Crew Scheduling in a Train Operating Company –

Fertagus Case Study

Tiago Miguel de Pina Alves

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisor: Prof. António Ramos Andrade

Examination Committee

Chairperson: Prof. João Orlando Marques Gameiro Folgado

Supervisor: Prof. António Ramos Andrade

Member of the Committee: Prof. Amilcar José Martins Arantes

June 2019

Acknowledgments

First of all, I would like to thank my dissertation supervisor, Professor António Ramos Andrade. At all times, his availability and helpfulness, keeping me in the right path through hard times, were essential to the conclusion of this research.

Furthermore, I would like to address both Engineers João Duarte and João Grossinho from Fertagus railway company. Both had the time to answer all my doubts, to provide all the Fertagus data essential to the present case study, whether it was through personal meetings, emails or phone calls, as well as, making it possible the access to company's maintenance yard.

I would also like to acknowledge FICO Xpress software for the academic license of FICO Xpress Optimization software, used to run this model.

I would also like to thank my university, Instituto Superior Técnico, for all the values and opportunities given all over these years, which became essential in my personal and professional development.

Finally, I would like to say a word to my family, especially my mother, father and brothers, my girlfriend and lastly to all my friends who motivated me in difficult times, providing all the support and motivation necessary to complete this dissertation.

Abstract

Essential to any metropolitan area, public transportation services aim to meet the passenger demand. Lately, emergent and disruptive technologies are being used for the railway sector. In line with this, the initial aim of this dissertation is to conceive a mathematical model which outputs an optimal daily maintenance crew schedule for a train operating company, Fertagus. One of the major contributions of this model is related with the introduction of skillsets for each maintenance crew member. Later, it was decided to integrate the driving crew scheduling problem to the initial model, and thus, obtain an optimal daily schedule comprehending both crews. The present work follows the works by Mira (2018) and Méchain (2017). This way, a mixed-integer linear programming model, considering the preventive maintenance actions scheduled for each day of the week, is presented, assigning both the maintenance crew and drivers to a daily scheduling, while minimizing the associated costs. The model outputs a data file indicating where and when each worker carries out the respective maintenance action and something similar concerning drivers and tasks. The program is initially validated by an illustrative example and is then applied to a real case scenario. While it was possible to obtain an optimal schedule for the maintenance crew, concerning Fertagus case study, for the driving crew, only a medium-size problem was possible to solve, and so, larger size instances are left for further research.

Key Words: Railway Management, Maintenance Crew Scheduling, Driving Crew Scheduling, Mixed-Integer-Linear Programming, Optimization

Resumo

Essenciais para qualquer metrópole, os serviços de transporte público procuram responder às exigências dos seus passageiros. Ultimamente, tecnologias emergentes e disruptivas têm vindo a ser usadas para o setor ferroviário. Assim, o objetivo inicial desta dissertação é conceber um modelo matemático que forneça um cronograma diário para a equipa de manutenção de uma empresa ferroviária, Fertagus. Uma das grandes contribuições deste modelo está relacionada com a introdução de competências para cada trabalhador. Mais adiante, decidiu-se integrar o problema do agendamento dos maquinistas ao modelo inicial, obtendo-se assim um cronograma diário compreendendo ambas as tripulações. O trabalho continua estudos realizados anteriormente por Mira (2018) e Méchain (2017). Neste sentido, um modelo de programação linear inteira mista, que considera as ações de manutenção preventiva a ser realizadas para cada dia da semana, é apresentado. O cronograma diário obtido, tanto para a tripulação de manutenção como para os motoristas, tem como pressuposto minimizar os custos associados. O modelo gera um ficheiro de dados indicando onde e quando cada trabalhador realiza a respectiva ação de manutenção e algo semelhante em relação aos motoristas e tarefas. O programa é inicialmente validado por um exemplo ilustrativo e depois é então aplicado ao caso real. Embora tenha sido possível obter um planeamento ideal para a equipa de manutenção para o caso de estudo da Fertagus, para a tripulação de maquinistas, foi apenas possível solucionar um problema de tamanho médio e assim, instâncias de maior escala são deixadas trabalho futuro.

Palavras Chave: Planeamento Ferroviário, Planeamento da Equipa de Manutenção, Planeamento da Equipa de Maquinistas, Programação Linear Inteira Mista, Otimização

Contents

Acknowledgments ii
Abstract iv
Resumo vi
Contents viii
List of Figures x
List of Tables xi
List of Acronyms xii
1 Introduction 1
1.1 Context 1
1.1.1 Transportation Network in Lisbon metropolitan area 1
1.1.2 New technologies for the Railway Sector 3
1.2 Problem Definition and Methodology 4
1.3 Document Structure 5
2 State of the Art 7
2.1 Maintenance scheduling in transportation companies 7
2.2 Maintenance crew scheduling in transportation companies 8
2.3 Driving crew scheduling in transportation companies 9
2.4 Contribution of the Research 11
2.5 Research Gaps and Opportunities 12
3 A Mixed-Integer Linear Programming Model 13
3.1 A Mixed-Integer Linear Programming Model 13
3.2 The Crew Scheduling Problem Definition 14
3.2.1 Maintenance Crew Scheduling Problem Definition 14
3.2.2 Driving Crew Scheduling Problem Definition 15
3.3 Indexes 16
3.4 Sets 17
3.5 Parameters 17
3.6 Constants 17
3.7 Pre-Processing Data 18
3.8 Decision Variables 20
3.9 Objective Function 21
3.10 Constraints 21
3.10.1 Maintenance Crew Constraints 21
3.10.2 Driving Crew Constraints 23
3.10.3 Decision Variables Constraints 26

4 Application – Model Implementation and Validation ... 27
4.1 Model Implementation in FICO Xpress Optimization Software ... 27
4.2 Illustrative Example ... 29
4.3 Results of the Optimization Model for the Illustrative Example ... 35
5 Case Study – Fertagus ... 41
5.1 Fertagus Train Operating Company ... 41
5.2 Input Parameters ... 42
6 Results and Discussion ... 52
6.1 Maintenance Crew Scheduling ... 52
6.2 Driving Crew Scheduling ... 55
6.2.1 Fertagus Case Study ... 55
6.2.2 Medium size Fertagus Case Study ... 57
7 Conclusion ... 60
7.1 Contributions ... 60
7.2 Limitations ... 62
7.3 Future Research ... 63
8 References ... 64
9 Appendix ... 66
A1 – Driving Crew Scheduling of the medium size Fertagus case study ... 66

List of Figures

Figure 1.1 – Mobility System in Lisbon Municipality Area (Retrieved from: Carris website) 1
Figure 1.2 – Railway map of Lisbon metropolitan area (R.f.: CP website) 3
Figure 3.1 – Present dissertation relation's with previous researches 13
Figure 3.2 – Example of maintenance crew scheduling problem with 2 maintenances actions scheduled to be carried out between i and j. 15
Figure 3.3 – Example of a task i, with all the respective information. 15
Figure 3.4 – Illustrative driving worker schedule, entering to service before task i_1 and exiting after i_2 16
Figure 3.5 – Example of a decoupling between tasks i and j 19
Figure 3.6 – Example of a coupling between tasks i and j 19
Figure 3.7 – Example of two simultaneous tasks i and j 20
Figure 3.8 – Diagram concerning constraints (6.1), (6.2), (7) and (8) 22
Figure 3.9 – Example of two consecutive maintenance actions, successfully performed by the same worker 23
Figure 3.10 – Example of two simultaneous maintenance actions, carried out on different units by two different workers 23
Figure 3.11 – Example of two consecutive maintenance actions, carried out on different units, successfully performed by a single maintenance worker 23
Figure 3.12 – Diagram concerning constraints 12.1 to 14.2 25
Figure 4.1 – Optimization Process 27
Figure 4.2 – Rolling-stock plan used in this illustrative example (Adapted from Mira, 2018) 32
Figure 4.3 – Minimum cost obtained in the illustrative example 35
Figure 4.4 – Results for the maintenance crew scheduling problem regarding the toy problem 36
Figure 4.5 – Results for the driving crew scheduling problem regarding the toy problem 36
Figure 4.6 – Illustrative example's computational data 39
Figure 5.1 – Fertagus' railway map (R. f.: Fertagus website) 41
Figure 5.2 – Fertagus' actual rolling-stock schedule for: Monday, Tuesday, Wednesday and Thursday 47
Figure 6.1 – Minimum cost obtained for maintenance crew concerning Fertagus case study ... 52
Figure 6.2 – Case Study's model computational data 52
Figure 6.3 – Results for the maintenance crew scheduling problem concerning Fertagus case study 53
Figure 6.4 – Example of a non-optimal driving crew scheduling concerning a decoupling 55
Figure 6.5 – Example of a non-optimal driving crew scheduling concerning a coupling 56
Figure 6.6 – Excerpt of the Fertagus rolling-stock schedule used in the medium size study 57
Figure 6.7 – Results for the driving crew scheduling problem concerning Fertagus medium size case study 58
Figure 6.8 – Computational data concerning medium size case study's model 59

List of Tables

Table 1.1 – Transportation systems in Lisbon metropolitan area 2
Table 2.1 – Summary of the analysis of the papers on crew and maintenance scheduling 11
Table 4.1 – Constants used 30
Table 4.2 – Maintenance crew competences........ 30
Table 4.3 – Amount of work and duration of maintenance actions 31
Table 4.4 – Information on units going to depot to perform maintenance 31
Table 4.5 – Information on maintenance actions scheduled for each unit........ 31
Table 4.6 – Information regarding stations........ 32
Table 4.7 – Information about tasks........ 33
Table 4.8 – Information on tasks carried out by each unit 33
Table 4.9 – Pair of tasks linked by each unit........ 34
Table 4.10 – Information on the unit and pair of tasks between each maintenance action is completed........ 35
Table 4.11 – Cost of employing a maintenance worker........ 35
Table 4.12 – Cost of employing a driving worker........ 35
Table 4.13 – Distribution of actions for maintenance crew member and tasks for driver 38
Table 4.14 – Calculus of the matrix column 40
Table 5.1 – Constants used 44
Table 5.2 – Maintenance crew competences........ 44
Table 5.3 – Amount of work and duration of maintenance actions........ 45
Table 5.4 – Information on units going to depot to perform maintenance 45
Table 5.5 – Information on maintenance actions scheduled for each unit........ 46
Table 5.6 – Information concerning stations 48
Table 5.7 – Information about tasks........ 48
Table 5.8 – Information on tasks carried out by each unit 49
Table 5.9 – Pair of tasks linked by each unit........ 50
Table 5.10 – Information on the unit and pair of tasks between each maintenance action is completed........ 51
Table 5.11 – Cost of employing a maintenance worker........ 51
Table 5.12 – Cost of employing a driving worker........ 51
Table 6.1 – Distribution of activities for maintenance worker........ 54
Table 6.2 – Calculus of the matrix column........ 56

List of Acronyms

AI	Artificial Intelligence
COP	Constraint Optimisation Programming
CP	Constraint Programming
EU	European Union
FVCSP	Flexible Vehicle and Crew Scheduling Problem
ILP	Integer Linear Programming
MILP	Mixed Integer Linear Programming
MIP	Mixed Integer Programming
S2R	Shift2Rail
VNS	Variable Neighbourhood Search
FRA	Fix-and-Relax Algorithm

1 Introduction

In this first chapter, the research topic of this study is presented, as well as a brief contextualization. In a first instance, the Lisbon metropolitan area transportation network is briefly presented and described. Later, the emergence of new technologies for the railway sector, namely in Artificial Intelligence, is mentioned. Afterwards, the research study is outlined, as the proposed approach and its methodology, and finally, the objectives of the research and the document's structure are presented.

1.1 Context

1.1.1 Transportation Network in Lisbon metropolitan area

Lisbon, as Portugal's capital, is one of the main focus of population in the country whether it is due to professional or personal reasons furthermore is nowadays also a highly rated city for many travellers all across the world. It is in fact becoming a busier city, not only due to the many thousands of tourists that come and go every day, but also consequence of the slight local population growth over the years according to the *United Nations World Urbanization Prospects* (UN 2018). To satisfy passengers high demand across the whole Lisbon metropolitan area, several transportation systems had to be created. In fact, there is a wide range of options to choose from, naming buses, subway, boats and trains. All these different means of transport together, when articulated form an integrated mobility system, essential to any avant-garde city. Figure 1.1 presents an example of a smaller scale integrated system, implemented in the Lisbon Municipality area where the flux of passengers is higher. More lately, a new era of transportation is surging, with the shared means of transport, such as scooters and bikes, etc.

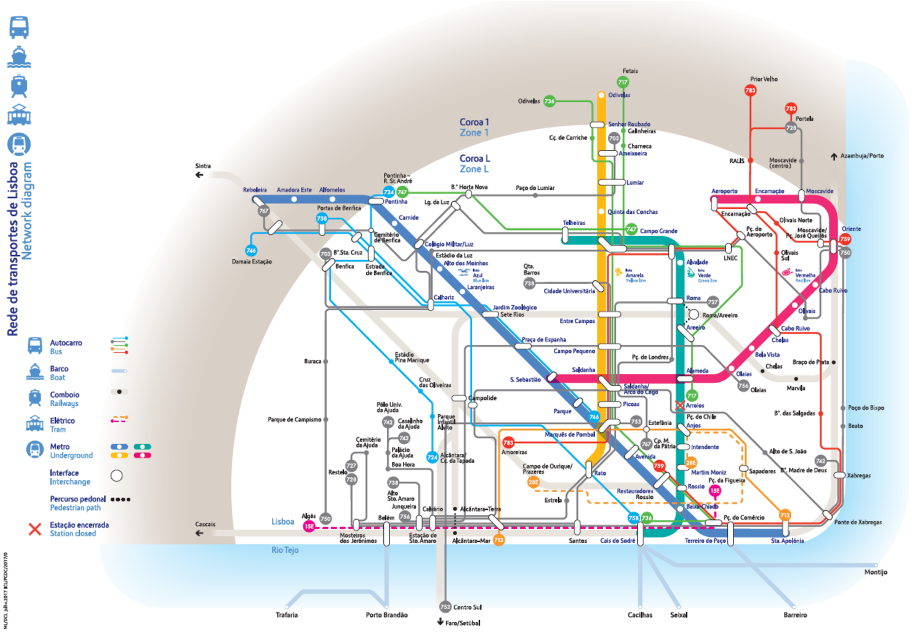


Figure 1.1 – Mobility System in Lisbon Municipality Area (Retrieved from: Carris website)

Table 1.1 displays a brief background on the main operating transportation systems in Lisbon metropolitan area, indicating the operating company of each one, as well as some facts and numbers and finally some relevant features.

Table 1.1 – Transportation systems in Lisbon metropolitan area

Transportation System	Operating Company	Facts & Numbers
Bus/Tram	*Carris* *etc.*	2.112 employees; 651 total units; 79 routes; 122.4 million customers (2017)
Subway	*Metropolitano de Lisboa*	1414 workers; 333 units; 4 lines; 56 stations; 44.5km line extension; 162 million passengers (2017)
Boat	*Transtejo-Soflusa*	28 vessels; 5 routes; 9 terminals (2017)
Train	*Comboios de Portugal (C.P.)*	91 units; 4 routes; 67 stations; 83 million passengers (2017)
	Fertagus	18 units; 1 route; 14 stations; 54km line extension; 70.000 daily moves (2019)

Buses are operated by an extensive list of companies, which are sectioned by regions. For example, *Vimeca* company operates on the west side of the city, *Transportes Sul do Tejo* (TST) company is responsible for the Setúbal peninsula, etc. Here, the focus was on ***Carris***, the main operating company in the Lisbon metropolitan area, comprehending 1350 drivers of the total 2112 workers, as well as 603 bus units and 48 trams, allocated to 79 different routes. Apart from buses, Carris also offers a tram service in specific points of the city. According to Carris (2017), in 2017, approximately 122,4 million passengers used this service, demonstrating the huge impact this company has in people's life.

Metropolitano de Lisboa is responsible for managing the Lisbon metrorail system, with the aim to provide a public passenger transport service, focused on the customer and promoting a sustainable mobility. The operating rolling-stock is formed by 333 units, while 1414 workers form the working crew. The subway system' line extension has approximately 44.5 km, sectioned in 4 different lines, comprehending 56 stations. It is reported that in 2017, the company served 162 million customers (Metro, 2017).

Transtejo-Soflusa company provides a public service river transport, extremely important since it provides a way to cross the *Tejo* river to a large number of commuters. To meet customers' demand, the company has a fleet composed by 28 ships and besides transporting people, several ferries are also provided so that vehicles can also be on board. Furthermore, there are 9 terminals, forming 5 distinct routes to support the passengers' movement across both banks of the river. (Transtejo-Soflusa, 2017)

Finally, essentially connecting the city centre with its outskirts, Lisbon railway network is operated by two different train operating companies, ***Comboios de Portugal (C.P.)*** and ***Fertagus.*** The first, a larger public company that runs regional and inter-regional trains, operates 4 different routes within Lisbon metropolitan area, comprehending 67 stations. During 2017, the fleet composed by 91 units served 83 million passengers (R&C CP, 2017). On the other hand, Fertagus is a private company that uses IP *(Infrasestruturas de Portugal)* rail tracks, responsible for operating a single line connecting *Roma-Areeiro* to *Setúbal*, which extends for 54 km and serves 14 stations. The 18 train units that form Fertagus' fleet, allow the company to be accountable for 70.000 daily moves (Fertagus, 2019). Figure 1.2 presents the whole Lisbon metropolitan area railway system in a diagram containing all the actual operating lines.

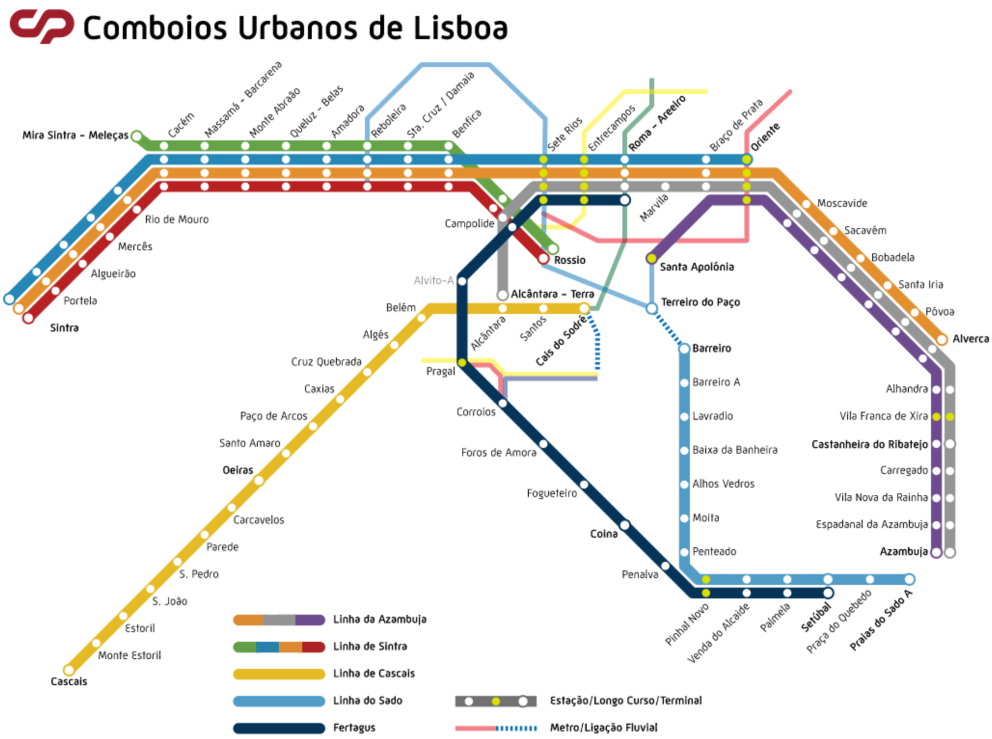


Figure 1.2 – Railway map of Lisbon metropolitan area (R.f.: CP website)

1.1.2 New technologies for the Railway Sector

In an era where technology evolves at a frenetic pace and consequently, new researches and approaches, arise to solve more complex problems, railway systems are not an exception. The *"rising traffic demand, congestion, security of energy supply and climate changes"* (EC, 2019) are some of the challenges that the European Union faces, so these new technologies can play a major role influencing the way future rail automation and maintenance are organised. Thus, an European rail initiative named Shift2Rail (S2R) was created in 2009, *"when key European rail* *sector players, under the coordination of the Association of the European Rail Industry (UNIFE), began investigating a policy instrument that could facilitate a step change for the European rail system."* (S2R, 2019). This initiative actually acts as a test bed for new technology developments, not only helping enhance the railway industry competitiveness edge, as helping railway systems to establish a new and broader part in transport markets. It is clear that to fulfil the defined EU transport policy and climate change goals, a big investment in rail research is necessary.

Nowadays, already under the *Horizon 2020,* a research and innovation program that promotes innovation, further researches have been conducted, for example, on arising and promising disruptive technologies such as Artificial Intelligence (A.I.) and its use for the railway sector *(S2R AI, 2019)*. The integration of A.I. as an extension in existing/future systems, can be regarded as an opportunity, making such systems more flexible, i.e. able to deal with real time problems, where conditions may change at a high pace. It is proposed to find ways to adapt existing Artificial Intelligence developed for other sectors, that may be easily used on the railway networks. There are several areas where A.I. may be applied, namely to crew scheduling management, where the automatization of crew scheduling and rostering can lead to a decrease of the operational costs. Furthermore, there are some expected impacts resulting from these A.I. related researches. Mainly, it is expected that railway systems' flexibility is improved, as well as a reduction of the complexity of the problem. A simplified supervision and a faster problem resolution would hopefully be obtained, leading to higher efficiency and therefore, enhanced performance. *(S2R AI, 2019)*

In line with these ideas, the current work aims to use such techniques to solve crew scheduling problem for the case study of a train operating company in Lisbon.

1.2 Problem Definition and Methodology

As stated, the emergence of new and impactful technologies may have a big impact on how the transportation systems are conceived and optimized. Therefore, in order to remain a viable and competitive solution, the railway sector must keep up with all these developments. One way to accomplish this is through carefully planned schedules, which must satisfy not only the customers, but also workers and company's requirements, ranging from rolling-stock timetables to crew scheduling, this is, both maintenance and driving crew scheduling.

The ultimate objective of this dissertation is to develop a model that minimizes the costs associated with crew scheduling for a railway company and that creates its daily planning for a given week. To achieve this goal, a company's rolling-stock timetable and previously scheduled maintenance activities must be considered. Thus, it is proposed to apply the decision model to the case study of Fertagus operating rolling-stock schedule, adapting a previously obtained maintenance scheduling plan from (Mira, 2018), that outputs a daily maintenance and driving crew scheduling, considering the maintenance actions that have to be performed for each day of the week with the lowest associated costs possible.

To achieve the proposed goal, the following steps were executed:

- Literature review on maintenance and crew scheduling, namely in transportation systems;

- Development and validation of a decision model based on an optimization model described in the literature review;

- Data collection and model implementation in a mixed-integer linear programming model;

- Analysis of results and discussion;

- Take conclusions, recognize limitations and define future work.

Based on this multi-step approach, both models (Mira, 2018) and (M. Pour *et al.*, 2018) from the studied articles were chosen to serve as starting points of the decision model to be created. Although the literature review was helpful to comprehend the area developments and some approaches to the operational crew scheduling problem, some modifications were necessary, as well as some contributions were added, namely, the inclusion of maintenance crew competences and driving crew scheduling. Besides this, some meetings and correspondence with the maintenance director and the maintenance supervisor of Fertagus made it possible to understand how the company works and its inherent operational constraints, as well as to collect all the data necessary to conceive the optimization model, and build the case study.

1.3 Document Structure

The present document is organized in seven chapters:

1. Introduction – In this first chapter, the research topic of this study is presented, as well as a brief contextualization. In a first instance, the Lisbon metropolitan area transportation system is briefly presented and described. Later, the emergence of new technologies for the railway sector, namely Artificial Intelligence, is mentioned. Afterwards, the research study is outlined, as the proposed approach and its methodology, and finally, the objectives of the research and the document's structure are presented.

2. State of the Art – The following chapter summarizes the most relevant papers analysed during this thesis. In a first instance, a brief introduction to studies and methods applied to maintenance scheduling is presented. Besides, work carried out on crew scheduling and on driver crew scheduling in several transportation companies is also reviewed. In the last section of this chapter, all the reviewed work is summarized in a table, providing in this way a wide overview on the work done up to today and on the research gaps and opportunities that need research.

3. A Mixed-Integer Linear Programming Model – In chapter 3, the decision model is presented and clarified in detail. The operational crew scheduling problem is defined and then all the data, decision variables, objective function and constraints associated with this study are explained.

4. Application – In the following chapter, an illustrative example is presented and explained in detail so that the mathematical model concepts presented in the previous chapter are easily understood. In addition, with this implementation it is expected to test and validate the model. In the first section, the implementation in FICO Xpress software is described, while the illustrative example and its parameters are specified in the second section. Finally, in the third section, the results of the model are discussed and explained.

5. Case Study – Fertagus – In this chapter, a brief introduction and description of Fertagus train operating company is initially presented. Further on, the model is applied to the case study, and the specific case study inputs are displayed and discussed.

6. Results and Discussion – In chapter 6, the results of crew scheduling for the Fertagus case study are presented and analysed. The whole problem was here divided in two models and ran separately, one for the maintenance crew scheduling and the other for the driving crew scheduling. This way results are also sectioned in two subchapters regarding the respective scheduling problems. It is also important to mention that both models were executed on the same computer, equipped with a memory of 8 GB (RAM), a 2.30 GHz processor and a 64-bit Operating System.

7. Conclusion – In this last final chapter the contributions of the study are provided, the limitations identified and some aspects which may be enhanced are pointed out and left for further research. It is always kept in mind the objective of clearly clarifying what was possible to achieve and what may be used as groundwork for further work.

2 State of the Art

The following chapter summarizes the most relevant papers analysed during this thesis. In a first instance, a brief introduction to studies and methods applied to maintenance scheduling is presented. Besides, work carried out on crew scheduling and on driver crew scheduling in several transportation companies is also reviewed. In the last section of this chapter, all the reviewed work is summarized in a table, providing in this way a wide overview on the work done up to today and on the research gaps and opportunities.

2.1 Maintenance scheduling in transportation companies

Haghani and Shafahi (2002) deals with the problem of scheduling bus maintenance. Its aim is to design a daily schedule that minimizes the number of unavailability hours for each vehicle. Hereby, it is desired that as many inspections as possible are carried out through idle time, this is when buses are out of service, and so maximize the usage of maintenance resources. Using an integer programming approach, the model outputs a maintenance schedule for each bus, in addition to the minimum number of maintenance lines that should be assigned for each type of action.

Bazargan (2015) develops a mathematical model to examine a flight school's aircraft dispatching strategy, compares it to the existing practices and eventually proposes improved strategies. It is desired to achieve a minimization of costs and/or an improvement of aircraft availability. Several models were presented and analysed and it is curious to note that the chosen strategy was the one which minimizes the number of maintenance actions and therefore increases the aircraft availability, even if the cost reduction was not so significant. The degree of difficulty concerning the implementation of each strategy was relevant as users would normally reject hard implementing approaches even if the results were interesting. Finally, it is also important to note that mathematical model has enough flexibility to be adapted to different instances or means of transport.

Méchain (2017) focuses on a problem of maintenance planning for a train operating company, Fertagus. A mixed integer linear programming (MILP) model is conceived, taking into account the many technical and infrastructure constraints regarding the company. The aim of this study is to develop a model that outputs a technical maintenance plan for a time horizon of 52 weeks, while minimizing the cost related with preventive maintenance. Besides this, the optimization model defines which maintenance actions need to be carried out each week, the maintenance line where the maintenance takes place and also the number of spare parts necessary to fulfil the technical plan.

Mira (2018) develops a mixed-integer linear programming decision model which provides a weekly, optimal and robust rolling-stock schedule, capable of including the maintenance actions, considering a previously scheduled preventive maintenance plan for each of the different weeks of the year. A robust model is meant to deal with disturbances and resist to limit delay propagation with relative ease. This model is validated for a small scale illustrative example and later applied to the Fertagus train operating company. However, the inclusion of maintenance tasks in the model lead to a significant increase of the number of decision variables, and thus, due to computational capacity limitations only a 3-day schedule was possible to execute instead of the weekly one. The results show that by rearranging the operating rolling-stock schedule, it is possible to reduce meaningfully the deadheading distance covered by train units. However, it is important to note that this reduction might not be feasible as it might collide with slots available for other train operators. Moreover, maintenance actions were successfully included and the 3-day schedule was obtained, indicating which units and when they should carry out maintenance, as well as the maintenance type to be performed. Finally, some analysis is conducted and it is possible to conclude that the solution obtained is not sensitive to variations on the weight of the different components of the objective function.

2.2 Maintenance crew scheduling in transportation companies

M. Pour *et al.* (2018) addresses an hybrid Constraint Programming/Mixed Integer Programming framework to solve a signalling maintenance crew scheduling problem for a section of the Danish railway system. This hybrid framework is split in two parts. First, in the construction phase, initial feasible solutions are obtained through a Constraint Programming (CP) model. After, in the improvement phase, CPLEX 12.4, a Mixed Integer Programming (MIP) solver, is used for further improvement of these initial solutions. Accordingly, this "hybridised framework is a contribution to the development of integration between MIP and CP, where CP greatly reduces the time required by the MIP to produce a solution" (M. Pour *et al.*, 2018). The model is based on the problem faced by Banedanmark's planning team, a company responsible for most of the railway infrastructure in Denmark, which provided the model formulation. The main aim of this research is to find feasible solutions for larger instances of the maintenance crew scheduling problem. While a general purpose MIP solver can only deal with a maximum period of two weeks due to an extensive number of real-life attributes and constraints, and a Constraint Optimisation Problem (COP) model does not get improved solutions, this hybrid framework is capable of generating good results for planning horizons up to eight weeks. Using this method, only feasible solutions previously obtained by CP will be later improved, reducing the time required by the improvement phase to produce an improved solution. The results of this hybrid framework are later presented and then compared with both the results of modelling the problem as a Constraint Optimisation Problem (COP), and the results of solving the MIP directly. To sum up, it was possible to verify that the proposed hybrid CP/MIP framework outputs better results than both solving the problem as a MIP problem directly and using COP to improve the initial solutions found by CP.

Martins (2018) proposes a mathematical model focused on the scheduling of preventive maintenance actions. It attempts to find a way to reduce the costs and time associated with maintenance, through an optimization of the bus availability, resources and infrastructures. For this a mixed-integer linear programming model is developed, validated using an illustrative example and then applied to a real scenario concerning a bus operating company, Carris. The fact that it integrates the scheduling of preventive maintenance actions with the maintenance teams of the bus company, constitutes a key contribution of this work, as a certain complexity is introduced since workers have different functions. After the results of this model are presented, some analysis and comparisons are made and it is interesting to conclude that an increase in the number of workers results in a more efficient occupation of maintenance lines; in a decrease of the number of working days leading to an increase of buses availability and so reducing the total preventive maintenance cost. The technical plan provided successfully covers all the maintenance actions while reducing the associated costs, the bus unavailability and last but not least, increasing the maintenance crew productivity.

Fuentes, *et al.* (2018) approaches a crew scheduling problem, regarding rapid transit networks, i.e. networks where distances are not that big but the service frequency is extremely high. Several methods are studied, naming branch-and-bound, heuristics and a fix-and-relax algorithm (FRA), and then applied to a portion of RENFE's train operating company's rapid transit network. The later showed to be the better option, obtaining the best performance in terms of computational time, the major concern since the problem deals with rapid transit networks. Using the fix-and-relax metaheuristic, near-optimal solutions are obtained and compared with commercial software, showing that computational times are reduced more than 94% while maintaining a good optimality gap. In this way, it is possible to show that this approach might be useful for the crew scheduling of rapid transit networks.

2.3 Driving crew scheduling in transportation companies

Valouxis and Housos (2002) present an approach for the combined bus and driver scheduling problem. Through a quick heuristic scheduling (QS) model, the problem is successfully solved, and in fact is viable for several bus companies in Greece. Furthermore, a column generation procedure (CGQS) is also presented, showing that it uses a Linear Programming solver and the QS process's solution. It could be concluded that the QS algorithms are essential to improve the performance of the CGQS algorithm. In fact, for most of the instances studied, the solutions found by the CGQS algorithm significantly improved the solution found by the QS algorithm.

Boyer *et al.* (2018) presented an integrated approach for the Flexible Vehicle and Crew Scheduling Problem (FVCSP), common in urban bus companies. In fact, the development of model integrating both Vehicle Scheduling and Crew Scheduling problems in a single approach is one of the main contributions of this work. The aim of this problem focus on minimizing the costs related with vehicles usage and drivers wage. Besides dealing with several constraints related to both drivers and vehicles, the model also has the flexibility required to deal with scenarios where the number of available vehicles/drivers can change daily. First, a mixed-integer linear programming model is proposed, and then a Variable Neighbourhood Search (VNS) metaheuristic approach, capable of solving larger instances, closer to real-life situations. Results showed that approaching the FVCSP with a commercial mixed-integer linear programming solver CPLEX 12.7, as expected, feasible solutions could only be returned, in reasonable time, for small instances. On the other hand, using the VNS approach, for much larger cases, good results were obtained in a practical time. It is finally suggested, for future work, the combined optimization of both the timetabling with the vehicle and crew scheduling problem.

Kang *et al.* (2019) developed three distinct Integer Linear Programming (ILP) models regarding public transport bus and driver scheduling problems with mealtime windows, for a unique bus route. Using CPLEX solver, the results obtained required an excessive computation time to reduce the optimality gap to at most 0,5%, so in order to improve computational efficiency, Kang *et al.* (2019) developed a valid inequality approach. Its computational efficiency is later validated using real world instances. Furthermore, it is also developed a self-adaptive search method to determine the upper and lower bounds of driver group as well as bus fleet sizes. Results obtained, compared with the single use of CPLEX, show that the valid inequality approach is able to solve larger-scale problems in a reasonable computational time. Since this problem has a single route system and a single depot, it is left for further research the analysis of a multi-route and/or multi-depot bus system. It is also proposed for future work, an integrated model of bus timetabling and driver scheduling.

2.4 Contribution of the Research

After reviewing what was considered to be the most relevant contributions on maintenance and crew scheduling for transportation companies, this section provides a summary of the contribution of each research, arranged on Table 2.1

Table 2.1 – Summary of the analysis of the papers on crew and maintenance scheduling

Authors (date)	Keywords	Proposed Technique	Contribution
Haghani and Shafahi (2002)	Bus transit, Heuristic algorithm, Optimization model, Scheduling	MILP	Daily inspection schedule minimizing the number of hours spent performing maintenance
Valouxis and Housos (2002)	Crew scheduling, Crew assignment, Column generation, Combinatorial optimization, Bus and Driver Scheduling	CGQS, LP, QS	Comparison between several approaches for a combined bus and driver scheduling problem
Bazargan (2015)	Mathematical model, Optimization, Integer Programming, Dispatching	IP	Aircraft dispatching strategies focused on cost minimization / aircraft availability maximization
Méchain (2017)	Maintenance optimization, Train operating companies, Mixed Integer Linear Programming.	MILP	Preventive maintenance weekly planning for 52 weeks under maintenance yard constraints
Mira (2018)	Railway Management, Rolling-Stock Planning, Maintenance Scheduling, Robustness, Optimization, Mixed Integer Linear Programming	MILP	Rolling-stock planning for 1 week, including maintenance actions from a technical plan
M. Pour et al. (2018**)**	Transportation Scheduling, Constraint Programming, Mixed Integer Programming, Hybrid Approaches	CP, MIP	Large scale maintenance crew scheduling
Martins (2018)	Bus transit, Maintenance Planning, Maintenance Scheduling, Optimization, Mixed Integer Linear Programming	MILP	Scheduling of preventive maintenance actions integrated with maintenance teams in a bus company
Boyer *et al.* (2018)	Crew Scheduling Problem, Vehicle Scheduling Problem, Mixed-Integer Problem, Variable Neighbourhood Search Algorithm,	MILP, Variable Neighbourhood Search Algorithm	Flexible vehicle and crew scheduling for large instances
Fuentes, *et al.* (2018)	Crew Scheduling, Fix & Relax, Matheuristic, Rapid Transit Networks, Computational time	ILP, FRA	Crew scheduling for rapid transit networks
Kang *et al.* (2019)	Bus & driver scheduling, Integer linear programming, Mealtime windows, Valid inequality approach	ILP, Valid Inequality Approach	Bus & driver scheduling with mealtime windows for a single public transport bus route

2.5 Research Gaps and Opportunities

Although the initial model (M. Pour *et al.*, 2018) on the preventive signalling maintenance crew scheduling for a Danish railway company, was considered appropriate, the final version of this model was completely modified and adapted to meet Fertagus' case study requirements, as well as to fit information extracted from (Mira, 2018) model. While in M. Pour *et al.* (2018) model the crew members dislocate to different technical places, in the present study there is a single technical place, the depot, and therefore there are no trips associated with the maintenance crew. The main focus concerns with the idea of defining a skillset for each maintenance worker. Regarding crew members' competences, in M. Pour *et al.* (2018) model, each task requires at most one competence, while in this research, each task may require more than one competency. Furthermore, in this case, maintenance tasks do not last longer than a shift, so there is only one set of crew members assigned to each task, carrying it out throughout all its duration. In later stages of this research, it was also decided to implement the driving crew scheduling and therefore to conceive a model capable of outputting the complete crew scheduling for a train operating company. After reviewing all the work done on these areas, some of the information was selected and gathered to be used in the current dissertation, fulfilling some of the opportunities presented by them.

3 A Mixed-Integer Linear Programming Model

In chapter 3, the decision model is presented and explored in detail. The operational crew scheduling problem is defined, and all the data, decision variables, objective function and constraints associated with this study are explored.

3.1 A Mixed-Integer Linear Programming Model

The present mathematical model is an adaptation of the model presented by (M. Pour *et al.*, 2018) on the preventive signaling maintenance crew scheduling problem. However, to comprehend the Fertagus case study and to integrate the information associated with the maintenance model extracted from (Mira, 2018), the present model had to be basically built from scratch. Furthermore, it also contemplates an integrated driving crew scheduling approach. This model is intended to be initially validated using a small-size illustrative example, and later applied to a real case scenario, aiming to reduce the cost associated with both drivers and maintenance crew scheduling.

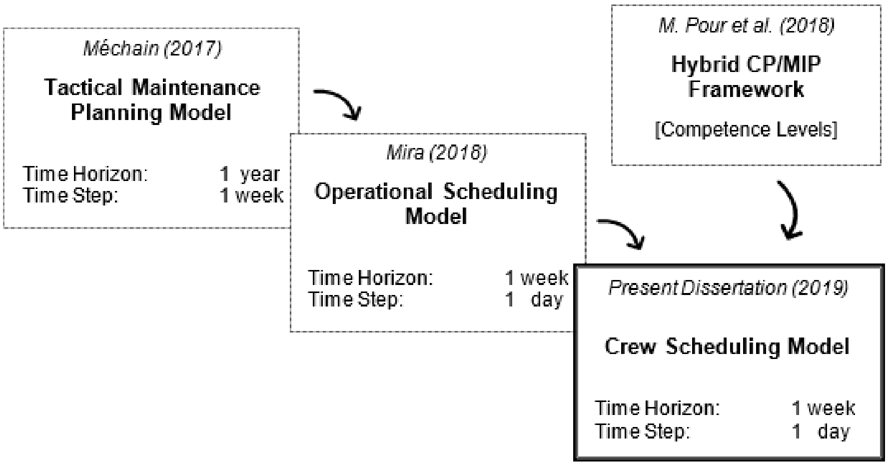


Figure 3.1 – Present dissertation relation's with previous researches

In fact, this study is in a certain way a continuation of the researches carried out by Méchain, (2017) and Mira, (2018), as Figure 3.1 suggests. The first outputs a preventive maintenance planning for all the 52 weeks of a given year. Some data is then extracted from this first model, to be used by Mira, (2018) as inputs of his own model, which obtains an operational scheduling plan, including an integrated rolling-stock and preventive maintenance scheduling. It is also important to note that both of these researches analysed the same case study – Fertagus train operating company, suggesting the existing relation between them. While the results obtained by Mira, (2018) integrate a new maintenance schedule with a new rolling-stock schedule, the current research uses the actual Fertagus company operating rolling stock schedule, as well as an adapted schedule of preventive maintenance actions for a given week. It had to be secured that both schedules were compatible, i.e. maintenance actions performed for the right units, within the right time intervals established by the rolling stock timetable. In this way, maintenance crew members can be assigned to the previously scheduled maintenance actions, obtaining a daily maintenance crew schedule. It is also important to mention that the workers' competences/skills were added to the problem, meaning that a maintenance worker requires the right competence for a given maintenance action. Additionally, a daily driving crew schedule is obtained based on the current Fertagus rolling stock timetable. Driving crew members will be assigned to the scheduled tasks so that all units are carried in time, while optimizing the costs related to employing drivers, i.e. the number of driving workers required for a given day should be minimized. All of the constraints associated with infrastructure, crew, rolling stock and other temporal and logical constraints are considered and respected while defining the model, as well as the aim to meet the company's objective of providing both the crew and the passengers a better service. While from Méchain, (2017) and Mira, (2018), some results were analysed and adapted to be used as inputs in the present study; whereas from M. Pour *et al.*, (2018), some constraints were adapted and implemented, namely the competences related constraints.

The final model is a Mixed-Integer Linear Programming (MILP) model. It is a decision model capable of using the weekly preventive maintenance planning and the rolling-stock, to build a daily crew schedule for a train operating company (Fertagus) and assign workers to daily tasks.

This final optimization model can be decomposed in two models: i) one related to the maintenance crew and ii) the other related to the driving crew. They use several different decision variables in order to reduce the associated operational costs, and indicating when and which tasks/actions are carried out by the crews. Further on, these decision variables are presented and explained. The daily costs related to both maintenance and driving workers was provided by Fertagus.

The next sections fully present and describe the problem in detail as well as all the necessary data, decision variables, objective function and constraints.

3.2 The Crew Scheduling Problem Definition

In the following subsubsections, a brief explanation on the crew scheduling problem is presented, detailing some of the most relevant aspects necessary to understand how the problem was approached. The crew scheduling problem was sectioned in two parts: i) the maintenance crew scheduling problem and then ii) the driving crew one.

3.2.1 Maintenance Crew Scheduling Problem Definition

A **maintenance action *m*** is carried out on a unit *k* and it takes place during a maintenance slot existing between two tasks (*i,j*). It has an associated **duration** required to be carried out, $\boldsymbol{MT_m}$ and to cover each action, one or more **maintenance workers mw**, may be assigned, depending on the total amount of work required to perform each action $\boldsymbol{AWt_m}$. This relation was previously defined by Mira (2018) as it follows: $MT_m = \frac{AWt_m}{nº\ workers}$. It is important to mention that in the present study, the amount of work is defined for each competence, $\boldsymbol{AW_{m,c}}$, which may be required to carry out a maintenance task.

Logically, to execute a maintenance task, workers have to master at least one required **competence *c***, otherwise they are not able to perform it. Besides this, maintenance crew is characterized by the starting and finishing times $\boldsymbol{t1m_{mw,k,m,d}}$, $\boldsymbol{t2m_{mw,k,m,d}}$, respectively, respecting the established maintenance action duration. Once unit k arrives or leaves, $\boldsymbol{Da_i}$ or $\boldsymbol{Dd_j}$, the depot, a **set up time**, $\boldsymbol{t_{man}}$, is required before and after performing a maintenance action. Moreover, when considering real life scenarios in a train operating company, different units can perform maintenance tasks simultaneously while **maintenance workers can only perform one action at a time**, and so, more complex instances happen and more difficult it becomes to schedule and assign the crew members. One relevant scenario has to do with the crew members scheduling when actions are carried out in different units. In this scenario, maintenance crew members need **time to move from one unit to another**, $\boldsymbol{t_{min}}$. Later in this chapter, these scenarios will be presented and explained in detail. Here, in Figure 3.2 a simple example of a maintenance crew scheduling problem is displayed next.

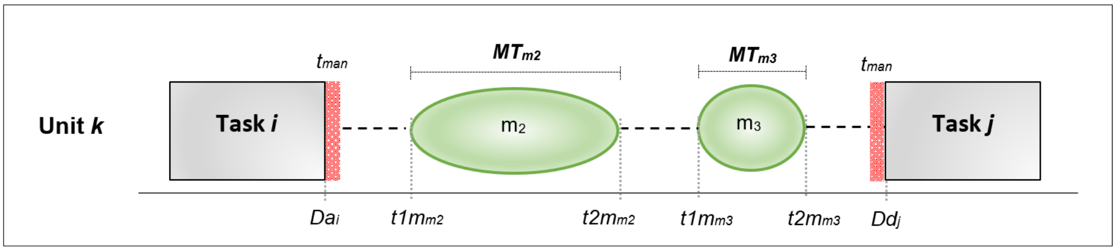


Figure 3.2 – Example of maintenance crew scheduling problem with 2 maintenances actions scheduled to be carried out between i and j.

3.2.2 Driving Crew Scheduling Problem Definition

Mira (2018) defined a **task** $\boldsymbol{T_i}$, *"as a non-splittable trip to be realized between one **departure station** $\boldsymbol{Sd_i}$ and one **arrival station** $\boldsymbol{Sa_i}$."* Additionally, tasks are also characterized by a **departure** and an **arrival time**, $\boldsymbol{Dd_i}$ and $\boldsymbol{Da_i}$ respectively. Figure 3.3 presents a simple task example.

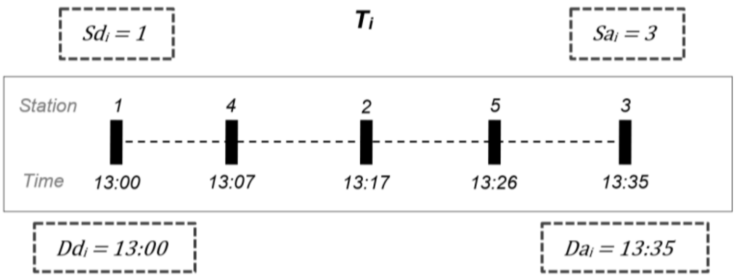


Figure 3.3 – Example of a task i, with all the respective information.

The list of all tasks that must be performed is then arranged such that tasks can be linked. From the previous researches it was established that two successive tasks, T_i and T_j can only be linked, if the arrival station of the first task matches the departure station of the latter. This arranged timetable comprehending all tasks forms the rolling-stock-schedule, which serves as input for the current research in order to obtain the driving crew scheduling.

Drivers are characterized by their **entry**, ***$eD_{dw,i}$*** **and exit, *$sD_{dw,i}$***, related with the first and last tasks carried out by them. However, for scheduling purposes, **entry and exit times, *$t1i_{dw,i}$* and *$t2i_{dw,i}$***, are also defined. There is also a gap, ***Δ_m***, between the drivers' entry time and the actual start of their first task, the same way there is for their exit time and the actual end of their last task, necessary to initiate and finalize the workday. It is defined **that each task can only be carried out by one driving worker, *dw***, whereas the same driver may be assigned to several tasks, provided that the maximum daily working time, ***TMS***, is not exceeded. Figure 3.4 helps understanding the previous concepts.

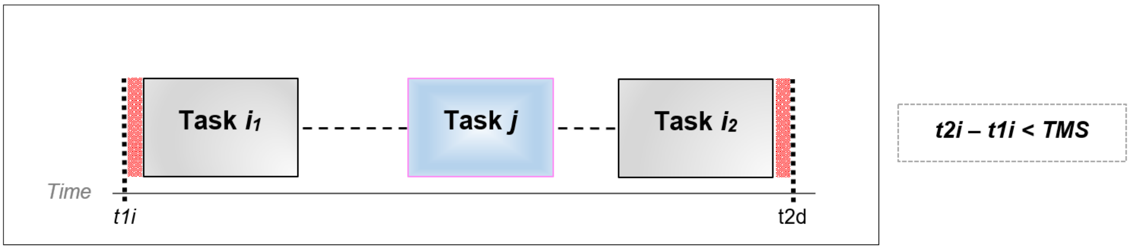


Figure 3.4 – Illustrative driving worker schedule, entering to service before task i_1 and exiting after i_2

Situations where drivers change unit can happen, although for the current study some temporal and infrastructure logical constraints are not fully considered and in fact, they are left for further research. Consequently, it is intended to minimize this number of changes, so that drivers rather perform successive tasks on the same unit *k*.

3.3 Indexes

k	train unit
s	station
i	task
j	task
m	maintenance action
c	competence
d	day
mw	maintenance worker
dw	driving worker

3.4 Sets

K	set of train units *k*
S	set of stations *s*
T	set of tasks *i*
MM	set of maintenance actions *m*
CC	set of competences *c*
D	set of days *d*
MW	set of maintenance workers *mw*
DW	set of driving workers *dw*

3.5 Parameters

Cmw_{mw}	daily cost of each maintenance worker *mw*
Cdw_{dw}	daily cost of each driving worker *dw*
Sd_i	departure station of task *i*
Sa_i	arrival station of task *i*
Dd_i	departure time of task *i*
Da_i	arrival time of task *i*
MT_m	duration of maintenance action *m* (minutes)
AWt_m	total amount of work required for each maintenance *m* (minutes)
$AW_{m,c}$	amount of work *per* competence *c*, required for each maintenance *m* (minutes)
$X_{k,i}$	tasks *i,* carried out on unit *k*
$Y_{k,i,j}$	pair of tasks *(i,j)* linked by unit *k*
$YM_{k,i,j,m}$	maintenance actions *m* performed on unit k, between pair of tasks *(i,j)*
$KM_{k,m}$	maintenance actions *m* that need to be performed on each unit *k*
$ZM_{k,d}$	units *k* that cover any maintenance action, on a given day *d*
$MWC_{mw,c}$	competences *c* mastered by each maintenance worker *mw*

3.6 Constants

NU	total number of units
NS	total number of stations

NMW	total number of maintenance workers
NCC	total number of competences
NDW	total number of driving workers
NM	total number of different kinds of maintenance actions
NT	total number of tasks to cover
ND	number of days
TMS	daily maximum service time for driving workers (minutes)
Δ_m	gap between drivers' entry/exit hour and beginning of the first/end of the last task, respectively (minutes)
t_{min}	gap required by maintenance workers when changing unit in successive maintenance actions (minutes)
t_{man}	gap required by units to set up for maintenance after arriving and before departing the depot (minutes)
LN	large number

3.7 Pre-Processing Data

While conceiving this MILP model some new data had to be created and implemented mainly due to their constraints. This pre-processing data is here presented and explained, in order to be understood and comprehended within the whole context of the problem. Later some of this data will be used in some constraints regarding drivers' availability.

Nk_i number of units *k* used to perform task *i*

$$ft_i = \begin{cases} 1, \text{ if tasks } i \text{ is the earliest task of unit} \\ 0, \text{ otherwise} \end{cases}$$

$$lt_i = \begin{cases} 1, \text{ if tasks } i \text{ is the latest task of unit} \\ 0, \text{ otherwise} \end{cases}$$

$$Li_{i,j} = \begin{cases} 1, \text{ if pair of tasks } (i,j) \text{ are linked} \\ 0, \text{ otherwise} \end{cases}$$

As presented above, the number of units *k* used to perform a task i, Nk_i, may change from task to task. From Mira, (2018) model we know that **each task can be performed by a minimum of one unit and a maximum of two units *k***. Therefore, if a pair of tasks *(i,j)* is successfully linked, meanwhile the number of units *k* used to perform task *i,* Nk_i, differs from the number of units *k* used to perform task *j,* Nk_j, then one of two scenarios occur:

(1) $Des_{i,j} = \begin{cases} 1, \text{ if two units performing task } i, \text{ were decoupled between task } i \text{ and task } j \\ 0, \text{ otherwise} \end{cases}$

(2) $Aco_{i,j} = \begin{cases} 1, \text{ if two units performing task } j, \text{ were coupled between tasks } i \text{ and } j \\ 0, \text{ otherwise} \end{cases}$

In the first scenario, when $Nk_i > Nk_j$, task *i* is performed by two units, while the successive task *j,* is performed by only 1 unit. This means that a decoupling has to be carried out between these two tasks. Figure 3.5 below sums up this scenario.

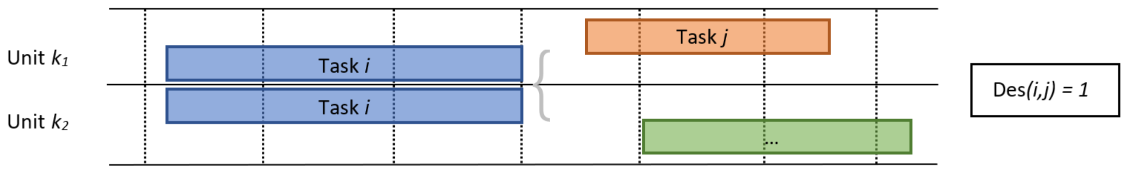


Figure 3.5 – Example of a decoupling between tasks i and j

In the second scenario, when $Nk_i < Nk_j$, task *i* is completed by one unit, while the next task *j* requires two units to be performed. Thus, two units need to be coupled, so that task *j* can be carried out. Figure 3.6 represents this second scenario.

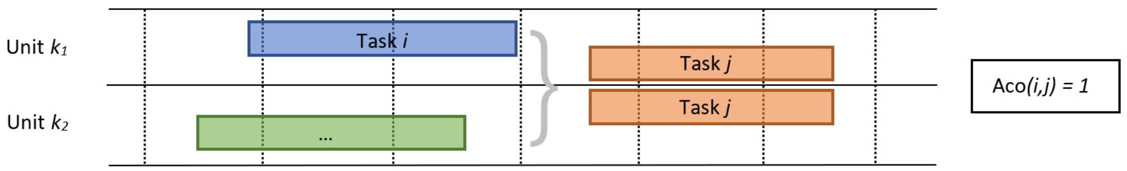


Figure 3.6 – Example of a coupling between tasks i and j

Finally, due to the high number of tasks and units associated with a railway company, many tasks can and in fact, are performed simultaneously:

$$Sim_{i,j} = \begin{cases} 1, \text{ if task } i \text{ and task } j \text{ are preformed simultaneously} \\ 0, \text{ otherwise} \end{cases}$$

Hereby, we define that if the departure time of task *j,* is comprehended between the departure and arrival time of task *I, i.e.* $\mathbf{Dd_i \geq Dd_j \geq Da_i}$**,** then we can assume that these tasks are performed simultaneously at least at one moment in time. Figure 3.7 is a schematic representation of such circumstance.

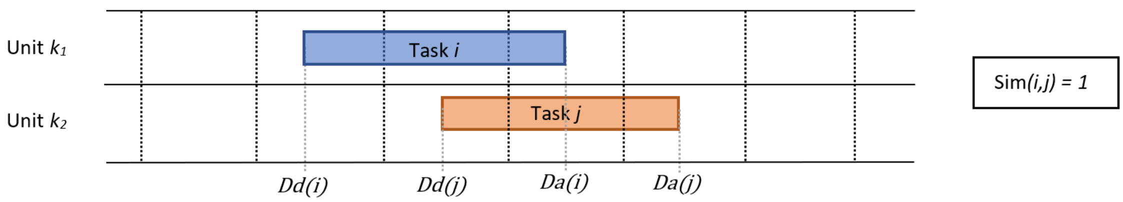


Figure 3.7 – Example of two simultaneous tasks i and j

3.8 Decision Variables

This section identifies the decision variables in the MILP formulation.

$$wMd_{mw,d} = \begin{cases} 1 \text{ if maintenance worker } mw\text{, works on day } d \\ 0 \text{ otherwise} \end{cases}$$

$$wM_{mw,k,m,d} = \begin{cases} 1 \text{ if maintenance worker } mw\text{, performs maintenance action } m \text{ on unit } k\text{, on day } d \\ 0 \text{ otherwise} \end{cases}$$

$t1m_{mw,k,m,d}$ maintenance worker *mw* **starting time**, performing maintenance action *m,* on unit *k*, on day *d*

$t2m_{mw,k,m,d}$ maintenance worker *mw* **ending time**, performing maintenance action *m,* on unit *k*, on day *d*

$$wDd_{dw} = \begin{cases} 1 \text{ if driving worker } dw\text{, works this day} \\ 0 \text{ otherwise} \end{cases}$$

$$wDt_{dw,i} = \begin{cases} 1 \text{ if driving worker } dw \text{ performs task } i \\ 0 \text{ otherwise} \end{cases}$$

$$eD_{dw,i} = \begin{cases} 1 \text{ if driving worker } dw \textbf{ enters} \text{ service on task } i \\ 0 \text{ otherwise} \end{cases}$$

$$sD_{dw,i} = \begin{cases} 1 \text{ if driving worker } dw \textbf{ exits} \text{ service on task } i \\ 0 \text{ otherwise} \end{cases}$$

Two additional variables that are linear dependent on other decision variables were also defined with associated linear constraints defined later in section 3.10:

$t1i_{dw,i}$ driving worker *dw,* **entry hour** for task *i*

$t2i_{dw,i}$ driving worker *dw,* **exit hour** for task *i*

3.9 Objective Function

This model aims to minimize the operational costs related to both maintenance crew and drivers scheduling, therefore, the objective function is sectioned in two terms, the first one addressing the maintenance crew costs and the second one related to the driving crew costs. To accomplish this, the model tries to minimize the number of workers of both crews, working on a given day *d*, as well as the cost of the assigned worker. For the maintenance crew, this value depends on the worker, however for the driving crew all members have the same daily cost.

Minimize: $\sum_{mw \epsilon MW} \sum_{d \epsilon D} \mathrm{Cm}w_{\mathrm{mw}} \,.\, w\mathrm{M}d_{\mathrm{mw,d}} + \sum_{\mathrm{dw} \epsilon DW} \mathrm{Cd}w_{\mathrm{dw}} \,.\, w\mathrm{D}d_{\mathrm{dw}}$

3.10 Constraints

In order to implement the necessary specifications and requirements of the problem, the objective function must respect several constraints. These constraints are grouped in two parts: maintenance crew constraints and driving crew constraints.

3.10.1 Maintenance Crew Constraints

$$wM_{mw,k,m,d} = 0 \qquad \forall\ mw \in MW, k \in K, m \in M, d \in D \mid KM_{k,m} = 0 \qquad (1)$$

$$wM_{mw,k,m,d} = 0 \qquad \forall\ mw \in MW, k \in K, m \in M, d \in D \mid ZM_{k,m} = 0 \qquad (2)$$

$$wM_{mw,k,m,d} \leq wMd_{mw,d} \qquad \forall\, mw \in MW, k \in K, m \in M, d \in D \qquad (3)$$

$$t1m_{mw,k,m,d} + t2m_{mw,k,m,d} \leq LN \times wM_{mw,k,m,d} \qquad \forall\, mw \in MW, k \in K, m \in M, d \in D \qquad (4)$$

$$t2m_{mw,k,m,d} \geq t1m_{mw,k,m,d} \qquad \forall\, mw \in MW, k \in K, m \in M, d \in D \qquad (5)$$

$$t1m_{mw,k,m,d} \geq (Da_i + t_{man}) \times wM_{mw,k,m,d} \qquad \forall\, i \in T, j \in T, mw \in MW, k \in K, m \in M, d \in D \mid YM_{k,i,j,m} = 1 \qquad (6.1)$$

$$t2m_{mw,k,m,d} \leq \left(Dd_j - t_{man}\right) \times wM_{mw,k,m,d} \qquad \forall\, i \in T, j \in T, mw \in MW, k \in K, m \in M, d \in D \mid YM_{k,i,j,m} = 1 \qquad (6.2)$$

$$t2m_{mw,k,m,d} \geq t1m_{mw,k,m,d} + MT_m - \left(1 - wM_{mw,k,m,d}\right) \times LN \qquad \forall\, i \in T, j \in T, mw \in MW, k \in K, m \in M, d \in D \mid YM_{k,i,j,m} = 1 \qquad (7)$$

$$t2m_{mw,k,m,d} - t1m_{mw,k,m,d} \leq MT_m \times wM_{mw,k,m,d} \qquad \forall\, i \in T, j \in T, mw \in MW, k \in K, m \in M, d \in D \mid YM_{k,i,j,m} = 1 \qquad (8)$$

$$t1m_{mw,k,m2,d} \geq t2m_{mw,k,m1,d} - LN \times \left(1 - wM_{mw,k,m2,d}\right) \qquad \forall\, i \in T, j \in T, mw \in MW, k \in K, m \in M, d \in D \mid YM_{k,i,j,m1} = 1 \wedge YM_{k,i,j,m2} = 1 \wedge m_1 \neq m_2 \qquad (9)$$

$$t1m_{mw,k2,m,d} \geq t2m_{mw,k1,m,d} + t_{min} - LN \times \left(1 - wM_{mw,k2,m,d}\right) \qquad \forall\, mw \in MW, k_1 \in K, k_2 \in K, m \in M, d \in D \mid KM_{k1,m} = 1 \wedge KM_{k2,m} = 1 \wedge k_1 \neq k_2 \qquad (10.1)$$

$$t1m_{mw,k2,m2,d} \geq t2m_{mw,k1,m1,d} + t_{min} - LN \times \left(1 - wM_{mw,k2,m2,d}\right) \quad \forall\, mw \in MW, k_1 \in K, k_2 \in K, m_1 \in M, m_2 \in M, d \in D \mid KM_{k1,m1} = 1 \wedge KM_{k2,m2} = 1 \wedge k_1 \neq k_2 \wedge m_1 \neq m_2 \qquad (10.2)$$

$$AW_{m,c} \leq \sum_{mw \in MW \mid MWC_{mw,c}=1} \left(t2m_{mw,k,m,d} - t1m_{mw,k,m,d}\right) \quad \forall\, i \in T, j \in T, k \in Km \in M, c \in CC, d \in D \mid YM_{k,i,j,m} = 1 \qquad (11)$$

Constraints (1) and (2) guarantee that if a unit *k* does not go to depot to perform maintenance actions on day *d,* i.e. if $ZM_{k,d} = 0$, then no maintenance worker *mw*, will be performing any maintenance on that unit *k*. Furthermore, if a maintenance action *m*, is not previously scheduled to be performed on a unit *k,* i.e. if $KM_{k,m} = 0$, then no maintenance worker *mw*, will be assigned to perform it. Constraint (3) establishes that if a maintenance worker performs any maintenance action on a day *d,* then he/she is assigned to work on that day.

Constraint (4) secures that if a maintenance worker *mw* is assigned, then its starting/finishing time must be greater than zero, meanwhile constraint (5) assures that the finishing time of a maintenance action, logically must be greater than its starting time.

Constraint (6.1) and (6.2) assure that maintenance tasks are performed within the right time gap. More precisely, constraint (6.1) guarantees that maintenance actions performed between pair of tasks *(i,j)* can only start after the arrival time of the unit Da_i to the depot, plus a gap required by units to set up for maintenance t_{man}. On the other hand, constraint (6.2) assures that maintenance actions end before the departure time of the unit Dd_j from the depot, less t_{man}. Constraint (7), guarantees that, if $wM_{mw,k,m,d} = 1$, i.e. if a maintenance worker *mw* is assigned to a maintenance action *m*, to be performed on unit *k*, on day *d*, then, the finishing hour of a maintenance action m, $t2m_{mw,k,m,d}$, must be greater than the starting hour $t1m_{mw,k,m,d}$, plus the duration necessary to carry out the maintenance action, MT_m. Additionally, constraint (8) ensures that the previously defined duration of a maintenance action MT_m, cannot be exceeded. These four constraints are respected in an illustrative example presented in Figure 3.8.

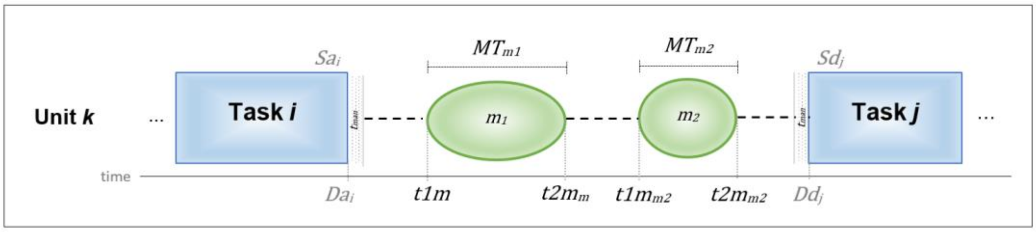


Figure 3.8 – Diagram concerning constraints (6.1), (6.2), (7) and (8)

The following constraints (9), (10.1) and (10.2) assure that if a maintenance worker is assigned to two different maintenance actions m_1 and m_2, he/she can only start another maintenance action m_2, after finishing the one that was started first. While constraint (9) guarantees that this happens for two maintenance actions performed on the same unit *k*, constraints (10.1) and (10.2) impose that this temporal coherence is established for two different units k_1 and k_2, whether the maintenance actions to be performed are the same or not. The following Figures (3.9), (3.10) and (3.11) aim to explain these constraints through some representative schemes.

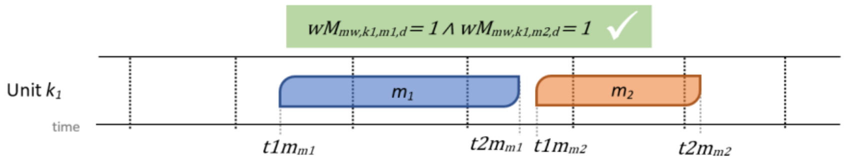


Figure 3.9 – Example of two consecutive maintenance actions, successfully performed by the same worker

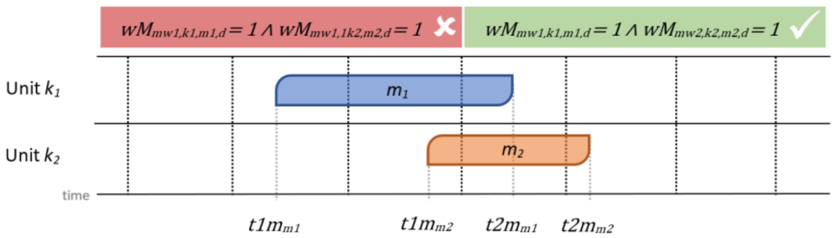


Figure 3.10 – Example of two simultaneous maintenance actions, carried out on different units by two different workers

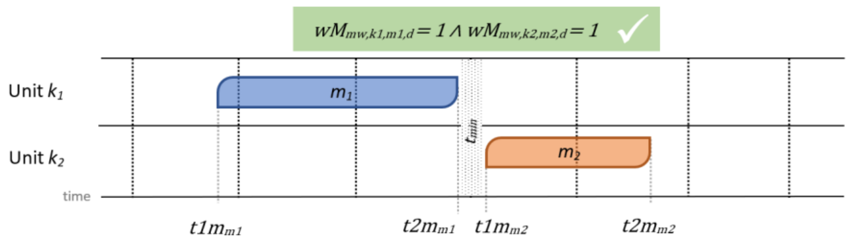


Figure 3.11 – Example of two consecutive maintenance actions, carried out on different units, successfully performed by a single maintenance worker

Finally, constraint (11) states that the amount of work per competence required by each one of the maintenance actions, $AW_{m,c}$, must be satisfied by the maintenance workers assigned to the respective maintenance action. It is also important to note that a maintenance worker *mw,* can only be assigned to a maintenance action *m*, in case he/she possesses at least one of the required competences to carry it out, i.e. $MWC_{mw,c} = 1$. Therefore, this competence related constraint imposes that the sum of the working times for all workers *mw*, assigned to a maintenance action *m*, must satisfy the required $AW_{m,c}$, for that maintenance action *m.*

3.10.2 **Driving Crew Constraints**

$$\sum_{i \in T} eD_{dw,i} \leq 1 \qquad \forall\, dw \in DW \qquad (12.1)$$

$$\sum_{i \in T} sD_{dw,i} \leq 1 \qquad \forall\, dw \in DW \qquad (12.2)$$

$$\sum_{j \in T} sD_{dw,j} = \sum_{i \in T} eD_{dw,i} \qquad \forall\, dw \in DW \qquad (12.3)$$

$$eD_{dw,i} \leq wDt_{dw,i} \qquad \forall\ dw \in DW, i \in T \qquad (13.1)$$

$$sD_{dw,i} \leq wDt_{dw,i} \qquad \forall\ dw \in DW, i \in T \qquad (13.2)$$

$$t1i_{dw,i} = (Dd_i - \Delta_m) \times eD_{dw,i} \qquad \forall\ dw \in DW, i \in T \qquad (14.1)$$

$$t2i_{dw,i} = (Dd_i + \Delta_m) \times sD_{dw,i} \qquad \forall\ dw \in DW, i \in T \qquad (14.2)$$

$$\sum_{dw \in DW} wDt_{dw,i} = 1 \qquad \forall\, i \in T \qquad (15)$$

$$wDt_{dw,i} \leq wDd_{dw} \qquad \forall\ dw \in DW, i \in T \qquad (16)$$

$$\sum_{dw \in DW} eD_{dw,i} = 1 \qquad \forall\, i \in T \mid ft_i = 1 \qquad (17.1)$$

$$eD_{dw,i} = wDt_{dw,i} \qquad \forall\, dw \in DW, i \in T \mid ft_i = 1 \qquad (17.2)$$

$$\sum_{dw \in DW} sD_{dw,i} = 1 \qquad \forall\, i \in T \mid lt_i = 1 \qquad (18.1)$$

$$sD_{dw,i} = wDt_{dw,i} \qquad \forall\, dw \in DW, i \in T \mid lt_i = 1 \qquad (18.2)$$

$$LN \times (1 - eD_{dw,i}) \geq \sum_{j \in T \mid Dd_j \leq Da_i \wedge i \neq j} wDt_{dw,i} \qquad \forall\, dw \in DW, i \in T \qquad (19.1)$$

$$LN \times (1 - sD_{dw,i}) \geq \sum_{j \in T \mid Dd_j \geq Da_i \wedge i \neq j} wDt_{dw,i} \qquad \forall\, dw \in DW, i \in T \qquad (19.2)$$

$$t2i_{dw,j} - t1i_{dw,i} \leq TMS + LN \times \left(2 - eD_{dw,i} - sD_{dw,j}\right) \qquad \forall\, dw \in DW, i \in T, j \in T \qquad (20)$$

$$wDt_{dw,i} + wDt_{dw,j} \leq 1 \qquad \forall\, dw \in DW, i \in T, j \in T \mid Sim_{i,j} = 1 \wedge i \neq j \qquad (21)$$

$$wDt_{dw,i} \leq LN \times \sum_{j \in T \mid X_{k,j}=1} eD_{dw,j} \qquad \forall\, dw \in DW, k \in K, i \in T \mid X_{k,i} = 1 \wedge Nk_i = 1 \qquad (22.1)$$

$$wDt_{dw,i} \leq LN \times \sum_{j \in T \mid X_{k,j}=1} sD_{dw,j} \qquad \forall\, dw \in DW, k \in K, i \in T \mid X_{k,i} = 1 \wedge Nk_i = 1 \qquad (22.2)$$

$$wDt_{dw,i} \leq LN \times \sum_{j \in T \mid X_{k_1,j}=1 \vee X_{k_2,j}=1} eD_{dw,j} \qquad \forall\, dw \in DW, k_1 \in K, k_2 \in K, i \in T \mid X_{k_1,i} = 1 \wedge X_{k_2,i} = 1 \wedge k_1 \neq k_2 \wedge Nk_i = 2 \qquad (23.1)$$

$$wDt_{dw,i} \leq LN \times \sum_{j \in T \mid X_{k_1,j}=1 \vee X_{k_2,j}=1} sD_{dw,j} \qquad \forall\, dw \in DW, k_1 \in K, k_2 \in K, i \in T \mid X_{k_1,i} = 1 \wedge X_{k_2,i} = 1 \wedge k_1 \neq k_2 \wedge Nk_i = 2 \qquad (23.2)$$

Constraints (12.1) and (12.2) assure that each driving worker *dw,* can only enter and exit to service once, this is one entry task and one exit task. Additionally, constraints (12.3) guarantee that if *dw* enters to service, then, logically he/she must also exit. Constraints (13.1) impose that each driving worker *dw*, must perform the task *i,* in which his/her shift starts, i.e. if he/she enters to service on that same task *i*. In line with (13.1), constraint (13.2) assures the same happens regarding the exit task. Constraints (14.1) and (14.2) define the entry and exit hour for each driver

$t1i_{dw,i}$, $t2i_{dw,i}$, respectively, keeping in mind that the set up time required by drivers before starting his/her first task and after finishing his/her last task is Δ_m. (Figure 3.12)

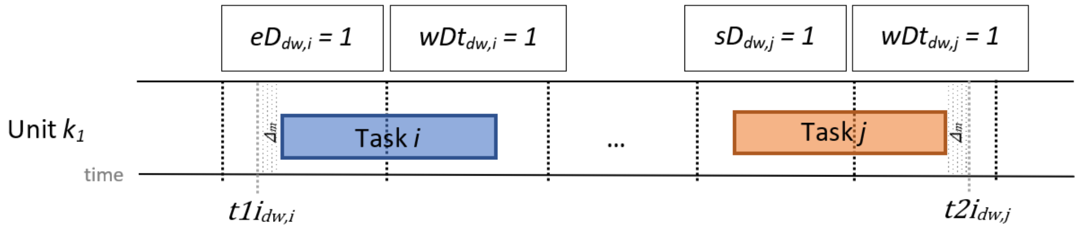


Figure 3.12 – Diagram concerning constraints 12.1 to 14.2

Constraints (15) establish that every task must be carried by only one driving worker, additionally, constraints (16) states that if a driver *dw* is assigned to any task, logically he/she works on that day.

Constraints (17.1) and (17.2) have to do with first tasks, ft_i. First, constraints (17.1) assure that every first task requires a driver *dw* to enter service, while constraints (17.2) assure that the driving worker who entered to service on that task i, also carries it out. Similarly, constraints (18.1) and (18.2) express parallel ideas regarding last tasks, i.e. every last task lt_i, implies that the driver *dw* who performed it, exits service after that task.

Constraints (19.1) assure that a driver *dw* cannot carry out any task *j* that occurs before his entry task *i*, while constraints (19.2) guarantee that no task *j* can be performed after the driver's exit task *i*. Constraints (20) secure that each driver *dw* does not work longer than the established maximum service time *TMS* (540 minutes).

Constraints (21) state that two different tasks performed simultaneously (Figure 3.7) require two driving workers to carry them out.

Finally, for single unit tasks, i.e. tasks with $Nk_i = 1$, constraints (22.1) assure that all tasks that require a train unit *k,* can only be carried out by a driving worker *dw,* that has entered to service on that same unit *k*, i.e. his first task has to be done on the same unit. Similarly, constraints (22.2) assure that a driver *dw* must perform tasks carried out on the same unit *k*, that he/she is finishing service, i.e. his/her last task. Analogously, constraints (23.1) and (23.2) guarantee that this also occurs for tasks that require two units k_1 and k_2, i.e. tasks with $Nk_i = 2$. These tasks can only be carried out by a driver *dw* who enters and exits service on one of those train units k_1 or k_2.

These last four constraints altogether assure that drivers do not change unit which is in accordance with what was mentioned above in subsection 3.2.2. In fact, changes of units are not considered in this model, and are left for further research, in adapting such model to this possibility.

3.10.3 **Decision Variables Constraints**

$$wMd_{mw,d} \in \{0,1\} \qquad \forall\, mw \in MW, d \in D \tag{24}$$

$$wM_{mw,k,m,d} \in \{0,1\} \qquad \forall\, mw \in MW, k \in K, m \in MM, d \in D \tag{25}$$

$$t1m_{mw,k,m,d} \geq 0 \qquad \forall\, mw \in MW, k \in K, m \in MM, d \in D \tag{26}$$

$$t2m_{mw,k,m,d} \geq 0 \qquad \forall\, mw \in MW, k \in K, m \in MM, d \in D \tag{27}$$

$$wDd_{dw} \in \{0,1\} \qquad \forall\, dw \in DW \tag{28}$$

$$wDt_{dw,i} \in \{0,1\} \qquad \forall\, dw \in DW, i \in T \tag{29}$$

$$eD_{dw,i} \in \{0,1\} \qquad \forall\, dw \in DW, i \in T \tag{30}$$

$$sD_{dw,i} \in \{0,1\} \qquad \forall\, dw \in DW, i \in T \tag{31}$$

The variables present in constraints (24), (25), (28), (29), (30) and (31) are binary variables.

4 Application – Model Implementation and Validation

In the following chapter, an illustrative example is presented and explained in detail so that the mathematical model concepts presented in the previous chapter are easily understood. In addition, with this implementation it is expected to test and validate the model. In the first section, the implementation in FICO Xpress software is described, while the illustrative example and its parameters are specified in the second section. Finally, in the third section, the results of the model are discussed and explained.

4.1 Model Implementation in FICO Xpress Optimization Software

Optimization is key to find the best solution possible for any problem, whether it is a daily question or work-related choices. It is indeed, a very important resource when it comes to decision making since it provides refined solutions. “*Optimization is the mathematical process of finding the best decision for a given business problem within a defined set of constraints—and it can be the difference between success and failure in today’s highly competitive marketplace.*” (FICO, 2019). This process consists in a continuous search for a better solution, combining a real world instance with an algorithm that replicates it and outputs a solution to be then interpreted under it, as Figure 4.1 suggests next.

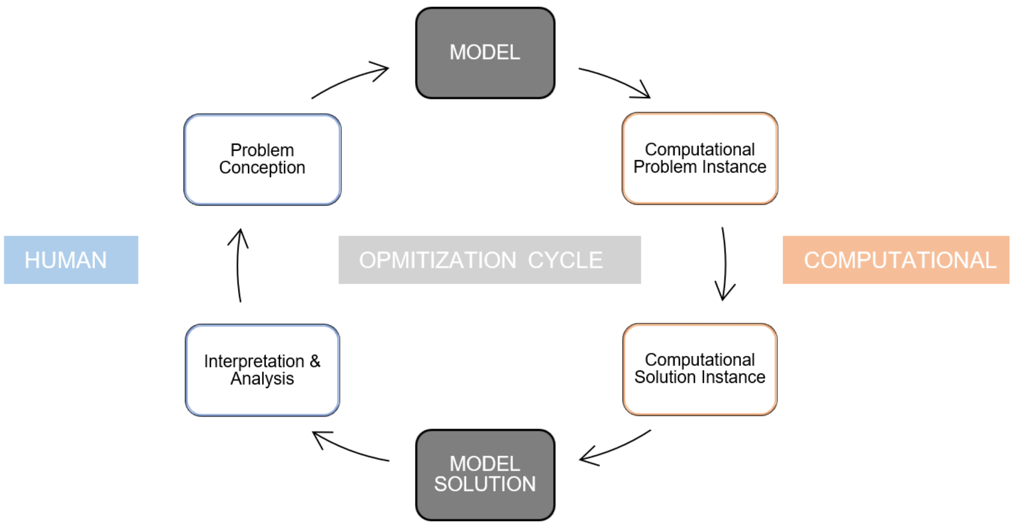


Figure 4.1 – Optimization Process

While on one hand, the constant technology evolution leads to an increasing complexity of these problems, on the other, it also provides software capable of modelling and solutioning even the most complex ones. In fact, optimization plays a major role in railway systems. Operational costs related to both maintenance and drivers crew have a big impact on their expenses and optimization solvers can provide better solutions and thus, lead to reduced costs. In order to find an optimized solution, at least one function must be either maximized or minimized, which is called the objective function. While there is a wide variety of software with optimization solvers on the market from Excel to CPLex, for this particular study, FICO Xpress was the chosen one.

"FICO Xpress Solver helps customers solve their most difficult, complex business problems by providing the widest breadth of industry leading optimization algorithms and technologies." (FICO, 2019). This software is equipped with different solvers for the model, such as, nonlinear, quadratic and mixed-integer linear solvers. *"FICO's powerful and versatile algorithms solve for large-scale, linear and mixed integer problems, as well as non-linear problems."* (FICO, 2019). For this research, a mixed integer linear solver was selected since both objective function and constrains are linear, adding up to the decision variables which are integers.

After selecting the solver, the mathematical algorithm was written in software's own language, Mosel, an advanced modelling and solving code. This way the relation between the mathematical formulation and the software is established. The model itself is divided into several groups: declarations, initializations, pre-processing, objective function, constraints, outputs. These sections can also be grouped into subsections in order to obtain an easier reading algorithm.

To begin with, the constants, sets and its indexes, parameters and decision variables are presented and defined in the declarations. In the next section, initializations, all the data necessary related to case study is imported through data files which will then be directly read by the software. This information is grouped by categories in several data files making it a lot easier to access and edit. In pre-processing some conditions are established so that the formulation of certain parameters is defined. After this, the objective function is defined followed by all the constraints required to respect the requirements of this particular case. Finally, the program presents the results for the decision variables that define the crew scheduling plan. However, to make the model more user-friendly, a results data file is created presenting the outputs in a simpler way. By doing this, a lot of time is saved, otherwise needed to understand that output.

This results file contains information related to both maintenance and driving crew, so it makes sense separating them. From the decision variable values, the program will output different sentences that contain and explain the results obtained. First, for the maintenance crew, if $wMd_{mw,d,}$ is equal to 0, then a sentence for all maintenance workers will be written, stating "*The maintenance worker #mw, doesn't work on Day [d]".* On the other hand, if $wMd_{mw,d,}$ is equal to 1, then "*The maintenance worker #mw, works on Day [d]".* Additionally, when $wM_{mw,k,m,d}$ is equal to 1, for all maintenance workers allocated, a sentence will pop up, stating *"Maintenance Activity: (m); Unit: [k]; Starting Hour (min):* $t1m_{mw,k,m,d}$*; Finishing Hour (min):* $t2m_{mw,k,m,d}$*"*, indicating which maintenance activities are performed on which unit, by each worker, as well as their starting and finishing times in minutes. Then, for the driving crew, something similar will happen. If wDd_{dw} is equal to 0, then a sentence for all driving workers will be written, stating *"The driving worker #dw, doesn't work on this day".* On the other hand, if wDd_{dw}, is equal to 1, then *"The driving worker #dw, works on this day".* Additionally, if $wDt_{dw,i}$ is equal to 1, then for all assigned drivers a sentence is written stating *"Task i;[*Dd_i*-*Da_i*]; Unit:(k); Entry Hour (min):* $t1i_{dw,i}$ *(*Sd_i*); Exit Hour (min):*

$t2i_{dw,i}$ *(Sa_i)",* highlighting which tasks each driver carries out, the respective unit, departure and arrival times and stations, as well as the drivers' entry and exit times in minutes. All of this data combined will form the optimized operational crew schedule.

After analysing the output of the model, it is good practice to check if all the constraints are respected, as well as if the values obtained make sense. After confirming that the model is coherent and respects every aspect required, we can test and validate it in a smaller scale problem or illustrative example, before analysing the actual case study.

4.2 Illustrative Example

As stated above, in order to validate the model in a more efficient way, an illustrative example is used. This illustrative example, or "toy problem", has the same sets and parameters as the case study, only in a much smaller scale. In this way, the computational time required is much shorter leading to a faster analysis and validation, as well as an easier understanding of the model.

In this example, 3 train units cover 9 tasks, in which one of them has to perform 2 different maintenance actions previously scheduled, while another unit has to perform a single one. Each action requires certain competences, so the amount of work required to perform it depends not only on the maintenance action but also on the competences needed, AWm,c. There are 5 distinct competences which a worker can master or not. Finally, the maintenance crew team is formed by 6 workers, and the driving crew by 5 drivers.

The ultimate aim is to obtain the best operational crew scheduling possible, i.e. one that minimizes the costs related to employing both maintenance and driving crew.

The following tables 4.1 to 4.12 provide all the parameter and its values used for this illustrative example.

Table 4.1 presents all the constants used in this "toy problem". Following the respective order, the number of units *(NU)*, the number of stations *(NS)*, the number of maintenance workers *(NMW)*, the number of different competences *(NCC)*, the number of driving workers *(NDW)*, the number of types of maintenance actions *(NM)*, the number of tasks *(NT)*, the number of days of the time horizon considered *(ND)* and the number of maximum working minutes in a day *(TMS)*. Then, the gap between the entry/exit time of a driver and respectively the starting/finishing time of the first/last task (Δ_m), the gap required by the maintenance crew when changing work units (t_{min}), a set up time required by units after arriving and before leaving the depot (t_{man}) and finally, a large number used in some of the constraints regarding maintenance *(LN)*.

Table 4.1 – Constants used

Constants	Unit	Value
NU	-	**3**
NS	-	**4**
NMW	-	**6**
NCC	-	**5**
NDW	-	**5**
NM	-	**3**
NT	-	**9**
ND	day	**1**
TMS	min	**540**
Δ_m	min	**15**
t_{min}	min	**5**
t_{man}	min	**5**
LN	-	**10000**

In Table 4.2, *mw* and *c*, represent the maintenance workers and the competences respectively. If a specific competence is mastered by a maintenance worker*, then* $MWC_{mw,c}$ is equal to 1, otherwise, its value is equal to 0. In this case, maintenance workers 1, 2 and 3 do not master only one competence, c_3, whereas maintenance worker 5 does not have competences 2 and 3. Maintenance workers 4 and 6 master all the competences possible in this particular case.

Table 4.2 – Maintenance crew competences

$MWC_{mw,c}$		*c*				
		c_1	c_2	c_3	c_4	c_5
mw	1	**1**	**1**	0	**1**	**1**
	2	**1**	**1**	0	**1**	**1**
	3	**1**	**1**	0	**1**	**1**
	4	**1**	**1**	**1**	**1**	**1**
	5	**1**	0	0	**1**	**1**
	6	**1**	**1**	**1**	**1**	**1**

Table 4.3 provides information on the maintenance actions' duration (MT_m) and the amount of work per competence $(AW_{m,c})$ required by each one of them. All maintenance actions have an associated duration, which is presented in the second column and an amount of work *per* competence, which is specified in the next columns. Logically, if $AW_{m,c}$ is equal to 0, then the respective maintenance action does not require that specific skill to be performed. It can also be assumed, that the total amount of work for each action, AWt_m, is equal to the maximum value of $AW_{m,c}$, as presented in the last column.

Table 4.3 – Amount of work and duration of maintenance actions

m	*MT (min)*	Amount of Work *per Competence, $AW_{m,c}$ (min)*					AW_t *(min)*
		c_1	c_2	c_3	c_4	c_5	
1	**186**	**744**	0	0	**300**	**300**	**744**
2	**53**	**210**	**100**	0	**50**	**50**	**210**
3	**60**	**60**	**40**	**20**	**35**	0	**60**

In Table 4.4, information about whether a unit *k* goes to the depot to perform maintenance actions on a given day *d* is presented. For the present example, in which the time horizon is just 1 day, units 1 and 2 go to the depot to perform maintenance work, thus their $ZM_{k,d}$ value is equal to 1.

Table 4.4 – Information on units going to depot to perform maintenance

$ZM_{k,d}$		*d*
		1
k	**1**	**1**
	2	**1**
	3	0

Table 4.5 is relative the previously scheduled maintenance actions for each train unit. If $KM_{k,m}$, is equal to 1, then a specific unit must perform the respective maintenance action. For instance, in this illustrative example, train 1 is set to perform a maintenance action 3 and unit 2 must complete maintenance actions 2 and 3.

Table 4.5 – Information on maintenance actions scheduled for each unit

$KM_{k,m}$		*m*		
		1	**2**	3
k	**1**	0	0	**1**
	2	0	**1**	**1**
	3	0	0	0

Next, in figure 4.2 the rolling-stock schedule used in this example is displayed. To introduce all the information related to this timetable in the model, several tables were conceived containing all the data present in the figure. Tables were grouped in a way information can be understood more easily.

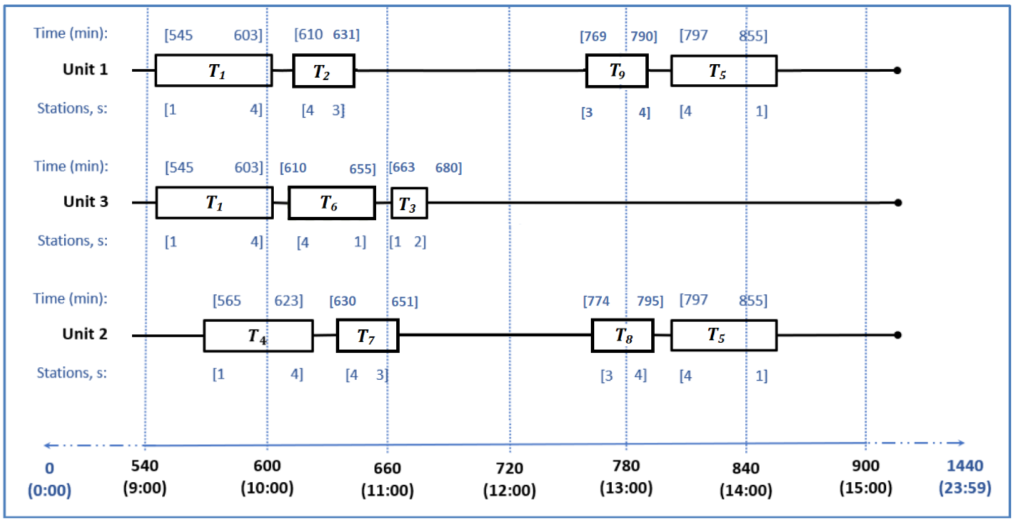


Figure 4.2 – Rolling-stock plan used in this illustrative example (Adapted from Mira, 2018)

Table 4.6 simply indicates the correspondence between each station name and number. In the first column, it is presented the station name, followed by the respective number in the second column. While *Roma-Areeiro*, *Pragal* and *Setúbal* are stations where passengers can enter or exit, *PMC* or depot, is where trains off duty, i.e. without any passengers, can access to perform maintenance.

Table 4.6 – Information regarding stations

Station Name	Station Number
Roma-Areeiro	1
Pragal	2
PMC (Depot)	3
Setúbal	4

Table 4.7 provides information on the tasks that need to be covered and its respective stations of departure and arrival *(Sd_i; Sa_i),* as well as departure and arrival times *(Dd_i; Da_i)*, in minutes. For instance, task 1, departs from station 1 at 9h05 and arrives station 4, at 10h03. It should also be pointed out that some tasks may be performed simultaneously, as it happens for example with tasks 1 and 4. When this happens, logically the same unit cannot perform both tasks and therefore another unit is required.

Table 4.7 – Information about tasks

Task	Sd_i	Sa_i	Dd_i	Da_i
1	1	4	545	603
2	4	1	610	631
3	1	2	663	680
4	1	4	565	623
5	4	1	797	855
6	4	1	610	655
7	4	3	630	651
8	3	4	774	795
9	3	4	769	790

Table 4.8 is also related to tasks, more precisely, it presents information explaining which units will perform each task. If $X_{k,i}$ is equal to 1, then the given unit *k* performs the respective task *i*. Otherwise, that task must be performed by a different unit. It is important to note that each task can be performed by 1 or 2 units. For instance, in the present example there are two tasks performed by two different units: task 1 is carried out by units 1 and 3, whereas task 5 is completed by units 1 and 2.

Table 4.8 – Information on tasks carried out by each unit

$X_{k,i}$		*i*								
		1	2	3	4	5	6	7	8	9
k	1	**1**	**1**	0	0	**1**	0	0	0	**1**
	2	0	0	0	**1**	**1**	0	**1**	**1**	0
	3	**1**	0	**1**	0	0	**1**	0	0	0

In Table 4.9 it is possible to examine which consecutive tasks *i* and *j* are linked by unit *k*. In case $Y_{k,i,j}$, is equal to 1 then tasks *i* and *j,* are linked by the respective unit k. On the other hand, if this does not happen, then tasks *i* and *j* are linked by a different unit or simply are not linked. For instance, hereby we can conclude that: unit k_1 links tasks 1 and 2, tasks 2 and 9 and also tasks 9 and 5; unit k_2 links tasks 4 and 7, tasks 7 and 8 as well as tasks 8 and 5; finally, unit k_3 links tasks 1 and 6, and tasks 6 and 3. It should also be pointed out that two successive tasks *i* and *j* cannot be linked by more than 2 units, as Mira (2018) previously established.

Table 4.9 – Pair of tasks linked by each unit

$Y_{k,i,j}$				*j*								
				1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
			1	0	**1**	0	0	0	0	0	0	0
			2	0	0	0	0	0	0	0	0	**1**
			3	0	0	0	0	0	0	0	0	0
			4	0	0	0	0	0	0	0	0	0
	1	***i***	**5**	0	0	0	0	0	0	0	0	0
			6	0	0	0	0	0	0	0	0	0
			7	0	0	0	0	0	0	0	0	0
			8	0	0	0	0	0	0	0	0	0
			9	0	0	0	0	**1**	0	0	0	0
			1	0	0	0	0	0	0	0	0	0
			2	0	0	0	0	0	0	0	0	0
			3	0	0	0	0	0	0	0	0	0
k			**4**	0	0	0	0	0	0	**1**	0	0
	2	***i***	**5**	0	0	0	0	0	0	0	0	0
			6	0	0	0	0	0	0	0	0	0
			7	0	0	0	0	0	0	0	**1**	0
			8	0	0	0	0	**1**	0	0	0	0
			9	0	0	0	0	0	0	0	0	0
			1	0	0	0	0	0	**1**	0	0	0
			2	0	0	0	0	0	0	0	0	0
			3	0	0	0	0	0	0	0	0	0
			4	0	0	0	0	0	0	0	0	0
	3	***i***	**5**	0	0	0	0	0	0	0	0	0
			6	0	0	**1**	0	0	0	0	0	0
			7	0	0	0	0	0	0	0	0	0
			8	0	0	0	0	0	0	0	0	0
			9	0	0	0	0	0	0	0	0	0

Whenever consecutive tasks *i* and *j* are successfully linked by a unit *k*, i.e. when $Y_{k,i,j}$, is equal to 1, there is the possibility that a maintenance action *m* is performed between them. If this happens, then $YM_{k,i,j,m} = 1$. From Table 4.10, it can be concluded that unit k_1 is under maintenance between tasks 2 and 9 to carry out maintenance action m_3 and unit k_2 performs actions m_2 and m_3 between the pair of tasks 7 and 8. It is also noteworthy to mention that in order to perform a given maintenance action *m* between two consecutive tasks *i* and *j*, logically its duration MT_m must be lower than the gap between the end of task *i* and the start of task *j*.

Table 4.10 – Information on the unit and pair of tasks between each maintenance action is completed

$YM_{k,i,j,m}$		m = 1	m = 2	m = 3
$Y_{k,i,j} = 1$	$Y_{1,1,2}$	0	0	0
	$Y_{1,2,9}$	0	0	**1**
	$Y_{1,9,5}$	0	0	0
	$Y_{2,4,7}$	0	0	0
	$Y_{2,7,8}$	0	**1**	**1**
	$Y_{2,8,5}$	0	0	0
	$Y_{3,1,6}$	0	0	0
	$Y_{3,6,3}$	0	0	0

Finally, Tables 4.11 and 4.12 present information related to the daily cost of employing a maintenance and/or a driving worker. While for the maintenance crew this value depends on the worker, its years of service and its skills, for the drivers this value is equal for the entire crew.

Table 4.11 – Cost of employing a maintenance worker

mw	1	2	3	4	5	6
$\boldsymbol{Cmw_{mw}}$	43	43	43	66	46	66

Table 4.12 – Cost of employing a driving worker

dw	1	2	3	4	5
$\boldsymbol{Cdw_{dw}}$	54	54	54	54	54

4.3 Results of the Optimization Model for the Illustrative Example

After the simulation is concluded, the program presents a solution of the minimum cost found for the crew scheduling, 423 monetary units (Figure 4.3). This solution is valid for the given time horizon of the present example, which is 1 day.

```
The minimum total working cost is : 423
Begin running model
End running model
```

Figure 4.3 – Minimum cost obtained in the illustrative example

Additionally, a data file is also created, comprehending the information regarding crew assignment and scheduling for both crews. Figure 4.4 presents the plan for the maintenance crew, highlighting if workers are assigned to work on this day, and if that's true, which maintenance activities are carried out by each one of them, as well as the unit under maintenance. Herewith, it is also displayed the starting and finishing hour, in minutes, for each activity carried out by each crew member. Figure 4.5 displays the driving crew scheduling information necessary to create the timetable, indicating whether drivers work on this day, and if so, the tasks, as well as departure and arrival times, carried out by them on the respective unit. Last but not least, the entry and exit hour for each driver are also presented. It is important to remember that each driver can only entry/exit service once, so logically some tasks do not have associated entry/exit times. However, for the sake of understanding the driver assignment, all tasks carried out are presented. After being displayed, these outputs are explained in detail for a better understanding of the solution.

```
[MAINTENANCE CREW]

Maintenance Worker #1 works on Day [1]
Maintenance Activity: (2); Unit: [2]; Starting Hour (min): 656; Finishing Hour (min): 709

Maintenance Worker #2 works on Day [1]
Maintenance Activity: (2); Unit: [2]; Starting Hour (min): 656; Finishing Hour (min): 709

Maintenance Worker #3 works on Day [1]
Maintenance Activity: (2); Unit: [2]; Starting Hour (min): 656; Finishing Hour (min): 709

Maintenance Worker #4 works on Day [1]
Maintenance Activity: (3); Unit: [1]; Starting Hour (min): 636; Finishing Hour (min): 696

Maintenance Worker #5 doesn't work on Day [1]

Maintenance Worker #6 works on Day [1]
Maintenance Activity: (2); Unit: [2]; Starting Hour (min): 656; Finishing Hour (min): 709
Maintenance Activity: (3); Unit: [2]; Starting Hour (min): 709; Finishing Hour (min): 769
```

Figure 4.4 – Results for the maintenance crew scheduling problem regarding the toy problem

```
[DRIVING CREW]

Driving Worker #1 doesn't work on Day [1]

Driving Worker #2 doesn't work on Day [1]

Driving Worker #3 works on Day [1]
Task4;[565-623]; Unit:(2); Entry Hour:550 (1); Exit Hour:0 (4)
Task5;[797-855]; Unit:(1); Entry Hour:0 (4); Exit Hour:870 (1)
Task5;[797-855]; Unit:(2); Entry Hour:0 (4); Exit Hour:870 (1)
Task7;[630-651]; Unit:(2); Entry Hour:0 (4); Exit Hour:0 (3)
Task8;[774-795]; Unit:(2); Entry Hour:0 (3); Exit Hour:0 (4)

Driving Worker #4 works on Day [1]
Task2;[610-631]; Unit:(1); Entry Hour:595 (4); Exit Hour:0 (3)
Task9;[769-790]; Unit:(1); Entry Hour:0 (3); Exit Hour:805 (4)

Driving Worker #5 works on Day [1]
Task1;[545-603]; Unit:(1); Entry Hour:530 (1); Exit Hour:0 (4)
Task1;[545-603]; Unit:(3); Entry Hour:530 (1); Exit Hour:0 (4)
Task3;[663-680]; Unit:(3); Entry Hour:0 (1); Exit Hour:695 (2)
Task6;[610-655]; Unit:(3); Entry Hour:0 (4); Exit Hour:0 (1)
```

Figure 4.5 – Results for the driving crew scheduling problem regarding the toy problem

First of all, it is important to highlight that all the maintenance activities and tasks were successfully executed. Concerning the maintenance crew, it is possible to conclude that 5 crew members are required to carry out all maintenance actions, while for the driving crew, only 3 drivers work on this day.

Regarding the maintenance crew scheduling problem, according to the relation presented earlier, maintenance action m_2 requires at least 4 maintenance workers, as $\frac{AWt_{m2}}{MT_{m2}} = 4$, while maintenance activity m_3 needs a single worker, $\frac{AWt_{m2}}{MT_{m2}} = 1$. Logically maintenance workers must have some specific competences required for each action and to meet the expected minimization of costs, and thus, reduce the number of working members preferably, the assigned workers should cover all the required skills, so that no other member has to be assigned. Here, all workers but one, mw_5, master all the skills required to carry out m_2, while regarding maintenance activity m_3, only crew members mw_4 and mw_6, have the required complete set of skills. First, in Figure 4.4 it is pretty clear to see that maintenance worker $\boldsymbol{mw_5}$ **is not assigned** to this given day. Maintenance workers $\boldsymbol{mw_1}$, $\boldsymbol{mw_2}$ **and** $\boldsymbol{mw_3}$ are assigned to a single maintenance activity m_2 to be performed on unit k_2, from 10h56 to 11h49. Maintenance worker $\boldsymbol{mw_4}$ performs maintenance actions m_3, on unit k_1, from 10h36 to 11h36. Finally, maintenance worker, $\boldsymbol{mw_6}$**,** carries out to two maintenance actions m_2 and m_3, to be performed on **the same train unit** k_2, meaning no gap time is necessary to change unit. This way, after executing m_2 from 10h56 to 11h49, m_3 can start right away from 11h49 to 12h49. In this way, all the maintenance activities are performed within the right time frame, meaning that the rolling stock units can proceed to their next duties without any type delay.

It is also important to note that any worker assigned to action m_2 could not perform m_3 on unit k_1, due to time constraints. In other words, maintenance activity m_2 only ends at 11h49, so in order to perform maintenance in a different unit *k,* the maintenance worker would require a certain amount of time to change units, t_{min} *(5 min)*, and thus m_3 could only be started on k_1 at 11h54 *(11h49 + 5 min)* and finished one hour later, since the duration of this activity is equal to 60 min. However, this is not possible, since unit k_1 has to finish maintenance at 12h44 maximum to carry out task 9 previously scheduled to start at 12h49. In this way, logically a different maintenance worker with the required skills had to be assigned to work on this day to perform that maintenance.

Regarding drivers scheduling in Figure 4.5, it is possible to observe that drivers $\boldsymbol{dw_1}$ and $\boldsymbol{dw_2}$ **are not required** to successfully carry out the planned duties. Driving worker $\boldsymbol{dw_3}$ is assigned to tasks $\boldsymbol{T_4}$, $\boldsymbol{T_7}$, $\boldsymbol{T_8}$ and $\boldsymbol{T_5}$, meaning that he/she covers tasks carried out on unit $\boldsymbol{k_2}$, including $\boldsymbol{T_5}$, which is also carried out by unit $\boldsymbol{k_2}$. Therefore, it is possible to conclude that he/she is scheduled to enter service before $\boldsymbol{T_4}$, at 9h10, and exit after $\boldsymbol{T_5}$, at 14h30, resulting in a total service time of 320min, this is, 5h20, and thus respecting the daily maximum service time, *TMS (540 min or 9 hours)*. Driver $\boldsymbol{dw_4}$ realizes tasks $\boldsymbol{T_2}$ and $\boldsymbol{T_9}$, on unit $\boldsymbol{k_1}$, and thus, it starts working at 9h55 (before $\boldsymbol{T_2}$) and finishes at 13h25 (after $\boldsymbol{T_9}$*).* Once again, as desired, *TMS* is respected since, 13h25 – 9h55 ≤ *TMS*. Finally, the remaining tasks $\boldsymbol{T_1}$, $\boldsymbol{T_3}$ and $\boldsymbol{T_6}$, carried out on unit $\boldsymbol{k_3}$, as well as $\boldsymbol{k_1}$ required to perform ***T₁***, are performed by driver ***dw₅***, whose entry and exit times are 8h50 and 11h35, respectively, resulting in 2h45 of working time. With this output it is possible to note that the changes of units are minimized, since each driver carries out tasks on the same unit, during all times, excepting the cases where tasks require two train units.

For an easier comprehension of the assignment and scheduling of each worker, Table 4.13 presented next, was conceived in order to achieve a wider overview of the whole problem. By doing this, the data files outputted by the model are complemented with a schematic visual diagram, making it easier to consult the whole crew schedule. It is also important to note that for comprehension reasons, the hours respective to each action/task are not displayed, however, these can be known, when integrated with the data files mentioned above.

Table 4.13 – Distribution of actions for maintenance crew member and tasks for driver

Maint. Crew	MAINT. ACTIONS			Driving Crew	TASKS								
	m_1	m_2	m_3		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
UNIT k_1													
mw_1	-	-		dw_1									
mw_2	-	-		dw_2									
mw_3	-	-		dw_3					x				
mw_4	-	-	x	dw_4		x							x
mw_5				dw_5	x								
mw_6	-	-											
UNIT k_2													
mw_1	-	x		dw_1									
mw_2		x		dw_2									
mw_3	-	x		dw_3				x	x		x	x	
mw_4	-			dw_4									
mw_5				dw_5									
mw_6	-	x	x										
UNIT k_3													
mw_1	-	-	-	dw_1									
mw_2				dw_2									
mw_3	-	-	-	dw_3									
mw_4	-	-	-	dw_4									
mw_5				dw_5	x		x			x			
mw_6	-	-	-										

In this table the data related to both maintenance crew and activities is presented in orange while tasks and drivers are displayed in a green colour. The maintenance actions/tasks assigned to a given worker are marked with an *"x"*. Also note that for crew members that are not required to work on this day, every respective action/task is shown in grey, meaning that they cannot carry them out. With this arrangement it is clearer to see the intention of minimizing the number of working crew members, and consequently, minimizing the objective function.

Regarding this illustrative example, the model took less than a tenth of a second to output an optimized solution, although for larger instances where the number of variables is much larger, the computational time required will logically increase. Figure 4.6 is an output from the sofware used, and presents some helpful information regarding the matrix' size, such as the number of columns or variables, before and after the pre-solving stage, along with the data regarding the final solution, such as the best bound, best solution and gap.

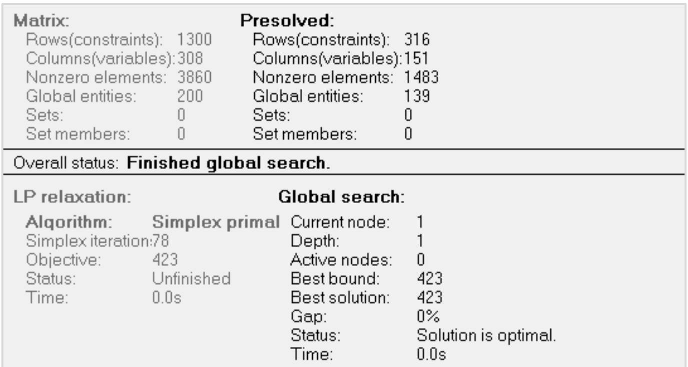


Figure 4.6 – Illustrative example's computational data

As it can be observed, for the current illustrative example, the best solution is 423 monetary units, which is indeed optimal as the gap is 0% (Figure 4.6). For larger instances where the required computational time may be extremely larger and non-practical, this value presents a good indicator on the quality of the obtained solution. In fact, the gap is by definition the ratio between the difference between the best solution and the best bound (in percentage), as presented in equation 32:

$$gap = \frac{best\ solution - best\ bound}{best\ solution} \times 100\ \% \qquad (32)$$

Finally, Table 4.14 exhibits the relation between the columns of the matrix and the size of the decision variables. In fact, the size of each decision variable is equal to the product of the size of each set of the corresponding indices. Note that, as mentioned before, the lower the number of columns, less is the computational time required. In this example, before the pre-solving stage, which is able to reduce the size of the initial matrix, the matrix is formed by 308 columns. After some redundant data is dismissed, and therefore the size of the initial matrix reduced to 151 columns. However, this process is already out of the scope of this study.

Table 4.14 – Calculus of the matrix column

Set	Size	Set	Size	Size of the Columns
$k = \{1, \ldots, 3\}$	3	$wMd_{mw,d}$	$6 \times 1 = 6$	$6 + 54 + 54 + 54 + 5 + 45 + 45 + 45 = 308$
$s = \{1, \ldots, 5\}$	5	$wM_{mw,k,m,d}$	$6 \times 3 \times 3 \times 1 = 54$	
$i = \{1, \ldots, 8\}$	9	$t1m_{mw,k,m,d}$	$6 \times 3 \times 3 \times 1 = 54$	
$m = \{1, \ldots, 3\}$	3	$t2m_{mw,k,m,d}$	$6 \times 3 \times 3 \times 1 = 54$	
$d = \{1\}$	1	wDd_{dw}	5	
$c = \{1, \ldots, 5\}$	5	$wDt_{dw,i}$	$5 \times 9 = 45$	
$mw = \{1, \ldots, 6\}$	6	$eD_{dw,i}$	$5 \times 9 = 45$	
$dw = \{1, \ldots, 5\}$	5	$sD_{dw,i}$	$5 \times 9 = 45$	

5 Case Study – Fertagus

In this chapter, a brief introduction and description of Fertagus train operating company is initially presented. Further on, the model is applied to the case study, and the specific case study inputs are displayed and discussed.

5.1 Fertagus Train Operating Company

Fertagus is a private train operating company, a branch of the group Barraqueiro, which links Setúbal, in the south bank of Lisbon, to Roma-Areeiro, up north the *Tejo* river. In fact, it was the first private company to assure the commercial concession of a railway line in Portugal, meaning to use the existing infrastructures, a fee is charged by the infrastructure manager, *Infraestruturas de Portugal,* (IP). Herewith, it is also important to note that Fertagus' trains are not the only ones using the rail track between Roma-Areeiro and Setúbal, a fact that may constitute a problem since not every train unit has the same requirements. Besides operating the railway line, the company is also accountable for the maintenance of the rolling-stock units as well as the maintenance of some railway stations, while IP is responsible for the railway infrastructure maintenance.

The railway line has an extension of 54 kilometres, split into 3 different routes *"Linha de Cintura", "Linha do Sul"* and *"Linha do Sado"*, as shown in Figure 5.1. The company serves 14 different stations, linking both banks of the river. While 10 of these stations are located in the South, where the company's headquarters are based, namely in Coina, where the maintenance yard is located, the remaining 4 are situated on the North side of the line. The trip linking one extreme to another, i.e. Roma-Areeiro to Setúbal, has an approximated duration of 57 minutes.

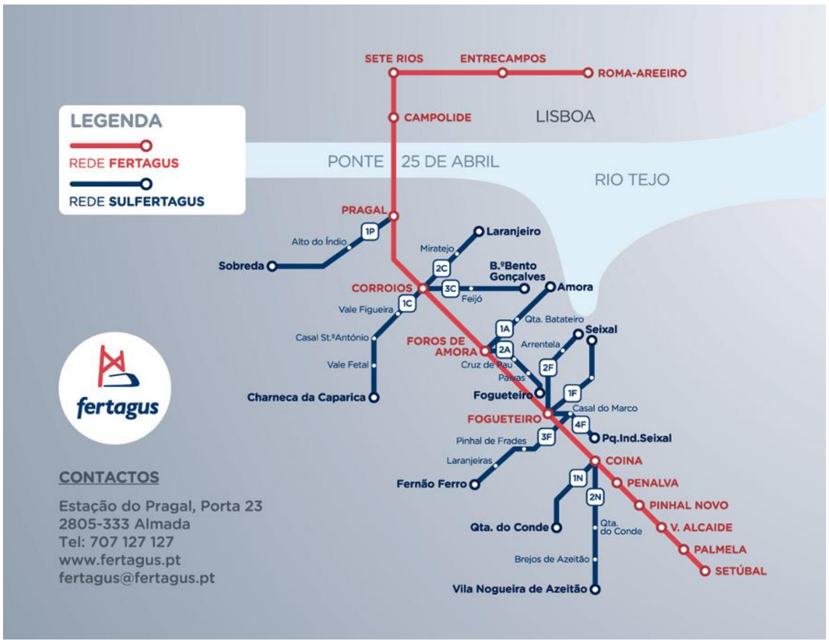


Figure 5.1 – Fertagus' railway map (R. f.: Fertagus website)

Although only 17 units are necessary to carry out the scheduled duties, the train fleet is formed by 18 units. This happens so that there is a train unit, which is rotatively in the depot, since no train can be pulled out of service to perform maintenance activities in case there is no other train available to replace it. This approach was implemented in order to meet the availability clauses established in the current contract.

According to R&C Fertagus (2018), by the end of 2018, the company had a total of 157 contributors, from directors to maintenance staff. Additionally, the company's maintenance yard has several lines with different functions. There are lines for parking units, lines to carry out some tests and the most relevant to the present study, lines where maintenance activities are carried out, i.e. rooftop-covered lines. More precisely, there are 3 lines where maintenance may be executed.

Finally, Fertagus, which is responsible for approximately 70,000 daily moves, reports that its service played a major role improving customers' quality of life considerably. Indeed, it states that many of its customers have nowadays more free time, more flexibility due to the numerous trips existing, as well as finding the trips more comfortable. In the future, Fertagus intends to continuously prove that this public transportation service is a valid and reliable alternative, with the drive to establish a better coordinated transportation plan integrated with other transport systems.

5.2 Input Parameters

After validating the model using a smaller-size illustrative example, it is time to apply it to a real world problem, the Fertagus case study. In order to collect all the information and details necessary to the study, several meetings were arranged, as well as visits to the maintenance yard, for an easier understanding on how the maintenance yard works.

In this case study, 17 train units are supposed to cover 196 daily tasks, in which some of them have to perform maintenance actions previously scheduled. There are 14 different types of maintenance activities and each one requires a certain set of competences/skills, so the amount of work required to perform it depends not only on the maintenance action but also on the competences needed, $AW_{m,c}$. There are 10 distinct competences which a worker can master, and each worker may use several competences at once. Finally, the maintenance crew team is formed by 16 workers, whereas the number of drivers is equal to 29. The ultimate aim is to obtain the best operational crew scheduling possible for one day of the week under a short computational time. One that minimizes the costs related to employing both maintenance and driving crew.

While most of the specifications and requirements of the real problem are represented in the model, some details had to be adapted and/or added, so the solution obtained may not represent the real optimal solution. For instance, the actual Fertagus maintenance crew is flexible and well-rounded, i.e. basically any worker can perform any action. However, in this model it is intended to implement some complexity and so a set of skills was conceived for every maintenance worker according to some data files provided by the Fertagus maintenance director. Furthermore, the amount of work, previously defined by Mira (2018), is here defined as the total amount of work, as such discretization is made across all competences, herewith represented by the amount of work per competence. The respective values were created since no specific inputs were provided by the company. Moreover, as mentioned before, the actual Fertagus operating rolling stock schedule is here adopted, combined with the outputs presented by Mira (2018), considering the necessary adaptations.

Additionally, it is important to clarify that some logical limits and constraints are already implied in the inputs provided by Mira (2018). For example, as stated in the previous section, there are only 3 different maintenance lines. Therefore, no more than 3 maintenance actions can be carried out at a time. In the current study, that is already included in the inputs, where the user defines which maintenance actions are scheduled to be performed, and logically, respects that constraint. Additionally the number of type of maintenance activities used differs from the one applied by Mira (2018), based on the information provided by Fertagus.

The model outputs a daily crew schedule, so we can obtain the optimal crew scheduling for any day of the week in a short time, by changing the inputs to the respective rolling stock timetable and maintenance plan. It is left for further research a weekly crew scheduling model, that can cope with different maintenance actions and/or tasks for each day.

Lastly, in line to what was done in the illustrative example, Tables 5.1 to 5.12, present all the parameters and the respective values used in this case study.

Table 5.1 presents the constants used in the present case study. Following the respective order, it can be read, the number of units *(NU)*, the number of stations *(NS),* the number of maintenance workers *(NMW)*, the number of different competences *(NCC)*, the number of driving workers *(NDW)*, the number of types of maintenance actions *(NM)*, the number of tasks *(NT)*, the number of days of the time horizon considered *(ND)* and the number of maximum working minutes in a day *(TMS)*. Then, it is displayed the gap between the entry/exit time of a driver and respectively the starting/finishing time of the first/last task (Δ_m), the gap required by the maintenance crew when changing work units (t_{min}), a set up time required by units after arriving and before leaving the depot (t_{man}) and finally, just simply a large number used in some of the constraints regarding maintenance *(LN)*. The latter is not related to any parameter values in the example.

.

Table 5.1 – Constants used

Constants	Unit	Value
NU	-	**17**
NS	-	**15**
NMW	-	**16**
NCC	-	**10**
NDW	-	**29**
NM	-	**14**
NT	-	**196**
ND	day	**1**
TMS	min	**540**
Δ_m	min	**15**
t_{min}	min	**5**
t_{man}	min	**5**
LN	-	**10000**

In Table 5.2, the maintenance crew members and the corresponding competences are presented. If a specific competence is mastered by a maintenance worker*, then* $MWC_{mw,c}$ is equal to 1, otherwise its value is equal to 0. In this case, some crew members are noteworthy of mention, namely worker mw_6, which only owns competence c_{10}. On the other hand, there are two maintenance workers, mw_{13} and mw_{15}, that possess every single skill. In fact, the last four workers are considered coordinators, fact reflected on the higher number of mastered competences.

Table 5.2 – Maintenance crew competences

$MWC_{mw,c}$		*c*									
		c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
mw	1	**1**	**1**	**1**	**1**	**1**	**1**	0	0	**1**	**1**
	2	**1**	**1**	**1**	**1**	**1**	**1**	0	0	**1**	**1**
	3	**1**	0	0	0	0	**1**	0	0	**1**	**1**
	4	**1**	**1**	**1**	0	0	**1**	0	0	**1**	**1**
	5	**1**	0	0	0	0	0	0	0	**1**	**1**
	6	0	0	0	0	0	0	0	0	0	**1**
	7	**1**	0	**1**	**1**	**1**	**1**	0	0	**1**	**1**
	8	**1**	**1**	**1**	**1**	**1**	**1**	0	0	**1**	**1**
	9	**1**	0	0	0	0	0	0	0	**1**	**1**
	10	**1**	**1**	**1**	**1**	**1**	**1**	0	0	**1**	**1**
	11	**1**	0	0	0	0	0	0	0	**1**	**1**
	12	**1**	**1**	**1**	**1**	**1**	0	0	0	**1**	**1**
	13	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	14	**1**	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	15	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	16	**1**	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Table 5.3 has to do with the maintenance actions' duration *(MT_m)*, the total amount of work *(AWt_m)* and the amount of work per competence *($AW_{m,c}$)* required by each one of them. The maintenance action duration is displayed on the second column, while the total amount of work is specified in the last one. In between, all maintenance actions and associated amount of work *per* competence, is specified . Once again, if $AW_{m,c}$ is equal to 0, then the respective maintenance action does not require that specific skill to be performed, otherwise a crew member with the respective skill must be assigned.

Table 5.3 – Amount of work and duration of maintenance actions

m	*MT (min)*	Amount of Work *per Competence, $AW_{m,c}$ (min)*										*AWt_m (min)*
		c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	
1	**150**	**744**	0	0	0	0	0	0	0	**300**	**250**	**744**
2	**420**	**1680**	0	0	0	0	0	0	0	**420**	**210**	**1680**
3	**210**	**840**	0	0	0	0	0	0	0	**175**	**420**	**840**
4	**210**	**840**	0	0	0	0	0	0	0	**525**	**100**	**840**
5	**276**	**840**	0	0	0	0	**420**	0	0	**200**	**100**	**840**
6	**186**	**744**	0	0	0	0	0	0	0	**300**	**400**	**744**
7	**186**	**744**	0	0	0	0	0	0	0	**250**	**150**	**744**
8	**186**	**744**	0	0	0	0	0	0	0	**150**	**70**	**744**
9	**186**	**744**	0	0	0	0	0	0	0	**125**	**75**	**744**
10	**186**	**744**	0	**700**	**150**	**150**	0	0	0	**200**	**100**	**744**
11	**420**	**840**	**840**	0	**100**	**100**	0	**210**	0	**200**	**100**	**840**
12	**53**	**210**	0	0	0	0	0	**210**	0	**100**	**50**	**210**
13	**53**	**210**	0	0	**200**	0	0	0	**100**	**100**	0	**210**
14	**60**	**60**	0	0	0	**60**	0	0	**20**	**20**	0	**60**

From Table 5.4, it is possible to know which units go to the depot to perform maintenance action on this day. As mentioned above, no more than 3 units can perform maintenance in the same day. Hereby, units k_1, k_3 and k_9, are scheduled to perform maintenance, so their $ZM_{k,d}$ value is equal to 1.

Table 5.4 – Information on units going to depot to perform maintenance

$ZM_{k,d}$		*k*																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
d	1	**1**	0	**1**	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0

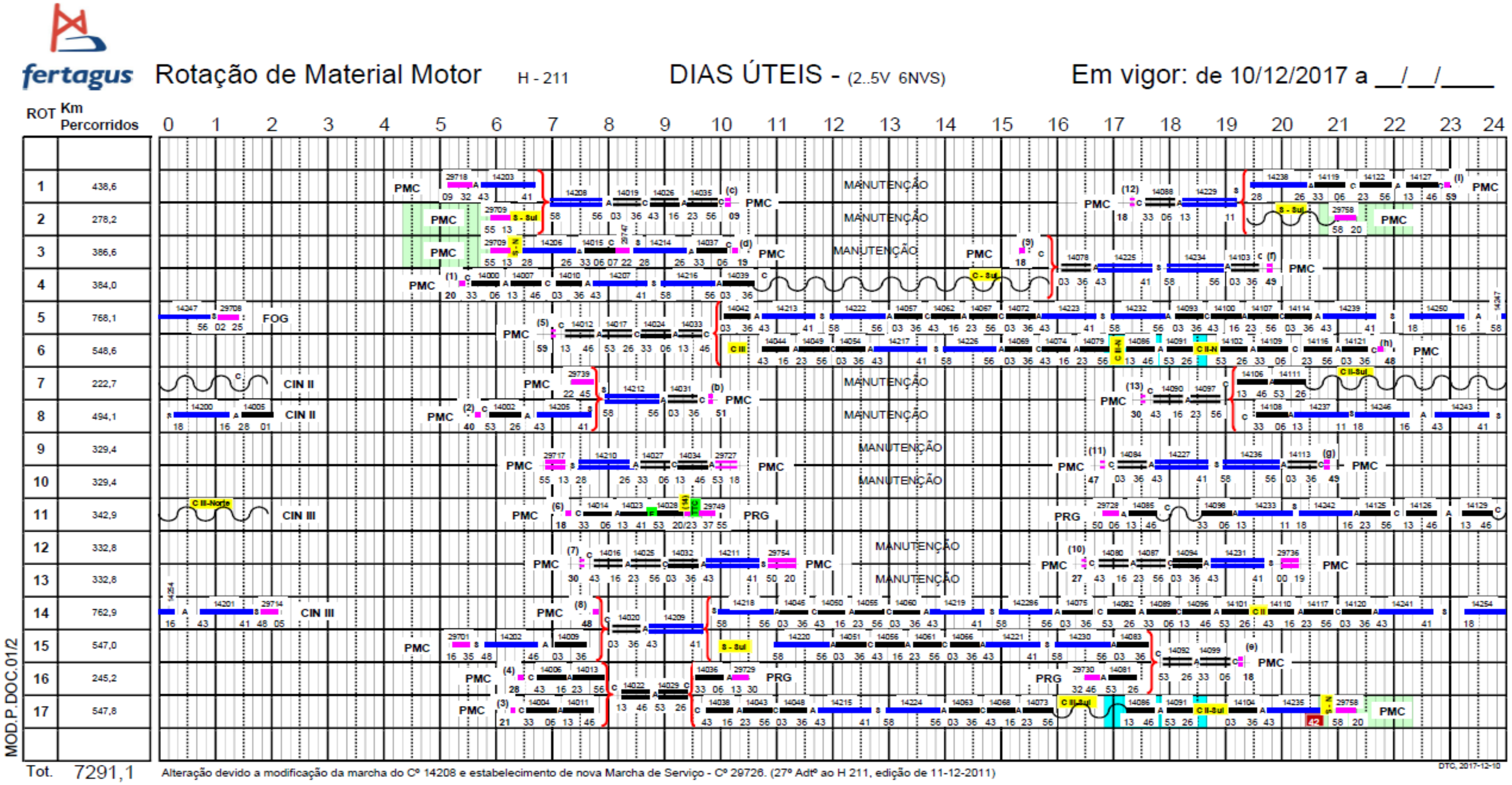
Table 5.5 indicated which rolling-stock units will perform a maintenance activity. If $KM_{k,m}$, is equal to 1, then a unit *k,* performs a maintenance action *m*. For instance, in this case study, only three train units are set to perform maintenance actions. More precisely, unit k_1, which must perform m_3 and m_{13}; unit k_3, which must carry out m_6; and finally, unit k_9, which must complete m_1 and m_9.

Table 5.5 – Information on maintenance actions scheduled for each unit

$KM_{k,m}$		m													
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
k	**1**	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	**1**	0	0	0	0	0	0	0	**1**	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Next, in Figure 5.2 part of the Fertagus current rolling-stock schedule is displayed. This plan covers 4 days of the week, namely, Monday, Tuesday, Wednesday and Thursday. The rest of the days of the week present some variations and so, this study do not focus on them. To understand this scheme it must be considered that there are 3 different types of tasks. Service tasks, represented in blue or black lines, dead-headings in pink lines and finally reserve tasks, characterized by black wavy lines. Additionally, tasks may be carried out by one or two train units, here represented by the number of lines representing a single task, i.e. one line if a single unit is required and a double line if two units are necessary.

All this information is then split and transcripted into several tables, which altogether contain all the data related to the rolling-stock plan necessary to the computational model. However, it should be mentioned that due to the large extension of certain inputs, several tables were shortened and present only an excerpt of the data.



Legenda: - RAR / SAL (1 UQE) - RAR / CIN (1 UQE) - RAR / SAL (2 UQE) - RAR / CIN (2 UQE) - Marchas em vazio

C - Estação de Coina (CIN); S - Setúbal (SAL); R - Roma/Areeiro (RAR); D - Depósito = PMC - Parque de Material de Coina - UQE de reserva

(1) - Cº nº 29703 (05H20-05H22)
(2) - Cº nº 29707 (05H40-05H42)
(3) - Cº nº 29711 (06H21-06H23)
(4) - Cº nº 29753 (06H28-06H30)
(5) - Cº nº 29719 (06H59-07H01)
(6) - Cº nº 29721 (07H18-07H20)
(7) - Cº nº 29723 (07H30-07H32)
(8) - Cº nº 29725 (07H48-07H50)
(9) - Cº nº 29751 (15H18-15H20)
(10)- Cº nº 29731 (16H27-16H29)
(11)- Cº nº 29745 (16H47-16H49)
(12)- Cº nº 29735 (17H18-17H20)
(13)- Cº nº 29733 (17H30-17H32)
(14)- Cº nº 29726 (09H21-09H23)
(a) - Cº nº 29716 (02H13-02H15)
(b) - Cº nº 29720 (09H49-09H51)
(c) - Cº nº 29722 (10H07-10H09)
(d) - Cº nº 29724 (10H17-10H19)
(e) - Cº nº 29732 (19H16-19H18)
(f) - Cº nº 29734 (19H47-19H49)
(g) - Cº nº 29738 (20H47-20H49)
(h) - Cº nº 29742 (21H46-21H48)
(i) - Cº nº 29744 (22H57-22H59)

Figure 5.2 – Fertagus' actual rolling-stock schedule for: Monday, Tuesday, Wednesday and Thursday (R.f.: Fertagus)

Table 5.6 simply indicates the correspondence between each station name and number. In the first column it is presented the station name, followed by the respective number in the second column. Passengers can enter or exit on all stations but one, *PMC* or depot, where only trains off duty, i.e. without any passengers go to perform maintenance activities.

Table 5.6 – Information concerning stations

Station Name	**Station Number**
Roma-Areeiro	1
Entrecampos	2
Sete-Rios	3
Campolide	4
Pragal	5
Corroios	6
Foros de Amora	7
Fogueteiro	8
PMC (Depot)	9
Coina	10
Penalva	11
Pinhal-Novo	12
Venda do Alcaide	13
Palmela	14
Setúbal	15

Table 5.7 is related to the tasks that need to be covered, its respective stations of departure and arrival $(Sd_{i;}\ Sa_i)$, as well as departure and arrival times $(Dd_{i;}\ Da_i)$ in minutes. Due to its huge extension, only a part of it is displayed here. For instance, task 1 departs from station 9 at 5h09 and arrives to station 1 at 5h32. It should also be pointed out that some tasks may be performed simultaneously and when this happens, since the same unit cannot perform both tasks, logically another train is required. Additionally, there are 3 different types of tasks, namely, service tasks, dead-headings, and reserve tasks. However, this does not constitute a relevant variable to the present study and thus it is not presented.

Table 5.7 – Information about tasks

Task	Sd_i	Sa_i	Dd_i	Da_i
1	9	1	309	332
2	1	15	343	401
3	15	1	418	476
4	1	10	483	516
5	10	1	523	556
6	1	10	563	596
(…)				

192	10	1	1033	1066
193	1	10	1073	1106
194	10	1	1143	1176
195	1	15	1183	1242
196	15	9	1258	1280

Table 5.8 relates tasks with train units, more precisely, it exposes information on whether a task is carried out by unit *k*. If $X_{k,i}$ is equal to 1, then unit *k* performs the respective task *i*. Otherwise, that task is performed by another unit. As mentioned above, only a part of this table is presented.

Table 5.8 – Information on tasks carried out by each unit

$X_{k,i}$		*i*												
		1	2	3	4	5	6		191	192	193	194	195	196
k	1	0	**1**	**1**	**1**	**1**	**1**	(…)	0	0	0	0	0	0
	2	0	0	**1**	**1**	**1**	**1**		0	0	0	0	0	0
	3	0	0	0	0	0	0		0	0	0	0	0	0
	4	0	0	0	0	0	0		0	0	0	0	0	0
	5	0	0	0	0	0	0		0	0	0	0	0	0
	6	0	0	0	0	0	0		0	0	0	0	0	0
	7	0	0	0	0	0	0		0	0	0	0	0	0
	8	0	0	0	0	0	0		0	0	0	0	0	0
	9	0	0	0	0	0	0		0	0	0	0	0	0
	10	0	0	0	0	0	0		0	0	0	0	0	0
	11	0	0	0	0	0	0		0	0	0	0	0	0
	12	0	0	0	0	0	0		0	0	0	0	0	0
	13	0	0	0	0	0	0		0	0	0	0	0	0
	14	0	0	0	0	0	0		0	0	0	0	0	0
	15	0	0	0	0	0	0		0	0	0	0	0	0
	16	0	0	0	0	0	0		0	0	0	0	0	0
	17	0	0	0	0	0	0		**1**	**1**	**1**	**1**	**1**	**1**

In table 5.9 it is possible to examine which **consecutive** tasks *i* and *j* are linked by unit *k*. In case $Y_{k,i,j}$ is equal to 1, then tasks *i* and *j* are linked by the respective unit *k*. However, if this is not the case, then tasks *i* and *j* are linked by a different unit or are not linked at all. It is also important to note that the order of the tasks is relevant. For instance, while *(1,2)* is considered a pair of consecutive tasks, that is not true for *(2,1)*. Therefore, it is possible to see some of the tasks performed by units k_1, k_2, and k_{17}.

Table 5.9 – Pair of tasks linked by each unit

$Y_{k,i,j}$		j														
k	i	1	2	3	4	5	6	7	(…)	190	191	192	193	194	195	196
1	1		**1**	0	0	0	0	0		0	0	0	0	0	0	0
1	2	0		**1**	0	0	0	0		0	0	0	0	0	0	0
1	3	0	0		**1**	0	0	0		0	0	0	0	0	0	0
1	4	0	0	0		**1**	0	0		0	0	0	0	0	0	0
1	5	0	0	0	0		**1**	0		0	0	0	0	0	0	0
1	6	0	0	0	0	0		**1**		0	0	0	0	0	0	0
1	7	0	0	0	0	0	0			0	0	0	0	0	0	0
1	(…)															
1	190	0	0	0	0	0	0	0			0	0	0	0	0	0
1	191	0	0	0	0	0	0	0		0		0	0	0	0	0
1	192	0	0	0	0	0	0	0		0	0		0	0	0	0
1	193	0	0	0	0	0	0	0		0	0	0		0	0	0
1	194	0	0	0	0	0	0	0		0	0	0	0		0	0
1	195	0	0	0	0	0	0	0		0	0	0	0	0		0
1	196	0	0	0	0	0	0	0		0	0	0	0	0	0	
2	1		0	0	0	0	0	0		0	0	0	0	0	0	0
2	2	0		0	0	0	0	0		0	0	0	0	0	0	0
2	3	0	0		**1**	0	0	0		0	0	0	0	0	0	0
2	4	0	0	0		**1**	0	0		0	0	0	0	0	0	0
2	5	0	0	0	0		**1**	0		0	0	0	0	0	0	0
2	6	0	0	0	0	0		**1**		0	0	0	0	0	0	0
2	7	0	0	0	0	0	0			0	0	0	0	0	0	0
2	(…)															
2	190	0	0	0	0	0	0	0			0	0	0	0	0	0
2	191	0	0	0	0	0	0	0		0		0	0	0	0	0
2	192	0	0	0	0	0	0	0		0	0		0	0	0	0
2	193	0	0	0	0	0	0	0		0	0	0		0	0	0
2	194	0	0	0	0	0	0	0		0	0	0	0		0	0
2	195	0	0	0	0	0	0	0		0	0	0	0	0		0
2	196	0	0	0	0	0	0	0		0	0	0	0	0	0	
(…)																
17	1		0	0	0	0	0	0		0	0	0	0	0	0	0
17	2	0		0	0	0	0	0		0	0	0	0	0	0	0
17	3	0	0		0	0	0	0		0	0	0	0	0	0	0
17	4	0	0	0		0	0	0		0	0	0	0	0	0	0
17	5	0	0	0	0		0	0		0	0	0	0	0	0	0
17	6	0	0	0	0	0		0		0	0	0	0	0	0	0
17	7	0	0	0	0	0	0			0	0	0	0	0	0	0
17	(…)															
17	190	0	0	0	0	0	0	0			**1**	0	0	0	0	0
17	191	0	0	0	0	0	0	0		0		**1**	0	0	0	0
17	192	0	0	0	0	0	0	0		0	0		**1**	0	0	0
17	193	0	0	0	0	0	0	0		0	0	0		**1**	0	0
17	194	0	0	0	0	0	0	0		0	0	0	0		**1**	0
17	195	0	0	0	0	0	0	0		0	0	0	0	0		**1**
17	196	0	0	0	0	0	0	0		0	0	0	0	0	0	

From Table 5.4, it is possible to identify which units *k* perform maintenance on this day, while in Table 5.5, it is specified which maintenance activities *m* are carried out on each unit that is scheduled to go to the depot. Whenever a pair of consecutive tasks *(i,j)* is successfully linked by a unit *k*, i.e. when $Y_{k,i,j}$ is equal to 1, then there is the possibility a maintenance action *m* is performed between them. Hereby, Table 5.10 specifies between which pair of tasks *(i,j)*, each rolling-stock unit *k,* performs the scheduled maintenance actions *m*. For these cases, $YM_{k,i,j,m} = 1$. In this way, the maintenance slots are established, in accordance to the actual Fertagus rolling-stock plan. From above, it can be concluded that unit k_1 is under maintenance between tasks 7 and 8 to carry out maintenance actions m_3 and m_{13}; unit k_3 performs activity m_6 between pair of tasks 23 and 24; and unit k_9 performs activity m_9 between pair of tasks 99 and 100.

Table 5.10 – Information on the unit and pair of tasks between each maintenance action is completed

$YM_{k,i,j,m}$		m													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
$Y_{k,i,j} = 1$	…	(…)													
	$Y_{1,7,8}$	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0
	…	(…)													
	$Y_{3,23,24}$	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0
	…	(…)													
	$Y_{9,99,100}$	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0

Finally, Tables 5.11 and 5.12 present information regarding the daily cost of employing a maintenance and/or a driving worker, respectively. While for the maintenance crew the value varies depending on the worker, for the drivers this value is constant.

Table 5.11 – Cost of employing a maintenance worker

mw	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Cmw_{mw}	43	43	43	46	40	40	43	43	43	43	43	36	66	60	66	60

Table 5.12 – Cost of employing a driving worker

dw	1	2	3	4	(…)	26	27	28	29
Cdw_{dw}	54	54	54	54		54	54	54	54

6 Results and Discussion

In chapter 6, the results of crew scheduling for the Fertagus case study are presented and analysed. The whole problem is here divided in two models and run separately, one for the maintenance crew scheduling and the other for the driving crew scheduling. In this way, results are also sectioned in two sections regarding the respective scheduling problems. It is also important to mention that both models were executed on the same computer, equipped with a memory of 8 GB (RAM), a 2.30 GHz processor and a 64-bit Operating System.

6.1 Maintenance Crew Scheduling

First of all, it is important to recall that the Fertagus maintenance crew members have different sets of skills. In reality, any worker is able to carry out almost every maintenance action, and thus, no such specific planning is done. Furthermore, and due to the adaptations made in this study, it does not make sense to draw a comparison of the results obtained with the actual scenario.

The maintenance crew scheduling model was executed for a specific day of a given week, adapting the scheduled maintenance actions from Mira (2018) model and using the actual Fertagus rolling-stock schedule. The minimum cost obtained for this problem was 454 monetary units as presented below in Figure 6.1.

```
The minimum total working cost is : 454
Begin running model
End running model
```

Figure 6.1 – Minimum cost obtained for maintenance crew concerning Fertagus case study

It is also possible to observe in Figure 6.2 that the computational time required to run this model is considerably small, 12.6 seconds more precisely. Consequently, and due to the fact that the output of this model is a daily schedule, if desired, it is also possible to obtain a weekly or even a monthly schedule in a reasonable time by modifying the inputs and running the model for each day.

```
Matrix:                            Presolved:
  Rows(constraints):   11954         Rows(constraints):   312
  Columns(variables):11440           Columns(variables):143
  Nonzero elements:  28976           Nonzero elements:   929
  Global entities:    3824           Global entities:     95
  Sets:               0              Sets:                0
  Set members:        0              Set members:         0
Overall status:  Finished global search.
LP relaxation:                        Global search:
  Algorithm:      Simplex primal      Current node:     17406
  Simplex iteration:224               Depth:            1
  Objective:      239.93              Active nodes:     0
  Status:         Unfinished          Best bound:       454
  Time:           0.0s                Best solution:    454
                                      Gap:              0%
                                      Status:           Solution is optimal.
                                      Time:             12.6s
```

Figure 6.2 – Case Study's model computational data

The fact that in a single day, only three different train units can go to the depot to perform maintenance, reduces the size of the problem significantly. In fact, this is one reason to the massive decrease of the number of variables after the pre-solving stage, resulting in this short computational time.

The results obtained for the maintenance crew scheduling are presented in a data file displayed in Figure 6.3. It is possible to observe that from the whole crew of 16 maintenance workers, only 10 are required to successfully carry out all maintenance actions and so, 6 of them are not assigned to work on this day. The maintenance actions performed by each worker, the respective unit, starting and finishing times are also presented here. Since the objective function focuses on minimizing the cost of employing workers, logically, the ones with an associated lower cost will be assigned, if they have the required competences.

```
[MAINTENANCE CREW]

Maintenance Worker #1 works on Day [1]
Maintenance Activity: (3); Unit: [1] Starting Hour (min): 778; Finishing Hour (min): 988
Maintenance Activity: (1); Unit: [9] Starting Hour (min): 623; Finishing Hour (min): 773

Maintenance Worker #2 works on Day [1]
Maintenance Activity: (3); Unit: [1] Starting Hour (min): 778; Finishing Hour (min): 988
Maintenance Activity: (1); Unit: [9] Starting Hour (min): 623; Finishing Hour (min): 773

Maintenance Worker #3 doesn't work on Day [1]

Maintenance Worker #4 doesn't work on Day [1]

Maintenance Worker #5 works on Day [1]
Maintenance Activity: (6); Unit: [3] Starting Hour (min): 624; Finishing Hour (min): 810
Maintenance Activity: (9); Unit: [9] Starting Hour (min): 815; Finishing Hour (min): 1001

Maintenance Worker #6 doesn't work on Day [1]

Maintenance Worker #7 works on Day [1]
Maintenance Activity: (6); Unit: [3] Starting Hour (min): 624; Finishing Hour (min): 810

Maintenance Worker #8 works on Day [1]
Maintenance Activity: (6); Unit: [3] Starting Hour (min): 625; Finishing Hour (min): 811
Maintenance Activity: (9); Unit: [9] Starting Hour (min): 816; Finishing Hour (min): 1002

Maintenance Worker #9 works on Day [1]
Maintenance Activity: (3); Unit: [1] Starting Hour (min): 778; Finishing Hour (min): 988
Maintenance Activity: (1); Unit: [9] Starting Hour (min): 623; Finishing Hour (min): 773

Maintenance Worker #10 works on Day [1]
Maintenance Activity: (13); Unit: [1] Starting Hour (min): 980; Finishing Hour (min): 1033
Maintenance Activity: (1); Unit: [9] Starting Hour (min): 623; Finishing Hour (min): 773
Maintenance Activity: (9); Unit: [9] Starting Hour (min): 789; Finishing Hour (min): 975

Maintenance Worker #11 doesn't work on Day [1]

Maintenance Worker #12 works on Day [1]
Maintenance Activity: (3); Unit: [1] Starting Hour (min): 770; Finishing Hour (min): 980
Maintenance Activity: (13); Unit: [1] Starting Hour (min): 980; Finishing Hour (min): 1033

Maintenance Worker #13 doesn't work on Day [1]

Maintenance Worker #14 works on Day [1]
Maintenance Activity: (13); Unit: [1] Starting Hour (min): 980; Finishing Hour (min): 1033
Maintenance Activity: (6); Unit: [3] Starting Hour (min): 624; Finishing Hour (min): 810

Maintenance Worker #15 doesn't work on Day [1]

Maintenance Worker #16 works on Day [1]
Maintenance Activity: (13); Unit: [1] Starting Hour (min): 980; Finishing Hour (min): 1033
Maintenance Activity: (1); Unit: [9] Starting Hour (min): 623; Finishing Hour (min): 773
Maintenance Activity: (9); Unit: [9] Starting Hour (min): 773; Finishing Hour (min): 959
```

Figure 6.3 – Results for the maintenance crew scheduling problem concerning Fertagus case study

For a faster visualization of the solution obtained, Table 6.1 was created. Integrated with the starting and finishing times previously presented above in the data file, Table 6.1. allows a broader overview of the solution, through a clearer schematic planning sectioned by unit.

Table 6.1 – Distribution of activities for maintenance worker

Maint. Crew	MAINT. ACTIONS													
	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}
	UNIT k_1													
mw_1			x											
mw_2			x											
mw_3														
mw_4														
mw_5														
mw_6														
mw_7														
mw_8														
mw_9			x											
mw_{10}													x	
mw_{11}														
mw_{12}			x										x	
mw_{13}														
mw_{14}													x	
mw_{15}														
mw_{16}													x	
	UNIT k_3													
mw_1														
mw_2														
mw_3														
mw_4														
mw_5						x								
mw_6														
mw_7						x								
mw_8						x								
mw_9														
mw_{10}														
mw_{11}														
mw_{12}														
mw_{13}														
mw_{14}						x								
mw_{15}														
mw_{16}														
	UNIT k_9													
mw_1	x													
mw_2	x													
mw_3														
mw_4														
mw_5									x					
mw_6														
mw_7														
mw_8									x					
mw_9	x													
mw_{10}	x								x					
mw_{11}														
mw_{12}														
mw_{13}														
mw_{14}														
mw_{15}														
mw_{16}	x								x					

Since it is previously known which train units go to the depot for maintenance on this day, for the sake of understanding, only those units are presented in Table 6.1.

It is important to note that while the same maintenance worker cannot perform two different maintenance activities simultaneously, there are cases where, for the same unit, two different activities are carried out at the same time. For instance, unit k_1, scheduled to perform maintenance actions m_3 and m_{13}, starts the latter before the initial one is completed. This has to do with the fact that from the four maintenance workers that carry out m_3, workers mw_1, mw_2, and mw_9, previously carry out maintenance on unit k_9 ending at **12h53**, and so, to change unit, t_{min} is required. In this way, workers mw_1, mw_2, and mw_9, can only start m_3 on unit k_1, 5 minutes later, at **12h58,** while mw_{12} starts at **12h50**. This means that mw_{12}, as the only worker of the referred ones executing both actions on unit k_1, finishes m_3 at **16h20** (12h50 + MT_{m3}) and starts m_{13} right away, while workers mw_1, mw_2, and mw_9 conclude m_3 at **16h28** (12h58 + MT_{m3}) and finish service.

Finally, while it would help understanding the context of the problem and its solution, it does not make sense to highlight the maintenance scheduling since each unit performing maintenance is continuously on the depot during that time. This way, each action may be performed on different times by different workers. Additionally, as it can be observed in the results, different maintenance workers can start the same maintenance activity m, with some minutes of interval, and so, that action m may have several starting and finishing times, as it happens for action m_3 executed on train unit k_1.

6.2 Driving Crew Scheduling

In the following subchapters, the results for the driver crew scheduling problem are presented and explained for several instances, namely the Fertagus case study and a medium size problem which includes part of the Fertagus actual rolling-stock schedule.

6.2.1 Fertagus Case Study

As stated before, the initial aim of the model regarding the driving crew was to output a solution for the whole Fertagus' rolling-stock fleet, i.e. for all of the 17 units. However, after running the model it was possible to observe that the results obtained do not present enough quality to be presented, i.e. there are some specific cases where the driver assignment do not replicate the desired solution, namely the coupling/decoupling of units.

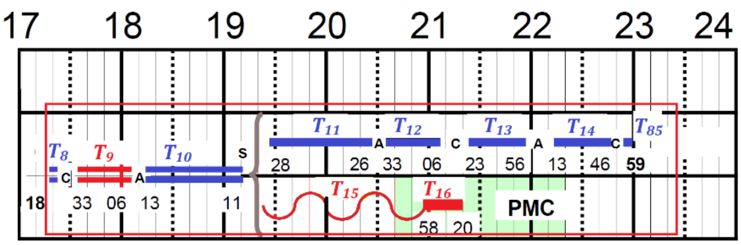


Figure 6.4 – Example of a non-optimal driving crew scheduling concerning a decoupling

To clarify this situation, Figure 6.4 is displayed, presenting an excerpt of the Fertagus actual rolling, where the assignment of the driving workers is not considered optimal. For instance, as it can be observed, there is a decoupling after T_{10}. Additionally, to carry out this set of tasks, two drivers are necessary, being that one of them dw_1 performs tasks presented in blue *(*T_8, T_{10}, T_{11}, T_{12}, T_{13}, T_{14}, T_{85}*)*, while dw_2 carries out the ones presented in red *(*T_9, T_5, T_{16}*)*. In this way, it is clear to see that, contrary to what is desired, drivers to not perform consecutive tasks. Something similar to this also occurs when two units are coupled, as it is displayed in Figure 6.5.

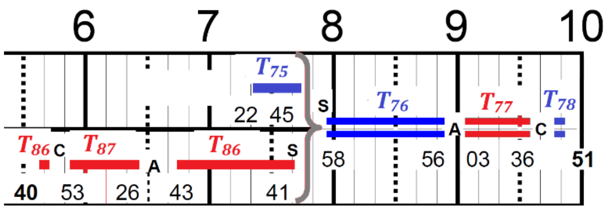


Figure 6.5 – Example of a non-optimal driving crew scheduling concerning a coupling

It is believed that the cause of this issue is related to constraints (22.1), (22.2), (23.1) and (23.2), which may not restrict the model enough or in the right way, so that these situations do not happen for a real-life and larger-scale scenario. Additionally, part of the pre-processing in section 3.7 also may be helpful to develop and implement in more specific constraints, namely parameters $Des_{i,j}$ and $Aco_{i,j}$. Unfortunately, since there was no time left, no further improvements were made in terms of the robustness, testing and viability, and so it is left for further research.

Furthermore, it is also important to mention that the computational time required to run this model was superior to 10 hours, making it non practical at all for large scale instances. Consequently, a medium size model problem was approached, containing part of the actual Fertagus rolling-stock schedule and its results are presented in the next section. Table 6.2 presents the calculus of the matrix columns for the Fertagus case study, which will logically be reduced in the next section.

Table 6.2 – Calculus of the matrix column

Set	Size	Set	Size	Size of the Columns
$k = \{1, \ldots, 17\}$	17	wDd_{dw}	29	$29 + 5684 + 5684 + 5684 = 17081$
$s = \{1, \ldots, 15\}$	15	$wDt_{dw,i}$	$29 * 196 = 5684$	
$i = \{1, \ldots, 196\}$	196	$eD_{dw,i}$	$29 * 196 = 5684$	
$dw = \{1, \ldots, 29\}$	29	$sD_{dw,i}$	$29 * 196 = 5684$	

6.2.2 **Medium size Fertagus Case Study**

In this section, a medium size scale study results are presented and described. It is important to note that some constant values and inputs had to be modified in order to recreate the rolling-stock schedule displayed below in Figure 6.6.

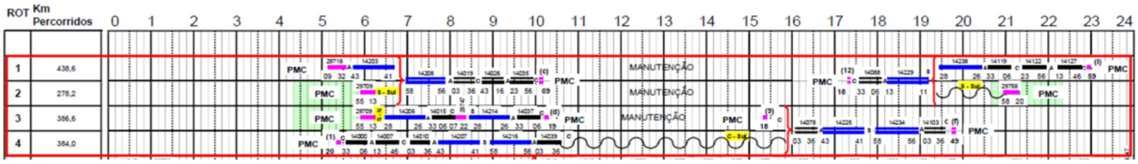


Figure 6.6 – Excerpt of the Fertagus rolling-stock schedule used in the medium size study

The total number of tasks was reduced to 84, with the attention that previously task T_{85}, is now T_{38}. All the other tasks remained unchanged matching the ones from the complete planning. Only 4 rolling stock units are considered and the driving crew is now formed by 8 members. The inputs related with the linkage between tasks by a given train unit, $Y_{k,i,j}$, the assignment of tasks to units, $X_{k,i}$, and the maintenance slot assigned to each unit, $YM_{k,i,j,m}$, were also reduced.

The data file obtained for the current case is displayed in Figure 6.7. At a first glance, it is possible to assess that only 7 driving workers are necessary to carry out all of the tasks, resulting in a total cost of 378 monetary units (Figure 6.8). Additionally, it can be observed that, as desired, drivers successfully carry out consecutive tasks, without changing units.

```
Driver Worker #1
Driving Worker [1] works on this day.
Task8;[1038-1040]; Unit: (1); Entry Hour (min):1023 (9); Exit Hour (min):0 (10)
Task8;[1038-1040]; Unit: (2); Entry Hour (min):1023 (9); Exit Hour (min):0 (10)
Task9;[1053-1086]; Unit: (1); Entry Hour (min):0 (10); Exit Hour (min):0 (1)
Task9;[1053-1086]; Unit: (2); Entry Hour (min):0 (10); Exit Hour (min):0 (1)
Task10;[1093-1151]; Unit: (1); Entry Hour (min):0 (1); Exit Hour (min):0 (15)
Task10;[1093-1151]; Unit: (2); Entry Hour (min):0 (1); Exit Hour (min):0 (15)
Task11;[1168-1226]; Unit: (1); Entry Hour (min):0 (15); Exit Hour (min):0 (1)
Task12;[1233-1266]; Unit: (1); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task13;[1283-1316]; Unit: (1); Entry Hour (min):0 (10); Exit Hour (min):0 (1)
Task14;[1333-1366]; Unit: (1); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task38;[1377-1379]; Unit: (1); Entry Hour (min):0 (10); Exit Hour (min):1394 (9)

Driver Worker #2
Driving Worker [2] works on this day.
Task1;[309-332]; Unit: (1); Entry Hour (min):294 (9); Exit Hour (min):0 (1)
Task2;[343-401]; Unit: (1); Entry Hour (min):0 (1); Exit Hour (min):0 (15)
Task3;[418-476]; Unit: (1); Entry Hour (min):0 (15); Exit Hour (min):0 (1)
Task3;[418-476]; Unit: (2); Entry Hour (min):0 (15); Exit Hour (min):0 (1)
Task4;[483-516]; Unit: (1); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task4;[483-516]; Unit: (2); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task5;[523-556]; Unit: (1); Entry Hour (min):0 (10); Exit Hour (min):0 (1)
Task5;[523-556]; Unit: (2); Entry Hour (min):0 (10); Exit Hour (min):0 (1)
Task6;[563-596]; Unit: (1); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task6;[563-596]; Unit: (2); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task7;[607-609]; Unit: (1); Entry Hour (min):0 (10); Exit Hour (min):624 (9)
Task7;[607-609]; Unit: (2); Entry Hour (min):0 (10); Exit Hour (min):624 (9)

Driver Worker #3
Driving Worker [3] works on this day.
Task37;[637-962]; Unit: (4); Entry Hour (min):622 (10); Exit Hour (min):977 (10)

Driver Worker #4
Driving Worker [4] works on this day.
Task24;[918-920]; Unit: (3); Entry Hour (min):903 (9); Exit Hour (min):0 (10)
Task25;[963-996]; Unit: (3); Entry Hour (min):0 (10); Exit Hour (min):0 (1)
Task25;[963-996]; Unit: (4); Entry Hour (min):0 (10); Exit Hour (min):0 (1)
Task26;[1003-1061]; Unit: (3); Entry Hour (min):0 (1); Exit Hour (min):0 (15)
Task26;[1003-1061]; Unit: (4); Entry Hour (min):0 (1); Exit Hour (min):0 (15)
Task27;[1078-1133]; Unit: (3); Entry Hour (min):0 (15); Exit Hour (min):0 (1)
Task27;[1078-1133]; Unit: (4); Entry Hour (min):0 (15); Exit Hour (min):0 (1)
Task28;[1143-1176]; Unit: (3); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task28;[1143-1176]; Unit: (4); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task29;[1187-1189]; Unit: (3); Entry Hour (min):0 (10); Exit Hour (min):1204 (9)
Task29;[1187-1189]; Unit: (4); Entry Hour (min):0 (10); Exit Hour (min):1204 (9)

Driver Worker #5
Driving Worker [5] works on this day.
Task17;[355-373]; Unit: (2); Entry Hour (min):340 (9); Exit Hour (min):0 (15)
Task17;[355-373]; Unit: (3); Entry Hour (min):340 (9); Exit Hour (min):0 (15)
Task18;[388-446]; Unit: (3); Entry Hour (min):0 (15); Exit Hour (min):0 (1)
Task19;[453-486]; Unit: (3); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task20;[487-502]; Unit: (3); Entry Hour (min):0 (10); Exit Hour (min):0 (15)
Task21;[508-566]; Unit: (3); Entry Hour (min):0 (15); Exit Hour (min):0 (1)
Task22;[573-606]; Unit: (3); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task23;[617-619]; Unit: (3); Entry Hour (min):0 (10); Exit Hour (min):634 (9)

Driver Worker #6
Driving Worker [6] works on this day.
Task30;[320-322]; Unit: (4); Entry Hour (min):305 (9); Exit Hour (min):0 (10)
Task31;[333-366]; Unit: (4); Entry Hour (min):0 (10); Exit Hour (min):0 (1)
Task32;[373-406]; Unit: (4); Entry Hour (min):0 (1); Exit Hour (min):0 (10)
Task33;[423-456]; Unit: (4); Entry Hour (min):0 (10); Exit Hour (min):0 (1)
Task34;[463-521]; Unit: (4); Entry Hour (min):0 (1); Exit Hour (min):0 (15)
Task35;[538-596]; Unit: (4); Entry Hour (min):0 (15); Exit Hour (min):0 (1)
Task36;[603-636]; Unit: (4); Entry Hour (min):0 (1); Exit Hour (min):651 (10)

Driver Worker #7
Driving Worker [7] works on this day.
Task15;[1152-1257]; Unit: (2); Entry Hour (min):1137 (15); Exit Hour (min):0 (15)
Task16;[1258-1280]; Unit: (2); Entry Hour (min):0 (15); Exit Hour (min):1295 (9)

Driver Worker #8
Driving Worker [8] doesn't work on this day.
```

Figure 6.7 – Results for the driving crew scheduling problem concerning Fertagus medium size case study

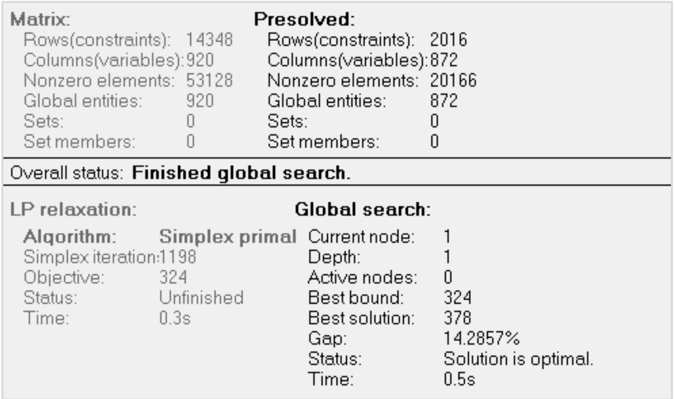


Figure 6.8 – Computational data concerning medium size case study's model

The computational time required to run this model is less than a 1 second, although logically it is expected that for larger instances this value increases significantly, as it happened for the whole real case scenario mentioned in the previous section. Additionally, the optimization gap obtained is 14%, meaning that the best bound uses one driver less than the best solution obtained. Nevertheless, it should be equal to zero, as the Status field show a message "Solution is optimal."

Moreover, in line to what was done for the maintenance crew in Table 6.1, the results for this driving crew scheduling problem are condensed in a Table presented in Appendix A1, due to its size. For an easier reading of the table, each driver as a different colour, so that it is easier for each one of them to assess which tasks they were assigned to. In addition, it is advised to consult these results with the help of the rolling-stock schedule for a better understanding of the planning.

Even though it is tempting to draw comparisons, it is important to remind that the present model has some real world aspects and constraints that were not considered, such as, the existence of unit changes, the drivers' rest time, the possibility of trips off service to reach a certain destination, the fact that entry and exit stations should match and even the fact that reserve tasks may be carried out by several drivers, although here every task is defined to be performed by a single driving worker. Some of these limitations will be assessed on section 7.2, and in the future, these aspects may be implemented in future researches.

A **sensitivity analysis** of the weight associated with maintenance and driving workers employment cost in the objective function is not carried out since these are values that have the same units, and both were established by the Fertagus company. Moreover, as it could be noticed, the computational time required to run both models is really low, so there is no point in presenting an **optimality gap analysis** for neither case. However, it would be possible to run models for an increasing number of units and study the evolution of the computational time and the respective optimality gap.

7 Conclusion

In this last chapter, the contributions of the study are provided, the limitations identified and some aspects which may be enhanced are pointed out and left for further research. It is always kept in mind the objective of clearly clarifying what was possible to achieve and what may be used as groundwork for further work.

7.1 Contributions

Once the analysis of results is concluded, it is now time to assess the whole conception of this study, walkthrough it and expose what was achieved with success (and what was not). The problems and setbacks encountered during the creation of the model are exposed in next section 7.2.

Following the reviewed work, with a higher focus on Mira (2018) dissertation, it was decided to follow some thoughts exposed on his future research section, namely the *“crew scheduling that takes into account the different skills of maintenance technicians”*. This idea was in fact the first main objective for the present dissertation: a maintenance crew scheduling model that considers different skillsets for each worker and that could be applied to a train operating company, Fertagus. In later stages of the conception of such model, it was also suggested to integrate the company’s driver crew scheduling with the previous maintenance crew plan, so that a broader model would provide an optimal daily schedule concerning both crews.

In this way, the main goal of this dissertation was to conceive a model that would provide an optimal daily crew scheduling plan for a train operating company (e.g. Fertagus), while minimizing the costs associated with workers’ employment. Information related to maintenance had to be gathered from Mira (2018) and Méchain (2017) models and dissertations, as the current work is in fact a continuation of both of these works, as explained in Figure 3.1. Moreover, some ideas implemented regarding the crew competences were inspired on M. Pour *et al.* (2018), and so it is also noteworthy to mention. Actually, to the best of our knowledge, it was not found a mathematical model that executes the maintenance crew scheduling with a skillset associated to each worker, across all the reviewed work and published work. This constitutes one of the main contributions of the present dissertation, an innovative maintenance crew scheduling that takes into account the workers’ different set of competences, as Mira (2018) suggested for future research.

The mathematical model conceived, built from scratch, follows a mixed integer linear programming formulation. The main goal was partially accomplished as, though the problem concerning Fertagus case study was successfully solved for the maintenance crew scheduling, no robust results were obtained for the driving crew scheduling problem. While the focus of this study is the Fertagus company and its specific conditions, the model is flexible enough to be adapted to different instances. To do that, one just needs to modify the inputs and some of the constraints related to infrastructure.

For the maintenance crew, it was possible to create and successfully solve the model for the Fertagus case study and therefore respect its constraints, some of them previously defined by Mira (2018) and Méchain (2017), and integrated in some of the inputs of the current study.

Regarding the driving crew problem, models were run for large instances. However, problems could be found, specifically for the decoupling and coupling of units. Since there was no time left, no further model improvements could be carried out, and thus, it is left for further research the enhancements and/or additions of some parameters in the mathematical model concerning these larger scale scenarios. Some of these limitations are presented next in section 7.2. On the other hand, a model was conceived outputting an optimal driving crew schedule for a medium-size problem, containing part of the actual Fertagus rolling-stock schedule. This medium-size driving crew scheduling model is another contribution of this research. It can even be used as groundwork for future work, keeping in mind the need to solve the identified problems concerning the units' coupling and decoupling.

Additionally, the computational time required to run both models is significantly small, as expected, since the output consists on a daily schedule. In this way, if desired, it is possible to obtain a weekly schedule in a practical time by running the model for each day of the respective week. No sensitivity analysis on the weight of the workers' employment costs was made, as these values are established by Fertagus company and are not meant to be modified. Altering their values would mean changes on workers' wages, and logically, as the intention of the model is to minimize costs, these changes would mean a lower income for crew members. Finally, an optimality gap analysis was also not carried out since the computational time necessary to run the models is not large enough to make it relevant.

All in all, it was possible to accomplish most of the proposed objectives by creating a mixed integer linear programming model that was conceived during a long and iterative process of restricting it enough, but not too much so that non-feasible solutions would be outputted. A lot of difficulties occurred, the program was either too restrictive or not restrictive enough, many unfeasible and non-optimal solutions were found. In the end, the mathematical model is considered to answer its purpose and solve to optimality some real instances concerning crew scheduling for a train operating company.

7.2 Limitations

As mentioned along this dissertation, many difficulties occurred during the conception of the model, some of which were not possible to overcome and so, are considered limitations, which are pointed out next.

Firstly, one of the major limitations of this model is the fact that it is very dependent on the user inputs, i.e. the inputs defined by the user must accurately characterize the real-world situation. Otherwise, even if a solution is obtained, it is hardly optimal as it does not provide an answer to the real instance. This means that all data must be precisely collected and meticulously introduced in the model, so that the output does not mislead the user.

Secondly, since the objective function is a cost-minimization linear function, only focused on the financial variables related with the workers' wages, which is previously established by the company, no sensitivity analysis on the weight of these parameters was carried out. Additionally, as the computational time required to run the model is generally reduced, it did not make sense to do an optimality gap analysis function of time, as it is not large enough.

Moreover, concerning the maintenance crew scheduling problem, some of the constraints and limitations were beforehand defined by Mira (2018) and Méchain (2017). For example, the number of units that can simultaneously stay in the depot for maintenance is limited to 3, therefore the size of this Fertagus case study is somehow limited and no larger scale instances were analysed.

Lastly, there are also several limitations concerning the driving crew scheduling problem. The first one is related to the conception of a model that outputs an optimal solution for the Fertagus case study, as only a medium-size problem was successfully solved and its solution validated. In line with this, situations where units couple or decouple might have some incoherencies and so it is a matter left for further research, mentioned next in section 7.3. Furthermore, comparisons and analysis could not be made with the real instance, since some of the real constraints and limitations were not taken into account, such as: the drivers' rest time; the entry station for a given driver matching with the exit station; trips off service to reach a certain destination; drivers changing unit, and the fact that there are some tasks, namely reserve tasks (characterized by black wavy lines on the Fertagus rolling-stock schedule), which can actually be carried out by several drivers, while for the current study, all tasks can only use one single driving worker. Finally, one last aspect important to refer is the value of the optimality gap obtained for the driving crew scheduling. Contrary to what happened in the maintenance crew scheduling, this value is different from zero, which as an indicator of optimality of the solution, may constitute another limitation, depending on the importance given by the decision maker to this parameter.

7.3 Future Research

A major improvement to this study would be the proposal of an approach to overcome many of the limitations exposed in section 7.2.

Due to some limitations presented above, the initial model for the whole crew scheduling had to be split in two, so that it was possible to study the maintenance crew scheduling for the Fertagus case study and the driving crew scheduling concerning a medium-size problem, part of the actual Fertagus instance. In this way, it would make sense to integrate both models in a single one, so that all the crew scheduling can be obtained by running a single model. Moreover, the output obtained is a daily schedule. As the computational time is really low, it is possible to obtain a weekly schedule if the model is run for several days. However, it is thought it may be useful to model a program that is able to output a weekly schedule in a single run. Logically this would result in a significantly increase of the number of variables and consequently, computational time. Implementing data in the pre-processing section may be useful, as all the parameters which do not respect the established requirements are beforehand eliminated, leading to a computational time decrease.

As stated before, the maintenance crew scheduling problem is a continuation of researches carried out by Mira (2018) and Méchain (2017), and thus, some limitations were already established beforehand. That is the case for the number of units that can go to depot on a single day, which is limited to three. However, for larger train companies, this number may be much larger and so the application of this model to a larger maintenance instance is suggested. Additionally, even though in this dissertation maintenance actions do not have any type of relation between them, it is usual that, for other instances, maintenance activities are related and for example, one action cannot be carried out while a different one is not finished. This kind of specific relations may be implemented, if that occurs for the studied case. Moreover, as future work, it might be interesting to define and obtain values for the amount of work per competence carried out by each maintenance worker, i.e. skills would be interpreted as *"sub-tasks"* executed by maintenance workers, so that the model would output, the time spent by each worker for each competence. This way it would be possible to analyse the most and least required skills, and so, a specialization of the working crew could be carried out.

The main suggested improvement, however, is related with the coupling and decoupling of units, understandably concerning the drivers scheduling problem. It is believed that by solving this detail it would be possible to assign drivers in a more efficient way and so optimal solutions for even larger cases than the present Fertagus case study could be obtained. Furthermore, some managerial, temporal and infrastructure restrictions were not considered, such as, drivers' rest time, the coherence between entry and exit stations, along with some others pointed out in section 7.2. In this way, it is suggested for future work, the application of some constraints related with this, so that a real instance is better characterized in the mathematical model.

8 References

Bazargan, M. (2015) 'An optimization approach to aircraft dispatching strategy with maintenance cost - A case study', *Journal of Air Transport Management*. Elsevier Ltd, 42, pp. 10–14. doi: 10.1016/j.jairtraman.2014.07.008.

Boyer, V., Ibarra-Rojas, O. J. and Ríos-Solís, Y. (2018) 'Vehicle and Crew Scheduling for Flexible Bus Transportation Systems', *Transportation Research Part B: Methodological*. Elsevier Ltd, 112, pp. 216–229. doi: 10.1016/j.trb.2018.04.008.

Fuentes, M., Cadarso, L. and Marín, Á. (2018) 'A Fix & Relax matheuristic for the Crew Scheduling Problem', *Transportation Research Procedia*. Elsevier B.V., 33, pp. 307–314. doi: 10.1016/j.trpro.2018.10.107.

Haghani, A. and Shafahi, Y. (2002) 'Bus maintenance systems and maintenance scheduling: Model formulations and solutions', *Transportation Research Part A: Policy and Practice*, 36(5), pp. 453–482. doi: 10.1016/S0965-8564(01)00014-3.

Kang, L., Chen, S. and Meng, Q. (2019) 'Bus and driver scheduling with mealtime windows for a single public bus route', *Transportation Research Part C: Emerging Technologies*. Elsevier, 101(February), pp. 145–160. doi: 10.1016/j.trc.2019.02.005.

M. Pour, S. *et al.* (2018) 'A hybrid Constraint Programming/Mixed Integer Programming framework for the preventive signaling maintenance crew scheduling problem', *European Journal of Operational Research*. Elsevier B.V., 269(1), pp. 341–352. doi: 10.1016/j.ejor.2017.08.033.

Martins, R. A. (2018) 'Planning maintenance in bus operating companies CARRIS case study', (June).

Méchain, M. (2017) 'Optimization of maintenance actions in train operating companies - Fertagus case study', (June), p. 44.

Mira, L. (2018) *Optimization of Maintenance and Availability in a Train Operating Company - Fertagus case study*.

Valouxis, C. and Housos, E. (2002) 'Combined bus and driver scheduling', *Computers and Operations Research*, 29(3), pp. 243–259. doi: 10.1016/S0305-0548(00)00067-8.

UN (2018), United Nations Revision of World Urbanization Prospects. Retrieved from: https://population.un.org/wup/

EC (2019), European Commission Rail Research and Shift2Rail.Retrieved from: https://ec.europa.eu/transport/modes/rail/shift2rail_en

S2R (2019), Shift2Rail History of the Initiative. Retrieved from: https://shift2rail.org/about-shift2rail/history-of-the-initiative/

S2R AI (2019), Shift2Rail Artificial Intelligence for the Railway Sector. Retrieved from: https://www.up2europe.eu/calls/artificial-intelligence-a-i-for-the-railway-sector-s2r-oc-ipx-01-2019_3451.html

Carris (2017) Retrieved from Carris: http://www.carris.pt/pt/indicadores-de-atividade/

Metro (2017), Metropolitano de Lisboa. Retrieved from Carris: https://www.metrolisboa.pt/institucional/conhecer/

Transtejo-Soflusa (2017) Retrieved from Transtejo-Soflusa: https://ttsl.pt/terminais-e-frota/

R&C CP (2017), Relatório e Contas 2017, Comboios de Portugal. Retrieved from Comboios de Portugal: https://www.cp.pt/StaticFiles/Institucional/1_a_empresa/3_Relatorio_Contas/2017/relatorio-contas-2017.pdf

Fertagus (2019) Retrieved from Fertagus: https://www.fertagus.pt/pt/teste---quem-somos/quem-somos

FICO (2019), Fico-Xpress-Solver. Retrieved from Fico: https://www.fico.com/en/products/fico-xpress-solver

R&C Fertagus (2018), Relatório e Contas 2018, Fertagus. Retrieved from Fertagus: https://www.fertagus.pt/Portals/0/RelatorioContas/RC%20Fertagus%202018-low.pdf?ver=2019-05-06-092042-613&timestamp=1557134817563

9 Appendix

A1 – Driving Crew Scheduling of the medium size Fertagus case study

Driving Crew	TASKS																																				
	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}	T_{19}	T_{20}	T_{21}	T_{22}	T_{23}	T_{24}	T_{25}	T_{26}	T_{27}	T_{28}	T_{29}	T_{30}	T_{31}	T_{32}	T_{33}	T_{34}	T_{35}	T_{36}	T_{37}	T_{38}
UNIT k_1																																						
dw_1								x	x	x	x	x	x	x																								x
dw_2	x	x	x	x	x	x	x																															
dw_3																																						
dw_4																																						
dw_5																																						
dw_6																																						
dw_7																																						
dw_8																																						
UNIT k_2																																						
dw_1								x	x	x																												
dw_2			x	x	x	x	x																															
dw_3																																						
dw_4																																						
dw_5																	x																					
dw_6																																						
dw_7															x	x																						
dw_8																																						
UNIT k_3																																						
dw_1																																						
dw_2																																						
dw_3																																						
dw_4																								x	x	x	x	x	x									
dw_5																	x	x	x	x	x	x	x															
dw_6																																						
dw_7																																						
dw_8																																						
UNIT k_4																																						
dw_1																																						
dw_2																																						
dw_3																																					x	
dw_4																								x	x	x	x	x	x									
dw_5																																						
dw_6																														x	x	x	x	x				
dw_7																																						
dw_8																																						