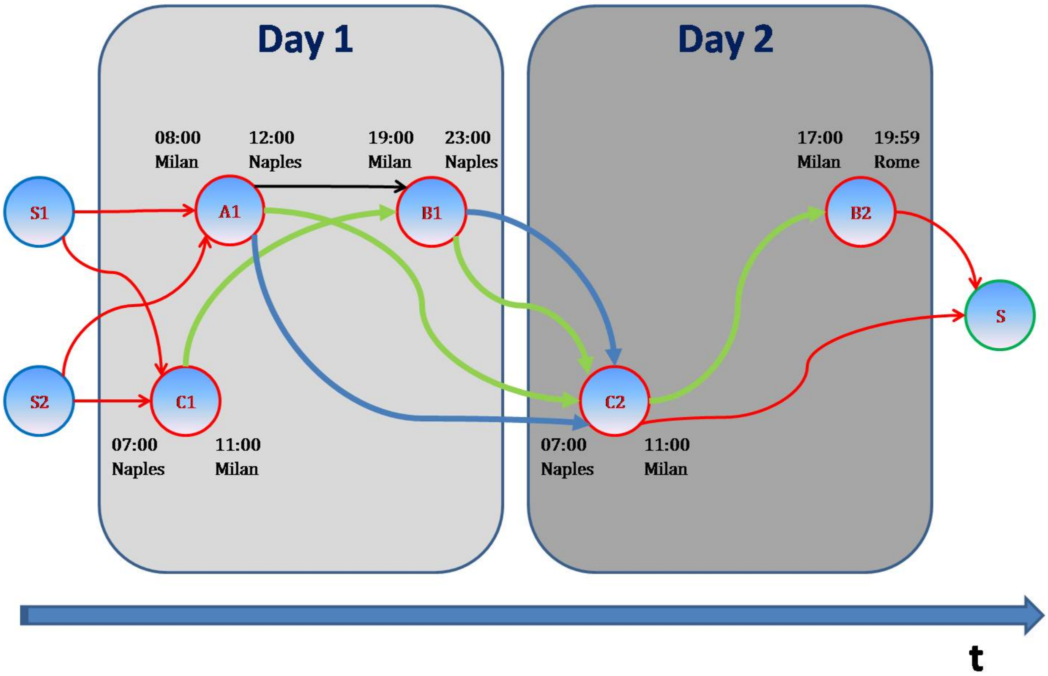

Figure 4.20: Non-Cyclic Timetable Graph 1

From the mathematical point of view, we need to solve a large mixed integer problem.
We provide a list of notation before to go so far.

- S_o is the set of source nodes,
- V is the set of nodes related to train-day services,
- V^+ is the set of nodes given by $V \bigcup S_o$,
- V^p is the set of nodes given by $V \bigcup \{sn\}$,
- A is the set of edges of the graph,
- A^- is the set of empty runs edges,
- M is the set of maintenance tasks,
- L is the set of maintenance levels,
- sn is the sink node,
- l is number of maintenance levels,
- $m \mapsto l$ means that the task m delete kilometres of level l,
- $Res_{s,l}$ are initial kilometres related to maintenance level l for the asset unit s,

- $timeleg_{i,j}$ is the timeleg between the services i and j,
- Km_i are the kilometres of train service i,
- β_l is the lower bound on the kilometres traveled by a train between consecutive executions of maintenance task including the level l,
- γ_l is the upper bound on the kilometres traveled by a train between consecutive executions of maintenance task including the level l,
- BM is a BIG integer (i.e. a sufficiently large number),
- α is a bound related to the maximum number of empty rides allowed in a solution.

Formulation 1

Problem variables
This formulation considers two types of variables:

- X is a set of binary variables such that $x_{ijz} \in X$ is equal to 1 if arc (i, j, z) belongs to the solution (0 otherwise);
- G is a set of real variables related to kilometres performed.

Accordingly with the cyclic formulation, the variable $g^m_{ijz} \in G$ is used to assure that the kilometres traveled by an asset unit between two consecutive executions of tasks involving the same maintenance level l is always between β_l and γ_l. In this formulation we will use an edge for each maintenance task allowed, for empty runs and for waiting times. We will have a variable g for each edge and for each maintenance level (we have seen the same thing in the cyclic timetable formulation). In the next formulations we neglect the case of empty runs with also maintenance operations.

Objective function
The target to achieve is the minimization of train units required to cover all train services for all days in the period of interest. Therefore, the objective function to minimize, is expressed as the number of paths found on the graph.

$$\sum_{(i,j,m) \in S_o x V x M} x_{ijm}$$

Path constraints
For each train-day node we have to add the set of Constraints 4.12 in order to build paths on the graph involving all nodes related to train services. A node must appears in one and only one path.

$$\left.\begin{array}{r} \sum_{(j,m)\in VxM} x_{ijm} = 1 \\ \sum_{(j,m)\in VxM} x_{jim} = \sum_{(k,z)\in VxM} x_{ikz} \end{array}\right\} \forall\ i \in V \tag{4.12}$$

The set of Equatione 4.12 can not be applied to source nodes. For each source node we have to add the simpler set of Constraints 4.13 because sources have no incoming edges and they can not belong to solution.

$$\sum_{(j,m)\in VxM} x_{ijm} \leq 1 \quad \forall\ i \in S_o \tag{4.13}$$

Bound on maximum number of empty runs
We have also to impose the upper bound on number of expensive empty runs in the solution.

$$\sum_{(i,j,m)\in A^-} x_{ijm} \leq \alpha \tag{4.14}$$

Maintenance constraints
To control efficiency of maintenance programming, we use variables g as in the model shown for cyclic timetable to store the kilometres cover by an asset unit during its path. Maintenance constraints could be written in the follow manner

$$g_{ijm}^{l} \leq \gamma_l \; x_{ijm} \quad \forall \; (i, j, m, l) \in VxV^{p}xMxL \tag{4.15}$$

Constraints 4.15 impose a limit to maximum number of kilometres run by an asset unit between two maintenance tasks involving the level l. Variables $g_{i,j,m}$ will be 0 if the edge (i, j, m) does not belong to the solution.

$$g_{ijm}^{l} \geq \beta_l \; x_{ijm} \quad \forall \; (i, j, m, l) \in VxV^{p}xMxL : m \mapsto l \tag{4.16}$$

Constraints 4.16 impose a limit to minimum number of kilometres run by an asset unit between two maintenance tasks involving the level l. We can not provide maintenance involving the level l before the unit has run the minimum number of kilometres β_l.

$$\sum_{(j,a) \; \in \; V^{p}xM} g_{ija}^{l} = Km_i + \sum_{(k,b) \; \in \; V^{+}xM: \; not \; b \; \mapsto l} g_{kib}^{l} \quad \forall \; i \in V \tag{4.17}$$

Constraints 4.17 are used to increase the kilometers covered by a train unit asset after a train service i.

$$g_{sim}^{l} = Res_{s,l} \; x_{sim} \quad \forall \; (s, l) \in S_o \; x \; L \tag{4.18}$$

We have also to consider initial status for each train asset unit in terms of kilometres covered for each maintenance level. So we will add the set of Equations 4.18.

Formulation 2

Formulation 1 can be improved from the point of view of efficiency. We can use variables g as node variables instead of edge variables. Generally the number of edges is very much greater than the number of nodes and therefore we can decrease the number of variables by this change. In this case there will be a drastic reduction of real variables. In the Formulation 1 real variables were $\propto VxV$ while in this case they are $\propto V$. This change could have a very big impact on lp model writing and solving time. On the other hand we will add a new set of binary variables. We can also neglect the sink node because variables g measuring the kilometres performed after a service i are now represented on a node i. In this case we do not need any final path edge to report the final condition of the asset units (see Figure 4.21).

Problem variables
This formulation considers a new type of variables h_i^l (with $i \in V$ and $l \in L$) that are equal to 0 if a maintenance task $m : m \mapsto l$ is provided in the node preceding i in the path (otherwise 0). We will define H as a set of real variables h_i^l.

Objective function
We will use the same target to aim and the same objective function to minimize of the Formulation 1.

$$\sum_{(i,j,m)\in S_o x V x M} x_{ijm}$$

Path constraints
These equations are quite similar to path constraints we have seen in the previous formulation. For each train-day node we must assure it belongs to one and only one path of the solution.

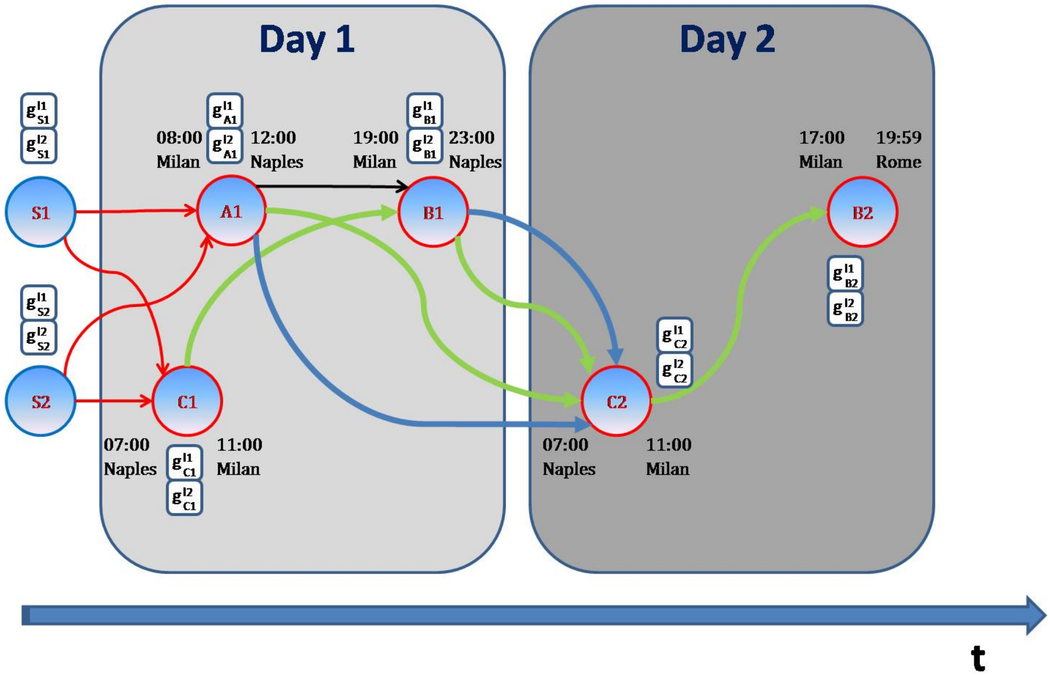

Figure 4.21: Non-Cyclic Timetable Graph 2

$$\left.\begin{array}{r}\sum_{(j,m)\in VxM} x_{ijm} \leq 1 \\ \sum_{(j,m)\in VxM} x_{jim} = 1 \quad \forall\, i \in V\end{array}\right\} \forall\, i \in V \tag{4.19}$$

A first difference with the previous model is on the first part of Constraints array 4.19. In this case we do not need to close the path with the sink node. In other words the final node for a path has not any outcoming edge.

$$\sum_{(j,m)\in(VxM)} x_{ijm} \leq 1 \quad \forall\, i \in S_o \tag{4.20}$$

Accordingly with Formulation 1, we will add for each source node the set of Constraints 4.20 to avoid using the same unit asset in more paths.

Bound on maximum number of empty runs
We use again the following inequalities to check the respect of upper bound on empty runs

$$\sum_{(i,j,m)\in A^-} x_{ijm} \leq \alpha \tag{4.21}$$

Maintenance constraints
These constraints are very different with respect previous formulation. The new real variables h are used to calculate kilometres performed after a provided service i. These variables are related to maintenance operations provided before the service i.

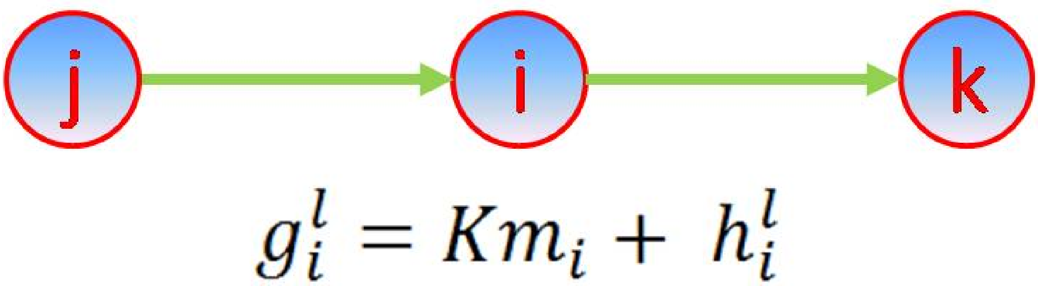

Figure 4.22: Relationship between variables g and h

The constraints sets 4.22 to 4.25 are used to assure that v_i^l is 0 if the solution edge incoming in the node i provides maintenance $m \mapsto l$ (otherwise v_i^l must be equal to variable g_j^l where node j precedes node i).

$$g_i^l = Km_i + h_i^l \quad \forall\, i \in V \tag{4.22}$$

Equation 4.22 shows how variables g and h are linked. We can calculate the value of g_i^l by the sum of kilometres of train service i (Km_i) and v_i^l (see Figure 4.22).

$$h_j^l \leq (1 - x_{ijm})\ \gamma_l \ \ \forall\ (i,j,m,l) \in V^+x\ Vx\ Mx\ L : m \mapsto l \tag{4.23}$$

Equation 4.23 gives an upper bound to the value of variable h_j^l.

$$h_j^l \geq g_i^l - BM(1 - x_{ijm}) \ \ \forall\ (i,j,m,l) \in V^+x\ Vx\ Mx\ L : not\ m \mapsto l \tag{4.24}$$

$$h_j^l \leq g_i^l + BM(1 - x_{ijm}) \ \ \forall\ (i,j,m,l) \in V^+x\ Vx\ Mx\ L : not\ m \mapsto l \tag{4.25}$$

Equations 4.24 and 4.25 impose that h_j^l is equal to value of g_i^l (where node i precedes the node j) if they are linked by a maintenance edge in the solution (otherwise h_j^l = 0).

$$g_i^l \geq \beta_l \sum_{(i,j,m) \in (V^+ \cup V \cup M):\ m \mapsto l} x_{ijm}\ \forall\ (i,l) \in V^+\ x\ L \mapsto l \mapsto l \tag{4.26}$$

We use the Equations 4.26 to verify the respect of the minimum maintenance efficiency when a train unit come to workshop.

$$g_s^l = Res_{s,l} \quad \forall\ (s,l) \in S_o\ x\ L \tag{4.27}$$

We have at last to consider initial status for each train asset unit. So we will add the equations 4.27.

Formulation 3

Formulation 2 and Formulation 1 have too many binary variables. For this reason sometimes commercial tools could not be able to solve quickly this kind of problem. We can improve these formulations trying to reduce as much as possible the number of decisional variables. In the previous formulations we have had a binary variable x_{ijm} for each edge of the graph representing the problem. Generally for each couple of linkable nodes we have more edges. The number of binary variables was $\propto$ $(number\ of\ couple\ of\ nodes) * (number\ of\ maintenance\ tasks)$.
We can reduce the problem using just one edge for each linkable nodes pairs. We can also use, accordingly with the Formulation 2, the variables g as node variables.

Problem variables
This formulation considers a new type of variables: P_j^l is a set of binary variables that are equal to zero if the level of maintenance l is provided on the edge solution incoming in the node j. In this case variables $x_{i,j,m}$ become $x_{i,j}$ because the maintenance tasks are not edges features.

Objective function
Our goal is the minimization of train units required to cover all train services for all days in the period of interest. Similarly to Formulation1 and Formulation 2 we would use less train unit we can.

$$\sum_{(i,j)\in S_o x V} x_{ij}$$

Path constraints

$$\left.\begin{aligned} \sum_{j \in V} x_{ij} \leq 1 \\ \sum_{j \in V} x_{ji} = 1 \end{aligned}\right\} \forall\, i \in V \tag{4.28}$$

For each train-day node we have to add the next set of constraints in order to build paths on the graph. Each train-day node must appear in one and only one path of the solution. To do this we introduce the constraints array 4.28.

$$\sum_{j \in V} x_{ij} \leq 1 \quad \forall\, i \in S_o \tag{4.29}$$

To assure that each asset unit could run at most on one path, we add the constraints set 4.29.

Bound on maximum number of empty runs
We have to add the constraint on the maximum number of empty runs.

$$\sum_{(i,j) \in A^-} x_{ij} \leq \alpha \quad \forall\, i \in S_o \tag{4.30}$$

Maintenance constraints
Following constraints sets 4.31 to 4.35 are used to assure that h_i^l is 0 if the solution edge incoming in node i provide maintenance

$m \mapsto l$ (otherwise h_i^l must be equal to variable g_j^l).

$$g_i^l = Km_i + h_i^l \quad \forall\ i\ \in V \tag{4.31}$$

Equation 4.31 are the same we have used in th Formulation 2. The value of g_i^l is given by the sum of kilometres of train service i (Km_i) and v_i^l (see Figure 4.22).

$$h_j^l \geq g_i^l - (1 - x_{ij}) * BM - p_j^l * BM \ \ \forall\ (i, j, l) \in V^+\ x\ V\ x\ L \tag{4.32}$$

$$h_j^l \leq g_i^l + (1 - x_{ij}) * BM + p_j^l * BM \ \ \forall\ (i, j, l) \in V^+\ x\ V\ x\ L \tag{4.33}$$

$$h_j^l \leq (1 - p_j^l)\ \gamma_l \ \ \forall\ (j, l) \in Vx\ L \tag{4.34}$$

Constraints 4.32 to 4.34 impose that variable h_j^l is equal to g_i^l if on the edge (i, j) there is not any maintenance task involving the level l. These constraints use a BIGM approach to active themselves only if some conditions are true.

$$p_j^l \leq \frac{g_i^l}{\beta_l} + (1 - x_{ij})BM \tag{4.35}$$

We use the Equations 4.35 to verify the respect the minimum maintenance efficiency when a train unit come to workshop.

$$g_i^l = Res_{i,l} \quad \forall\ (i, l) \in S_o\ x\ L \tag{4.36}$$

We have also to consider initial status for each train asset unit as we have seen in the Formulation 1 and Formulation 2. For this reason we will add the set of equations 4.36.
We have worked only on maintenance level and never on maintenance tasks. This is a very important statement to improve this complex formulation. Probably we can not be sure that a specific combination could be provided within the time between two following train services. In the formulations we have seen before we added a maintenance edge if and only if there was enough time to provide it. But in this model we have just one edge for all maintenance tasks. For this reason we have to add another constraints set. The possible alternatives are:

- if the duration of a maintenance job can be calculated by the sum of the durations t^l of maintenance levels involved, we can use the inequality 4.37 to verify the feasibility of the solution

$$\sum_{l \in L} p_j^l * t^l \leq \sum_{i \in V^+} x_{ij} * timeleg_{i,j} \quad \forall\ j \in V \tag{4.37}$$

- If the larger maintenance levels include the smaller, we can opt for the following set of constraints checking that the time window at our disposal is enough to provide the most expensive level

$$timeleg_{i,j} \geq -(1-p_j^l)BM + t_l - (1-x_{ij})BM \quad \forall\ (i,j) \in V^+xV \tag{4.38}$$

We will use these constraints set in next tests.

Models comparison

We try to give a short comparison on efficiency of the three models we have seen so far. Let's consider the following instance

- non-cyclic timetable with 100 trains per day;
- the reference period has 30 days;
- feasible interval for pairing is from 60' to 4320';
- three maintenance levels;
- the maximum number of empty runs in the roster is 1;
- the maximum empty run kilometres are 300 (so only between Rome and Naples);
- only one workshop (located in Naples);
- empty runs speed average 50 Km/h.

We have compared the three models using number of constraints, number of binary variables and real variables. It appears (see Table 4.6) that the third model is the best because the number of decisional variables and of constraints is smaller than the others.

Table 4.6: Comparison between models part1

Formulation	Num Constraints	Num Binary Variables	Num Real Variables
Model 1	963014	194000	582000
Model 2	647037	194000	3600
Model 3	542418	54200	3600

We have solved the same instance using Cplex 12 and the results are shown in the Table 4.7. As we can observe, the writing model time and solving time are very short using Formulation 3. We will use the formulation 3 to solve a workshop optimization in the new parts of this thesis.

Table 4.7: Comparison between models part 2

Formulations	Writing Model Time (sec)	Solve Time (sec)	Total
Model 1	603	26	629
Model 2	535	9	544
Model 3	146	4	150

Chapter 5

Workshop Optimization

This chapter describes new mixed-integer linear-programming formulations for the maintenance scheduling problem faced by Trenitalia (train operating company) managers, with input data taken from the rolling stock rostering plan. The computational results are carried out on a Trenitalia's maintenance site located in Naples. The solutions computed via a commercial MILP solver are compared with practical solutions. A relevant cost reduction is possible by using the proposed formulations. We also show how the proposed method can be used as an effective tool to absorb real-time timetable perturbations while respecting the agreed level of service. The strategic relevance of maintenance scheduling, is due to the reduction of needs (such as platforms) and to the enhancement of quality standards (such as vehicle reliability and cleaning). This section studies how to manage the maintenance operations in order to increase availability and reliability of railway services.

The Naples workshop is one of the key points of the Italian railway system for several maintenance services. The layout of the studied workshop consists of various indoor/outdoor tracks for light maintenance, wheel profiling, washing, cleaning, inspection and other services (see Figure 5.1).

Each row of Table 5.1 indicates a typical maintenance activity at the workshop with its identifier (Column 1), description (Column 2) and expiry (Column 3) while in the Table 5.2 are shown mainte-

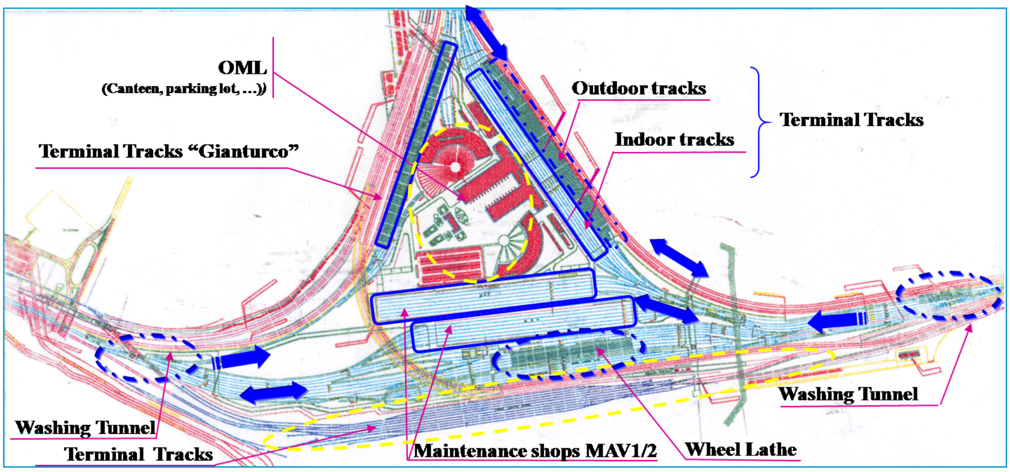

Figure 5.1: Naples workshop layout

nance activities mean durations (Column 2). The activities named I, II, and III are similar in terms of maintenance operations and duration, but have three different expiries. The activities named IV, V and VI have expiries that are multiples of 60.000 km. Specifically, Activity V incorporates Activity IV and extends it with an addition of ultrasonic flaw detection. Similarly, Activity VI extends Activity V by adding the turning of wheel flanges.

Table 5.1: Typical maintenance activities

ID	Short Description	Expiry
I	Pantograph and Bogies Check	Each return to workshop
II	Pantograph and Bogies Check	7.500 Km (+/- 10%)
III	Pantograph and Bogies Check	30.000 Km (+/- 10%)
IV	Pantograph, Bogies and Wheel Flanges Check	60.000 Km (+/- 10%)
V	Ultrasonic Flaw Detection	180.000 Km (+/- 10%)
VI	Wheel Truing	360.000 Km (+/- 10%)

Figure 5.2 shows two different routes in the workshop, named Routes #1 and #2.
Both routes visit parts of the workshop where it is possible to provide maintenance operations (indoor terminal tracks for both routes), washing (washing tunnel) and cleaning (outdoor terminal tracks for Route #1 and terminal Gianturco tracks in Route #2).

Table 5.2: Typical maintenance activities durations

ID	Duration [h]
I	8
II	8
III	8
IV	24
V	72
VI	108

Note that Route #1 has two movements between workshop sites, while Route #2 has three movements.

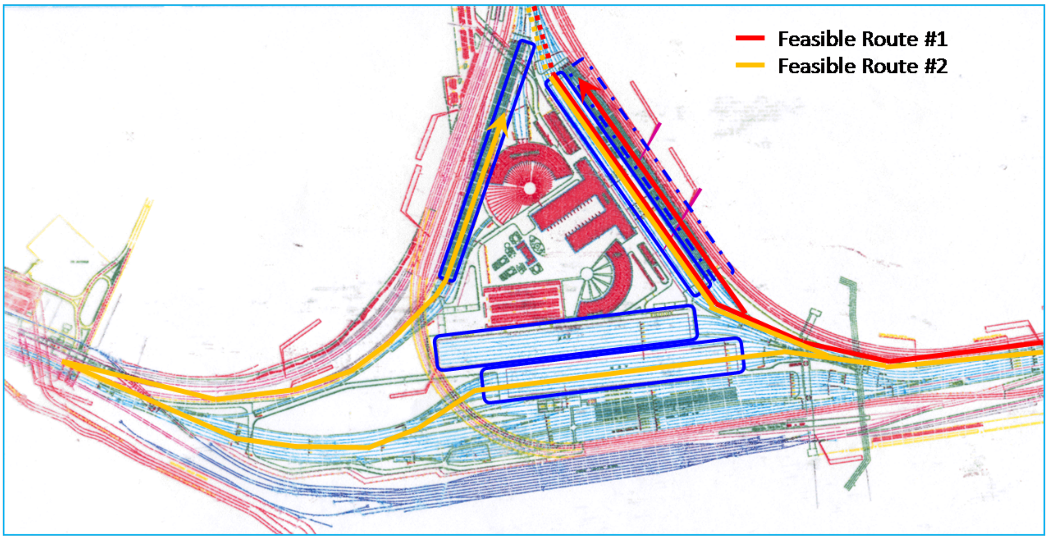

Figure 5.2: Typical routes in the workshop

In practice, the high complexity of the workshop problem requires to divide it into smaller sub-problems. Since rail companies tend to schedule first the resources with a long time of acquisition (e.g. rail tracks or asset units) and then the resources with higher degree of flexibility (e.g. human resources), this work is focused on the former type of resources.
We have considered also parallel tasks (see Figure 5.3). So we can manage the slack time of the shortest activities in a maintenance package. Generally a maintenance project solution is a sequence of more than one maintenance packages.

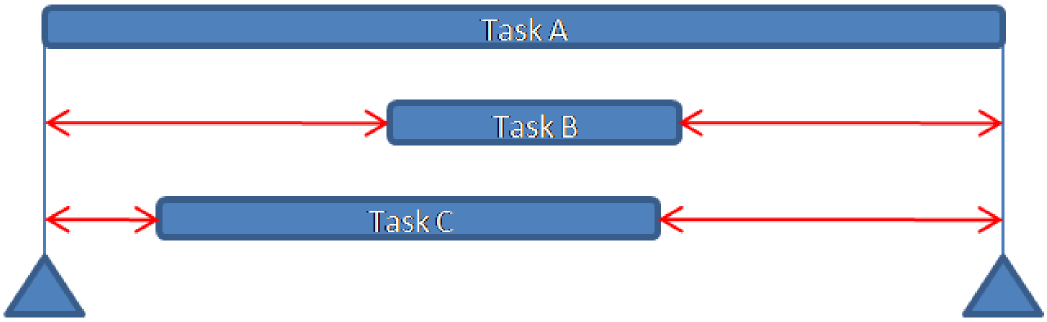

Figure 5.3: Parallel tasks

The general problem addressed here can be defined as follows: Given timetables, rolling stock assets, maintenance operations, a rolling stock circulation solution has to be computed with minimum costs, that are expressed in terms of the number of used asset units, empty runs, and train movements between station platforms and the workshop.
Figure 5.4 shows the framework proposed to solve the problem. Each coloured module (i.e. roster, station and workshop) represents a specific sub-problem and is divided in input and outputs data, that are exchanged with the other modules. A sequential method is developed in order to integrate the solutions provided by each module. Our approach is to first solve the rostering problem for each asset unit type and then solve the other two modules involving passenger stations and workshop operations. An automatic procedure is under development to manage the interaction between the modules and a feedback information is returned in case an infeasibility is provided by some modules. In the following, we briefly describe each module and the models we developed to solve it.

5.1 Rolling stock rostering

The rostering module computes a rolling stock roster that covers a set of commercial services and minimizes the costs related to the

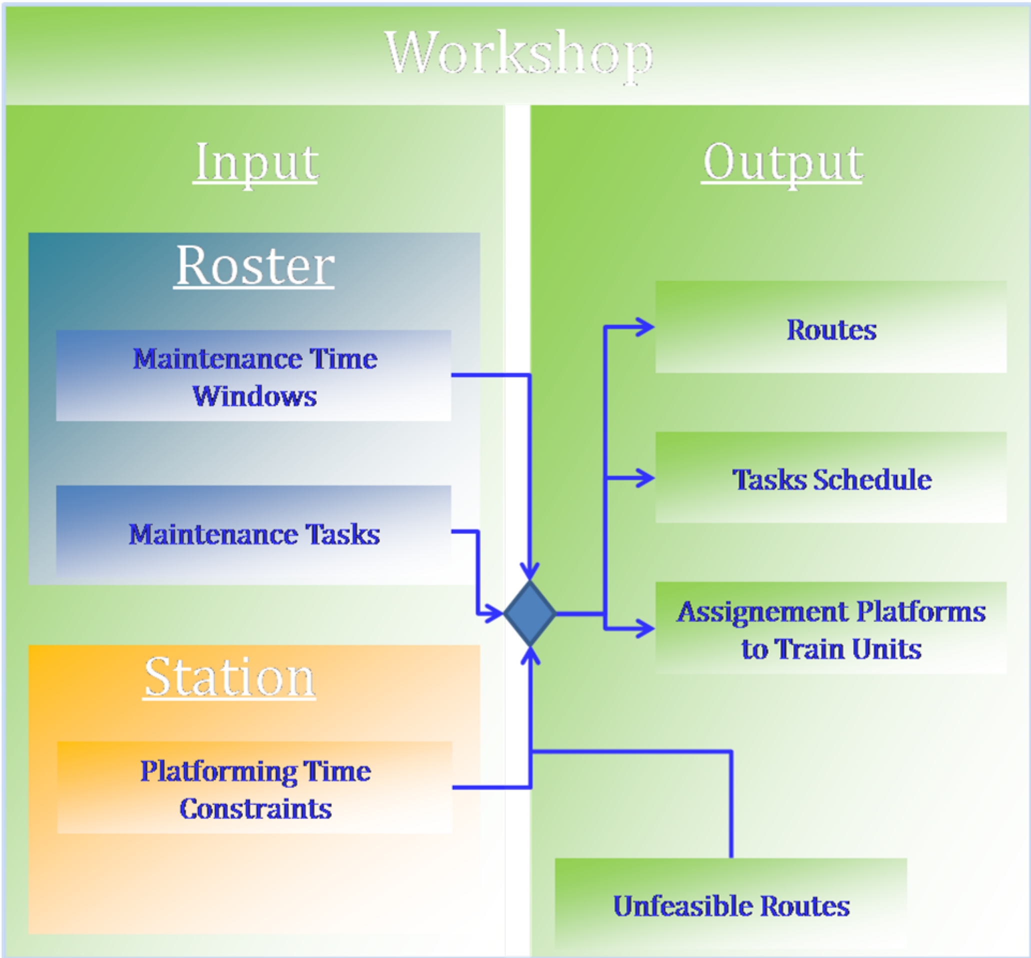

Figure 5.4: Interaction between the roster, station and workshop modules

asset units, including the empty runs. Specifically, this module optimizes the distance run by asset units of various types between consecutive maintenance operations. We suppose the timetable is cyclic and the assignment of commercial services per asset kind follows brand promise principles. As described in Giacco et al. [7] and in the previous section, the rolling stock rostering problem corresponds to find a Hamiltonian path on a graph made by commercial services (nodes) and by feasible pairs (edges), representing service or maintenance activities to be provided. This problem is treated as a kind of traveling salesman problem with additional constraints and variables to guarantee the respect of maintenance expiry and to guarantee maintenance efficiency. The output is a cyclic roster including the schedule of maintenance activities, as time windows $[t_u^{ini}, t_u^{fin}]$ to provide the required maintenance, and an assigned workshop location.

5.2 Station and workshop scheduling

Maintenance operations have to be scheduled in passenger stations and workshops. There are constraints on the maintenance activities that have to be performed by each train and on the time windows that are defined in the rolling stock rostering plan. A workshop must manage some asset units simultaneously and must be able to absorb small perturbations of the circulation. A time window for the maintenance operations between two commercial services is thus considered. Generally, the time windows are bigger than the sum of all activities and the workshop has some recovery time. Also, the workshop could be considered as a space buffer to store trains. An objective function of practical interest is the minimization of the number of train movements in the workshop area.

The following section will present a new formulation for the workshop maintenance problem with minimization of train movements within the workshop. The system robustness will be investigated, i.e. the feasibility of the workshop schedule in presence of disturbances.

5.3 First approach

This section describes a big M formulation for the workshop scheduling module. We call *job* a sequence of scheduled waiting times and maintenance operations. The problem variables are the initial and final times for each maintenance activity. We also model *alternative jobs* as additional variables.

List of notations

- $A = [a_1, a_2, \dots a_a]$ is the activities set that the workshop can provide,
- a is the number of the activities belonging to *A*,
- $U = [u_1, u_2, \dots, u_u]$ is the set of asset units we have to analyze,
- u is the number of asset units belonging to *U*,
- $R = [r_1, r_2, \dots, r_r]$ is a feasible route (or sequence) in the workshop,
- r is the number of feasible routes in the workshop,
- t_u^{ini} is the time in which the rolling stock *u* arrives at the workshop,
- t_u^{fin} is the time in which the rolling stock *u* should leave the workshop,
- $j_{ru} = [r, (t_u^{ini}, t_u^{fin})]$ is the job on route *r* for the asset unit *u* with time window (t_u^{ini}, t_u^{fin}),
- $J = [j_1, j_2, \dots, j_j]$ is the set of feasible jobs,
- n is the number of feasible jobs,
- $B = [b_1, b_2, \dots, b_b]$ is the set of workshop resources (platforms),
- b is the number of resources of the workshop,
- x_{ruob} is a binary variable that assumes value 1 if the *o*-th task in route *r* of asset unit *u* is assigned to resource *b*,
- o_r^{max} is the maximum number of elements in the sequence of route *r*,
- y_{ru} is a binary variable that assumes value 1 if route *r* is chosen for asset unit *u*,

- BM is a BIG integer (i.e. a sufficiently large number),
- s_{ac} is a binary variable that assumes value 1 if activity *a* if scheduled before *c*, in which *a* and *c* do not belong to the same asset unit,
- w_{ac} is the binary variable that assumes value 1 if the end of activity *a* precedes the start of activity *c* by more than 1440 minutes (i.e. 1 day),
- w_{ca} is the binary variable that assumes value 1 if the end of activity *c* precedes the start of activity *a* by more than 1440 minutes (i.e. 1 day),
- k_r is the number of movements within the route *r*,
- $t^i_{ruo}(t^f_{ruo})$ is a non-negative variable that represents the start (end) time of the *o*-th task in route r of unit u begins (ends),
- $ar(o)$ is the *o*-th task of route *r*,
- $pr(a)$ is the position of activity *a* in the route *r*,
- $dar(o)$ is the maximum allowed duration of activity $ar(o)$.

Constraints

Model constraints are of four types:

- one and only one alternative job is chosen for each asset unit,
- no overlap between activities is allowed on each resource,
- only one assignment is possible for each maintenance activity,
- time windows and maximum allowed duration of activities are respected.

We next show the inequalities that model the various types of constraints.

$$\sum_{r \in R} y_{ru} = 1 \quad \forall\ u \in U \tag{5.1}$$

Inequality (5.1) imposes that one and only one route must be chosen for each train.

$$\sum_{b \in B} x_{ruob} = y_{ru} \quad \forall \ (r, u, o) \in \ R \times U \times O \tag{5.2}$$

Inequality (5.2) constrains all the activities of a chosen route to be assigned to workshop resources

$$\begin{aligned} & t^f_{ruo} - t^i_{ruo} \geq dar(o) \sum_{b \in B} x_{ruob} \quad with \quad dar(o) = max(d_j) \\ & \forall \ (r, u, o) \in \ R \times U \times O \quad with \ j \ parallel \ tasks \ on \ the \ route \ \mathsf{r} \\ & and \ in \ the \ position \ \mathsf{o} \end{aligned} \tag{5.3}$$

for parallel activities in route *r* and position *o*.
Inequality (5.3) is to check whether a sufficient processing time is given to each maintenance activity. The duration of activities is 0 if route *r* is not chosen for asset unit *u*, otherwise it represents the processing time for unit *u* on the platform *b*.

$$t^i_{ruo} = t_{ini_u} \ with \ o = 1 \tag{5.4}$$

In Constraint (5.4), workshop must manage the rolling stock constraints delivered by the rostering optimizer on the time window start, including scheduled waiting times.

$$t^f_{ruo} = t^{fin}_u \ with \ o = o^{max}_r \tag{5.5}$$

In onstraints (5.5), the workshop must manage the rolling stock constraints delivered by the rostering optimizer on the time window end.

$$t^i_{r_1u_1c} \geq t^f_{r_2u_2a} - BM(1 - s_{ac}) - BM(1 - x_{r_2u_2pr(a)b}) - BM(1 - x_{r_1u_1pr(c)b})$$
$$\forall \quad b, r_1, r_2, u_1, u_2 \; with \; u_1 \neq u_2 \tag{5.6}$$

$$t^i_{r_2u_2a} \geq t^f_{r_1u_1c} - BMs_{ac} - BM(1 - x_{r_2u_2pr(a)b}) - BM(1 - x_{r_1u_1pr(c)b})$$
$$\forall \quad b, r_1, r_2, u_1, u_2 \;\; with \; u_1 \neq u_2 \tag{5.7}$$

Inequalities (5.6) and (5.7) are used to avoid overlaps on each resource. If two activities are scheduled on the same resource and activity a precedes activity c, the start of c must be scheduled after the end of a. For this reason, given a resource b, Constraint (5.6) guarantees that $t^i_{r_1u_1c} \geq t^f_{r_2u_2a}$ if a is scheduled before c ($s_{ac} = 1$), a is assigned to resource b ($x_{r_2u_2pr(a)b} = 1$) and c is assigned to resource b ($x_{r_1u_1pr(c)b} = 1$). Similarly, Constraint (5.7) guarantees that $t^i_{r_2u_2a} \geq t^f_{r_1u_1c}$ if c is scheduled before a ($s_{ac} = 0$), a is assigned to resource b ($x_{r_2u_2pr(a)b} = 1$) and c is assigned to resource b ($x_{r_1u_1pr(c)b} = 1$). Let's now suppose that overlaps of activities beyond the end of a work day should be avoided. For modelling this situation, Inequalities (5.8) and (5.9) are necessary:

$$t^f_{r_1u_1c} - 1440 \leq t^i_{r_2u_2a} + BM(1 - s_{ac}) + BM(1 - x_{r_2u_2pr(a)b})$$
$$+BM(1 - x_{r_1u_1pr(c)b}) \; \forall \; b, r_1, r_2, u_1, u_2 \; with \; u_1 \neq u_2 \tag{5.8}$$

$$t^f_{r_2u_2a} - 1440 \leq t^i_{r_1u_1c} + BMs_{ac} + BM(1 - x_{r_2u_2pr(a)b}) + BM(1 - x_{r_1u_1pr(c)b})$$
$$\forall\ b, r_1, r_2, u_1, u_2\ with\ u_1 \neq u_2 \tag{5.9}$$

If activity a precedes c, both activities are scheduled on resource b and c ends after midnight ($t^f_{r_1u_1c} \geq 1440$), we have to check that the end of c and the start of a do not overlap. To this aim, Inequality (5.8) checks if the completion of c minus 1440 is scheduled before the start of a. This constraints set is valid even if $t^f_{r_1u_1c} < 1440$.
Inequalities (5.9) are used when activity c is scheduled before activity a. These constraints only work if two consecutive activities on the same resource are separated by less than 1440 minutes. Alternatively, Inequalities (5.8) and (5.9) have to be replaced by Inequalities (5.10) – (5.15):

$$w_{ac} > (t^i_{r_1u_1c} - t^f_{r_2u_2a} - 1440)/2880\ \forall\ r_1, r_2, u_1, u_2, a, c \tag{5.10}$$

$$w_{ac} \leq 1 + (t^i_{r_1u_1c} - t^f_{r_2u_2a} - 1440)/2880\ \forall\ r_1, r_2, u_1, u_2, a, c \tag{5.11}$$

With Inequalities (5.10) and (5.11), the variable w_{ac} is set to 1 if the end of activity a and the start of activity c are separated by more than 1440 minutes.

$$w_{ca} > (t^i_{r_2u_2a} - t^f_{r_1u_1c} - 1440)/2880\ \forall\ r_1, r_2, u_1, u_2, a, c \tag{5.12}$$

$$w_{ca} \leq 1 + (t^i_{r_2u_2a} - t^f_{r_1u_1c} - 1440)/2880 \; \forall \; r_1, r_2, u_1, u_2, a, c \tag{5.13}$$

Similarly, with Inequalities (5.12) and (5.13), the variables w_{ac} are set to 1 if the end of activity c and the begin of activity a are separated by more than 1440 minutes.

$$\begin{aligned} t^f_{r_1u_1c} - 1440 \;\; &\leq \;\; t^i_{r_2u_2a} + BM(1 - s_{ac}) + BM(1 - x_{r_2u_2pr(a)b}) \\ &+BM(1 - x_{r_1u_1pr(c)b}) + BMw_{ac} \; \forall \; b, r_1, r_2, u_1, u_2 \end{aligned} \tag{5.14}$$

$$\begin{aligned} t^f_{r_1u_1a} - 1440 \;\; &\leq \;\; t^i_{r_2u_2c} + BMs_{ac} + BM(1 - x_{r_2u_2pr(a)b}) + BM(1 - x_{r_1u_1pr(c)b}) \\ &+BMw_{ca} \; \forall \; b, r_1, r_2, u_1, u_2 \end{aligned} \tag{5.15}$$

To avoid overlaps of activities on a same resource, we use Inequalities (5.14) and (5.15). If activities a and c are both scheduled on resource b, a is scheduled before c and there are more than 1440 minutes between the end of a and the start of c, then the end of a - 1440 must be previous of the start of c.

Objective functions

The objective function is expressed in terms of the minimization of the number of train movements. The number of train movements is computed as the number of changes of resources during maintenance routing in the workshop. Since this objective function can be formulated by using either the variables x_{ruob} or the variables y_{ru}, in our test experiments both options will be evaluated:

- $Approach1$ Min : $\sum x_{ruob}$ $where\ ar(o)\ is\ a\ movement$

- $Approach2$ Min : $\sum k_r y_{ru}$

5.3.1 Computational experiments

The results are based on real instances taken from a Trenitalia's maintenance site located in Naples. We analyze 100 working days and we compare the programs used in real life with our model solutions. We consider all asset units types and all maintenance types. Waiting time longer than one day is considered not as a variable but as a constraint on resources availability. A pre-processing was also performed in order to reduce the problem size to a sequence of one-day problems, so we generated 100 one-day cases. For each train, we consider all possible routes in the workshop. These instances are solved by CPLEX 12 on a PC with 2.27 GHz and 4 GB of RAM. Ten minutes of computation are allowed for the branch and bound (see [58], [84] and [10]) code of CPLEX.

5.3.2 Practical versus model solutions

This section compares the solutions found by our maintenance scheduling formulation with the practical solutions at Naples workshop. Table 5.3 reports the average results on the 100 instances obtained for the two approaches (i.e. the two definitions of the same objective function) of Section 3.3: the percentage of solutions that have been improved via optimization compared to the practical solutions, the percentage of optimal solutions, the average saving in terms of train movements reduction and the total computation time of CPLEX (in seconds).

From the results of table 5.3, $Approach1$ offers the best percentage of improved solutions with a large computation time. However, $Approach2$ guarantees the best average gain, good percentage of improved solutions and a considerable smaller computation time. So, the former approach gives the largest % improvement while the latter approach can be better used for real-time maintenance control applications.

Table 5.3: Comparison between models

Approach	1	2
Improved Solutions	60	43
Optimal Solutions	65	48
Average Train Movements Saving	1.8	2.6
Total Computation Time (sec)	531.2	152.9

Furthermore, the initial problem could be also divided in two simpler sub-problems put in sequence. The first sub-problem is to find the minimum number of movements choosing a set of alternative routing/processes in the workshop while the second one is to make a schedule for routing/processes chosen.

$$\sum_{r \in R} y_{ra} = 1 \quad \forall\ a \in U \tag{5.16}$$

The formulation for the first sub-problem is based on the set of constraints (5.16) to select one and only one route/process for each train unit.

$$\sum_{r \in R} y_{ra} \quad \forall\ a \in U \tag{5.17}$$

To exclude feasible solutions from the solutions found by the solver in a previous step, we add a set of constraints (5.17). For each run i, if the second sub-problem does not give an optimal solution, we append constraints (5.17) to the first model. Note that if the problem one is unfeasible then the global problem is unfeasible.

The objective function of the first sub-problem to minimize could be expressed as

$$\sum_{r \in R} k_r y_{ra} < u \quad \forall\, a \in U \tag{5.18}$$

The second sub-problem is the global maintenance problem with variables y set from the first sub-problem. While the second problem is feasible we solve an instance of first problem with some added constraints that avoid to find solutions already found. When we solve the second problem we are sure to have found the optimal solution for our global problem. We add a time limit on computation time to avoid to lock the process. The commercial solver often has big troubles to proof the unfeasibility of an instance. This approach is very quick (14 seconds to find an optimal solution) when the optimal solution has few movements. Unfortunately in most of cases (84%) this condition in not true and therefore the solver is not able to solve the problem.
The remaining experiments of this paper are dedicated to a further evaluation of the potential impact of $Approach1$. We use the same set of instances but with a reduced set of routes available for each train. Figure 5.5 shows a deeper average comparison between the real world solutions and our model solutions in terms of the percentage of saving between 0 and 5 train movements. In the 95.8% of the 100 one-day cases a better solution is obtained by our model, while in 4.2% of cases the practical solutions are identical to the model solutions. This result can be considered as remarkable since a better solution is very often achieved via optimization. Specifically, the number of savings is often between 2 and 3 train movements. We observe that this analysis is done only for unscheduled train movements between workshop and passenger stations, that are required for the computation of a feasible train schedule.

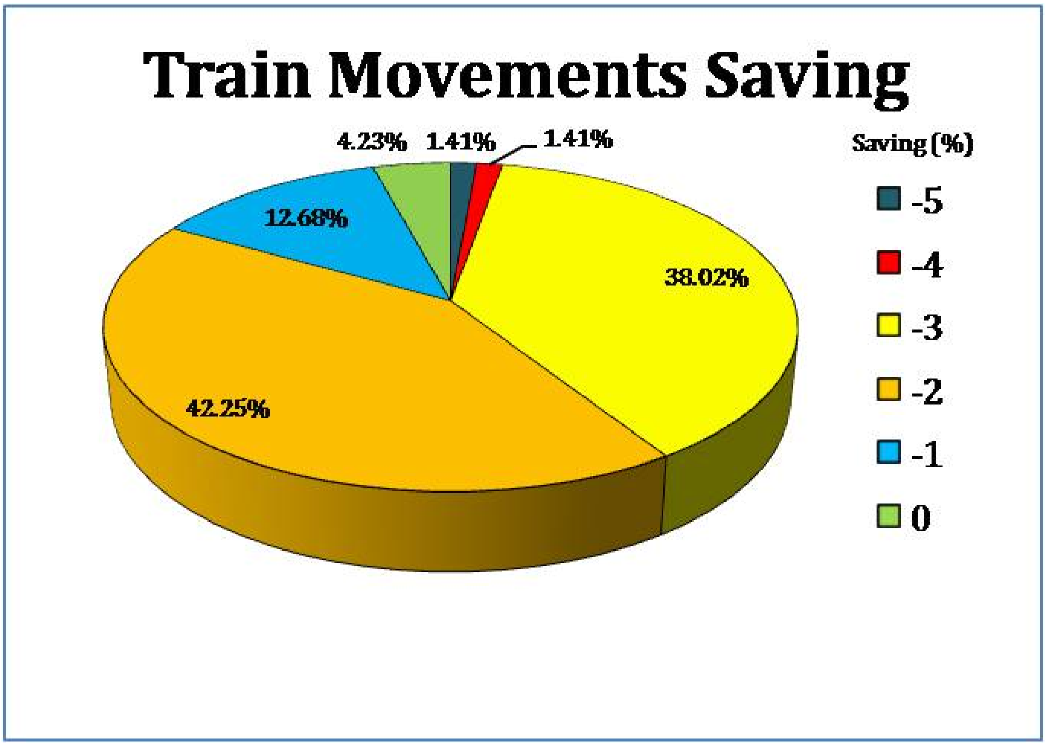

Figure 5.5: Savings of our model solutions compared to the practical solutions

5.3.3 Assessment of the computation time

Figure 5.6 shows the CPLEX performance when varying the number of jobs. The time to compute the CPLEX solution is up to around 1 minute, so we can conclude that the formulation can be quickly solved by CPLEX for real-world instances. Practical use of the proposed method can be envisaged both during maintenance planning and operational phases.

5.3.4 Robustness analysis

A robustness analysis is provided in order to evaluate the schedule quality in case of medium-term traffic disturbances that alter the off-line plan of operations. The objective is to re-balance the use of workshop resources to compute of a feasible schedule, i.e. a schedule of all workshop operations within the time windows given by

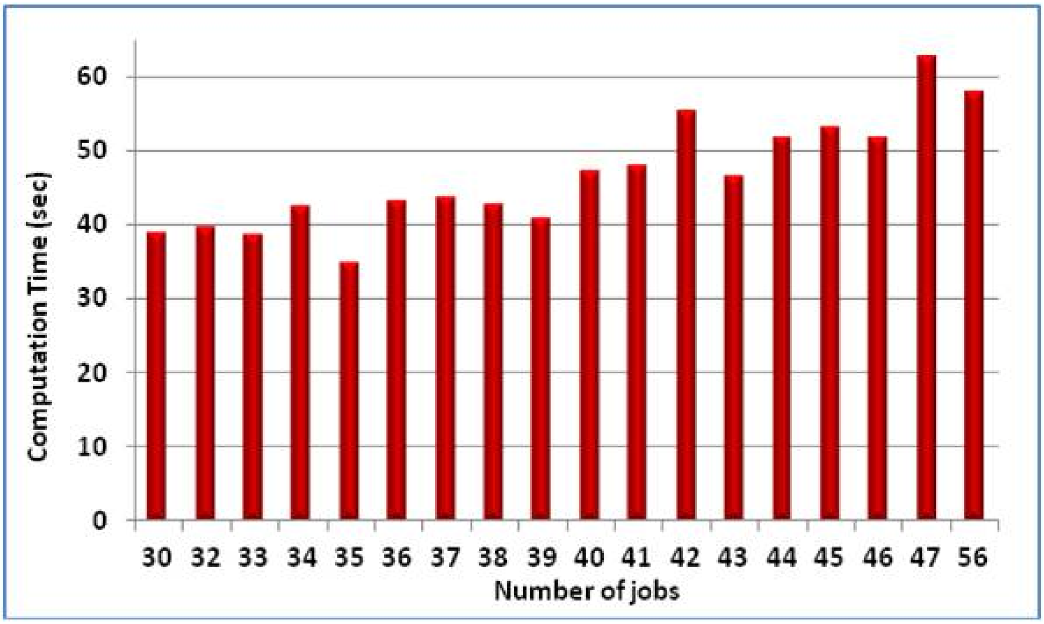

Figure 5.6: Computation time for an increasing number of jobs in the workshop

the rolling stock rostering planner and perturbed by traffic disturbances.
Figure 5.7 and 5.8 represent a sample job, made by three activities, to be scheduled in the workshop. We distinguish between a not compressed schedule, i.e. a schedule with some recovery time, and compressed schedule, i.e. a schedule with less recovery time or without recovery time. Specifically, we analyze the case in which the time window start is delayed by unforeseen events and the schedule needs to be compressed accordingly, without reducing the duration of the maintenance activities, in order to schedule all activities within the given time window.
Let's now define a compression factor

$$CF = \frac{recovery\ time\ to\ be\ compressed}{total\ available\ recovery\ time} \tag{5.19}$$

where total available recovery time = $(t_u^{fin} - t_u^{ini}) - \sum d_i$ (30 minutes in the case of Figure 5.7) and recovery time to be compressed is a

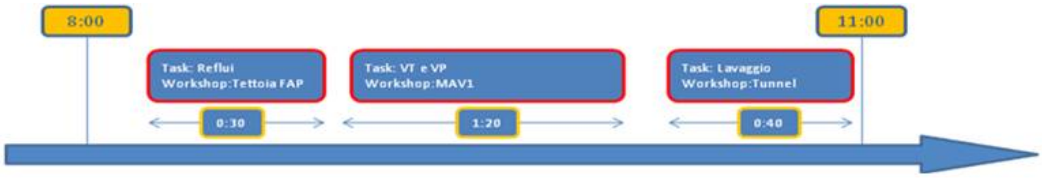

Figure 5.7: Time window not compressed

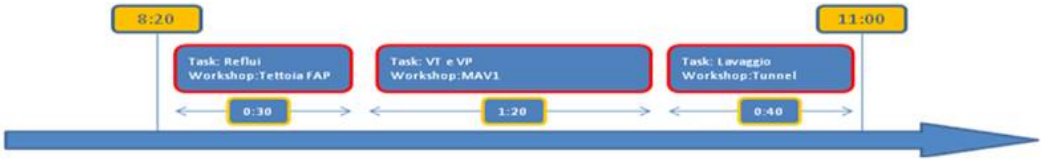

Figure 5.8: Time window compressed

reduction of time at workshop disposal due to traffic disturbances (20 minutes in the example case of Figure 5.8).
Figure 5.9 reports results on a subset of 63 instances for 9 different values of compressed time windows (0.1, 0.2, . . . , 0.9), obtained by delaying the start time of the time windows. For compression values equal to 0.1 – 0.4 we always obtain a feasible solution, while for 0.5 and upper values the number of feasible solutions decreases with the increase of the compression factor.
However, we did not find, as we expected, an exact correlation between the computation time and the compression factor, i.e. a systematic increase of computation time when increasing the compression factor (see Figure 5.10). Surprisingly, for some practical cases we have that compressed instances are easier to solve by CPLEX than uncompressed instances.

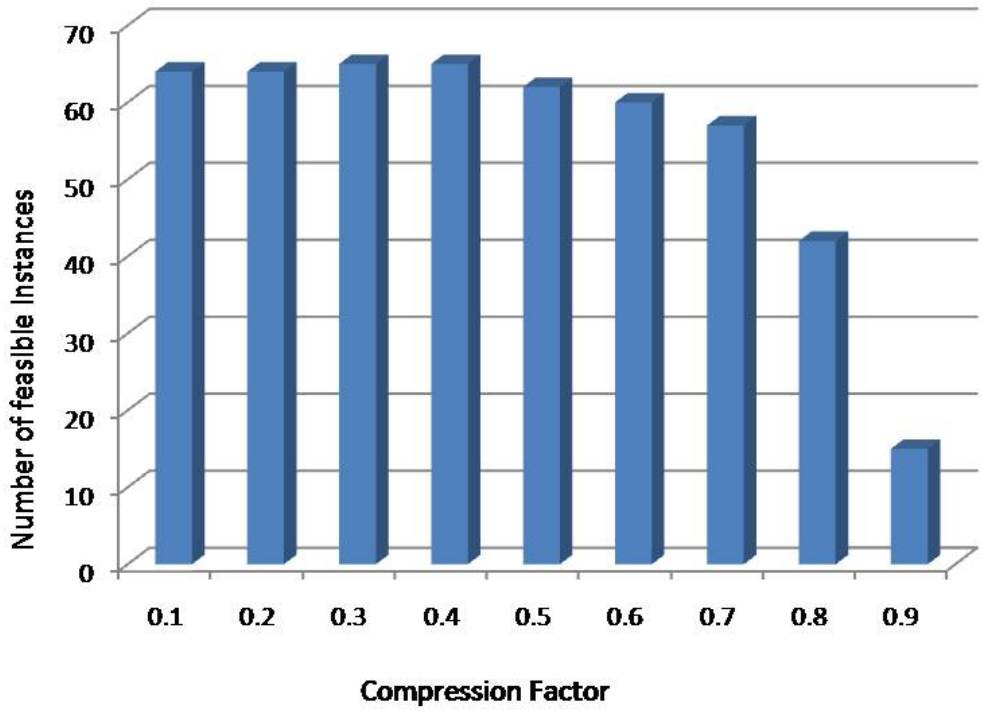

Figure 5.9: Study of the compression factor (feasible instances)

5.4 Second approach

Different approaches could be used to solve maintenance scheduling problem. In this section we will consider this problem as an implementation of standard open shop scheduling formulation (see [35], [15],[48], [8] and [61]). To explain this approach we will use three graphs.

The first graph, named *train maintenance site graph*, models the maintenance tasks (or operations) to be scheduled for all asset units. Each node of this graph represents a maintenance operation to be performed at the workshop on a specific resource. For each train unit we can have more alternative jobs because more different maintenance processes could satisfy maintenance needed. That means we could have also different routes in the workshops. We will have a node for each train unit, maintenance activity (oper-

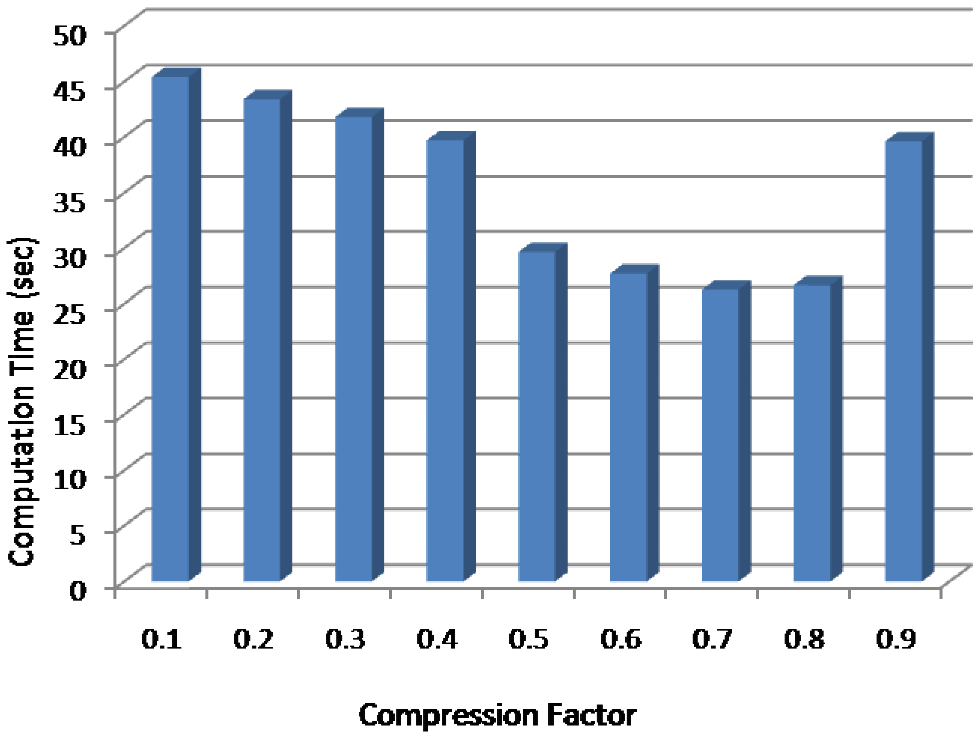

Figure 5.10: Study of the compression factor (computation time)

ation of a possible job) and resource. We will note this set of nodes MSN. The edges of graph, made only between nodes belonging to the same job, represent a possible sequence for the connected nodes (maintenance tasks). A path on this graph corresponds to a maintenance schedule for a given job and asset unit. Obviously not all nodes will belong to the path. We also add to the graph two virtual nodes for each train unit in order to link all its alternative jobs. These virtual nodes will be called $start$ and end nodes.
The second graph, named *maintenance graph*, models the maintenance operations with reference to workshop resources. This graph is made by the set of nodes MSN used in the *train maintenance site graph* and a different set of edges. The edges of graph, made only between nodes referred to the same workshop resource, represent a possible sequence for the connected nodes. Each path on this graph represents a possible schedule on a specific resource. We

also add two virtual nodes, called *start* and *end* nodes but different from those used before, for each workshop resource in order to link all possible nodes referred to this resource. Note that the paths on *train maintenance graph* and *maintenance graph* must involve the same subset of nodes.
The third graph, named *movements graph*, models the train movements to be performed on the infrastructure resources. Each node (movement) is a train occupation of a specific infrastructure resource. For each edge of *train maintenance site graph* involving different resources we add a node to *movements graph*. We are making a graph similar to *maintenance graph* but based on unreal resources to avoid having more movements in parallel in the workshop than allowed. For example if are at least allowed two movements in parallel, the *movements graph* will based on two unreal resources. Each edge, allowed between nodes referred to the same unreal resources, describes the order between two consecutive movements. Also in this case we will add two virtual nodes, *start* and *end* nodes, are created for each possible infrastructure resource in order link all nodes of the graph referred to the same unreal resource. Each path corresponds to a schedule for occupation of infrastructure resources due to the movements between two consecutive maintenance operations on different workshop resources or between the entrance/exit into/from the maintenance station and the first/last maintenance operations. A maintenance scheduling solution is a schedule of the maintenance tasks of each train (path on *train maintenance graph*), of the maintenance tasks at each workshop resource (path on *maintenance site graph*), and of the occupation time on the infrastructure resources (path on *movements graph*). A maintenance scheduling solution thus corresponds to a subset of paths for each graph.

List of notations

We now list the notation used for the MS problem.

- R is the set of maintenance resources,
- J is the set of train maintenance jobs,
- MV is the set of train movements on the infrastructure resources,
- RG is the maintenance site graph nodes set,
- RGA is the maintenance site graph edges set,
- rg is the cardinality of the set RG,
- JG is the train maintenance nodes set,
- JGA is the train maintenance edges set,
- jg is the cardinality of the set JG,
- MG is the movements graph nodes set,
- MGA is the movements graph edges set,
- mg is the cardinality of the set MG,
- dt_i is the duration of the task i,
- r_s is the virtual start node for the maintenance resource r,
- r_f is the virtual end node for the maintenance resource r,
- job_s is the virtual start node for the train maintenance job job,
- job_f is the virtual end node for the train maintenance job job,
- mov_s is the virtual start node for the movement mov,
- mov_f is the virtual end node for the movement mov,
- tws_j is the start time for the maintenance window of the train j,
- twf_j is the end time for the maintenance window of the train j,
- mm is maximum number of train movements at the same time,
- (i, j) is the edge between node (operation, movement or waiting time) i and node (operation, movement or waiting time) j,
- $u_{(i,j)}$ is a binary variable that is 1 if edge (i, j) belongs to a maintenance graph path, 0 otherwise,

- $v_{(i,j)}$ is a binary variable that is 1 if edge (i, j) belongs to a train maintenance graph path, 0 otherwise,
- $z_{(i,j)}$ is a binary variable that is 1 if edge (i, j) belongs to a movements graph path, 0 otherwise,
- $ul_{(i,j)}$ is an integer variable for edge (i, j) to avoid sub-loops on maintenance graph,
- $vl_{(i,j)}$ is an integer variable for edge (i, j) to avoid sub-loops on train maintenance graph,
- $zl_{(i,j)}$ is an integer variable for edge (i, j) to avoid sub-loops on movements graph,
- $ut^{I}_{(i,j)}$ is the start time of the node j on the maintenance site if (i, j) belongs to a path,
- $ut^{F}_{(i,j)}$ is the end time of the node i on the maintenance site if (i, j) belongs to a path,
- $vt^{I}_{(i,j)}$ is the start time of the node j on the train maintenance if (i, j) belongs to a path,
- $vt^{F}_{(i,j)}$ is the end time of the node i on the train maintenance if (i, j) belongs to a path,
- $zt^{I}_{(i,j)}$ is the start time of the node j on an infrastructure resource if (i, j) belongs to a path,
- $zt^{F}_{(i,j)}$ is the end time of the node i on an infrastructure resource if (i, j) belongs to a path,
- Tmr is the minimum setup time between consecutive operations performed at the same maintenance resource,
- Tmj is the minimum setup time between consecutive operations performed on the same train at different maintenance resources,
- Tmm is the minimum setup time between consecutive operations performed at different maintenance resources.

Numerical example

We present a numerical example to illustrate our MS problem formulation. The RSR solution provides two trains named $J1$ and $J2$. The train $J1$ has two tasks ($J11$ and $J12$). The time window in which we have to process $J1$ is from 8:00 AM till 6:00 AM of the next day. The train $J2$ has only one task ($J21$). The time window for $J2$ is from 8:00 AM to 12:00 AM. The tasks $J11$ and $J21$ must be processed on a maintenance resource of the type 1 ($R1$) while $J12$ on a maintenance site of the type 2 ($R2$). Our maintenance station has only one maintenance resource of type 1 and only one of type 2.

Table 5.4: Tasks duration

Train	Task	Duration (min)
$J1$	$J11$	100
$J1$	$J12$	100
$J2$	$J21$	100

Table 5.5: Parameters

Setup	Value	Description
Tmj	25	Setup time between two operations at the same site
Tmm	25	Setup time between consecutive operations on a resource

Table 5.6: Time for train movements between two maintenance sites

From maintenance site	To maintenance site	Time (min)
$R1$	$R2$	100
$Virtual\ start\ /\ end\ node$	$R1$	25
$Virtual\ start\ /\ end\ node$	$R2$	25

Table 5.4 shows the duration (in minutes) of all tasks and for each job. We assume that all tasks have the same duration. In Table 5.5, we present the values of parameters Tmj and Tmm. Table 5.6 gives the time to move a rolling stock unit between two different resources.

Figure 5.11 shows the maintenance graph for this example. For each resource, we have two virtual nodes ($R1_s$, $R1_f$ for $R1$ and $R2_s$, $R2_f$ for $R2$). A first maintenance resource operates both trains (i.e., nodes $J21$, $J11$), while the other resource operates only $J1$ (i.e., node $J12$). In our case we will use two resources (because we have two paths) to provide the maintenance coming from RSR. On the first track we will provide the operation J11 and J21 while on the second track we will provide the activity J12. Also the start and end times of each task are shown on the left and right sides of the corresponding node.

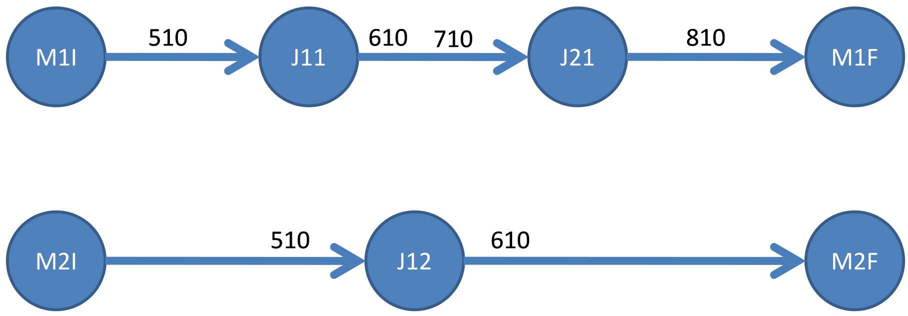

Figure 5.11: Maintenance graph

Figure 5.12 shows the train maintenance graph for this example. For each train maintenance, we have two virtual nodes ($J1_s$, $J1_f$ for $J1$ and $J2_s$, $J2_f$ for $J2$). The tasks of the first train maintenance are represented by nodes $J12$ and $J11$, while the task of the second train maintenance by node $J21$. The start and end times of each task are shown in the figure, including the start time of virtual nodes.

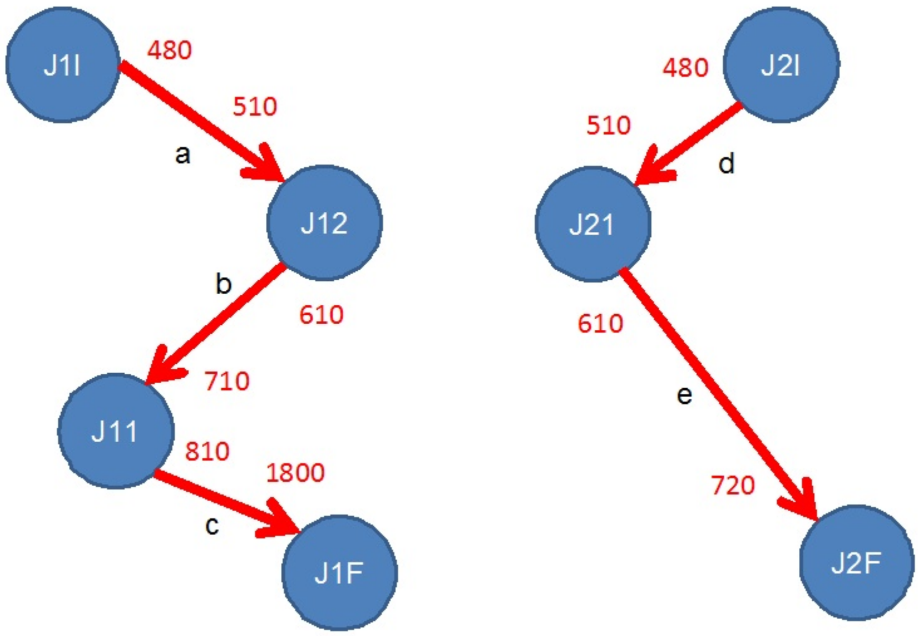

Figure 5.12: Train maintenance graph

Figure 5.13 shows the movements graph for this example. For each resource, we have two virtual nodes ($M1_s$, $M1_f$ for $M1$ and $M2_s$, $M2_f$ for $M2$). The tasks of the first resource are represented by nodes d, b and c, while the tasks of the second resource by nodes a and e. The start and end times of each task are shown in the figure.

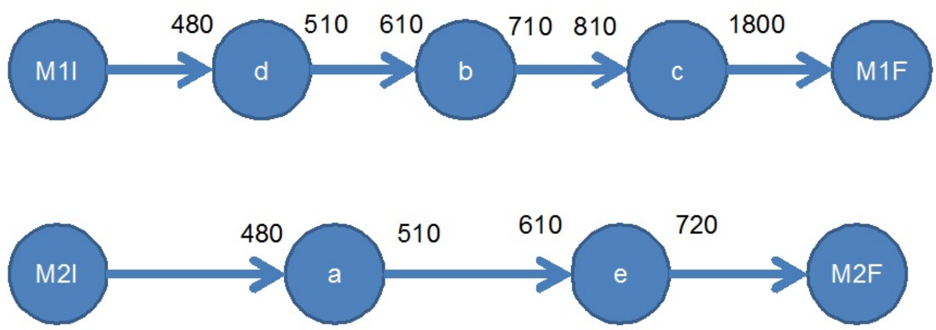

Figure 5.13: Movements graph

A comparison between Figures 5.11 and 5.12 shows that the train maintenance and maintenance site graphs are based on the same node sets. The idea is that each node represents a task (job part) on a specific resource. Obviously, with reference to maintenance graph and train maintenance graph, we will have two schedules with the same time for each node.

Problem variables

For each edge of maintenance graph we have the following variables:

- a binary variable u to define if an edge belongs to a solution path ($u = 1$),
- an integer variable ul with a value greater then 0 if the edge belongs to a solution path, showing the position of the edge in path,
- a real variable $ut^{I}_{(i,j)}$ showing the start time of the task, represented by the node in which the edge enters,
- a real variable $ut^{F}_{(i,j)}$ showing the end time of the task, represented by the node in which the edge exits.

Similar situation arises for the variables of the train maintenance graph (i.e., a binary variable v and an integer variable vl for each edge; $vt^{I}_{(i,j)}$ and $vt^{F}_{(i,j)}$ for each edge) and of the movements graph (i.e., a binary variable z and an integer variable zl for each edge; $zt^{I}_{(i,j)}$ and $zt^{F}_{(i,j)}$ for each edge).

Objective function

$$min \sum_{(i,k)\in MGA} z_{(i,k)}$$

The main objective of the MS problem is the minimization of the train movements in the maintenance area. This objective function corresponds to the number of paths in the movements graph.

Maintenance site constraints

$$\sum_{h\in RG} u_{(h,i)} = \sum_{k\in RG} u_{(i,k)} \quad \forall\ i \in RG \tag{5.20}$$

$$\sum_{h\in RG} u_{(i,h)} = 1 \ \ \forall\ i \in RG \tag{5.21}$$

$$\sum_{i\in RG} u_{(i,r_f)} <= 1 \quad \forall\ r \in R \tag{5.22}$$

Equations (5.20),(5.21) and (5.22) are used to define a schedule of all needed tasks at the workshop. Equation (5.20) is to model a path in the maintenance graph, while Equation (5.22) is to avoid conflicting train paths starting from the same node. Equation (5.21) is to manage alternative nodes. In the MS solution, a task must be processed only on a maintenance resource. That means we must have just one node belongs to solution.

$$\sum_{h \in RG} ul_{(i,h)} = \sum_{k \in RG} ul_{(k,i)} + \sum_{r \in R} u_{(r_s,i)} + \sum_{t \in RG} u_{(i,t)} \quad \forall\ i \in RG \tag{5.23}$$

$$ul_{(i,h)} <= rg * \ u_{(i,h)} \quad \forall\ (i,h) \in \ RGxRG \tag{5.24}$$

Equations (5.23) and (5.24) are mandatory to avoid sub-loops in the solution. The variable ul (if greater than 0) reports the order of edge in the solution path. These variables have an upper bound that is the number of nodes of the maintenance site graph.

$$u_{(i,j)} \quad binary \tag{5.25}$$

Variable $u_{(i,j)}$ is 1 if edge (i,j) belongs to a maintenance path, 0 otherwise.

Train maintenance constraints

In this section we will describe the main constraints to build paths on train maintenance graph.

$$\sum_{h \in JG} v_{(h,i)} = \sum_{k \in JG} v_{(i,k)} \quad \forall\ i \in JG \tag{5.26}$$

$$\sum_{i \in JG} v_{(i,j_f)} <= 1 \quad \forall\ j \in J \tag{5.27}$$

$$\sum_{h \in JG} v_{(i,h)} = \sum_{k \in RG} u_{(i,k)} \quad \forall\ i \in (JG \bigcap RG) \tag{5.28}$$

Equations (5.26),(5.27),(5.28) are mandatory to control the integrated jobs schedule. Equation (5.26) defines the basic rule to generate a path in the train maintenance graph, while Equation (5.27) avoids potential conflicting train paths starting in the same node. Among alternative nodes, we need to insert into a maintenance site path just one node. Equation (5.28) pairs maintenance resource variables with train maintenance variables. That means the maintenance site and train maintenance paths must include the same nodes set.

$$\sum_{h \in JG} vl_{(i,h)} = \sum_{k \in JG} vl_{(k,i)} + \sum_{j \in J} v_{(j_s,i)} + \sum_{t \in JG} v_{(i,t)} \quad \forall\ i \in JG \tag{5.29}$$

$$vl_{(i,h)} <= jg \ \ v_{(i,h)} \quad \forall\ (i,h) \in JGA \tag{5.30}$$

Equations (5.29) and (5.30) work similarly to Equations (5.23) and (5.24). The variable vl (if greater than 0) represents the edge order for each node. These variables have an upper bound that is the number of nodes of the train maintenance graph.

$$v_{(i,h)} \quad binary \tag{5.31}$$

Variable $v_{(i,h)}$ is 1 if (i,h) belongs to a train maintenance path, 0 otherwise.

Movements constraints

$$\sum_{h \in MG} z_{(h,i)} = \sum_{k \in MG} z_{(i,k)} \quad \forall\ i\ \in\ MG \tag{5.32}$$

$$\sum_{i \in MG} z_{(i,m_f)} <= 1 \quad \forall\ m \in MG \tag{5.33}$$

Similarly to the other types of paths, Equations (5.32) and (5.33) define the infrastructure paths.

$$\sum_{h \in MG} z_{(i,h)} = v_{(p,q)*} \quad \forall\ i \in MG \tag{5.34}$$

where (p,q)* belongs to JGA and corresponds to node i that belongs to MG.
Equation (5.34) combines the train maintenance variables with the movements variables. The basic idea is that each node of the movements graph is an edge for the train maintenance graph. Consequently, a specific node of the movements graph could belong to an movements path only if the train maintenance edge associated belongs to a train maintenance path.

$$\sum_{h \in MG} zl_{(i,h)} = \sum_{k \in MG} zl_{(k,i)} + \sum_{m \in M} z_{(m_s,i)} + \sum_{t \in MG} z_{(i,t)} \quad \forall\ i \in MG \qquad (5.35)$$

$$zl_{(i,h)} <= mg\ z_{(i,h)} \quad \forall\ (i,h) \in MGA \qquad (5.36)$$

In the movements graph, sub-loops are not permitted. This is achieved in our formulation by adding Equations (5.35) and (5.36).

$$z_{(i,j)} \quad binary \qquad (5.37)$$

Variable $z_{(i,j)}$ is 1 if (i, j) belongs to a infrastructure path, 0 otherwise.

Constraints on $ut^I_{(i,j)}$ and $ut^F_{(i,j)}$

We next describe the equations for the variables ut:

$$\sum_{j \in RG} ut^I_{(i,j)} = \sum_{k \in JG} vt^I_{(i,k)} \quad \forall\ i \in (JG \bigcap RG) \tag{5.38}$$

$$\sum_{j \in RG} ut^F_{(i,j)} = \sum_{k \in JG} vt^F_{(i,k)} \quad \forall\ i \in (JG \bigcap RG) \tag{5.39}$$

Equations (5.38) and (5.39) guarantee that start time and end time are scheduled accordingly on maintenance and on train maintenance graph.

$$\sum_{k \in RG} ut^F_{j,k} >= \sum_{i \in RG} ut^I_{(i,j)} + dt_j \sum_{t \in RG} u_{(t,j)} \quad \forall\ j \in R \tag{5.40}$$

Equation (5.40) imposes that the end of each task (node) must be greater than its start time plus its duration.

$$ut^I_{(i,j)} <= 2880\ u_{(i,j)} \tag{5.41}$$

$$ut^F_{(i,j)} <= 2880\ u_{(i,j)} \tag{5.42}$$

From Equations (5.41) and (5.42), the start and end times of each task have an upper bound that is 2880 (minutes) times $u_{(i,j)}$, due to the cyclic timetable hypothesis.

$$ut^F_{(i,j)} + Tmr\ u_{(i,j)} <= ut^I_{(i,j)} \quad \forall\ (i,j) \in RGA \tag{5.43}$$

From Equation (5.43), given two consecutive tasks, the second task must start only after a specific setup time from the completion of the first task. This setup time must include the movement of rolling stock units. Note that $Tmr > 0$ if two consecutive tasks are related to different trains ($Tmr < 1440$).

$$\sum_{i \in RG} ut^F_{(i,r_f)} - 1440 + Tmr <= \sum_{k \in RG} ut^I_{(r_s,k)} \quad \forall\ r \in R \tag{5.44}$$

Equation (5.44) is very important to avoid overlaps between operations when a scheduled job exceeds the midnight.
Figure 5.14 shows that the overlap between operations is avoided if ut^F (that could be greater than 1440 minutes) minus 1440 minutes is lower than ut^I.

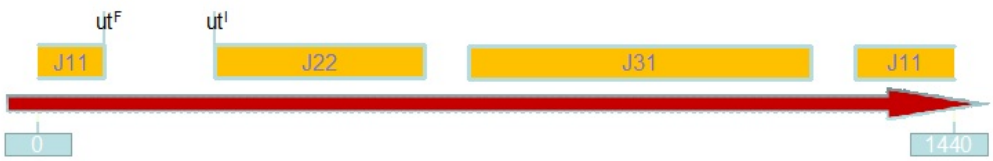

Figure 5.14: Example of an overlap-free job schedule on a maintenance site

Constraints on $vt^I_{(i,j)}$ and $vt^F_{(i,j)}$

We next describe the equations for the variables vt.

$$\sum_{k \in MG} vt^F_{(job_s,k)} = tws_{job} \quad \forall \; job \in J \tag{5.45}$$

$$\sum_{k \in MG} vt^I_{(i,job_f)} = twf_{job} \quad \forall \; job \in J \tag{5.46}$$

From Equations (5.45) and (5.46), the start and end times of each node of path must be included in the time window defined by the RSR solution. This represents a strong correlation between the rostering and maintenance scheduling problems.

$$vt^I_{(i,j)} >= vt^F_{(i,j)} + Tmj \; v_{(i,j)} \quad \forall \; (i,j) \in JGA \tag{5.47}$$

From Equation (5.47), two consecutive tasks of a job on different maintenance resources must be separated by more Tmj minutes.

$$vt^I_{(i,j)} <= 2880 \; v_{(i,j)} \tag{5.48}$$

$$vt^F_{(i,j)} <= 2880 \; v_{(i,j)} \tag{5.49}$$

From Equations (5.48) and (5.49) the start and end times of each task have an upper bound equal to 2880 (minutes) $v_{(i,j)}$.

Constraints on $zt^I_{(i,j)}$ and $zt^F_{(i,j)}$

We now describe a set of constraints for the variables zt and the relationship with the variables ut and vt.

$$\sum_{j \in MG} zt^{I}_{(p,q)*} = ut^{F}_{i} \quad \forall\ i \in MG \tag{5.50}$$

$$\sum_{j \in MG} zt^{F}_{(p,q)*} = ut^{I}_{i} \quad \forall\ i \in MG \tag{5.51}$$

From Equations (5.50) and (5.51), the variables $zt^{I}_{(i,j)}$ represent the beginning of a train movement, while $zt^{F}_{(i,j)}$ the end of a train movement. It is compulsory that the timing of these train movements is strong related to the value of variables $ut^{I}_{(i,j)}$ and $ut^{F}_{(i,j)}$, or $vt^{I}_{(i,j)}$ and $vt^{F}_{(i,j)}$.

$$zt^{I}_{(i,j)} <= \ 2880\ z_{(i,j)} \tag{5.52}$$

$$zt^{F}_{(i,j)} <= \ 2880\ z_{(i,j)} \tag{5.53}$$

From Equations (5.52) and (5.53), the variables $zt^{I}_{(i,j)}$ and $zt^{F}_{(i,j)}$ have also an upper bound equal to 2880 minutes.

$$zt^{I}_{(i,j)} >= zt^{F}_{(i,j)} + Tmm\ z_{(i,j)} \quad \forall\ (i,j) \in MGA \tag{5.54}$$

From Equation (5.54), a minimum setup time, called Tmm, must be respected between consecutive movements of different rolling stock units.

$$\sum_{i \in MG} zt^{F}_{(i,m_f)} - 1440 + Tmm <= \sum_{k \in MG} zt^{I}_{(m_s,k)} \quad \forall\ m \in MV \tag{5.55}$$

From Equation (5.55), the train movements on infrastructure resources must have no overlap when a scheduled movement exceeds the midnight.

We note that the maximum number of parallel movements mm is imposed building the movements graph.

5.4.1 Computational experiments

The Table 5.7 shows the results obtained using the second approach for the problem of maintenance optimization on several (but partial) real instances. The tested instances involve only cleaning and standard maintenance tasks. The workshop is reduced to a very limitated number of resources (tracks). First column represents the number of trains involved. In the second and third columns we report the number of resources of workshop (maintenance and cleaning platforms). The fourth column shows the computation time and the value of objective function. We impose a computation time limit (4400 seconds) to avoid to lock the commercial solver.

Table 5.7: Outcomes

Trains	Num. mainten. resources	Num. Cleaning resources	Time (sec)
2	2	2	0,09
2	2	3	0,08
2	2	4	0,08
2	2	5	1,37
2	2	6	1,53
2	2	7	1,98
2	2	8	7,35
2	2	9	234
2	2	10	2470
2	2	11	not solved in 4400
3	3	11	4392
3	3	12	not solved in 4400

The results are definitely good but unfortunately the performance of the solver does not allow us to use in practice this formulation despite the number of resources used are considerably lower than those of the practice. Cplex has not been able to solve compete real instances.

5.5 Third Approach

This approach assumes that the sequencing of the tasks of a job is already assigned. Each job is a sequence of maintenance operation ($real$) and waiting time/movements ($virtual$) (see Figure 5.15). If two following activities of the job are scheduled on the same resource then there will be no movements.

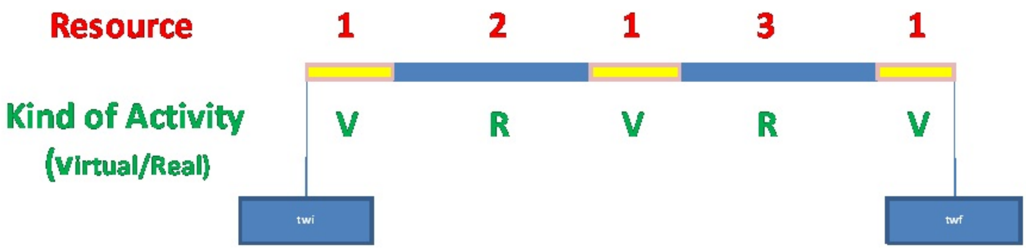

Figure 5.15: Job representation

We would schedule the real activities on the workshop resource minimizing the number of movements (virtual activities).
The following notations will be used in the model:

- A is the set of maintenance tasks to provide,
- $a_{i,j,r}$ is the task i of the job j to provide on the resource r,
- $d_{i,j}$ is the duration of the i of the job j,
- Twi_j is the initial time window of the job j,
- Twf_j is the final time window of the job j,
- R is the set of resources available,
- KR is a set of binary variables such that $kr_{i,j,r}$ is equal to 1 if the task i of the job j is provided on the resource r and zero otherwise,
- $tini_{i,j}$ is a real variable representing the initial time of the task i of the job j,
- $tfin_{i,j}$ is a real variable representing the final time of the task i of the job j,
- $tini_{I,j}$ is a real variable representing the initial time for the first task of the job j,

- $tfin_{F,j}$ is a real variable representing the final time for the last task of the job j,
- KF is a set of binary variables such that $kf_{(i,v),j,r}$ is equal to 1 if the virtual task (just waiting time) between the activity i and the activity v of the job j is provided on the resource r and 0 otherwise. This set includes variables such as $kf_{(I,v),j,r}$ for virtual tasks before the first task of the job (and $kf_{(v,U),j,r}$ for virtual tasks after the last task of the job),
- S is a set of binary variables such that $s_{i,j}$ is equal to 1 if i is scheduled earlier than v on the resource r and zero otherwise,
- BM is a BIG integer (i.e. a sufficiently large number).

$$\sum_{r \in R} kr_{i,j,r} = 1 \quad \forall \; real \; activity \; i \; and \; job \; j \tag{5.56}$$

Equation (5.56) assures that each maintenance operation is scheduled on one and only one resource.

$$\sum_{r \in R} kf_{(i,v),j,r} <= 1 \quad \forall \; job \; j \; and \; couple \; of \; real \; activities \; (i,v) \tag{5.57}$$

Equation (5.57) says that virtual activities are not mandatory.

$$tfin_{i,j} - tini_{i,j} >= d_{i,j} \quad \forall \; activity \; i \; and \; job \; j \tag{5.58}$$

Equation (5.58) checks that the duration of activities as not understimated.

$$tini_{I,j} >= Twi_j \quad \forall \; job \; j \tag{5.59}$$

$$tfin_{U,j} <= Twf_j \quad \forall \; job \; j \tag{5.60}$$

Equations (5.59) and (5.60) set the start/end time respecting time windows.

$$\sum_{r \in R} kf_{(i,v),j,r} * 2880 >= (tini_{v,j} - tfin_{i,j}) \; \forall \; virtual \; activity \; between \; (i,v) \tag{5.61}$$

$$\sum_{r \in R} kf_{(i,v),j,r} <= (tini_{v,j} - tfin_{i,j}) \; \forall \; virtual \; activity \; between \; (i,v) \tag{5.62}$$

Equations (5.61) and (5.62) assure that virtual activities are active only and only if $tini_{v,j}$ is greater than $tfin_{i,j}$.

$$tini_{v,j} >= tfin_{i,j}$$
$$\forall \; couple \; of \; real \; activities \; i \; and \; v \; of \; the \; same \; job \; j \;:\; v \; follows \; i \tag{5.63}$$

Given a couple of activities (i, v) with v following i, Equations (5.63) guarantees that the start time of v happens after the end time of i.

$$tini_{v,j} \ - \ tfin_{i,g} \ >= \ -BM \ * \ (1 - kr_{(i,g,r)})$$
$$-BM \ * \ (1 - kr_{(v,j,r)}) - BM \ * \ (1 - s_{i,v})$$
$$\forall \ resource \ r \ and \ couple \ of \ real \ activities \ we \ could \ schedule \ on \ \ r \tag{5.64}$$

Constraints (5.64) avoids overlaps on each resource. If two real activities v and i are on the same resource r ($(1 - kr_{(i,g,r)}) = 0$ and $(1 - kr_{(v,j,r)} = 0)$) and i precedes v then this constraints are activated (otherwise this equations are always respected).

$$tini_{i,g} - tfin_{v,j} >= -BM * (1 - kr_{(i,g,r)})$$
$$-BM * (1 - kr_{(v,j,r)}) - BM * s_{i,v} \tag{5.65}$$
$$\forall \ resource \ r \ and \ couple \ of \ real \ activities \ we \ could \ schedule \ on \ r$$

Constraints (5.65) work as Constraints (5.64) if v precedes i.

$$tfin_{l,g} - tini_{v,j} >= -BM * (1 - kf_{(i,v),j,r})$$
$$-BM * (1 - kf_{(l,m),g,r}) - BM * (1 - s_{((i,v),(l,m))})$$
$$\forall \ resource \ r \ and \ couple \ of \ virtual \ activities \ we \ could \ schedule \ on \ r \tag{5.66}$$

$$tfin_{i,g} - tini_{m,j} >= -BM * (1 - kf_{(i,v),j,r})$$
$$-BM * (1 - kf_{(l,m),g,r}) - BM * s_{((i,v),(l,m))}$$
$$\forall \ resource \ r \ and \ couple \ of \ virtual \ activities \ we \ could \ schedule \ on \ r \tag{5.67}$$

Constraints (5.66) and (5.67) work, at the same way of (5.64) and (5.65),for the virtual activities. In this case we have to take into account start and end time of activities l,m,i and v enclosing real activities.

$$\begin{aligned} tini_{l,g} - tfin_{i,j} >= -BM * (1 - kr_{(i,j,r)}) \\ -BM * (1 - kf_{(l,m),g,r}) - BM * (1 - s_{(i,(l,m))}) \\ \forall\ resource\ r\ and\ couple\ of\ a\ real\ activity\ and\ a\ virtual\ activity. \end{aligned} \quad (5.68)$$

$$\begin{aligned} tini_{i,j} - tini_{m,g} >= -BM * (1 - kr_{(i,j,r)}) \\ -BM * (1 - kf_{(l,m),g,r}) - BM * s_{(i,(l,m))} \\ \forall\ resource\ r\ and\ couple\ of\ a\ real\ activity\ and\ a\ virtual\ activity. \end{aligned} \quad (5.69)$$

Constraints (5.68) and (5.69) work for couples of activities made by a virtual and a real activity.

$$\begin{aligned} h_{i,v} >= (tini_{v,j} - tfin_{i,g} - 1440)/1440 \\ \forall\ couple\ of\ activities\ (i,v) \end{aligned} \quad (5.70)$$

$$\begin{aligned} h_{i,v} <= (tini_{v,j} - tfin_{i,g})/1440 \\ \forall\ couple\ of\ activities\ (i,v) \end{aligned} \quad (5.71)$$

To assure that there are no overlaps due to activities passing the midnight we define new binary variables $h_{i,v}$ that take the value of 1 if between the beginning of v and the end of the activity i there are more than 1440 minutes (0 otherwise). These variables and related Constraints (5.70) and (5.71) should be used also for virtual activities.

$$tfin_{v,j} - 1440 - BM*(1 - kr_{(i,g,r)}) - BM*(1 - kr_{(v,j,r)})$$
$$-BM*(1 - s_{i,v}) - BM*h_{i,v} <= tini_{i,g} \quad (5.72)$$
$$\forall\ resource\ r\ and\ couple\ of\ real\ activities$$

$$tini_{m,g} - 1440 - BM*(1 - kf_{(i,v),j,r}) - BM*(1 - kf_{(l,m),g,r})$$
$$-BM*(1 - s_{((i,v),(l,m))}) - BM*h_{((i,v),(l,m))} <= tfin_{i,v} \quad (5.73)$$
$$\forall\ resource\ r\ and\ couple\ of\ virtual\ activities$$

$$tini_{m,g} - 1440 <= tini_{i,j} - BM*(1 - kr_{(i,j,r)})$$
$$-BM*(1 - kf_{(l,m),g,r}) - BM*(1 - s_{(i,(l,m))}) - BM*h_{(i,(l,m))} \quad (5.74)$$
$$\forall\ resource\ r\ and\ couple\ of\ a\ real\ activity\ and\ a\ virtual\ activity$$

$$tfin_{i,g} - 1440 - BM*(1 - kr_{(i,g,r)}) - BM*(1 - kr_{(v,j,r)})$$
$$-BMs_{i,v} - BM*h_{i,v} <= tini_{v,j} \quad (5.75)$$
$$\forall\ resource\ r\ and\ couple\ of\ real\ activities$$

$$tini_{v,j} - 1440 - BM*(1 - kf_{(i,v),j,r}) - BM*(1 - kf_{(l,m),g,r})$$
$$-BMs_{((i,v),(l,m))} - BM*h_{((i,v),(l,m))} <= tfin_{l,g} \quad (5.76)$$
$$\forall\ resource\ r\ and\ couple\ of\ virtual\ activities$$

$$tfin_{i,j} - 1440 <= tfin_{l,g} - BM*(1 - kr_{(i,j,r)})$$
$$-BM*(1 - kf_{(l,m),g,r}) - BMs_{(i,(l,m))} - BM*h_{(i,(l,m))} \quad (5.77)$$
$$\forall\ resource\ r\ and\ couple\ of\ a\ real\ activity\ and\ a\ virtual\ activity$$

Constraints (5.72), (5.73) , (5.74) ,(5.75), (5.76) and (5.77) check that the last and the first task assigned to the resource r are not overlapped.

Note that $Tfin$ ($Tini$) becomes Twi_j (Twf_j) in case of initial (final) virtual activity.

For each couple of real activities (i, n) of job j we define a new binary variable $y_{(i,n,j,r)}$ that will be 0 if i and n are assigned to the same resource (otherwise $y_{(i,n,j,r)}$ will be 1).

$$y_{(i,n,j,r)} <= (kr_{(i,j,r)} + kr_{(n,j,r)})/2 \tag{5.78}$$

$$y_{(i,n,j,r)} >= kr_{(i,j,r)} + kr_{(n,j,r)} - 1 \tag{5.79}$$

Therefore variables y are linked to variables k by Constraints (5.78) and (5.79).
If we impose that virtual activity, between real activities i and n, is not provided on the same resource of i and n, the number of movements for each job is given by Equation (5.80).

$$n_j = \sum_{r \in R} kf_{(I,v),j,r} + \sum_{r \in R} kf_{(v,U),j,r} + \sum_{r \in R \ and \ i,n \in J} (2 * kf_{(i,n,j,r)}) + \sum w_{(i,n,j)} \tag{5.80}$$

where variable $w_{(i,n,j)}$, calculated by Constraints (5.81), says if real activities i e n need a movement ($w_{(i,n,j)} = 1$).

$$\begin{aligned} w_{(i,n,j)} &<= 1 - \sum_{r \in R} kf_{(i,n),j,r} \\ w_{(i,n,j)} &<= 1 - \sum_{r \in R} y_{(i,n,j,r)} \\ w_{(i,n,j)} &>= (1 - \sum_{r \in R} kf_{(i,n),j,r}) + (1 - \sum_{r \in R} y_{(i,n,j,r)}) - 1 \end{aligned} \tag{5.81}$$

Objective function to minimize could be written as sum of n_j (see

Equation (5.82)).

$$\sum_{j \in J} n_j \tag{5.82}$$

$$kf_{(i,v),j,r} + kr_{(i,j,r)} <= 1 \; \forall \; r \in R \; and \; \forall \; consecutive \; (i,v) \; \in J \tag{5.83}$$

$$kf_{(i,v),j,r} + kr_{(v,j,r)} <= 1 \; \forall \; r \in R \; and \; \forall \; consecutive \; (i,v) \; following \in J \tag{5.84}$$

We would avoid that a virtual task, between the real activities i and n, is scheduled on the same resource of i or n. To do that we add Constraints (5.83) and (5.84).
We can observe also in the Figure 5.16 different value for objective function for two cases represented. Furthermore feasibility of the case in top of the Figure 5.16 implies the feasibility of the case in bottom.
We note that, differently from the first approach, this model does not treat alternative jobs and calculates the movements at run time.

5.5.1 Computational experiments

In this section we present results obtained using the third approach for the problem of maintenance optimization. First column represents the number of trains involved. The second one shows the number of operations to be provided. In the third and fourth we report the number of resources of workshop (maintenance and cleaning platforms). The fifth and the sixth columns show the computation time and the value of objective function.

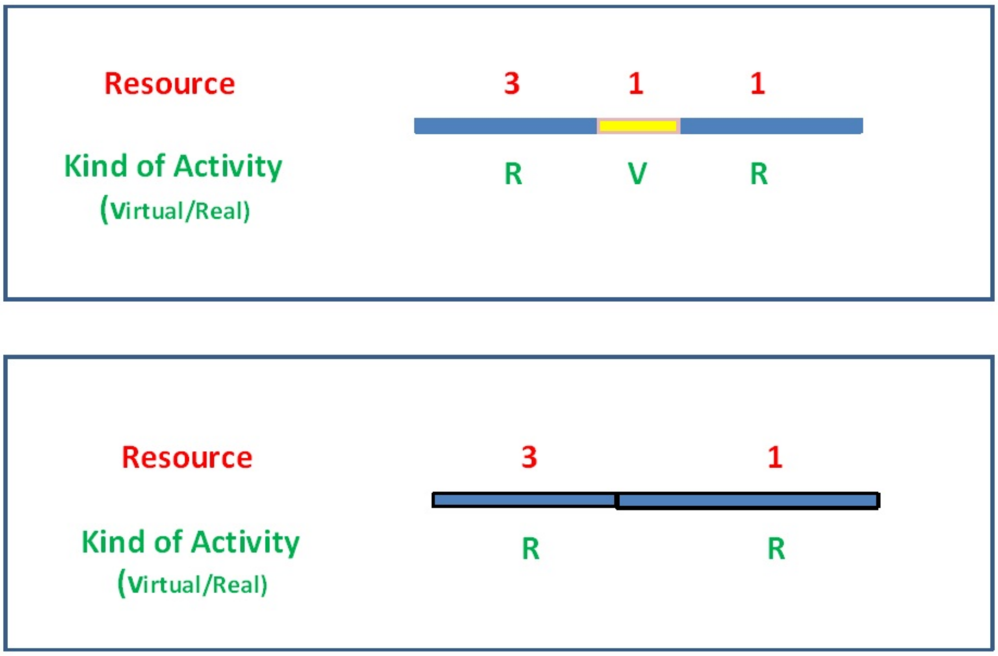

Figure 5.16: Objective Function

The results show a very good computation time despite the reduced complexity of instances. We tried to solve unfeasible instances but the solver was unable to stop the calculation in accordance with the time limits of 4400 seconds. Future research will be dedicated to develop a custom algorithm to check the feasibility of instances.

Table 5.8: Outcomes

Number of trains	Number of Tasks	Number of maintenance resources	Number of Cleaning resources	Time (sec)	obj function
4	10	5	5	0,17	5
4	10	4	4	0,13	5
4	10	3	3	0,08	5
4	10	2	2	0,67	7
4	10	1	2		-
4	10	2	1		-
5	11	5	5	0,22	6
5	11	4	4	0,18	6
5	11	3	3	0,16	6
5	11	2	2	0,16	6
5	11	1	2		-
5	11	2	1		-
3	6	5	5	0,09	3
3	6	4	4	0,08	3
3	6	3	3	0,05	3
3	6	2	2	0,09	3
3	6	1	2	0,03	4
3	6	2	1	0,08	3
3	6	2	1		-
6	10	5	5	0,22	3
6	10	4	4	0,14	3
6	10	3	3	0,13	3
6	10	2	2	0,13	3
6	10	1	2		-
6	10	2	1		-

Chapter 6

Alternative Framework

Sometimes, in case of a big workshop, a “Buffered" maintenance management could be suitable. However, we can still use the tools introduced in the previous chapters of this PhD thesis. The framework presented before is not totally able to solve the “Buffered" maintenance scheduling problem but likely we can adapt it. The question is : "How do we do that"?
Firstly, we could break in two the problem of rolling stock rostering. In a first step we could solve this problem neglecting maintenance constraints. In this formulation sub-loops elimination constraints are compulsory to avoid rosters involving only passenger station not close to the workshop. Therefore, this formulation will be focused only on the minimization of the number of train asset units needed to cover the given timetable and on the minimization of empty runs. In a second step we could find an optimal/feasible set of maintenance tasks to provide. At last we could find an optimal schedule for maintenance tasks selected in the previous formulation.
That means, we can solve the problem using for the step 1 a short formulation of rolling stock rostering and for the step 3 the first approach for the maintenance scheduling problem.
So, given a roster, we need to find a formulation that is able to answer, for each asset unit with end of its journey in a passenger station close to the maintenance footprint, if maintenance operations

must be provided and if the train must come back to workshop or not. For the step two we could use successfully the formulation used for rolling stock rostering problem in case of non-cyclic timetable (for instance we could use the most efficient model). In fact, if we consider roster blocks between successive stops in passenger stations close to workshop as commercial services to be delivered, the step two becomes a rolling stock rostering problem in which the objective function is the minimization of train units.

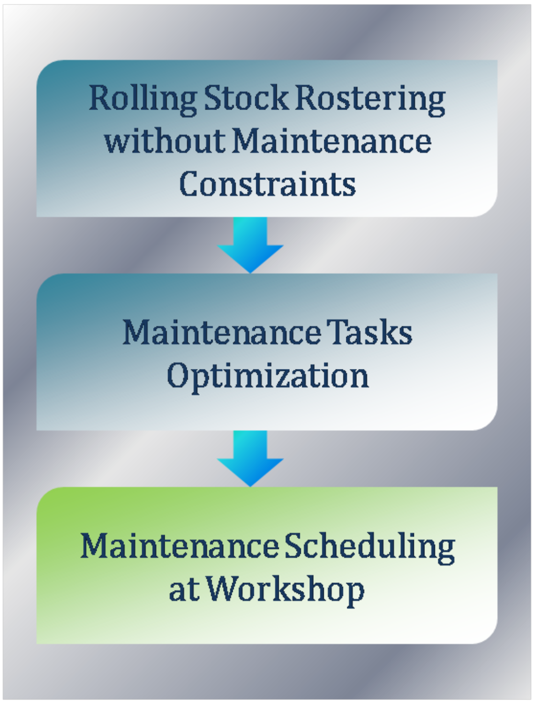

Figure 6.1: Alternative framework

Let's try to explain the Figure 6.1 with an example. Given a cyclic or non-cyclic timetable and an upper-bound on the number of empty runs allowed, we solve the rostering problem using the formulations we have shown so far. Our solution will be a train roster minimizing costs related to asset usage and costs to provide empty runs (see Figure 6.2).
We can brake the roster for each stop in Naples station (if the workshop in analysis is the IDP of Naples) obtaining the parts as shown

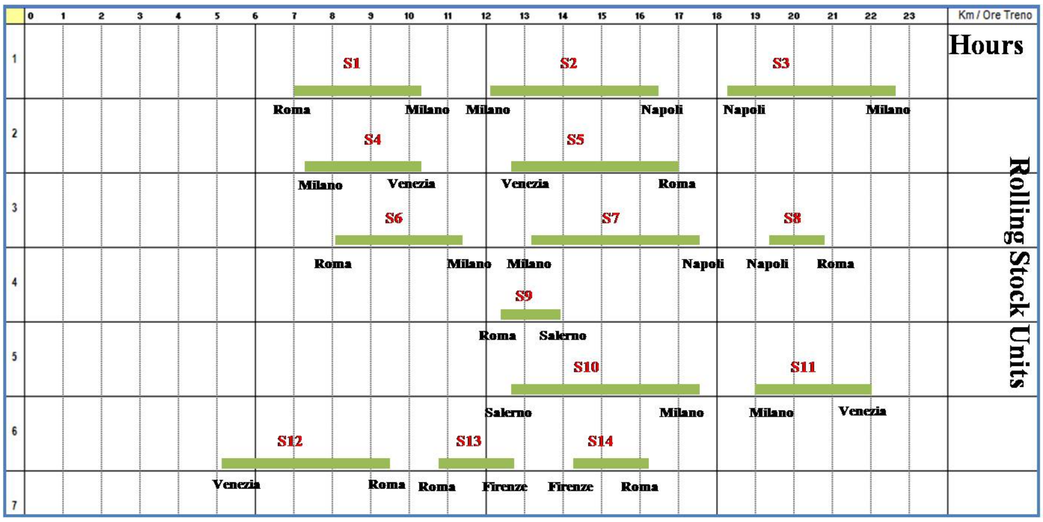

Figure 6.2: Roster example

in the Figure 6.3.
Each roster part will become a node for a new virtual non-cyclic rolling stock rostering problem and its kilometres are given by the sum of kilometres of services involved. In this problem we also can add on a function objective for example on the costs of maintenance operations. The solution of this sub-problem will be a set of maintenance jobs to scheduled in the workshop. At last we need to schedule these maintenance jobs at the workshop exactly as we have seen before.
It is not possible to say in advance if a “Buffered" maintenance management is better than “Non-Buffered" approach. We can observe that “Non-Buffered" approach is very efficient but not very flexible. When this approach is chosen, we need to remake rolling stock roster and maintenance scheduling in case of changes of timetables (very frequent event) or in case of delay due to workshop. If a “Buffered" approach is chosen maintenance management has less constraints and often delays not imply remaking of rolling stock rosters. In real word we observe that big workshop very often adopt a “Buffered" kind of maintenance management.

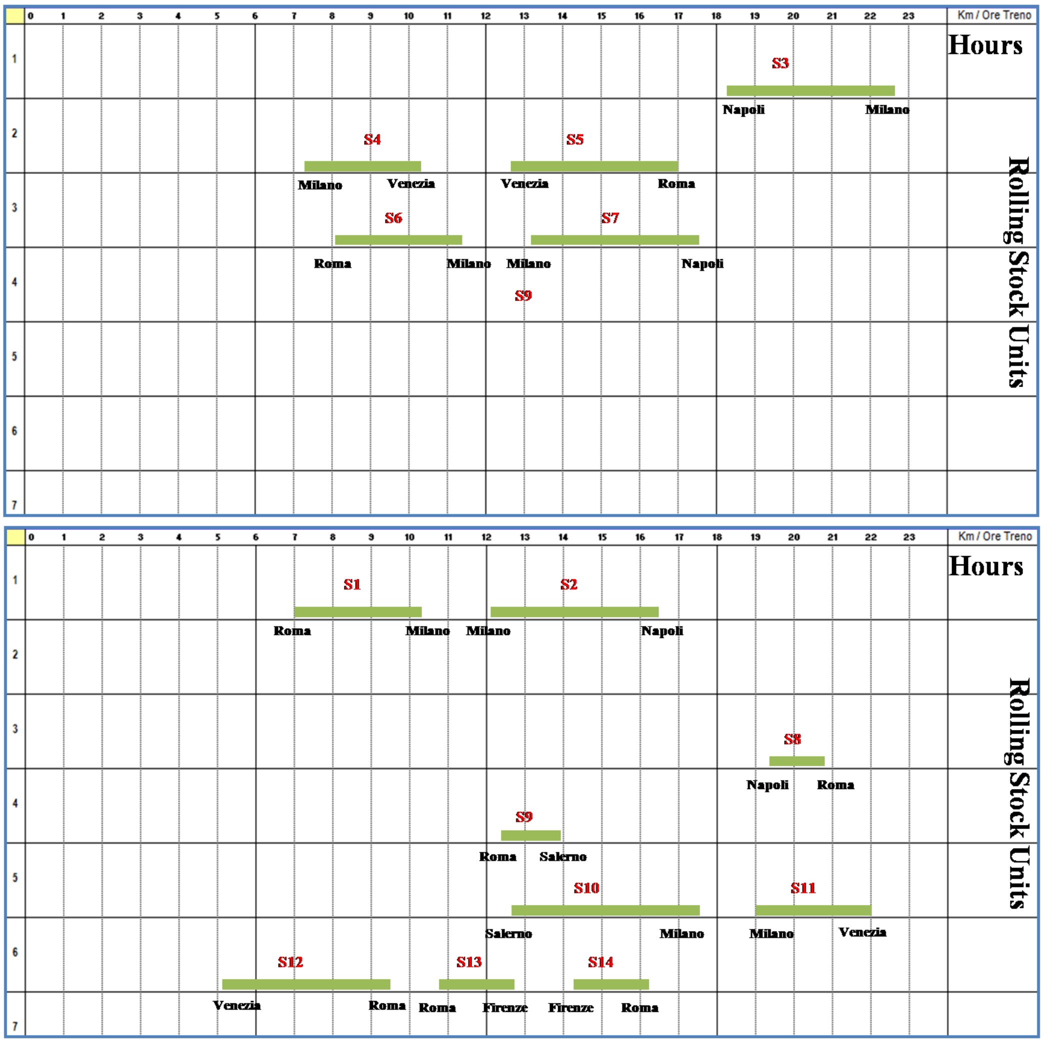

Figure 6.3: Roster decomposition example

Chapter 7

Conclusions

This PhD thesis presents a new framework for rolling stock rostering and maintenance scheduling problems. New formulations are proposed for integrating medium-term maintenance planning in the network-wide railway rolling stock circulation problem. These are key problems in railway planning that require to cover a given set of services and maintenance works with a minimum amount of rolling stock units and with a minimum efficiency of maintenance management. Additional objectives are to minimize the number of empty rides and to maximize the kilometres travelled by each train between two maintenance operations of the same type. The constraints of the rolling stock rostering problem require that the different types of maintenance operations must be carried out for each train periodically. The various maintenance tasks can only be done at a limited number of dedicated sites. The rostering and maintenance scheduling problems are formulated by a graph theoretical approaches.

New formulations are also proposed for the workshop maintenance sub-problem, that is to find feasible schedules with a minimum number of movements within the workshop. Experimental results on real-world scenarios from Trenitalia show that this integrated approach can reduce significantly the number of trains, empty runs and movements within workshop when compared with the current rolling stock circulation and maintenance plans. For a set

of timetables and rolling stock categories, we compare flexible versus rigid plans regarding the number of empty rides and maintenance kilometres. The computational evaluation presents the efficiency of the new solutions compared to the practical solutions. Our model solution with flexible empty rides presents up to 23% reduction in the number of trains needed to cover all services. Furthermore experimental results show an interesting reduction of movements at workshop during maintenance routing and also a very good quality of maintenance schedule in case of medium-term traffic disturbances that alter the off-line plan of operations.
A commercial solver is able to solve practical-size instances of this problem in a few minutes, so the proposed formulation can also be adopted to compute good quality solutions in real-time. Future research will be dedicated to the improvement of the framework, the analysis of its sensitivity to uncertainty, and the investigation of global formulations and algorithmic approaches.
We use a commercial a MIP solver in order to compute optimal solutions in a short computation time.

Bibliography

[1] Abril, M., Barber, F., Ingolotti, L., Salido,M.A., Tormos, P., Lova, A. (2008). An assessment of railway capacity. *Transportation Research Part E*, Vol. 44(5), pp. 774–806.

[2] Ahuja, R.K., Cunha, C.B., Şahin, G. (2005). Network Models in Railroad Planning and Scheduling. *TutORials in Operations Research*, H.J. Greenberg, J.C. Smith (Eds.), pp. 54–101.

[3] Albers,M. (2009). Freight Railway Crew Scheduling: Models, Methods, and Applications. Logos

[4] Alfieri, A., Groot, R., Kroon, L.G., Schrijver, A. (2006). Efficient circulation of railway rolling stock. *Transportation Science*, Vol. 40(3), pp. 378–391.

[5] Anderegg, L., Eidenbenz, S., Gantenbein, M., Stamm, C., Taylor, D.S., Weber, B., Widmayer, P. (2003). Train Routing Algorithms: Concepts, Design Choices, and Practical Considerations. *Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments (Proceedings in Applied Mathematics)*, R.E. Ladner (Ed.), pp. 106–118.

[6] Anily, S., Glass, C.A. , Hassin, R. (1998). The scheduling of maintenance service, Discrete Applied Mathematics, 82, pp. 27-42.

[7] Anily, S., Glass, C.A. , Hassin, R. (1999). Scheduling of maintenance services to three machines, Annals of Operations Research, 86, pp. 375-391.

[8] Applegate, D., Cook, W. (1991). A computational study of the job shop scheduling problem. ORSA Journal on Computing 1991; 3:149-156.

[9] Assad, A.A. (1980). Models for Rail Transportation. *Transportation Research A,14:205-220.*

[10] Berrada,M., Stecke, K. (1986). A Branch and Bound Approach for Machine Load Balancing in Flexible Manufacturing Systems. Management Science.

[11] Billionnet, A. (2004). Using integer programming to solve the train-platforming.

[12] Blasum, U., Bussieck, M., Hochst, W., Moll, C., Scheel, H., Winter, T. (2000). Scheduling Trams in the Morning.

[13] Borndörfer, R., Grötschel, M., Lukac, S., Mitusch, M., Schlechte, T., Schultz, S., Tanner, A. (2005). An auctioning approach to railway slot allocation.

[14] Borndörfer, R., Reuther, M., Schlechte, T., Weider, S. (2011). A Hypergraph Model for Railway Vehicle Rotation Planning. *Proceedings of the 11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, A. Caprara, S. Kontogiannis (Eds.), Saarbrücken, Germany.

[15] Brucker, P., Hurink, J., Jurisch, B., WOostmann, B. (1997). A branch and bound algorithm for the open shop problem. Discrete Applied Mathematics 1997; 76:43-59.

[16] Budai, G., Huisman, D., Dekker, R. (2006). Scheduling Preventive Railway Maintenance Activities. *Journal of the Operational Research Society*, Vol. 57(9), pp. 1035–1044.

[17] Bussieck, M., Winter T., Zimmermann, U. (1997). Discrete optimization in public rail transport.

[18] Cacchiani, V., Caprara, A., Toth, P. (2010). Non-Cyclic Train Timetabling and Comparability Graphs.

[19] Cadarso, L., MarÃn, A. (2009). An approach to rapid transit rolling stock assignment. Arrival-TR-0239.

[20] Cadarso, L., Marín, Á. (2011) Robust Rolling Stock and Routing Integration in Rapid Transit Networks.

[21] Caprara, A., Galli, L., Toth, P. Solution of the Train Platforming Problem.

[22] Caprara, A., Kroon, L.G., Monaci, M., Peeters, M., Toth, P. (2006). Passenger Railway Optimization. *Handbooks in Operations Research and Management Science*, C. Barnhart, G. Laporte (Eds.), pp. 129–187.

[23] Caprara, A., Monaci, M., Toth, P., Guida, P.L. (2006). A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics*, Vol. 154(5), pp. 738–753.

[24] Carter,M.W., Price,C.C. (2001). Operations Research: A Practical Introduction. CRC Press.

[25] Carey, M., Carville, S. (1998). Scheduling and platforming trains at busy complex stations.Transportation Research Part A 37 (2003) 195-224

[26] Confessore, G., Liotta, G., Cicini, P., Rondinone, F., De Luca, P. (2009). A simulation-based approach for estimating the commercial capacity of railways.*Proceedings of the 2009 Winter Simulation Conference*, M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin and R.G. Ingalls (Eds.), Austin, Texas, USA.

[27] Cordeau, J., Soumis, F., Desrosiers, J. (2001). Simultaneous assignment of locomotives and cars to passenger trains.

[28] Cornelsen, S., Di Stefano, G. (2007). Track assignment.

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

[30] Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M. (2010). Railway dynamic traffic management in complex and densely used networks. In R.R. Negenborn, Z. Lukszo, H. Hellendoorn (Eds.), *Intelligent Infrastructures, Intelligent Systems, Control and Automation: Science and Engineering* Vol. 42, pp. 377–404

[31] Dahlhaus E, Horak P, Miller M, Ryan J (2000). The train marshalling problem.

[32] D'Ariano, A., Pacciarelli, D., Pranzo, M. (2008). Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research, Part C*, Vol. 16(2), pp. 232–245

[33] De Luca Cardillo, D., Mione, N. (1999). k L-List T Colouring of Graphs.

[34] Desaulniers, G., Hamdouni, M., Soumis, F., Marcotte, O., Van Putten, M. (2006). Dispatching Buses in a Depot using Block Patterns.

[35] Dorndorf, U., Pesch, E., Phan-Huy, T. (2001). Solving the open shop scheduling problem. JOURNAL OF SCHEDULING J. Sched. 2001; 4:157-174 (DOI: 10.1002/jos.073).

[36] European Commission, Luxembourg: Office for Official Publications of the European Communities (2001). White paper - European Transport Policy: Time to Decide. *ISBN 92-894-0341-1* , 119 pp., pp. 22

[37] Erlebach, T., Gantenbein, M., Hürlimann, D., Neyer, G., Pagourtzis, A., Penna, P., Schlude, K., Steinhöfel, K., Taylor, D.S., P. Widmayer. (2001). On the Complexity of Train Assignment Problems. *Lecture Notes in Computer Science*, Volume 2223, pp. 390–402

[38] Eidenbenz, S., Pagourtzis, A., Widmayer, P. (2003). Flexible Train Rostering. *Lecture Notes in Computer Science*, Volume 2906, pp.615–624

[39] Fioole, P. , Kroon, L. , Maroti, G. , Schrijver, A. (2008). Circulation of railway rolling stock: a branch-and-price approach.

[40] Freling, R., Lentink, R.M., Kroon, L.G., Huisman, D. (2005). Shunting of passenger train units in a railway station.

[41] Fu, Z., Li, L., Li, Z. (2002). Models and Algorithms for the Optimization of Passenger Train Departure and Arrival Time Windows in Scheduling. Traffic and Transportation Studies (2002): pp. 492-497.

[42] Garg, A., Deshmukh, S.G. (2006). Maintenance management: literature review and directions. *Journal of Quality in Maintenance Engineering Vol. 12, N. 3, pp. 205-238.*

[43] Giacco, G., D'Ariano, A., Pacciarelli, D. (2013). Rolling stock rostering optimization under maintenance constraints.*Journal of Intelligent Transportation Systems (in print).*

[44] G.L. Giacco, A. D'Ariano, D. Pacciarelli (2012). Mixed-integer linear-programming formulations for a railway rolling stock circulation problem, *Proceedings of the 12th International Conference on Advanced Systems for Public Transport, Santiago, Chile.*

[45] Giacco, G., Carillo,D., D'Ariano, A., Pacciarelli, D. (2013). Short-term rolling stock rostering and maintenance scheduling optimization.*Ingegneria Ferroviaria (in print).*

[46] Giacco, G.L., D'Ariano, A., Pacciarelli, D. (2011). Rolling stock rostering optimization under maintenance constraints, Proceedings of the 2nd International Conference on Models and Technology for Intelligent Transportation Systems, MT-ITS 2011, pp. 1–5, 22-24 June 2011, Leuven, Belgium

[47] Goverde, R., Koelemeijer, G. (2000). Performance evaluation of periodic railway timetables: Theory and algorithms.

[48] Guneret, C., Jussien, N., Prins, C. (1998). Using intelligent backtracking to improve branch and bound methods: an application to open shop problems. Technical Report, N Ecole des Mines de Nantes, Nantes, 1998.

[49] Hansen, I.A., Pachl, J. (Eds.) (2008). *Railway Timetable and Traffic: Analysis, Modelling and Simulation*, Eurailpress, Hamburg, Germany

[50] Higgins, A. (1998). Scheduling of railway maintenance activities and crews, *Journal of the Operational Research Society, Vol. 49, N. 1, pp. 1026-1033.*

[51] Hrivnák, M., Križanová, A. (2007). Position of rail transport in EU common transport policy.

[52] Huisman, D., Kroon, L.G., Lentink, R.M., Vromans, M.J.C.M. (2005). Operations Research in Passenger Railway Transportation. *Statistica Neerlandica,59(4):467-497.*

[53] Ilog (2010). Cplex user's manual, ILOG.

[54] Khan,M.B., Zhou, X. (2010).Stochastic Optimization Model and Solution Algorithm for Robust Double-Track Train-Timetabling Problem. Intelligent Transportation Systems, IEEE Transactions on (Volume:11 , Issue: 1).

[55] Kontaxi, E., Ricci, S. (2010). Railway capacity analysis: Methodological framework and harmonization perspectives.*Proceedings of the 12th World Conference on Transport Research (WCTR 2010)*, J. Viegas, R. Macário (Eds.), Lisbon, Portugal.

[56] Kroon, L., Dekker, R., Maroti, G. , Vromans, M. (2006). Stochastic Improvement of Cyclic Railway Timetables.

[57] Kroon, L., Lentink, R.M., Schrijver, L. (2006). Shunting of Passenger Train Units: An Integrated Approach.

[58] Lawler,E. L., Wood, D. E. (1966). Branch-and-Bound Methods: A Survey.

[59] Lichtenegger, M. (1990). Der taktfahrplan (The periodic timetable).

[60] Lingaya, N., Cordeau, J., Desaulniers, G., Desrosiers, J., Soumis, F. (2001). Operational car assignment at VIA Rail Canada.

[61] Lopez, P., Roubellat, F., Christian Prins, C. (2010). Open Shop Scheduling. Wiley Online Library.

[62] Lubbecke, M., Zimmermann, U. (2003). Shunting minimal rail car allocation.

[63] Lusby, R.M., Larsen, J., Ehrgott, M., Ryan, D. (2011). Railway track allocation: models and methods. *OR Spectrum*, Vol. 33(4), pp. 843–883

[64] Marinov, M., Viegas, J. (2011). A mesoscopic simulation modelling methodology for analyzing and evaluating freight train operations in a rail network. *Simulation Modelling Practice and Theory*, Vol. 19(1), pp. 516–539

[65] Mároti, G., Kroon, L.G. (2005). Maintenance routing for train units: the transition model. *Transportation Science*, Vol. 39(4), pp. 518–525

[66] More, J.J.,Stephen, J.W. (1994). Optimization Software Guide. Society fo Industrial Applied Mathematics.

[67] Nachtigall, K. (1998). Periodic network optimization and fixed interval timetables.

[68] Nachtigall, K., Voget, S. (1996). A genetic algorithm approach to periodic railway synchronization.

[69] Nonner, T., Souza, A. (1998).Optimal Algorithms for Train Shunting and Relaxed List Update Problems.

[70] Oum, T.H., Waters,W. G., Yu, C. (1999). A Survey of Productivity and Efficiency Measurement in Rail Transport.Journal of Transport Economics and Policy Vol. 33, No. 1 (Jan., 1999), pp. 9-42.

[71] Peeters, M., Kroon, L. (2007). Circulation of railway rolling stock: a branch-and-price approach.

[72] Penicka, M., Strupchanska A., Bjorner, D. (2003). Train maintenance routing. *Symposium on Formal Methods for Railway Operation and Control Systems, Hungary.*

[73] Pietrantonio,L. D. , Pelkmans, J. The economics of eu railway reform. College of Europe, Bruges European Economic Policy Briefings, Sep. 2004.

[74] Ramani, K., Mandal, B. (1992). Operational planning of passenger trains in Indian Railways.

[75] Schlechte, T. (2011). Railway Track Allocation - Simulation and Optimization.

[76] Schöbel, A. (2006). Optimization in Public Transportation. Springer.

[77] Serafini, P., Ukovich, W. (1989). A mathematical model for periodic scheduling problems.

[78] He, S., Song, R. , Chaudhry,S. (2000). Fuzzy dispatching model and genetic algorithms for railyards operations.

[79] Sherali, H. (1994). A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes. Volume 28, Issue 3, June 1994, Pages 213-233.

[80] Silver, E.A. and Murphy, G.F. (1994). Cost analysis and extension of a simple maintenance scheduling heuristic, NRL (U.S.) 41(7), pp. 945 - 958.

[81] Sivarethinamohan,R. (2008). Operations Research. McGraw Hill.

[82] Srinivasan,G. (2010). Operations Research : Principles and Applications. PHI New Delhi.

[83] Sriskandarajah, C., Jardine, A. K. S., Chan, C. K. (1998). Maintenance Scheduling of Rolling Stock Using a Genetic Algorithm. *Journal of the Operational Research Society, Vol. 49, No. 11, pp. 1130-1145.*

[84] Thoft-Christensen,P., Murotsu,Y. (1986). The Branch-and-Bound Method.

[85] Tomii, N. , Zhou, L. (1999). Shunting scheduling problem.

[86] Tomii, N. , Zhou, L. (2000). Depot shunting scheduling using combined genetic algorithms.

[87] Waeyenbergh, G., and Pintelon, L. (2002) A framework for maintenance concept development. *International Journal of Production Economics, Vol. 77, N. 3, pp. 299-313.*

[88] Wang, L., Chu, J., Wu, J. (2007). Selection of optimum maintenance strategies based on a fuzzy analytic hierarchy process.*International Journal of Production Economics, Vol. 107, N. 1, pp. 151-163.*

[89] Wang, W., Chen, H., Bell, M.C. (2005). Vehicle breakdown duration modelling, Journal of Transportation and Statistics,8(1), pp. 75-84.

[90] Weigand, W. (1983). The man-machine dialogue and timetable planning.

[91] Wendler, E. (2006). The scheduled waiting time on railway lines.

[92] Zhou,W., Li-min,J.. The Theory and Method of Design and Optimization for Railway Intelligent Transportation Systems.

[93] Ziarati, K., Soumis, F., Desrosiers, J., Gelinas, S., Saintonge, A. (1997). Locomotive Assignment with Heterogeneous Consists at CN North America.

[94] Zwaneveld, P.J., Kroon, L.G., Romeijn, H.E., Salomon, M., Dauzere- Peres, S., Van Hoesel, C.P.M., Ambergen, H.W. (1996). Routing Trains Through Railway Stations: Model Formulation and Algorithms. Transportation Science, 30(3):181-194.

Appendix A

Glossary

We provide here a short explanation of main terminologies used in the previous chapters.

Buffered Maintenance:
A specific kind of maintenance management in which workshop manager could decide the train asset units to process without constraints.

Corrective Maintenance:
Maintenance work intended to correct an existing problem.

Cyclic Timetable:
A timetable not changing day by day for all period for which it is in force.

Duty:
The workload of an asset unit for a certain period.

Empty Run:
Train without commercial aims.

Infrastructure Manager (IM):
Any body or undertaking responsible for establishing and main-

taining railway infrastructure. This may also include the management of infrastructure control and safety systems. The functions of the Infrastructure Manager on a network may be assigned to different bodies or undertakings.

Maintenance:
Activity aiming to maintain something in good working order, prevent operational disturbance and/or uphold a given technical standard.

MIP Solver:
Software able to solve mixed integer programming problems.

Non-buffered Maintenance:
A specific kind of maintenance management in which workshop manager provide maintenance task accordingly to the rolling roster.

Pattern:
Sets of transports with the same characteristics (stops and travel distances) and repeated daily with a fixed frequency for a given time window.

Periodicity:
It's a description of the days in which a train service is provided.

Predictive Maintenance:
This kind of maintenance is based upon the actual condition of the equipment and a determination of when maintenance should be performed to minimize costs. New technology techniques such as ultrasound, infrared and vibration online testing make predictive maintenance a viable alternative in certain circumstances. However, for most equipment the complex metrics for making educated guesses (predictive) is provided by preventive maintenance

programs.

Preventive Maintenance:
The care of equipment and facilities in satisfactory operating condition by providing for systematic inspection, detection, and correction of incipient failures either before they occur or before they develop into major defects.

Railway undertaking (RU):
In the context of licensing, any private or public undertaking the principal business of which is to provide rail transport services for goods and/or passengers, with a requirement that the undertaking must ensure traction.

Rolling stock:
The collective term for the rail fleet; sometimes it is used for one vehicle. It describes all the vehicles that are used on a railway track. It usually includes both powered and unpowered vehicles, for example locomotives, hauled passenger vehicles and freight vehicles (coaches and wagons), diesel units, electric units and service stock. The term is sometimes used to refer only to non-powered vehicles, thus excluding locomotives. The term contrasts with fixed stock (infrastructure), which is a collective term for the track, signals, stations and buildings etc. necessary to operate a railway.

Rolling stock circulation:
The collection of the rolling stock duties for a certain period.

Route:
A (railway) route can be seen on a map and has a physical existence, unlike a (railway) path, which is part of a timetable.

Spot train:
Train not belonging to any pattern.

Station:
A place where trains stop, or where loading and unloading occurs, and where assistance may be available. Also a place where there can be points (facing or trailing) that make it possible for the train to use different routes.

Timetable:
A schedule listing the times at which certain events, such as arrivals and departures at a transport station, are expected to take place. The timetable defines all planned train and rolling-stock movements which will take place on the relevant infrastructure during the period for which it is in force.

Timetable changeover / timetable change:
The date on which the old timetable ceases to be valid, and the new timetable starts to take effect. The timetable change is regulated by the EU and has taken place in December every year since 2004 across Europe.

Timetable drafting:
The period of timetable development during which a draft timetable is prepared.

Timetable period:
A timetable period means the period of operation of a Working Timetable; it starts on the day of a timetable change (change date).

Timetable planning process / timetabling process:
A complex process of consultation and planning which defines the data relating to all train and rolling-stock movements that are expected to take place on the relevant infrastructure during the period of validity of the timetable. Detailed train timings are agreed by IMs and RUs.

Train:
One or more railway vehicles capable of being moved. It may consist of a locomotive (sometimes more than one) to provide power with various unpowered vehicles attached to it. It may consist of a multiple unit, i.e. several vehicles formed into a fixed formation or set, which carry their own power and do not require a locomotive. A train may be only a locomotive running light (deadheading) to a point elsewhere on the railway. A train may carry passengers, freight or, rarely nowadays, both.

Train-day:
It's a single train service running in a day.

Vehicle (railway):
Railway vehicle that runs on its own wheels on railway lines, with or without traction. A vehicle is composed of one or more structural and functional subsystems or parts of such subsystems.

Appendix B

List of abbreviations

CM	Corrective Maintenance
GA	Genetic Algorithm
IM	Infrastructure Manager
LP	Linear Programming
MIP	Mixed Integer Programming
MS	Maintenance Scheduling
OR	Operations Research
PESP	Periodic Event Scheduling Problem
PM	Predictive Maintenance
PSS	Passenger Station Scheduling
RSM	Rolling Stock Management
RSR	Rolling Stock Rostering
RU	Railway Undertaking
TTP	Train Timetabling Problem
TUSP	Train Unit Shunting Problem

Appendix C

Rostering numerical example lp file

Minimize $obj : X\_A\_C\_s + X\_A\_C\_MC\_NA\_IMC + X\_A\_D\_v$
$+X\_A\_D\_vi\_MC\_NA\_IMC + X\_B\_C\_s + 2X\_B\_C\_MC\_NA\_IMCv$
$+X\_B\_D\_v + 2X\_B\_D\_vi\_MC\_NA\_IMC + X\_C\_A\_s + X\_D\_A\_s$

Subject To
Path constraints

$vh1 : X\_A\_C\_s + X\_A\_C\_MC\_NA\_IMC + X\_A\_D\_v$
$+X\_A\_D\_vi\_MC\_NA\_IMC = 1$
$vh2 : X\_B\_C\_s + X\_B\_C\_MC\_NA\_IMC + X\_B\_D\_v$
$+X\_B\_D\_vi\_MC\_NA\_IMC = 1$
$vh3 : X\_C\_A\_s + X\_C\_B\_s = 1$
$vh4 : X\_D\_A\_s + X\_D\_B\_s = 1$
$vh5 : X\_C\_A\_s + X\_D\_A\_s = 1$
$vh6 : X\_C\_B\_s + X\_D\_B\_s = 1$
$vh7 : X\_A\_C\_s + X\_A\_C\_MC\_NA\_IMC + X\_B\_C\_s$
$+X\_B\_C\_MC\_NA\_IMC = 1$
$vh8 : X\_A\_D\_v + X\_A\_D\_vi\_MC\_NA\_IMC + X\_B\_D\_v$
$+X\_B\_D\_vi\_MC\_NA\_IMC = 1$

Optional subloops elemination constraints

$vl1 : Y\_A\_C + Y\_A\_D = 1$
$vl2 : Y\_B\_C + Y\_B\_D - Y\_C\_B - Y\_D\_B = 1$
$vl3 : -Y\_A\_C - Y\_B\_C + Y\_C\_B + Y\_C\_A = 1$
$vl4 : -Y\_A\_D - Y\_B\_D + Y\_D\_B + Y\_D\_A = 1$
$vl5 : -4X\_A\_C\_s - 4X\_A\_C\_MC\_NA\_IMC+$
$Y\_A\_C \leq 0$
$vl6 : -4X\_A\_D\_v - 4X\_A\_D\_vi\_MC\_NA\_IMC$
$+Y\_A\_D \leq 0$
$vl7 : -4X\_B\_C\_s - 4X\_B\_C\_MC\_NA\_IMC$
$+Y\_B\_C \leq 0$
$vl8 : -4X\_B\_D\_v - 4X\_B\_D\_vi\_MC\_NA\_IMC + Y\_B\_D \leq 0$
$vl9 : -4X\_C\_A\_s + Y\_C\_A \leq 0$
$vl10 : -4X\_C\_B\_s + Y\_C\_B \leq 0$
$vl11 : -4X\_D\_A\_s + Y\_D\_A \leq 0$
$vl12 : -4X\_D\_B\_s + Y\_D\_B \leq 0$

Maintenance constraints

$man1 : -220X\_A\_D\_v + G\_A\_C\_s\_MC + G\_A\_C\_MC\_NA\_IMC\_MC$
$+G\_A\_D\_v\_MC + G\_A\_D\_vi\_MC\_NA\_IMC\_MC$
$-G\_C\_A\_s\_MC - G\_D\_A\_s\_MC = 600$

$man2 : -220X\_B\_D\_v + G\_B\_C\_s\_MC$
$+G\_B\_C\_MC\_NA\_IMC\_MC + G\_B\_D\_v\_MC$
$+G\_B\_D\_vi\_MC\_NA\_IMC\_MC - G\_C\_B\_s\_MC$
$-G\_D\_B\_s\_MC = 600$
$man3 : -G\_A\_C\_s\_MC + G\_C\_A\_s\_MC - G\_B\_C\_s\_MC$
$+G\_C\_B\_s\_MC = 600$
$man4 : -220X\_A\_D\_vi\_MC\_NA\_IMC - 220X\_B\_D\_vi\_MC\_NA\_IMC$
$-G\_A\_D\_v\_MC + G\_D\_A\_s\_MC - G\_B\_D\_v\_MC$
$+G\_D\_B\_s\_MC = 480$
$man5 : -3000X\_A\_C\_s + G\_A\_C\_s\_MC \leq 0$
$man6 : -3000X\_A\_C\_MC\_NA\_IMC$

$+G\_A\_C\_MC\_NA\_IMC\_MC \leq 0$
$man7: -3000X\_A\_D\_v + G\_A\_D\_v\_MC \leq 0$
$man8: -3000X\_A\_D\_vi\_MC\_NA\_IMC$
$+G\_A\_D\_vi\_MC\_NA\_IMC\_MC \leq 0$
$man9: -3000X\_B\_C\_s + G\_B\_C\_s\_MC \leq 0$
$man10: -3000X\_B\_C\_MC\_NA\_IMC+$
$G\_B\_C\_MC\_NA\_IMC\_MC \leq 0$
$man11: -3000X\_B\_D\_v + G\_B\_D\_v\_MC \leq 0$
$man12: -3000X\_B\_D\_vi\_MC\_NA\_IMC$
$+G\_B\_D\_vi\_MC\_NA\_IMC\_MC \leq 0$
$man13: -3000X\_C\_A\_s + G\_C\_A\_s\_MC \leq 0$
$man14: -3000X\_C\_B\_s + G\_C\_B\_s\_MC \leq 0$
$man15: -3000X\_D\_A\_s + G\_D\_A\_s\_MC \leq 0$
$man16: -3000X\_D\_B\_s + G\_D\_B\_s\_MC \leq 0$
$man17: -1200X\_A\_C\_MC\_NA\_IMC$
$+G\_A\_C\_MC\_NA\_IMC\_MC \geq 0$
$man18: -1200X\_A\_D\_vi\_MC\_NA\_IMC$
$+G\_A\_D\_vi\_MC\_NA\_IMC\_MC \geq 0$
$man19: -1200X\_B\_C\_MC\_NA\_IMC$
$+G\_B\_C\_MC\_NA\_IMC\_MC \geq 0$
$man20: -1200X\_B\_D\_vi\_MC\_NA\_IMC$
$+G\_B\_D\_vi\_MC\_NA\_IMC\_MC \geq 0$

maximum number of empty runs constraint allowed

$er: X\_A\_D\_v + X\_A\_D\_vi\_MC\_NA\_IMC + X\_B\_D\_v$
$+X\_B\_D\_vi\_MC\_NA\_IMC \leq 1$

Bounds

$0 \leq X\_A\_C\_s \leq 1$
$0 \leq X\_A\_C\_MC\_NA\_IMC \leq 1$
$0 \leq X\_A\_D\_v \leq 1$

$0 \leq X\_A\_D\_vi\_MC\_NA\_IMC \leq 1$
$0 \leq X\_B\_C\_s \leq 1$
$0 \leq X\_B\_C\_MC\_NA\_IMC \leq 1$
$0 \leq X\_B\_D\_v \leq 1$
$0 \leq X\_B\_D\_vi\_MC\_NA\_IMC \leq 1$
$0 \leq X\_C\_A\_s \leq 1$
$0 \leq X\_C\_B\_s \leq 1$
$0 \leq X\_D\_A\_s \leq 1$
$0 \leq X\_D\_B\_s \leq 1$
$0 \leq Y\_A\_C \leq 4$
$0 \leq Y\_A\_D \leq 4$
$0 \leq Y\_B\_C \leq 4$
$0 \leq Y\_B\_D \leq 4$
$0 \leq Y\_C\_B \leq 4$
$0 \leq Y\_D\_B \leq 4$
$0 \leq Y\_C\_A \leq 4$
$0 \leq Y\_D\_A \leq 4$
$0 \leq G\_A\_C\_s\_MC \leq 3000$
$0 \leq G\_A\_C\_MC\_NA\_IMC\_MC \leq 3000$
$0 \leq G\_A\_D\_v\_MC \leq 3000$
$0 \leq G\_A\_D\_vi\_MC\_NA\_IMC\_MC \leq 3000$
$0 \leq G\_C\_A\_s\_MC \leq 30008$
$0 \leq G\_D\_A\_s\_MC \leq 3000$
$0 \leq G\_B\_C\_s\_MC \leq 3000$
$0 \leq G\_B\_C\_MC\_NA\_IMC\_MC \leq 3000$
$0 \leq G\_B\_D\_v\_MC \leq 3000$
$0 \leq G\_B\_D\_vi\_MC\_NA\_IMC\_MC \leq 3000$
$0 \leq G\_C\_B\_s\_MC \leq 3000$
$0 \leq G\_D\_B\_s\_MC \leq 3000$

Binaries

$X\_A\_C\_s$ $X\_A\_C\_MC\_NA\_IMC$ $X\_A\_D\_v$ $X\_A\_D\_vi\_MC\_NA\_IMC$
$X\_B\_C\_s$ $X\_B\_C\_MC\_NA\_IMC$ $X\_B\_D\_v$ $X\_B\_D\_vi\_MC\_NA\_IMC$

$X\_C\_A\_s$ $X\_C\_B\_s$ $X\_D\_A\_s$ $X\_D\_B\_s$

End