Computational Experiments of a Maintenance Scheduling Problem

Carris bus company Case Study

Francisco Manuel de Lemos Custódio Fernandes

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Prof. António Ramos Andrade

Prof. José Maria Campos da Silva André

Examination Committee

Chairperson: Prof. Carlos Frederico Neves Bettencourt da Silva

Supervisor: Prof. António Ramos Andrade

Member of The Committee: Prof. Cristina Marta Castilho Pereira Santos Gomes

August 2019

Acknowledgements

First of all, I would like to thank my dissertation supervisors, Prof. António Ramos Andrade and for his support and availability to aid me with all my questions and concerns. His contribution and his guidance were truly essential for the concretization of this work. And Prof. José Maria André for accepting to participate in this project.

Secondly, I would like to thank Rodrigo Arrais Martins for his previous work which represented the ground stone for this dissertation.

I am thankful to my colleagues António Amaral, Francisco Monteiro, João Alemão, João Barata, Miguel Tavares and Nuno Mendes for their support and companionship along this though road.

I am also grateful to FICO Xpress software for providing an academic license of FICO Xpress Optimization software, that made possible the optimization process and the achievement of results.

Moreover, I should recognize the preparation, of almost six years, that was provided by my university, Instituto Superior Técnico, which allowed me to develop personally and technically and led me to the concretization of this project.

Last but not least, I would like to express my gratitude to my family whose patience and support were critical during this phase of my life. And to my friends who were always there.

Abstract

In 2018, Rodrigo Arrais Martins developed a Mixed Integer Linear Program model that was implemented in FICO Xpress software, in order to optimize the bus maintenance scheduling of a bus operating company with the goal of reducing its maintenance costs. The results obtained, though improving the company's current schedule, were no great regarding the optimality of the solution and the computational time it took to reach it. The present dissertation searches a way to make improvements in those aspects.

A parallel solving multiple model approach based on the Dantzig-Wolfe decomposition was first attempted, resulting in the impossibility to generate results.

Then an alteration to the original model by introducing new restrictions, in order to guide the solver to the solution, was implemented. The results regarding computational time showed great enhancement to the original model, but the improvement in terms of optimality was scarce.

Lastly, a heuristic approach, in which the problem was solved sequentially for one bus at a time, was developed. This model showed great improvements such in computational time as in optimality. Showing a reduction of 99.7% in computational time and 8.9% in maintenance costs.

Both the heuristic approach and the alteration to the original model were validated through an illustrative example.

Keywords: Optimization, Maintenance Scheduling, Computational Experiment, Bus transport, Mixed Integer Linear Programming, Parallel Solving

Resumo

Em 2018, Rodrigo Arrais Martins desenvolveu um modelo de Programação Linear Inteira Mista implementado em FICO Xpress, com o objectivo de optimizar o planeamento da manutenção numa empresa operadora de autocarros, ultimamente reduzindo os custos de manutenção. Os resultados obtidos, apesar de apresentarem melhorias em relação às práticas da empresa, deixaram a desejar em termos de optimalidade da solução e tempo de computação. A presente dissertação procura melhorar esses aspectos.

Uma abordagem de resolução em paralelo, utilizando multiplos modelos e baseada na decomposição Dantzig-Wolfe, foi tentatada não tendo sido conseguido que o modelo gerasse resultados,

Seguidamente, foi implementada uma alteração ao modelo original, onde se introduziram novas restriçóes de forma a conduzir o modelo para a solução. Os resultados em termos de tempo computacional foram bastante satisfatórios, havendo pouca melhoria no valor da solução.

Por ultimo, foi desenvolvida uma abordagem heurística do problema, em que o planeamento de cada autocarro é resolvido sequencialmente. Este modelo apresentou grandes melhorias à solução original, apresentando uma redução de 99.7% em tempo computacional e de 8.9% na redução de custos.

A abordagem heurística assim como a alteração ao modelo original, foram validadas através de um exemplo ilustrativo.

Keywords: Optimization, Maintenance Scheduling, Computational Experiment, Bus transport, Mixed Integer Linear Program, Parallel solving

Table of Contents

List of Figures

List of Tables

List of Acronyms

GIS	Geographical Information System
MCS	Monte Carlo Simulation
LP	Linear Programming
ILP	Integer Linear Programming
MILP	Mixed Integer Linear Programming
VSM	Vehicle Scheduling Maintenance

List of Indices, Constants and Sets

b bus

c competence

d day

m maintenance type

t time period

v vehicle type

w worker

B set of buses

C set of competences

D set of days

M set of maintenance types

T set of time periods

V set of vehicle types

W set of workers

TO_b set of time periods in which maintenance activities cannot occur for bus b (e.g. time periods for operation)

TM_b set of time periods in which maintenance activities can occur for bus b (e.g. time periods for maintenance)

cc_w competence of worker w

cD_w daily cost of worker type w

g_{bmvc} amount of work that bus b needs to perform maintenance type m, in vehicle type v, with competence c

nw_c number of workers with competence c

v_b vehicle type of bus b

cU bus unavailability cost

nd number of days

nml number of maintenance lines

nt number of time periods

nw number of workers

nsl number of special lines (exclusive to articulated buses)

ntd number of time periods in a day

L large number

Gb_{NBus} set of buses in which they are ordered

1. Introduction

This first chapter provides an introduction on the research topic of the present dissertation. First, a context on the role of public transport and especially on bus transport is given. Then the research objective of this dissertation is introduced. And, finally, the structure of the document is presented.

1.1. Context

1.1.1. Public transport

Public Transport has always been seen as a solution, both for the environmental problems surrounding the cities' increasing pollution by CO2 emissions and the overwhelming traffic that we see nowadays in most large metropolis. As the Earth's population grows, the urban population grows along, since, according to the European Commission (EC, 2016), *"(...) 50% of the world's population lives in cities"*. This organization also states that "*they are responsible for three-quarters of the global energy consumption as well as approximately 80% of the global greenhouse gas emissions"*. Looking at the speedy rate that the population is growing (70% in four decades according to the United Nations), it is imperative that there are some changes and adaptations in the way we move, specially within our cities. There is a need to disrupt the predominance of the private car as the preferred choice of transportation, besides urban congestion, it is responsible for the emission of gases and the increase in noise pollution.

The solution to this problem is not to expand the road network; this strategy will only result in more traffic in the big cities, so other alternatives must be looked upon and conditions for a shift towards more sustainable forms of transportation must be provided. Therefore, the improvement and optimization of the public transports' operations will play a key role in facing the changes caused by the population and urban concentration growth.

Opportunities to use public transport as a lever towards the evolution of the use of energy in mobility are already being explored. The European Commission (EC, 2011) stated that "the objective for the next decade is to create a genuine single European transport area by eliminating all residual barriers between modes and national systems, easing the process of integration and facilitating the emergence of multinational and multimodal operators."

The focus of this dissertation is bus transportation, whose importance in urban mobility is undisputed, without reducing the significance of other transports (train, tram, metro, ships, ...) within the overall mobility system, towards a more efficient and sustainable mobility system in urban areas.

1.1.2. The importance of bus maintenance

Lisbon as a city faces some challenges, such as a decrease in air quality and an increase in the number of cars in circulation each day. There are about 370.000 cars per day in the centre of Lisbon. CARRIS bus operating company oversees the bus service in Lisbon and, according to the National Statistics Institute 2017 Inquiry, it is responsible for around 10% of the city's trips.

With the purpose of improving the quality, reliability, safety and life of the vehicle, and thus improving the buses' competitiveness against other solutions, CARRIS bus operating company depends on the service provided by its maintenance department. The bus maintenance can be divided into three categories: i) daily inspection, ii) corrective maintenance and iii) preventive maintenance.

Daily inspections are performed at the end of each daily run, by the driver or the operator responsible for parking the buses in the depot. The problems are recorded and, if necessary, the buses are immobilized for observation of the corrective maintenance team. Corrective maintenance occurs when buses are unexpectedly damaged. The buses sometimes need to be replaced for the remaining hours of their schedule, resulting in large additional expenses, in extreme cases, the corrective maintenance team must stop the activities to repair another vehicle. Preventive maintenance covers regular inspections for pre-specified mileage or time intervals. These inspections are scheduled as general interventions, and some worn out components can be replaced. The problem under analysis will focus on preventive maintenance and it will be provided a way to build a maintenance schedule for preventive maintenance.

1.2. Research Objective

The present dissertation follows a previous work conducted by my former colleague Rodrigo Arrais Martins who addressed this problem in 2018. As stated by Martins (2018), the objective of his dissertation was to develop a decision model that was able to minimize the cost of maintenance by bus companies; specifically, his goal was to create a technical maintenance plan that would minimize the total cost of preventive maintenance. The developed maintenance scheduling model was then adapted to the CARRIS, a Lisbon-based bus operating company, case study.

The following secondary objectives were also investigated to address the previous issue:

- Increase the availability of the buses;
- Increase the percentage of occupation of the facility;
- Sensitivity analysis of the cost of bus unavailability;
- Identify the resource limit between the number of elements in a maintenance crew and the size of the facility;
- Evaluation of the optimal maintenance plan as a support to the decision maker.

As it will be discussed in chapter 3, the results achieved by Martins proved that his approach was able to achieve a feasible solution for this type of problems, but it presented a large optimality gap, which is the relative difference between a best known solution and a value that bounds the best solution possible, for a long computational time. Therefore, the goal of this dissertation is to improve those results conducting several computational experiments with various approaches and trying different solving mechanisms, namely by restricting more the problem under analysis. In order to achieve this objective, three different methods were tried:

- The Dantzig-Wolfe decomposition (parallel solving);
- Heuristic approach;
- Adding constraints to the original problem;
- These will be the targets of this investigation, and the results of this research work will be provided and analysed throughout the document.

In order to achieve these goals, the following steps were pursued:

- Literature review on parallel solving mechanisms;
- Development of a new decision model based on parallel solving;
- Development of a heuristic approach to solve approximately the problem;
- Improvement of the Martins' model through restriction introduction;
- Comparison between the results obtained with the Martins' and the improved model;
- Conclusions, limitations and future research;

1.3. Document Structure

The present dissertation is structured in five chapters:

1. Introduction - The first chapter introduces the context of the problem, the importance of buses and public transportation, and the basis of this research. The objectives and methodology are presented and finally, the dissertation structure is outlined.

2. Related work – State of the art - In chapter 2, a summary of the most relevant papers, which were studied during this dissertation, is presented. It introduces the bus maintenance scheduling area, but it also provides a brief review of some of the parallel solving and improving options related to the goal of this project.

3. Martins model (2018), heuristic approach, Danztig-Wolfe decomposition and restriction introduction - Chapter 3 explores the methodology and the associated implementation. Section 3.1 provides a presentation of the Martins original model used as a basis of study for this project, and an illustrative example validating the model is given in section 3.2. Section 3.3 introduces the different approaches addressed in this dissertation. Section 3.3.1 describes briefly how the implementation of the model in the Mosel language of the FICO® Xpress Optimization software was made. In section 3.3.2, a heuristic approach is explained, and its implementation is briefly described. Lastly, in section 3.3.3 it is presented an approach where new restrictions to the original problem are introduced. An illustrative example is presented for all the approaches which were validated.

4. Computational experiments and model implementation (CARRIS case study) - In chapter 4, the experiments made to improve the previous models and its implementation and adaptations are discussed. In section 4.1, an exhaustive breakdown of the computational experiments made with the Dantzig-Wolfe problem is presented. Then, in section 4.2, the heuristic approach and its results are analysed. Section 4.3 provides a thorough analysis of the “introducing new restrictions to the original problem” approach. Finally, in section 4.4, the results of the two last approaches are compared between them and with the previously achieved results for the case study. The results are discussed afterwards.

5. Conclusion, Limitations and Further Research - this final chapter provides the conclusions of the research, identifies some limitations and points out further steps of improvements and enhancements to the research here conducted.

2. Related work – State of the Art

This dissertation follows the research conducted by Rodrigo Arrais Martins in 2018 and represents a continuation of his work. After a careful research, no relevant and recent articles on the subject addressed by Martins (2018) were found. Therefore, the research conducted in the present document also relies on his state of the art on topics such as: the bus maintenance planning and scheduling, and maintenance optimization in transport systems. This section will summarize the contribution of these articles and other relevant ones related to the computational experiments conducted in this dissertation.

2.1. Bus maintenance planning and scheduling, and maintenance optimization in transports

Bus maintenance and scheduling

Haghani and Shafani (2002) focused on finding a way to respond to the problem of scheduling bus maintenance. Based on bus operation schedules, maintenance and inspection needs, their goal is to design the daily supervision for buses that should be inspected, mostly during their idle time, to reduce the number of hours the vehicle is out of service, i.e. reduce unavailability. The optimization program suggests a solution to this problem, which returns the maintenance schedule for each bus that should be inspected, as well as the minimum number of maintenance lines that are used for each type of inspection, during the scheduled period.

Haghani et al. (2003) studied the feasibility and cost of three scheduling models of bus transit vehicles (Haghani, et al., 2003). The first two models used a single depot and the others multiple depots. All models were analysed using two factors: i) deadhead speed and ii) the maximum allowed block time. They tested several objective functions: first objective was to minimize the number of vehicles (which means reducing the fixed investment costs in the fleet cost of capital), the second was to decrease the total deadhead time and cost of operations, and the third was to decrease the combination of two previous objectives.

Zhou et al. (2004) proposed a multi-agent system model to solve a bus maintenance scheduling problem (Zhou, et al., 2004). The problem was divided into predictive and reactive scheduling. The optimized schedule was performed for all agents, with mutual collaboration between agents. This article provided relevant information for the construction of the constraints of the model proposed by Martins (2018).

Adonyi et al. (2013) developed a solution for the bus maintenance planning problem in public transportation (Adonyi, et al., 2013). In their model, it is ensured that there are enough buses available for the scheduled service and that maintenance and repair tasks can be applied in the bus's downtime during its service day, and thus avoiding maintenance only at night (which entails higher cost per hour). The model also manages to reduce maintenance costs because buses will only be repaired if required.

Kamlu and Laxmi (2016) considered that the Vehicle Scheduling Maintenance (VSM) could be affected by these three aspects: different terrain, amount of mileage and variable load and associated uncertainties (Kamlu & Laxmi, 2016). The VSM is designed through a fuzzy model, and the three previous aspects are used as a basis. The fuzzy model combines the uncertainties associated with each of the three factors to arrive at the best VSM and is also able to predict in advance which type of maintenance is required, and thus, it allows for adaptation and better planning of tasks. Furthermore, it reduces the stock of spare parts planned for each maintenance, consequently leading to a reduction in overall cost. This approach provides an exciting way to deal with uncertainty associated with these three different aspects associated with vehicle maintenance.

Through a real-life crew scheduling problem of public bus transportation, Öztop et al. (2017) studied the ideal number of crewmember drivers to perform a specific set of tasks with minimal cost (Öztop, et al., 2017). The most relevant point of this paper is the presentation of two constraints: i) drivers cannot exceed the maximum limit of total work time and ii) different crew capacities for different types of vehicle.

Maintenance optimization in transports

Sriram and Haghani (2003) studied how to minimize maintenance costs and how to minimize the cost associated with redistributing aircrafts to flights that were not originally intended (Sriram & Haghani, 2003). A mathematical formulation is used to solve the aircraft maintenance scheduling problem, as well as a heuristic method since it can obtain feasible solutions in a reasonable computing time. The optimization program provides a schedule with the different flights and information on which aircraft is assigned. The main point is to analyse the possibility of performing maintenance during flight inactivity, usually between the end of the night and the beginning of the next morning, considering the different types of maintenance (type A and B), the heterogeneity of aircrafts in the fleet, the location of the maintenance bases for different types of aircraft, amongst others.

Bazargan (2015) presented a maintenance optimization at a flight training school (Bazargan, 2015). A mixed-integer linear programming (MILP) model was introduced to uncover a strategy that minimizes total maintenance cost during the planning period and increases aircraft availability. Then, the optimization solutions were compared with different performance evaluation planning criteria, such as: closest to maintenance; furthest to maintenance; random maintenance; cheapest next maintenance; equal utilization. The non-optimized strategies mentioned above have, as the main advantage, their simplicity and ease of implementation, which are factors that are of interest to companies. Finally, a plan with a smaller number of maintenance activities was tested, which, despite having a higher associated cost, obtained better availability indicators, and thus becoming the chosen solution.

Doganay and Bohlin (2010) suggested a solution based on a MILP to answer the problem of optimizing maintenance scheduling with spare parts (Doganay & Bohlin, 2010). The goal is to minimize the cost of train maintenance, shunting work, extra work and used life on the horizon. The authors have concluded that by including the costs of the spare parts, they can significantly reduce total costs. They were also able to optimize the entire fleet of trains and at the same time preserve their constraints.

Pour et al. (2017) proposed a hybrid framework that uses feasible solutions generated by Constraint Programming, and then uses a mixed-integer programming approach to optimize those solutions (Pour, et al., 2017). The chosen model is intended to solve the programming issue of the preventive signal maintenance team in the Danish railway system. The objective function guarantees: i) the minimization of the number of business days to complete the plan, ii) all tasks are completed within the planning horizon and iii) the minimization of the penalty associated with assigning workers a task on non-consecutive days. It is important to highlight that, in this type of problem, there are several practical restrictions related to the type of tasks, the crew schedule, the daily management of tasks, crew competences, amongst others.

These were the articles that served as a basis for the Martins research (2018), which is the study object and the starting point to the present dissertation. Then, Martins (2018) developed a MILP model that tried to optimize the maintenance costs of a single Lisbon depot from a bus operating company. The model featured restrictions related to the crew availability, bus availability and maintenance line availability. The model also focused in bus availability as a major decision factor. Finally, it provided a bus maintenance schedule that was able to outperform the system already used by that company. The results of this work were used as a comparison basis for the present document, and the model itself was the object of the study here conducted.

2.2. Computational models and the Dantzig-Wolfe decomposition

Dantzig & Wolfe (1959) developed a technique for the decomposition of a linear program that permits the problem to be solved by alternate solutions of linear sub-programs representing its several parts and coordinating a program that is obtained from the parts by linear transformations. The coordinating program generates at each cycle new objective forms for each part, and each part generates in turn new activities for the interconnecting program. Such a problem can be studied by an appropriate generalization of the duality theorem for linear programming, which permits a sharp distinction to be made between those constraints that affect only a part of the problem and those that connect its parts. This leads to a generalization of the Simplex Algorithm, for which the decomposition procedure becomes a special case. Formally the prices generated by the coordinating program cause the manager of each part to look for a sub-program, which proposes to the coordinator as the best the coordinating program can do. The coordinator finds the optimum 'mix' of sub-programs (using new proposals and earlier ones) consistent with over-all demands and supply, and thereby generates new prices that again generates new proposals by each of the parts, etc.

Tebboth (2001) evaluated the computational merits of the Dantzig-Wolfe decomposition algorithm. He used modern computer hardware and software and have developed an efficient parallel implementation of the Dantzig-Wolfe decomposition. His work shows that if a reasonable block structure can be found, the decomposition method is worth trying. Dantzig-Wolfe decomposition will not rival mainstream techniques as an optimisation method for all LP problems. But the Dantzig-Wolfe decomposition has some niche areas of application: certain large-scale classes of primal block angular structured problems, and, in particular, where the context demands rapid results using parallel optimisation, or near optimal solutions with a guaranteed quality.

Colombani & Heipcke (2011), describe several examples of sequential and parallel solving of multiple models with FICO Xpress software and Mosel language. The examples showcase different uses of the module *mmjobs*, such as concurrent execution of several instances of a model, the (sequential) embedding of a sub-model into a master, and the implementation of decomposition algorithms (Dantzig-Wolfe and Benders decomposition). This article was studied to identify possible approaches to improve the model developed by Martins (2018).

These specific references formed a basis of research on deciding which approach to pursue and on how to implement the fundamentals of that approach into the Mosel Language.

2.3. Contributions of the references

Table 1 - Summary of the analysis of the papers on the maintenance and scheduling of buses

References	General Topic	Proposed technique	The contribution of the paper
(Adonyi, et al., 2013)	Optimal bus maintenance plan	P-graph framework	Minimization of costs in repairing schedules and maintenance tasks that allow only the necessary buses to be available in each period.
(Bazargan, 2015)	Aircraft maintenance and availability	Mixed integer linear programming	The trade-off between maintenance cost and availability fleet. Comparison between optimization and non-optimization solution
(Doganay & Bohlin, 2010)	Train maintenance plan optimization with spare parts	Mixed integer linear programming	The influence of spare parts on the maintenance total costs
(Haghani & Shafahi, 2002)	Bus maintenance scheduling	Mixed integer linear programming	Minimize the number of hours buses are taken from their scheduled service for inspection and maximize utilization of maintenance facilities.
(Haghani, et al., 2003)	Single depot multiple depots	Mixed integer linear programming	The importance of the deadhead speed and the maximum allowed block time in the choice of single or multiple depots in the vehicle scheduling models.
(Kamlu & Laxmi, 2016)	Vehicle scheduling maintenance	Fuzzy model, GIS, MCS	Analysis of the gap between maintenance provided a classification of the type of maintenance, which improves the operation and reduces the cost of the transport system.
(Öztop, et al., 2017)	Bus drivers crew scheduling	Binary programming	Crew members with different skills that cover all tasks with the lowest possible cost.
(Pour, et al., 2017)	Preventive maintenance crew scheduling	A hybrid constraint programming/mixed integer programming	Distribution, organization, and optimization of the crew in order to reduce operational costs of preventive maintenance
(Sriram & Haghani, 2003)	Aircraft scheduling maintenance	Mixed integer linear programming	Maintenance costs and the penalty associated with redistributing aircraft to flights that were not originally intended
(Zhou, et al., 2004)	Bus maintenance and dynamic events	Multi-agent system – linear programming	Ability to generate scheduling of bus maintenance tasks within a reasonable time and yet respond dynamically to unexpected events.
(Colombani & Heipcke, 2011)	Computational approaches	Multiple models and parallel solving	Comparison between various parallel solving techniques
(Dantzig & Wolfe, 1959)	Decomposition of linear programs	Dantzig-Wolfe decomposition	Provided the decomposition on which the first approach to the problem relies
(Martins, 2018)	Maintenance planning	Mixed integer linear programming	Base model for the experiments and comparison term for the results
(Tebboth, 2001)	Computational Approaches	Dantzig-Wolfe decomposition	Algorithm and computational implementation of the Dantzig-Wolfe decomposition

The references stated before the table's division represent the ones studied to develop the Martins' model (2018), while the bottom part of the table represent the specific references that were used exclusively for this dissertation.

3. Martins model (2018), heuristic approach, Dantzig-Wolfe decomposition and constraints introduction

As stated in the introduction of this document, my thesis will focus in the improvement of the model developed by my former colleague Rodrigo Martins (2018).

This chapter will start with a presentation of the Martins' Model (2018), revealing the indexes, sets, parameters, constants, decision variables, objective function and constraints which compose the model and an illustrative example of the model will be presented.

Afterwards, the computational implementation of the experiments made in this dissertation, a Dantzig-Wolf based approach, a heuristic approach and an alternative Martins' Model, will be addressed, being an illustrative example also presented for the approaches which were possible to validate.

3.1. The Martins' Model

The main purpose of this dissertation is to computationally improve the model described below, in order to apply it to the real case of the Lisbon based bus operating company, Carris, with the primary goals of decreasing the computational time it takes to get a reasonable solution and improve the value of the objective function obtained by Martins in 2018. With that in mind, a description of the model and its goals is presented below.

The following mathematical model delivers an optimal maintenance schedule that minimizes the costs associated with maintenance crew and the costs associated with bus unavailability. The formulation is an Integer Linear Programming (ILP) model that assigns workers to certain periods of time resulting on the scheduling of the maintenance activities (Martins 2018).

The optimization model adopts four types of binary decision variables:

- The first one indicates whether, during a certain time period, a particular bus carries out a maintenance task by a specific worker;
- The second variable indicates whether, for a given time, a specific bus is under maintenance;
- The third variable indicates whether a given worker is assigned during a specific day;
- The last decision variable indicates whether for a given day, a specific bus is under maintenance.

The decision variables, objective function and constraints of the problem will now be introduced and explained. They will be presented following the structure of the optimization model.

The mathematical model program implementation in FICO® Xpress Optimization is written in the Mosel language and can be divided into five stages: i) the sample declarations section, ii) the initialization from data files, iii) the objective function expression, iv) the choice of the constraints and v) the creation of files with the output values.

Decision variables

$$x_{bmtw} = \begin{cases} 1 & \text{if the maintenance type m is performed on bus b at time t by the worker w} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{wd} = \begin{cases} 1 & \text{if worker w is assigned at day d} \\ 0 & \text{otherwise} \end{cases}$$

$$z_{bt} = \begin{cases} 1 & \text{if bus unit b is under maintenance at time unit t} \\ 0 & \text{otherwise} \end{cases}$$

$$z_{bd} = \begin{cases} 1 & \text{if bus b is under maintenance at day d} \\ 0 & \text{otherwise} \end{cases}$$

Objective function

$$Minimize \sum_{w \in W} \sum_{d \in D} cD_w \,.\, y_{wd} + \sum_{b \in B} \sum_{d \in D} cU \,.\, z_{bd} \tag{1.1}$$

The objective function (1) is composed of two components: i) the crew maintenance costs, denoted by O_1 in equation 1.2; and ii) the buses' unavailability costs, denoted by O_2 in equation 1.3. These two components are explained in detail:

$$O_1 = \sum_{w \epsilon W} \sum_{d \epsilon D} cD_w * y_{wd} \tag{1.2}$$

The parameter cD_w corresponds to the daily cost of each worker w. Thus, crew maintenance costs (Equation 1.2) can be expressed as the sum of all maintenance costs performed by every worker at every day period until the end of the activities. As mentioned before, y_{wd} is a binary decision variable that indicates whether the worker w is assigned on day d (it is equal to one) or not (it is equal to zero).

$$O_2 = \sum_{b \epsilon B} \sum_{d \epsilon D} cU * z_{bd} \tag{1.3}$$

The cU corresponds to the cost associated with unavailability of buses and it is also an input from a data file. It is important to notice that, assigning a value for cDw is not an easy job, there are many factors that influence the importance of each bus, such as the type of bus, the route it takes and the number of passengers affected, facing this a constant value was attributed to cU. There are also a lot of factors, some really hard to quantify, which affect the size of this value, for example the loss of revenues, impacts on passenger's perceived satisfaction and reliability, the opportunity costs and regulatory penalties. Taking this into account one can understand that O_2 should pose as a heavier weight in the objective function than O_1. Preference must be given to *"making vehicles available in viable and safety conditions for the operations!"* (as stated by the maintenance director of CARRIS). The parameter O_2 (Equation 3), expresses the sum of all the unavailability costs per bus unit, for each day out of its regular service. As mentioned before, z_{bd} is a binary decision variable that indicates whether the bus, b, is assigned on day d (it is equal to one), or not (it is equal to zero).

Constraints

$$z_{bt} = 0, \quad \forall\, b \in B, t \in TO_b \tag{2}$$

$$x_{bmtw} = 0, \quad \forall\, b \in B, m \in M, t \in TO_b, w \in W \tag{3}$$

$$\sum_{b\in B}\sum_{m\in M} x_{bmtw} \leq 1, \quad \forall\, t \in T, w \in W \tag{4}$$

$$L * \left[1 + \left(x_{bmtw} - x_{bm(t-1)w}\right)\right] \geq \sum\nolimits_{t_0 \in TM_b:(t_0>t)} x_{bmt_0w}, \forall\, b \in B, m \in M, t \in T\backslash\{nt\}, w \in W \tag{5}$$

$$\sum_{b_0\in B}\sum_{m\in M}\sum_{w\in W:\, CC_w=c} x_{b_0mtw} \leq nw_c, \quad \forall\, b \in B, c \in C, t \in TM_b \tag{6}$$

$$\sum_{b_0\in B}\sum_{m\in M}\sum_{w\in W} x_{b_0mtw} \leq nw, \quad \forall\, b \in B, t \in TM_b \tag{7}$$

$$\sum_{t\in TM_b}\sum_{w\in W:\, CC_w=c} x_{bmtw} \geq g_{bmvc}, \quad \forall\, b \in B, c \in C, m \in M, v \in V \tag{8}$$

$$1 - \left(\sum\nolimits_{w\in W:(CC_w=3)} x_{bmtw}\right) + \sum\nolimits_{w\in W:(CC_w\neq 3)} x_{bmtw} \leq L * \left(1 - \sum\nolimits_{w\in W:(CC_w=3)} x_{bmtw}\right)$$

$$\forall\, b \in B, t \in TM_b, m = 3 \tag{9.1}$$

$$1 - \left(\sum\nolimits_{w\in W:(CC_w=3)} x_{bmtw}\right) + \sum\nolimits_{w\in W:(CC_w\neq 3)} x_{bmtw} \leq L * \left(1 - \sum\nolimits_{w\in W:(CC_w=3)} x_{bmtw}\right),$$

$$\forall\, b \in B, t \in TM_b, m = 4 \tag{9.2}$$

$$\sum\nolimits_{m\in M} x_{bmtw} \leq z_{bt}, \quad \forall\, b \in B, t \in TM_b, w \in W \tag{10}$$

$$\sum_{b_0 \in B} z_{b_0 t} \leq nml , \quad \forall\, b \in B, t \in TM_b \tag{11}$$

$$\sum_{b_0 \in B} \sum_{m \in M:(VV_{b_0}=2)} x_{b_0 mtw} \leq nsl , \quad \forall\, b \in B, t \in TM_b, w \in W \tag{12.1}$$

$$\sum_{b_0 \in B} \sum_{m \in M:(VV_{b_0} \neq 2)} x_{b_0 mtw} \leq nml - nsl , \quad \forall\, b \in B, t \in TM_b, w \in W \tag{12.2}$$

$$x_{bmtw} \leq y_{wd} , \forall\, b \in B, m \in M, w \in W, d \in D, \{t \in TM_b : \ ntd.(d-1)+1 \leq t \leq ntd.d\} \tag{13}$$

$$z_{bt} \leq zd_{bd} , \qquad \forall\, b \in B, d \in D, \{ t \in TM_b : \ ntd.(d-1)+1 \leq t \leq ntd.d \} \tag{14}$$

$$x_{bmtw} = \{0,1\} \qquad \forall\, b \in B, m \in M, t \in T, w \in W \tag{15}$$

$$y_{wd} = \{0,1\} \qquad \forall\, w \in W, d \in D \tag{16}$$

$$zd_{bd} = \{0,1\} \qquad \forall\, b \in B, d \in D \tag{17}$$

$$z_{bt} = \{0,1\} \qquad \forall\, b \in B, t \in T \tag{18}$$

The constraints were divided into four categories: i) management constraints; ii) crew and competences/skills constraints; iii) maintenance yard constraints; and iv) general constraints. The division is intended to facilitate understanding, though some constraints could be assigned to two or even three groups.

i. Management constraints:

Constraint (2) ensures that no bus is under maintenance during the regular service/operation time. Constraint (3) indicates that no maintenance activity m, no bus b, and no worker w can be scheduled during the regular service/operation time, i.e. there is no maintenance at any time of regular service/operation. Constraint (4) states that all workers at any given time can only perform a task at a time.

ii. Crew and competences constraints

Constraint (5) ensures that when a bus is under maintenance the same worker performs his/her task in consecutive time units, i.e. maintenance tasks cannot be split. Constraint (6) indicates that, for all maintenance times, the number of assigned workers with a specific skill ($CC_w = c$) must be lower or equal than the number of workers with that skill (nw_c). Constraint (7) The number of workers assigned must be lower or equal to the number of available workers (nw). Constraint (8) guarantees, for any bus b and maintenance m, that the total maintenance time for a type of worker is at least equal to the amount of scheduled maintenance work (g_{bmvw}) for this type of worker, this constrain will be significant in chapter 4.1. Constraints (9.1 and 9.2) are identical and specific. When a bus is carrying out maintenance of type $m = 3$ (9.1) or type $m = 4$ (9.2), workers of type $w = 3$ or $w = 4$ must labour alone until they finish, i.e. they must work without the presence of any other type of worker.

iii. Constraints related to the maintenance yard

Constraint (10) states that if any maintenance assignment is made, the bus must remain in the maintenance depot for the time t needed to complete the task. Constraint (11) imposes that, for all maintenance times, the number of buses in the depot is lower or equal to the number of maintenance lines (nml). Constraint (12.1) ensures that the number of maintenance activities assigned to buses of type two ($VV_b = 2$) at the same time period cannot exceed the number of available maintenance lines capable of receiving that type of vehicle (nsl). Constraint (12.2) limits for all the buses that are not of type two ($\mathrm{VV_b} \neq 2$), the number of available lines for maintenance activities as $nml - nsl$, i.e. the difference between the total number of maintenance lines and the number of available maintenance lines capable of receiving buses of type two.

iv. General constraints

Constraint (13) states the relation between x_{bmtw} and y_{wd} decision variables, a conversion from hours to days that must be made, to determine the workers assigned to each day, which is needed in the objective function for component O_1. Constraint (14) states the relation between z_{bt} and zd_{bd} decision variables, a conversion from hours to days that must be made, to determine which buses are unavailable in each day, which is needed in the objective function, namely in component O_2. Finally, constraint (15) states that x_{bmtw} is a binary variable for all bus units, maintenance activities, time units and workers; constraint (16) states that y_{wd} is a binary variable for all workers and days units; constraint (17) states that z_{bd} is a binary variable for all bus and days units and constraint (18) states that z_{bt} is a binary variable for all bus and time units.

The outputs from the solution: x_{bmtw}, z_{bt}, y_{wd} and z_{bd} will be the primary tool in the analysis of the possible solution of the model. Through the results, it is possible to see, if the constraints respect the conditions of the case study, as well as if the value of the solution from the optimization makes sense, and whether the decision maker found what he/she was seeking. If none of these are verified, it will be necessary to modify the model; otherwise, the model can be tested.

3.2. Illustrative Example

In this section, an illustrative example is presented. The model is run for a smaller, much simpler problem, for which the optimal solution is found. The point of the illustrative example is to validate the model, by verifying that its restrictions are not violated, and showing that the solution given by the model is optimal. The Martins' Model has already been validated, thus the point of this section is not to prove its validity, but to give a comparison term to the new models presented in this document. It is important to highlight that the illustrative problem presented is not the same presented by Martins in 2018. In fact, it was adapted, by changing the timeline and the number of buses, in order to better illustrate the issues being addressed by the computational experiments conducted during the present research.

The parameters of the reduced problem will now be introduced.

In *Table* 2, the value of the constants is presented. The problem was sized to three days, ins order to have best illustrated the computational time required to solve the problem. Therefore ND is 3 and NT to 72, the other constants are the same as in the Case Study (in the Appendix).

Table 2 - Constant definition

Constants	Description	Units	Values
cU	Bus unavailability costs	Monetary units	100
NT	Total number of hours	Hours	72
ND	Number of days	Days	3
NW	Number of workers	-	6
NML	Number of maintenance lines	-	7
SL	Number of special lines	-	1

Secondly, *Table 3* represents the periods of time where there are no maintenance activities, for these periods of time is then assigned the value of zero to all the relevant decision variables.

Table 3 - Operation shifts without maintenance

Bus \ Sb	1st operation shift without P. Maintenance		2nd operation shift without P. Maintenance		3rd operation shift without P. Maintenance		4th operation shift without P. Maintenance	
	ta1b	**ta2b**	**tb1b**	**tb2b**	**tc1b**	**tc2b**	**td1b**	**td2b**
1	1	9	18	33	42	57	66	72
2	1	9	18	33	42	57	66	72
3	1	9	18	33	42	57	66	72
4	1	9	18	33	42	57	66	72
5	1	9	18	33	42	57	66	72

In *Table 4*, the cost per worker per day is shown as a function of the worker's skills. In this problem there are six workers to be assigned, just like in the Case Study (Appendix).

Table 4 - Worker costs definition

Worker	w_typew	CCw	cDw
1	Mechanic_1	1	45
2	Mechanic_2	1	45
3	Mechanic_3	1	45
4	Lubricator	2	35
5	Electrician	3	45
6	Bodywork Mechanic	4	45

Table 5 provides information about the amount of work (in hours) which each type of worker is required to do in each bus. That amount of work depends in the conjugation of the vehicle type and the maintenance type. There are five types of buses.

Table 5 - Amount of Work definition

Bus unit (b)	ma_typem (m)	v_typev (v)	c_typec (c)	Amount of work hours G(b,m,v,c)
1	IV	A	Mechanic	16
			Lubricator	4
			Electrician	2
			Bodywork Mechanic	2
2	II	B	Mechanic	6
			Lubricator	1
			Electrician	1
			Bodywork Mechanic	2
3	I	C	Mechanic	2
			Lubricator	1
			Electrician	1
			Bodywork Mechanic	2
4	IV	D	Mechanic	14
			Lubricator	4
			Electrician	2
			Bodywork Mechanic	2
5	I	E	Mechanic	4
			Lubricator	1
			Electrician	1
			Bodywork Mechanic	2

Finally, in *Table 6*, it is shown which buses can go to each maintenance line. As it is noticeable only buses of type B (Bus 2) require a special maintenance line (maintenance line 2).

Table 6 - Maintenance Line definition

Vehichle Types / Maintenance Lines	A	B	C	D	E
1	1	0	1	1	1
2	0	1	0	0	0
3	1	0	1	1	1
4	1	0	1	1	1
5	1	0	1	1	1
6	1	0	1	1	1
7	1	0	1	1	1
NML=7		**SL=1**			

With all the relevant date already stated, the validated model results will now be presented, starting with the Objective Function , computational time and optimality gap, the factors under study.

Figure 1 - Original model illustrative example results

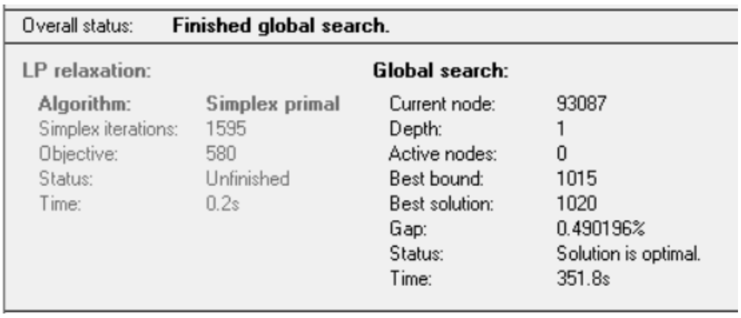
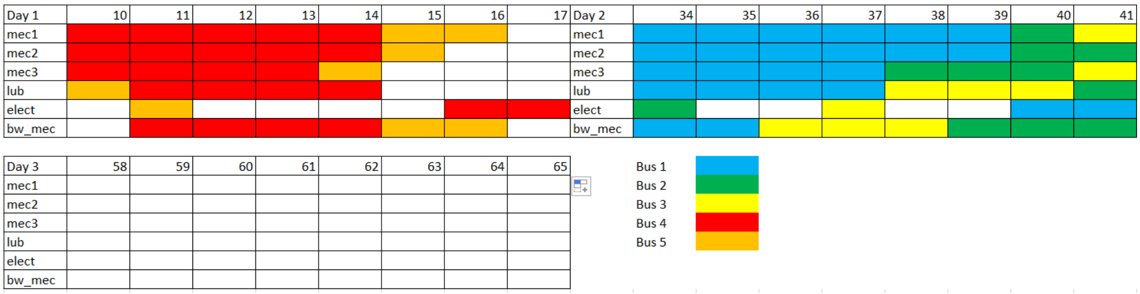


Figure 1 shows the result from the FICO Xpress software, where it is given that the optimal solution for this problem is of 1020 monetary units, which was achieved in 351.8 seconds. Although it presents an optimality gap of 0.49%, the solution is considered optimal (as shown by the Status message: “Solution is optimal.”).

These values will be used to validate the other approaches to the problem (if the same optimal solution is obtained) and comparing the present optimal solution with the ones obtained using the other models/approaches. Some indicators of their performance in a larger problem can then be obtained.

In order to illustrate the results, the scheduling solution given by the model is presented below.

Figure 2 - Original model illustrative example scheduling



3.3. Other approaches to solve the problem

Three different approaches were tried to solve the problem: i) an approach based on Dantzig-Wolf decomposition; ii) a heuristic approach and iii) an approach based on introducing new constraints to the original formulation. These approaches are described and tested in subsections 3.3.1, 3.3.2 and 3.3.3, respectively.

3.3.1. **Approach based on Dantzig-Wolf decomposition**

A few methods were considered for achieving a more computationally efficient modelling of the problem. At first sight, the one that that seemed more appropriate was the Dantzig-Wolf method. It was in fact the main target of the computational experiments described over this research work. In this chapter theory and the essence behind this method is addressed. The goal of this section is to provide an overall knowledge of the Dantzig-Wolfe algorithm and to present its main differences from the original model. A further detailed explanation of the concrete adaptations for the present problem, and a discussion of the decisions made when developing the model, will be given in section 4.1.

The Dantzig-Wolfe decomposition, as stated by Colombani and Heipcke in 2011, is a solution method for problems where, if a relatively small number of constraints were removed, the problem would fall apart into a number of independent problems. This means that it is possible to rearrange and decompose the original problem into a series of simpler sub-problems, being sometimes necessary to re-organize the constraint definitions, grouping them by common index (sub)sets such as time periods, products, plant locations, and so on.

The constraints (including the objective function) that represent a condition to all the sub-problems as a whole and therefore cannot be independently expressed in the sub-problems, are referred to as global, linking or master constraints. These constraints are left out of the sub-problems and constitute what is called the master problem. The sub-problems are solved as pricing problems, coordinated by the master problem. By solving the master problem, a solution to the original problem is obtained. Since the master problem has a large number of variables, the goal is to restrict the master problem to a small subset of variables. These variables are determined by solving the pricing sub-problems. The objective functions for the pricing problems are based on the dual values of the restricted master problem, therefore the value of the objective function at each extreme point is the price of the master problem variable relevant to that extreme point.

For minimization problems, such as this one, solving the modified pricing problems generates basic feasible solutions of minimum reduced cost. If the objective value at an extreme point is negative, then the associated master problem variable is added to the master problem; if the minimum objective value over all extreme points is positive, then no master problem variables exists to improve the current master problem solution.

The main reason as to why this approach was chosen lies in the fact that the Dantzig-Wolfe decomposition is able of performing a significant amount of the computational work in the sub-problems which are close to an order of magnitude smaller than the original problem and thus easier to solve. Resulting in a great advantage regarding computational time, since the sub-problems are independent of each other and may be solved at the same time.

A potential drawback of the decomposition approach is the huge size of the master problem, it has many more variables—though fewer constraints—than the original problem. Furthermore, numerical problems may occur through the dynamic generation of variables of the master problem.

When Tebboth (2001) studied the computational applications of this decompositions he came to the conclusion that many factors may influence the performance of the decomposition approach, so for a particular application, computational experiments are required to find out whether this solution method is suitable. As it will be later discovered, this method is not suitable for the present problem.

Experiments with this method included different ways of decomposing a given problem. These experiments will be related in further detail in section 4.1.

As stated above, the original problem is divided into sub-models which solve pricing problems and are then linked by the master problem, in order to give a solution to the master problem. The pricing problems are linked by being assigned a weight to each proposal, representing each a part of the final solution. For the present problem a decomposition of the problem by day was used, because using this decomposition only one linking constraint is required. The goal of this section is to provide an overall knowledge of the Dantzig-Wolfe.

The main difference from the original problem is that instead of solving a large LP problem, it solves several independent pricing problems at the same time. For these, two models are needed i) a Master model, which controls the sub-problems, verifies the general constraints and provides a final solution; ii) and a sub-model, which solves a reduced problem.

i) The Master model starts, as in the original model, by reading data and inputting it into the solver. But there some new additions to the previous elements, here it is also necessary to define a set of events (Phases, which control the master/sub-model relationship, a linear control regarding the linking constrains, the sets in which the proposals from the sub-model, and its prices are saved, a number of processes which represent the pricing solving, the optimization solving and the final solving, these are need since the model consists on several problems and not just one. Also, the sub-problem needs to be called upon. Then several iterative loops are created in order to update the pricing of each sub-problem, one for each process. The first process generates a first infeasible solution, the second verifies the feasibility of the linked solutions of the sub-problems, and the third one verifies the optimality of the solution and the last one provides the final solution to the original problem.

ii) The sub-model needs the same inputs as the master model and is where almost all the constraints of the original problem are stated. The sub-model communicates with the master through receiving event codes informing which process is being solved. It also receives information about the pricing updates generated by the master and provides the master with proposals having into account these updates. At the first stage the sub-model solves a zero linking constraint problem for each sub-problem which is then analysed by the master, pricing information is given to the sub-model and the second stage, where the first solution is updated, takes action until the master deems the overall solution feasible. After that, as stated in i), the master runs an optimality check, again updating the prices of the sub-problems now related to optimality and not feasibility, and lastly the sub-model provides proposals to those updated prices until the master finds the solution optimal. Then it is all up to the master model.

All the processes will be explained in depth in section 4.1, where an extensive discussion of the model implementation will be conducted.

Also, it was not possible to validate this model, therefore no illustrative example or results are provided. The model found the problem infeasible, thus no solution was discovered. Further insights on as to why this happens and what can be done to solve this will be presented in section 4.1 and chapter 5.

3.3.2. Heuristic Approach

This heuristic approach (as the next one) is highly based in the Martins' model. The great difference is that instead of solving one large ILP, it solves various smaller problems. However, instead of solving each smaller problem in a parallel way (as the Dantzig-Wolfe approach), this heuristic approach solves them sequentially. The heuristic, its main principles and the model itself will be discussed in further detail in section 4.2. This section will focus on the computational changes from the Martins' model, the model validation and the indicators of its behaviour in the large-scale problem.

This approach relies on solving the problem for one bus at a time, locking the solution of the previous bus, i.e. the problem is solved for bus 1, the time periods occupied the solution for bus 1 are withdrawn from the solution possibilities of the next bus and so on, until the problem is solved for all the buses.

Major computational changes come from: i) the introduction of a new set that dictates the order in which the buses are solved; ii) the introduction of a loop that annexes the buses orderly to the B set, solves and saves the solution to the problem; and iii) a new constraint that prevents a next bus from overwriting or substituting previous buses' allocations.

i) The new set Gb_{NBus} is defined, with values equal to the bus numbers and where i is the index that defines the order, i.e. Gb_1 is the first bus the problem is solved for. It can be defined manually by the user or decision maker, or by using a criteria for choosing the order. The order criteria defined was based on the amount of work needed for that bus. Nevertheless, an explanation on why it is inserted manually is provided in section 4.2.

ii) Using the "repeat" function of the Mosel language a loop where a new variable called $NBus$ is created with the value of one and increased by one at each iteration. This variable is then used to annex a new bus to the set B, being Gb_{NBus} added to that set. The problem formulated by Martins (Martins 2018) is now a procedure and not the only problem. At the end of the procedure the solution found is added to a new set $Prop\_x_{bmtw}$. This procedure happens within the loop, and it stops when the number of iterations ($NBus$) is equal to the number of buses present in the problem.

iii) Finally, a new constraint was added:

$$Prop\_x_{bmtw} = x_{bmtw}, \qquad \forall\, b \in B \backslash \{Gb_{NBus}\}, m \in M, t \in T, w \in W \qquad (21)$$

This restriction states that every solution already obtained must keep the same value, and thus preventing the new solutions to be allocated to those time slots. This new restriction will be explained in detail in section 4.2.

Performance of the heuristic approach in the illustrative example

For the illustrative example, the constants and parameters presented in section 3.2 remain the same, but the new Gb_{NBus} set also needs to be defined and it is presented below:

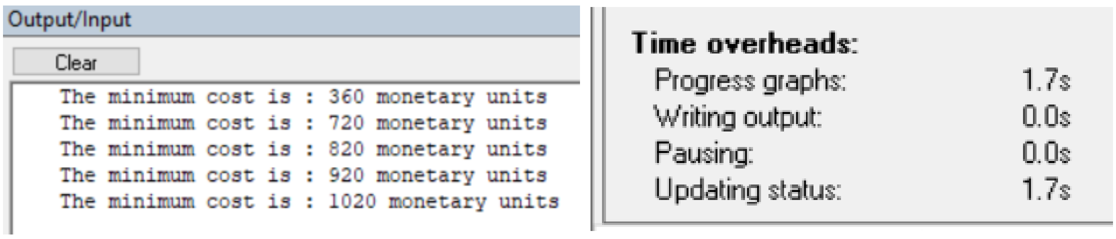
Table 7 – Solving order definition

Order (Gb)	Bus (b)
1	1
2	4
3	2
4	5
5	3

By checking *Table 5* in section 3.2, it is noticeable that the order in which the buses are solved for this problem follows, from the highest to the lowest, the amount of work needed for each bus, i.e. decreasing order in amount of work. The main intuition behind such choice is that "it is easier to fill smaller spaces with smaller objects than with bigger ones". Further details are provided in section 4.2.

Regarding the results obtained with this approach, it does not make sense to present them in the same way that they were presented in section 3.2, since there is not one, but five ILP problems to be solved. Here, it is not possible to obtain a value for the optimality gap of the overall problem, but the solution is obtained from the output and the computational time is also obtainable. They are shown in *Figure 3*.

Figure 3 - Heuristic approach illustrative example results

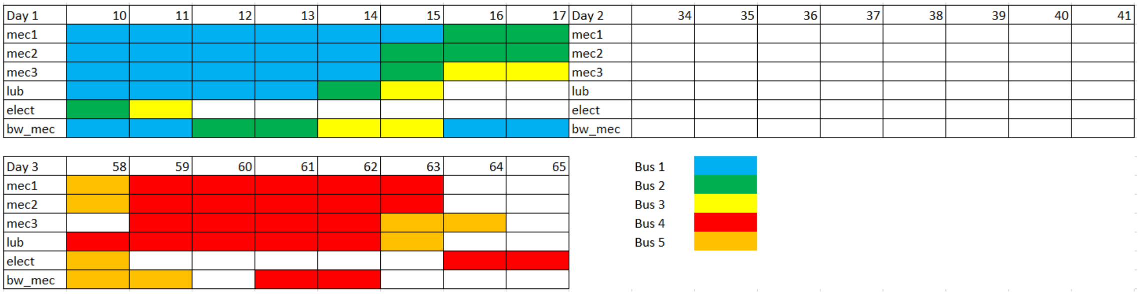


The solution given by this model is identical to the one provided by the original one, which is known to be optimal. Therefore, it is possible to conclude that the solution obtained using this heuristic approach is also optimal and thus the model is validated.

Not surprisingly, the computational time is much lower (99.5% lower), providing a very optimistic potential when applied in the large-scale problem.

Following there is the scheduling proposed by this model:

Figure 4 - Heuristic approach illustrative example scheduling



3.3.3. Introducing New Restrictions Approach

The third approach that was tried is even more similar to the original one than the previous one. This third approach is a small variation of Martins' formulation (2018), in which new restrictions are introduced to reduce the number of variables and nodes of the original problem. These new restrictions are based on characteristics that an optimal solution would have.

These new restrictions to the problem are the following:

$$\sum_{b \in B} \sum_{d \in D} zd_{bd} = NB \tag{22}$$

$$\sum_{m \in M} \sum_{w \in W} x_{1m10w} \geq 1 \tag{23}$$

$$\sum_{m \in M} \sum_{w \in W} x_{4m34w} \geq 1 \tag{24}$$

$$\sum_{m \in M} \sum_{w \in W} x_{12m58w} \geq 1 \tag{25}$$

$$\sum_{d1 \in D : d1 \leq d} zd_{3d1} \geq zd_{6d}, \qquad \forall\, d \in D \tag{26}$$

$$\sum_{d1 \in D : d1 \leq d} zd_{9d1} \geq zd_{10d}, \qquad \forall\, d \in D \tag{27}$$

$$\sum_{d1 \in D : d1 \leq d} zd_{5d1} \geq zd_{11d}, \qquad \forall\, d \in D \tag{28}$$

$$\sum_{b \in B} \sum_{m \in M} \sum_{w \in W} x_{bmtw} \geq \sum_{b \in B} \sum_{m \in M} \sum_{w \in W} x_{bmt-1w}, \qquad \forall\, t \in T: 10 > t \leq 17 \tag{29}$$

$$\sum_{b \in B} \sum_{m \in M} \sum_{w \in W} x_{bmtw} \geq \sum_{b \in B} \sum_{m \in M} \sum_{w \in W} x_{bmt-1w}, \qquad \forall\, t \in T: 34 > t \leq 41 \tag{30}$$

$$\sum_{b \in B} \sum_{m \in M} \sum_{w \in W} x_{bmtw} \geq \sum_{b \in B} \sum_{m \in M} \sum_{w \in W} x_{bmt-1w}, \qquad \forall\, t \in T: 58 > t \leq 65 \tag{31}$$

$$\sum_{b \in B} \sum_{m \in M} \sum_{w \in W} x_{bmtw} \geq \sum_{b \in B} \sum_{m \in M} \sum_{w \in W} x_{bmt-1w}, \qquad \forall\, t \in T: 82 > t \leq 89 \tag{32}$$

$$\sum_{b \in B} \sum_{m \in M} \sum_{w \in W} x_{bmtw} \geq \sum_{b \in B} \sum_{m \in M} \sum_{w \in W} x_{bmt-1w}, \qquad \forall\, t \in T: 106 > t \leq 113 \tag{33}$$

$$\sum_{b \in B} \sum_{m \in M} x_{bmtw} \geq L \times \sum_{b \in B} \sum_{m \in M} x_{bmtw-1}, \qquad \forall\, t \in T, w \in W:\ CC_w = 1\ \cap w > 1 \tag{34}$$

$$y_{w-1} \leq y_w, \qquad \forall\, d \in D, w \in W:\ CC_w = 1\ \cap w > 1 \tag{35}$$

$$\sum_{b \in B} \sum_{m \in M} \sum_{w \in W} \mathrm{x}_{\mathrm{bmtw}} \geq L \times \sum_{b \in B} \sum_{m \in M} \sum_{\mathrm{w} \in \mathrm{W}} \sum_{t \in T: t \leq 24} \mathrm{x}_{\mathrm{bmtw}}, \qquad t = 34 \tag{36}$$

$$\sum_{b \in B} \sum_{m \in M} \sum_{w \in W} \mathrm{x}_{\mathrm{bmtw}} \geq L \times \sum_{b \in B} \sum_{m \in M} \sum_{\mathrm{w} \in \mathrm{W}} \sum_{t \in T: 25 \leq t \leq 48} \mathrm{x}_{\mathrm{bmtw}}, \qquad t = 58 \tag{37}$$

$$\sum_{b \in B} \sum_{m \in M} \sum_{w \in W} \mathrm{x}_{\mathrm{bmtw}} \geq L \times \sum_{b \in B} \sum_{m \in M} \sum_{\mathrm{w} \in \mathrm{W}} \sum_{t \in T: 49 \leq t \leq 72} \mathrm{x}_{\mathrm{bmtw}}, \qquad t = 82 \tag{38}$$

$$\sum_{b \in B} \sum_{m \in M} \sum_{w \in W} \mathrm{x}_{\mathrm{bmtw}} \geq L \times \sum_{b \in B} \sum_{m \in M} \sum_{\mathrm{w} \in \mathrm{W}} \sum_{t \in T: 73 \leq t \leq 96} \mathrm{x}_{\mathrm{bmtw}}, \qquad t = 106 \tag{39}$$

$$\sum_{b \in B} zd_{\mathrm{b}d-1} \leq L \times \sum_{b \in B} zd_{\mathrm{bd}}, \qquad \forall\, d \in D: d > 1 \tag{40}$$

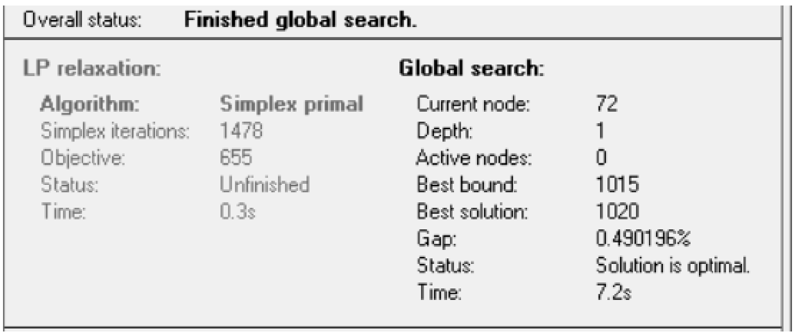
$$\sum_{w \in W} y_{\mathrm{w}d-1} \leq L \times \sum_{w \in W} y_{\mathrm{wd}}, \qquad \forall\, d \in D: d > 1 \tag{41}$$

This new set of restrictions and the reasons behind their construction will be explained in further detail in section 4.3. The goal of this section is only to provide enough understanding of the model so that the results of the illustrative example can be rightly interpreted, and the model validated. Moreover, one may use the performance of this third approach in the illustrative example to predict its expected behaviour in the large-scale problem.

Performance of the third approach in the illustrative example

The constants and parameters of this model are exactly the same presented in section 3.2. Therefore, they will not be presented again. Since this model is also a single ILP problem, the results can be presented in the same way as the original model.

Figure 5 – Additional restrictions approach illustrative example results

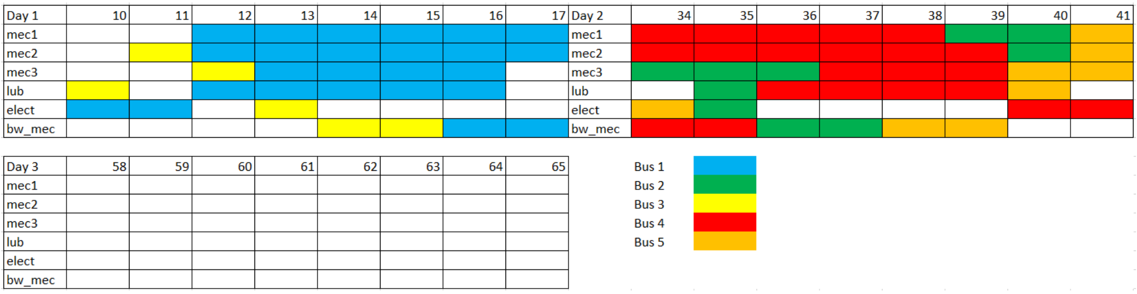


From *Figure 5*, it is possible to observe that the solution and the optimality gap are exactly the same as in the original model, thus clearly validating the model, and reporting in the Status field that the "Solution is optimal.". This was expected since in a problem of such a scale it was likely that the solution would be the same due to their similarity. What comes as quite as a surprise is the computational time, it was expected to be lower than the original model but regarding the simplicity of the problem a smaller difference in computational time was anticipated, not a drop of 98.9% in computational time. This presents a good indicator for this approach's performance in the large-scale problem.

It should be pointed out that from this illustrative example, one can state that the introduction of the additional constraints did not increase the objective value, but only improved the computational time to achieve the same optimal value.

Similar to the other approaches, the scheduling solution from this model is also presented below:

Figure 6 - Restrictions approach illustrative example scheduling



4. Computational experiments and model Implementation (Carris Case Study)

In this chapter, the computational experiments conducted using an approach based on the Dantzig-Wolfe method are explored. Afterwards, an alternative approach based on the introduction of new constraints is also explored, conducting a comparison between the original model and the final model with the new constraints. The general aspects of the implementation of the approach based on the Dantzig-Wolf method were already In the present chapter, a deep insight of the implementation details on the application of the Dantzig-Wolf decomposition to our problem will be discussed, namely the new parameters, declarations, initializations, objective function, decision variables and constraints. All model ingredients already discussed will not be fully explained again. The case study is presented in the Appendix A1.

4.1. Approach based on Dantzig-Wolf decomposition

As a starting point to the following computational experiments, a factory production example presented in “Multiple models and parallel solving with Mosel” (Colombani & Heipcke, 2011) was chosen to be ideally adapted to include the specificities of the bus maintenance scheduling model. However, there were two main differences between the factory production example and the bus maintenance scheduling model. The example solves a non-binary maximization problem, while the present model is a binary minimization problem. The binary essence of the bus maintenance scheduling problem creates difficulties that will be analysed later.

The first step was to decide how to decompose the problem. There were a the following options: i) decomposing by day, ii) decomposing by time period, iii) decomposing by bus or by worker. This would mean solving the scheduling for each day, time period, bus or worker in a parallel way.

The decomposition by time period can be easily excluded due to the fact that the dimension of each problem would be so small that almost all the constraints would be in the master problem, which is exactly the opposite of what is the aim of the decomposition.

At a first glance, one would be inclined to decompose the problem by bus, i.e. to schedule each bus. However, this option also raises two problems. First, if one looks at the constraints, one can see that there are six associated constraints that will have to be in the master problem, which might be doable but certainly not ideal. Nevertheless, this first problem, combined with the fact that only one part of the objective function depends directly on the buses, would make the implementation of this alternative decomposition very complex in terms of programming.

Applying the same argument, one can also notice that the decomposition by worker raises the exact same problems. Although these two decompositions make the model more complex, they are perfectly good decompositions that are worth being explored.

The decomposition by day was chosen. By doing such decomposition, each subproblem will be big enough and there will be only one associated/linking constraint (8) to be included on the master problem, and both components of the objective function depend directly on the days. However, not everything is perfect about this decomposition. The associated/linking constraint is a minimum limit constraint which is not typical in this type of algorithm and requires some adaptations. Moreover, it also depends on different indices than those that the decision variable controls, which causes some troubles when generating optimal proposes, as it will mess with all the values imposed by the control and not only the ones needed, which will be further explained ahead.

The implementation in FICO Xpress requires: i) a Master model and ii) a sub-model or slave model. The master model contains the linking constraints and provides the final solution to the original problem, while the sub-model will solve each subproblem (i.e. for each day) independently.

In order to decompose the problem in days a few adaptations to the previous model were made. The time T was set to 24 hours and the decision variables x_{bmtw} and z_{bt} are now x_{dbmtw} and z_{dbt}. The constraints assigning each time t to each day d are no longer necessary and thus the problem for each day is solved separately.

The linking constraints are the ones that depend on all subproblems and not only on a single one. If one looks at the constraints stated in Chapter 3, one can see that the only constraint that represents a sum in all the days is constraint number (8), which establishes the minimum amount of work (in hours) that should be put in bus b, maintenance activity m and vehicle type v by a worker with competence c:

$$\sum_{t \in \mathrm{TM_b}} \sum_{w \in W:\, \mathrm{CC_w = c}} x_{bmtw} \geq g_{bmvc}, \qquad \forall\, \mathrm{b \in B, c \in C, m \in M, v \in V} \tag{8}$$

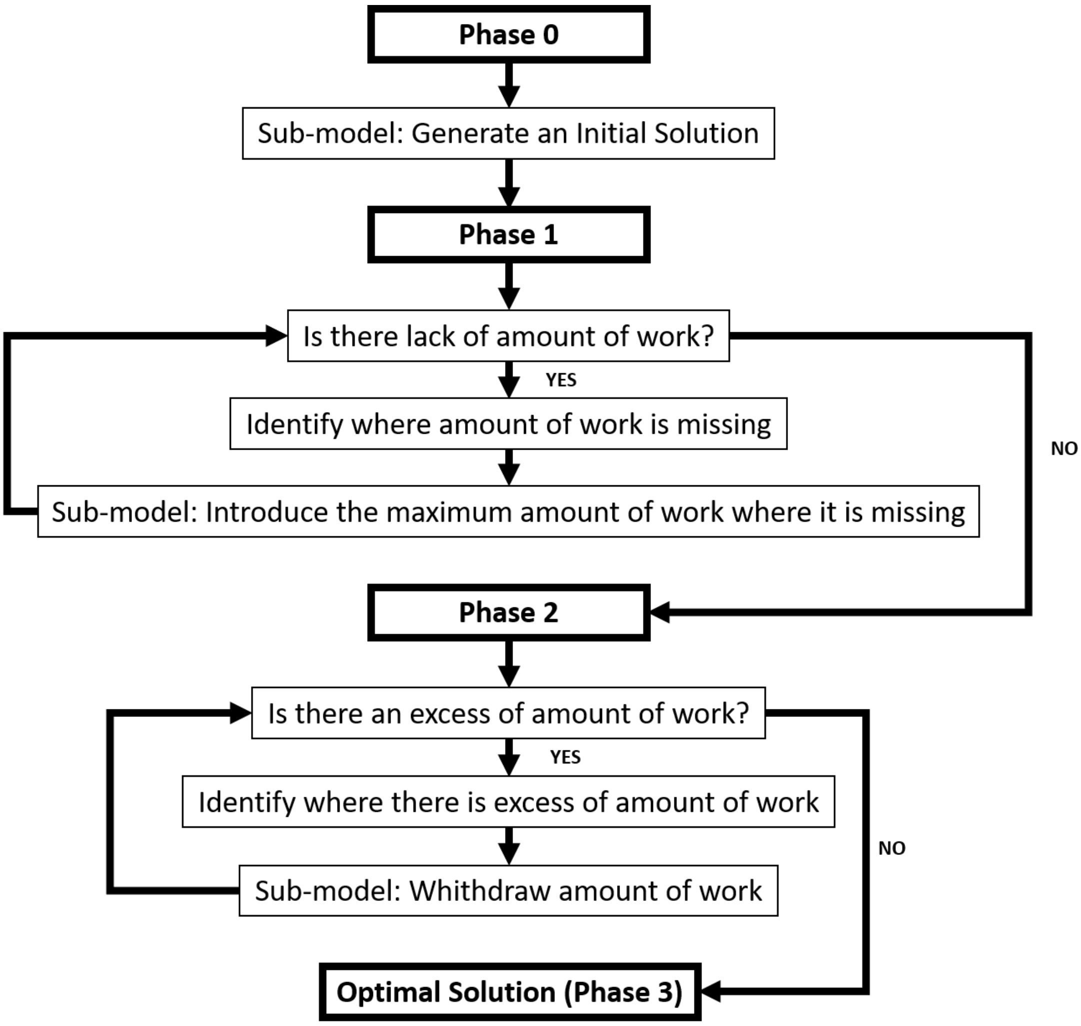
All the remaining constraints are stated in the sub-model adapted for each day, by removing the $\forall\, \mathrm{b \in B}$ statement where it is applied and the decision variables that previously depended on d no longer do, i.e. x_{dbmtw} is now x_{bmtw}, z_{dbt} is now z_{bt}, y_{wd} is now y_w and zd_{bd} is now zd_b in the sub-model.

There are four phases to the solving model, which are controlled by the master model:

- **Phase 0** – where a plausible solution is found;
- **Phase 1** – where a feasible solution is found, starting from the solution of Phase 0;
- **Phase 2** – where the feasible solution found in Phase 1 is optimized;
- **Phase 3** – where the final original is solved

Following there is a scheme illustrating the flow of information between the different phases.

Figure 7 - Dantzig-Wolfe phase communication



Phase 0 – where a plausible solution is found

Although **Phase 0** is not strictly necessary, it avoids starting from a random solution. In **Phase 0**, the sub-model will run without verifying the global constraints, its objective function is the same as the original problem and the idea is to find an infeasible solution that represents the optimal solution disregarding the linking/global constraint. As in this problem the linking constraint is the only one that imposes a minimum limit on the amount of work needed for each bus, the perfect solution for the problem, which is not controlled by this constraint, will be a solution in which no work is needed, and therefore, a solution where all the decision variables are equal to zero. The reader should keep in mind that **Phase 0** is an optional phase, whose only purpose is to give us a starting point that is not completely random.

Phase 1 – where a feasible solution is found, starting from the solution of Phase 0

After a solution for **Phase 0** is found, that solution is turned into a proposal for each day, example: $Prop\_x_{dbmtwk}$ and according to each iteration ($nPROP$). Then, the goal is for the master problem to combine the propositions in a way that the problem becomes feasible. This is achieved by assigning weights to each proposal. In **Phase 1**, the linking constraint is introduced. A linear control is created in order to determine which decision variables are in violation of the global constraint:

$$Gx_{bmvc} \coloneqq \sum_{d \in D} \sum_{t \in T} \sum_{k \in nPROP(d)} \sum_{w \in W:\, CC_w = c} Prop_{x_{dbmtwk}} \times weight_{dk} + excessS \;\geq\; g_{bmvc} \tag{42}$$

Therefore, the objective function of the **Phase 1** for the master model is to minimize a new variable called "*excessS*". This allows us to determine, in a first approach, if the constraint is being violated or not. And if it is not, the result for the *excessS* will be zero and the problem will go on to the next phase. Otherwise, i.e. if it is violated, the solution of the *excessS* will be larger than zero.

The analysis of the dual price of the linear control will show where the amount of work is missing, or at least where it is missing the most. A solution of the dual price is provided: $Price\_x_{bmvc}$ and it is sent to the subproblem.

In the subproblem, the objective function at this phase will maximize the number of hours assigned in order to fill the needs it got from the Price of the linear control:

$$Maximize \sum_{b \in B} \sum_{m \in M} \sum_{t \in T} \sum_{v \in V} \sum_{w \in W:\, CC_w = c} Price\_x_{bmvc} \times x_{bmtw} \tag{43}$$

By doing that, one can notice a disparity between our needs $bmvc$, and what it can provide, $bmtw$. In fact, this was one of the issues mentioned above associated with this decomposition. However, at this stage, it is not critically relevant yet, though it will become later in **Phase 2**.

After this maximization, the needed working hours discovered in the master problem are provided. But these are the working hours needed by one or some of the $bmvc$ combinations, it can also be for all of them. The master problem is run again and a new $excessS$ is found, this value will always be less or equal than the previous value and the process is repeated for the new-found needs. The program keeps doing this until the solution of the $excessS$ is equal to zero, and then it goes to **Phase 2**.

There is also a crash instruction when in the sub-model a solution to the objective function is equal to zero, so that the problem is deemed infeasible in such case. This happens because the solution of the sub-model being equal to zero would mean one of two things: i) all the $Price\_x_{bmvc}$ combinations are equal to zero or ii) the ones that are different from zero cannot be improved due to any constraint of the subproblem. Both situations would mean that there is no room for improvement of the previous solution, and knowing that the previous solution is not feasible, one can also assume that if it cannot be improved, it will never become feasible.

At this point in this Phase, an issue should be discussed as it becomes critical to the technique of this model. The prices given in the linear control Gx_{bmvc} strongly depend on the weights assigned to each proposal. In a different problem a weight that is not binary would be acceptable because, as stated above we impose the condition that the sum of weights selected for a day must be equal to one. While in a different problem that combination of weights multiplied by the proposed solution could be handled in a way that the sum of the weights times the proposed solutions would have to be an integer number. The issue with this problem is that all the solutions proposed are binary, which means that without enough proposals, it would be very often impossible to make a combination that gives an integer as a solution.

For example: let three proposals for x_{11411} be equal to: 0, 1 and 1. If a value different from zero or one is set for the first proposal, an integer solution will not be achieved. Note that this will happen with any number of proposals, any time of the day and in any day, even if only one of the zero-valued proposals is given a weight different from zero or one, the solution will not be an integer. So far there does not seem to be a problem here, as the weights that are one or zero can be chosen and it would work with that. However, the proposals do not cover all the possible combinations, as they are generated according to the needs of the amount of work missing. The goal here would be to get a little bit of every proposal and join them accordingly to the restrictions, but due to the binary properties of the problem, parts of a proposal cannot be selected and 'glued together' to form a feasible solution. As the proposals must be chosen as a whole, the ones chosen at **Phase 1** will never cover all the needs of the restriction, because upgrades to the proposals are generated as a set and not one by one. It could be done one by one, but this would go against one of the purposes of the present work which is to reduce computational time. Moreover, as a result of the obligation to choose whole proposals, there will come a point where the same set of proposals will be chosen twice in a row (in the present case this happens on the second iteration), and thus activating the crashing mechanism due to the lack of improvement on the final solution.

Therefore, using a pure pricing/weight Dantzig-Wolfe decomposition would not be a successful option for the present case study. However, a few adaptations/suggestions related to the weight scheme are proposed and left for further research. It is crucial to mention that this conclusion was reached already in an advanced stage of the present research (after several tests), and therefore no time for further experiments with adaptations was left. In fact, alternative approaches were pursued in order to get the optimization results in a more efficient way.

Phase 2 – where the feasible solution found in Phase 1 is optimized

As the binary issue discussed above was implemented further down the road in the experiments, the **Phase 2** of the problem had already been developed, not imposing the critical constraint to the weights. Although that matter has the exact same effect here has it had on **Phase 1**. On **Phase 2**, after a feasible (but not optimal) solution was found at the end of **Phase 1**, the goal is to reduce the excess of amount of work assigned in Phase 1. To do that a linear control and a new objective function have to be reformulated:

$$Gx_{bmvc} \coloneqq \sum_{d \in D} \sum_{t \in T} \sum_{k \in nPROP(d)} \sum_{w \in W:\, CC_w = c} Prop\_x_{dbmtwk} \times weight_{dk} \geq \; g_{bmvc} \tag{44}$$

$$minimize \sum_{d \in D} \sum_{k \in nPROP(d)} Prop\_cost_{dk} \times weight_{dk} \tag{45}$$

Here the main issue is not about the deficit of work assigned ($excessS$), but the concern shifts solely to prevent the chosen proposals to go lower the minimum limit. And the objective function is now similar to the objective function of the original problem. $Prop\_cost_{dk}$ is the solution of the daily adaptation of the original problem for each day and each proposal. In this phase, the process is quite similar to **Phase 1**, getting the price data for each combination of $bmvc$ and sending it to the sub-model. But now the sub-model has a different objective function:

$$Minimize \left(O_1 + O_2 - \sum_{b \in B} \sum_{m \in M} \sum_{t \in T} \sum_{v \in V} \sum_{w \in W:\, CC_w = c} Price\_x_{bmvc} \times x_{bmtw} \right) \tag{46}$$

Note that the goal is to minimize the difference between the solution of the daily adaptation of the original problem and the price data that we are trying to improve. When this value reaches zero in all sub-problems it means that there is no more room for improvement and, therefore, the solution is optimal.

And this will be repeated until that difference is equal to zero, and thus the optimal solution is reached.

Adding up to the problem described in **Phase 1**, here another issue appears. It has nothing to do with feasibility but with the increase of computational time in a way that makes no sense to use this approach. As pointed out before, the sets that $Price\_x_{bmvc}$ and x_{bmtw} depend on are not the same. This means that a change only in the variables needed cannot happen, it will require changing all the variables that are associated with that set.

Take as an example when $Price\_x_{1411} = \; 1$, and one is maximizing this component of the objective function. Then, ideally all the x_{bmtw} would be changed to one. The problem is that this change can be more drastic than wanted. For example for g_{1411}, $16\ hours\ of\ work$ would be needed, but translating it to the x_{bmtw} variable one gets $x_{14tw|CC(w)=1}$. As there are three workers with competence $c = 1$ and there are $8\ hours$ that fit the time frame (considering the restrictions of the subproblem), it will provide $24\ hours$ (3×8) of input work in the variable x_{bmtw}.

Due to that reason, it takes much more iterations to reach the value that is needed, and thus it significantly expands the computational time. It should be mentioned that the implementation process of this decomposition took more time than initially planned. An alternative approach (hopefully more straightforward) was pursued to provide comparable results with the original bus maintenance scheduling problem.

Phase 3 – where the final original is solved

As stated in the description, the original problem is solved, and the final solution is give.

4.2. Heuristic Approach

This alternative approach was idealized with the objective of reducing the decision variables and combinations for each solving cycle of the model, and thus reducing the computational time.

The basic structure of this approach relies in solving the scheduling for one bus, then saving it, and solving for the next bus removing the spaces occupied by the previous bus and repeat that sequentially until all the buses are assigned. It is as if the model is playing “a game of Tetris” but with the buses’ schedule as its figures. In this way the model focuses in solving only one problem (for a single bus) at the time for the yet available time periods, instead of solving every combination for every bus and every time period available at once. This results on a great reduction of computational time. However, although the optimality gap for each problem solved is equal to zero, there is no guarantee that the overall solution is optimal.

The order in which the buses’ scheduling is solved is very significant in this problem. Ideally the model would run every possible combination of ordered buses and choose the one with the best result. This would result in running $13!$, $6{,}23 \times 10^{9}$, different combinations. Making an estimate and spoiling some of the results, for one combination it takes $37\, seconds$, so that many combinations would take $2{,}3 \times 10^{11}$ seconds which are $6{,}4 \times 10^{7}$ hours, and since through this method the solution is always found, and only after all the combinations are run, it goes against the main goal of this dissertation, that it would pursue an approach that is already known to take more computational time. Moreover, the solution found does not have certain optimality.

To tackle this problem, it was decided to establish the order in which the model would schedule the buses based on a criteria. The order chosen was from the bus with the most need in maintenance hours to the one with the least. This choice is justified by the fact that it is much easier to allocate a small number of hours to an almost filled up time table than otherwise. A problem found was that, in the first buses, the model would assign some of them to different days, resulting in doubling the cost of a stopped unit, as it was saving on workers, not knowing it was going to need them anyway for the not yet assigned buses. This, though computationally accurate, would not result in the best solution. As it is known from Chapter 3, it is preferred that the buses stay the least time possible in the depot. We also know from Chapter 3 that the value of the cost per stopped unit in Martins’ model (2018) was set arbitrarily and the value was chosen in order to show that preference. It is clear that, in the model here discussed, that privilege is not shown. So, in order to fix this issue, a new value for cU, that represents its importance, was attributed. The value chosen was $260\, monetary\ units$, one notices that this is the cost of having all the workers assigned to a new day, thus affirming, undoubtably, the importance of this component of the objective function. For comparison purposes all the final values are presented with the cU equal to 100, as it was in the Martins' model (2018). Otherwise, comparisons with the original models would not be straightforward.

The heuristic approach results in a sequential model that solves thirteen problems (one for each bus). The problem solved for each bus is almost the exact same problem of the Martins' model (2018), same variables, same restrictions plus one and same objective value. First, the order in which the buses will be scheduled is established. For this problem, as stated before, they were ordered by the amount of work needed. Then a loop is established, and the B array is created. The B array starts only with one element and gains one more at each iteration of the loop.

The first subproblem is solved, this is the original problem for only one day, after that the $Prop\_x_{bmtw}$ array is created and the values of the decision variable x_{bmtw} are saved in there. For the second iteration, second bus, there is a new restriction to the problem which locks the values of x_{bmtw} previously assigned:

$$Prop\_x_{bmtw} = x_{bmtw}, \qquad \forall\, \mathrm{b} \in \mathrm{B} \backslash \{Gb_{NBus}\}, \mathrm{m} \in \mathrm{M}, t \in T, w \in W \qquad (47)$$

Where, Gb is an array containing the number of the bus by the defined order and $NBus$ represents the iteration number. It is required to take that element from the array because, as mentioned before, the increment of the B array was already done, and thus it is necessary to withdraw the bus being scheduled from the ones being "locked".

The process is repeated until all the buses are scheduled, so that the result of the last sub-problem (scheduling the last bus with all the others already assigned) is the final solution to the original problem.

Furthermore, it was tried to optimize the solution given by the model described above through some iterative approaches. The first was to make the problem go back to the first scheduled bus and repeat the loop until no better solution was found. Secondly, it was tried to do exactly the same but by changing two and then three buses at a time. None of these approaches resulted in a better final value for the objective function, and they, actually, significantly increased the computational time. Therefore, they were discarded.

The results obtained with this approach were very satisfactory. The solution was validated with an illustrative example for which the optimal solution was known, section 3.3.2.

As known a great improvement in the computational time was shown, even for a small problem as the illustrative example. For the whole case study this approach reached a minimum cost value of $2295\, monetary\ units$ in a surprising time of $36.7\, seconds$. This represents a massive improvement of the Martins' solution (2018), as his model reached a minimum value of $2520\, monetary\ units$ after a computation time of $11003.9\, seconds$. This represents a reduction of 8.93% in terms of costs and 99.7% in terms of computational time. These results are shown in *Figure 8*.

```
The minimum cost is : 360 monetary units
The minimum cost is : 720 monetary units
The minimum cost is : 1080 monetary units
The minimum cost is : 1180 monetary units
The minimum cost is : 1280 monetary units
The minimum cost is : 1380 monetary units
The minimum cost is : 1480 monetary units
The minimum cost is : 1750 monetary units
The minimum cost is : 1850 monetary units
The minimum cost is : 1995 monetary units
The minimum cost is : 2095 monetary units
The minimum cost is : 2195 monetary units
The minimum cost is : 2295 monetary units
```

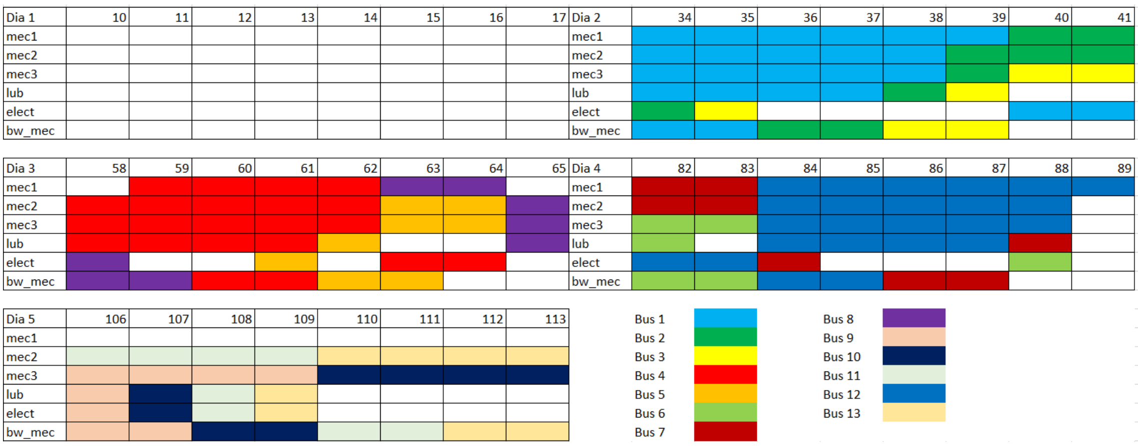
Time overheads:	
Progress graphs:	36.7s
Writing output:	0.0s
Pausing:	0.0s
Updating status:	36.7s

Figure 8 - Heuristic approach case study results

Given these results, it is fair to say that the main goal of these experiments, which was to optimize the maintenance costs and to drastically reduce the computational time was achieved mainly with a simple heuristic approach.

In *Figure 9* the scheduling proposed by this approach is shown.

Figure 9 - Heuristic approach case study scheduling



4.3. Introducing New Restrictions to the Original Problem

The main goal of introducing new restrictions to the original problem is to limit the research space of the problem, and thus having the model to test much less combinations in the search for the optimal solution.

This is achieved by "shaping the solution", i.e. by pointing out some aspects (ones more obvious than others) that would be very likely to be present in the optimal solution and allowing the solver to only test such solutions. These aspects or characteristics are inserted in the form of restrictions in the formulation of the problem.

The following restrictions were added to the Martins' model (2018) based on what is expected for the solution provided by that same model:

$$\sum_{b \in B} \sum_{d \in D} zd_{bd} = NB \tag{22}$$

which states that all the buses are assigned to only one day;

$$\sum_{m \in M} \sum_{w \in W} x_{1m10w} \geq 1 \tag{23}$$

$$\sum_{m \in M} \sum_{w \in W} x_{4m34w} \geq 1 \tag{24}$$

$$\sum_{m \in M} \sum_{w \in W} x_{12m58w} \geq 1 \tag{25}$$

These restrictions make sure that the three buses with the greatest need in terms of amount of work are scheduled in different days (which day is irrelevant) i.e. $Bus\ 1$ is scheduled for $day\ 1$ which starts at $time\ period\ 10$;

$$\sum_{d1 \in D: d1 \leq d} zd_{3d1} \geq zd_{6d}, \quad \forall\, d \in D \tag{26}$$

$$\sum_{d1 \in D: d1 \leq d} zd_{9d1} \geq zd_{10d}, \quad \forall\, d \in D \tag{27}$$

$$\sum_{d1 \in D: d1 \leq d} zd_{5d1} \geq zd_{11d}, \quad \forall\, d \in D \tag{28}$$

These restrictions define the order in which the buses that are due to the same maintenance activities and are the same type of vehicle, and therefore are interchangeable, are scheduled. The goal is to, instead of letting the model test and decide which goes first, decide for the model, avoiding that step, and in this way, reducing the size of potential feasible solutions to test.

$$\sum_{b\in B}\sum_{m\in M}\sum_{w\in W} x_{bmtw} \geq \sum_{b\in B}\sum_{m\in M}\sum_{w\in W} x_{bmt-1w}, \quad \forall\, t \in T{:}\, 10 > t \leq 17 \tag{29}$$

$$\sum_{b\in B}\sum_{m\in M}\sum_{w\in W} x_{bmtw} \geq \sum_{b\in B}\sum_{m\in M}\sum_{w\in W} x_{bmt-1w}, \quad \forall\, t \in T{:}\, 34 > t \leq 41 \tag{30}$$

$$\sum_{b\in B}\sum_{m\in M}\sum_{w\in W} x_{bmtw} \geq \sum_{b\in B}\sum_{m\in M}\sum_{w\in W} x_{bmt-1w}, \quad \forall\, t \in T{:}\, 58 > t \leq 65 \tag{31}$$

$$\sum_{b\in B}\sum_{m\in M}\sum_{w\in W} x_{bmtw} \geq \sum_{b\in B}\sum_{m\in M}\sum_{w\in W} x_{bmt-1w}, \quad \forall\, t \in T{:}\, 82 > t \leq 89 \tag{32}$$

$$\sum_{b\in B}\sum_{m\in M}\sum_{w\in W} x_{bmtw} \geq \sum_{b\in B}\sum_{m\in M}\sum_{w\in W} x_{bmt-1w}, \quad \forall\, t \in T{:}\, 106 > t \leq 113 \tag{33}$$

These restrictions, one for each day, define that the model should assign the work starting from the beginning of the day.

$$\sum_{b\in B}\sum_{m\in M} x_{bmtw} \geq L \times \sum_{b\in B}\sum_{m\in M} x_{bmtw-1}, \quad \forall\, t \in T, w \in W{:}\ CC_w = 1\ \cap w > 1 \tag{34}$$

In this restriction the workers with capacity $CC_w = 1$ are being assign in a predefined order, again saving computational time;

$$y_{w-1} \leq y_w, \quad \forall\, d \in D, w \in W{:}\ CC_w = 1\ \cap w > 1 \tag{35}$$

A second worker w with competence c is only assigned on day d if the previous worker w with competence c is already assigned on that day d;

$$\sum_{b\in B}\sum_{m\in M}\sum_{w\in W} \mathrm{x_{bmtw}} \geq L \times \sum_{b\in B}\sum_{m\in M}\sum_{\mathrm{w}\in \mathrm{W}}\sum_{t\in T:t\leq 24} \mathrm{x_{bmtw}}, \quad t = 34 \tag{36}$$

$$\sum_{b\in B}\sum_{m\in M}\sum_{w\in W} \mathrm{x_{bmtw}} \geq L \times \sum_{b\in B}\sum_{m\in M}\sum_{\mathrm{w}\in \mathrm{W}}\sum_{t\in T:25\leq t\leq 48} \mathrm{x_{bmtw}}, \quad t = 58 \tag{37}$$

$$\sum_{b\in B}\sum_{m\in M}\sum_{w\in W} \mathrm{x_{bmtw}} \geq L \times \sum_{b\in B}\sum_{m\in M}\sum_{\mathrm{w}\in \mathrm{W}}\sum_{t\in T:49\leq t\leq 72} \mathrm{x_{bmtw}}, \quad t = 82 \tag{38}$$

$$\sum_{b\in B}\sum_{m\in M}\sum_{w\in W} \mathrm{x_{bmtw}} \geq L \times \sum_{b\in B}\sum_{m\in M}\sum_{\mathrm{w}\in \mathrm{W}}\sum_{t\in T:73\leq t\leq 96} \mathrm{x_{bmtw}}, \quad t = 106 \tag{39}$$

There are only activities assigned for a determined day if the previous day already has activities;

$$\sum_{b\in B} zd_{\mathrm{bd}-1} \leq L \times \sum_{b\in B} zd_{\mathrm{bd}}, \quad \forall\, d \in D{:}\, d > 1 \tag{40}$$

$$\sum_{w\in W} y_{\mathrm{wd}-1} \leq L \times \sum_{w\in W} y_{\mathrm{wd}}, \quad \forall\, d \in D{:}\, d > 1 \tag{41}$$

A bus b (…) or a worker w (…) is only assigned to day d, if it was already assigned for the previous day.

With the introduction of all these new restrictions, a simplification to the problem is accomplished, by guiding the solver to the configuration of an optimal solution. Therefore, by following such an approach, some decisions are made for the model, in some of them the answer is being given to the model, such as in restriction (22). And in others, where either choice results in the same outcome, the choice is made for the model avoiding many combinations being tried when it is known that the outcome would be the same.

The results of this approach were positive. A better solution was found in a much smaller computational time, and with a smaller optimality gap. The solution found was of $2510\ monetary\ units$, which results in a reduction of 0.4% from the $2520\ monetary\ units$ found by Martins (2018). This solution was found in $1287.5\ seconds$, a decrease of 88.3% in computational time when compared to the $11003.9\ seconds$ from 2018. Regarding the optimality gap the values presented now was of 16.14%, minus 5.06% than the 21.20% obtained before.

Figure 10 - Restrictions approach case study results

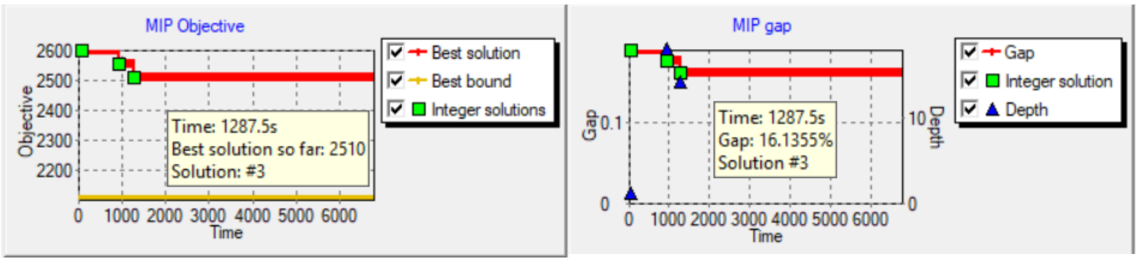


Figure 11 - Restrictions approach case study scheduling

4.4. Comparison of Results

In this section an analysis of the results obtained in both the approaches is executed by comparing them to the ones obtained by Martins (2018). A review of the analysis done by Martins (2018) to his results showed that when the model was ran with a cU greater than $250\ monetary\ units$, despite the objective value being higher, the optimization of the problem promised a better allocation of the elements. As in the heuristic approach an update of the cU was made to $260\ monetary\ units$ in the solver, although the result exhibited was adapted to $100\ monetary\ units$, it was deemed relevant to provide an analysis cadent of the same adaptation to the Martins model (2018). Therefore, the results from the model ran for cU equal to 260 and $275\ monetary\ units$, the best looking of the ones Martins (2018) tested, were included in this analysis. Those results were examined with the $100\ monetary\ units$' adaptation.

Firstly, an analysis of the final solution, computational time and optimality gap is conducted, followed by an analysis of the weight of the cost components. After that statistics regarding the bus unavailability, the days in which maintenance activities were assign, and the total working days of each type of worker, are presented. Lastly, the money loss in working hours vs paid hours is evaluated.

Table 8 - Objective value, computational time and optimality gap analysis

Model	Objective Value (cU=100, adapted)	Improv to Original	Comp Time (s)	Improv to Original	Optimaility gap	Improv to Original
Martins	2520	-	11003.9	-	21.20%	-
Restrictions	2510	0.40%	1287.5	88.30%	16.14%	5.06%
Martins (cU=260)	2465	2.18%	7373.1	33.00%	12.11%	9.09%
Martins (cU=275)	2295	8.93%	6840.6	37.83%	5.16%	16.04%
Heuristic	2295	8.93%	36.7	99.67%	5.16%	16.04%

From *Table 8* it is observed that both the Heuristic approach and the Martins ($cU = 275$) present the best solution to the problem with a value of $2295\ monetary\ units$. The optimality gap from Martins ($cU = 275$) suggests that this solution is close to optimal, and since the Heuristic approach presents the same value it is fair to assume that this applies to it too. This value represents an improvement of 8.93% from the original model and the optimality gap is reduced by 16.04%.

But where the Heuristic approach comes as an isolated champion in the computational time category, an astonishing reduction of 99.67% representing a value of $36.7\ seconds$ makes this approach the most efficient one. It is important to notice that both the approaches developed in this dissertation presented great reductions in computational time even when compared with adapted Martins models. This was one of the main goals of this research.

It is also relevant that both the cU adaptations of the Martins model (2018) present better solutions than the original one.

Table 9 - Cost component weight analysis

Cost Component	Martins		Restrictions		Martins (cU=260)		Martins (cU=275)		Heuristic	
	Value	% total	Value	% total	Value	% total	Value	% total	Value	% total
O_1	1120	44.4%	1210	48.2%	1085	45.5%	995	43.4%	995	43.4%
O_2	1400	55.6%	1300	51.8%	1300	54.5%	1300	56.6%	1300	56.6%

Regarding the cost components, considering the analysis performed by Martins (2018), an increasing of the O_2 component's weigh should evolve with the increase of the solution's optimality. This is not visible in *Table 9* from the Martins model (2018) to the Restrictions approach, this is due to a leap from $14\ assigned\ buses$ (one is repeated) to 13, this is evident in *Table 10.* Although, from the Restrictions approach forward it is possible to identify the evolution described above. It is also interesting to notice that the O_2 component plays a heavier role in all these models.

Table 10 - Bus availability and worker assignment

Aproach	Number of times the buses are unavailable	Number of Days	Number of work days by competence			
			mec (3)	lub	ele	bw
Martins	14	5	12	5	4	5
Restrictions	13	5	13	5	5	5
Martins (cU=260)	13	5	12	5	5	5
Martins (cU=275)	13	4	11	4	4	4
Heuristic	13	4	11	4	4	4

In *Table 10* it is evidenced the leap addressed above. The information provided allows an understanding of effects of the optimization, by the number of days and the work days by competence it is possible to observe that the best solutions present a more compact scheduling. It is also noticeable that if a new restriction to the restriction model, imposing that the solution only had four days, was added, a better solution could be achieved.

Table 11 - Money loss, working hours vs paid hours

Worker	Day 1	Day 2	Day 3	Day 4	Day 5	working hours/ week	paid hours/ week	hours lost %	cDw monetary units	cost/week monetary units	money loss/week (monetary units)
Original Model											
mec1	7	6	8	8	8	37	40	7.50%	45	225	16.88
mec2	6	4	8	8	0	26	32	18.75%	45	180	33.75
mec3	5	0	8	8	0	21	24	12.50%	45	135	16.88
lub	3	3	6	6	4	22	40	45.00%	35	175	78.75
elect	5	3	4	4	0	16	32	50.00%	45	180	90.00
bw_mec	6	6	6	6	2	26	40	35.00%	45	225	78.75
Restrictions									Total	1120	315.00
mec1	6	8	7	7	8	36	40	10.00%	45	225	22.50
mec2	5	4	7	5	7	28	40	30.00%	45	225	67.50
mec3	8	6	6	0	0	20	24	16.67%	45	135	22.50
lub	5	5	5	3	4	22	40	45.00%	35	175	78.75
elect	3	3	3	3	4	16	40	60.00%	45	225	135.00
bw_mec	4	4	4	6	8	26	40	35.00%	45	225	78.75
Martins (cU=260)									Total	1210	405.00
mec1	8	6	5	8	0	27	32	15.63%	45	180	28.13
mec2	7	7	7	8	0	29	32	9.38%	45	180	16.88
mec3	7	7	8	0	6	28	32	12.50%	45	180	22.50
lub	6	5	5	4	2	22	40	45.00%	35	175	78.75
elect	4	3	3	4	2	16	40	60.00%	45	225	135.00
bw_mec	6	4	4	8	4	26	40	35.00%	45	225	78.75
Martins (cU=275)									Total	1165	360.00
mec1	0	8	8	8	8	32	32	0.00%	45	180	0.00
mec2	0	7	8	8	7	30	32	6.25%	45	180	11.25
mec3	0	7	8	0	7	22	24	8.33%	45	135	11.25
lub	0	5	7	4	6	22	32	31.25%	35	140	43.75
elect	0	3	5	4	4	16	32	50.00%	45	180	90.00
bw_mec	0	4	8	8	6	26	32	18.75%	45	180	33.75
Heuristic									Total	995	190.00
mec1	0	8	6	8	0	22	24	8.33%	45	135	11.25
mec2	0	8	8	7	8	31	32	3.13%	45	180	5.63
mec3	0	8	8	7	8	31	32	3.13%	45	180	5.63
lub	0	6	6	6	4	22	32	31.25%	35	140	43.75
elect	0	4	4	4	4	16	32	50.00%	45	180	90.00
bw_mec	0	6	6	6	8	26	32	18.75%	45	180	33.75
									Total	995	190.00

Table 11 provides some interesting data, though no pattern is identified, the two best solutions, the most "compact", as seen on *Table 9*, present the lower value of money loss per week. This value, $190\,monetary\ units$, represents a 39.7% drop in relation to the original value, $315\,monetary\ units$.

After analysing all these results, it is concluded that the heuristic approach is superior to all the other studied models and is by far superior when compared to the original model, presenting great improvements in relation to it, especially regarding computational time.

5. Conclusions and Further Research

This final chapter presents the main conclusion of the dissertation, identifies several limitations and looks at possible future steps for further research.

5.1. Conclusions

The main objective of the present dissertation was to optimize in terms of computational time, optimality gap and final solution, the model created in 2018 by my former colleague Rodrigo Arrais Martins (2018) on the bus maintenance scheduling and applied to the Carris case study. As stated in Chapter 3, previous results in Martins (2018), though satisfying and ground breaking, still exhibited a large margin for improvement. The reported optimality gap of 20.15% after a computational time of $13\ hours$ is far from being ideal, and the present research work had the challenge to try to improve the computational time, optimality gap and final solution.

The initial idea was to work with parallel solving mechanisms in order to save computational time. A model was created from scratch, though it was based on the Factory Production example (Colombani & Heipcke, 2011) and using the constraints, decision variables and objective function proposed by Martins (2018), as well as the Dantzig-Wolfe decomposition for parallel solving. Other types of parallel solving mechanisms were considered, as its characteristics were seemingly aligned to those of the present problem, the Dantzig-Wolfe was the selected mechanism. The implementation of this model was a long process and was not possible to validate it for neither the illustrative example nor the real case. As it was explained in section 4.1. Upon reaching this conclusion and given that creating a new model for a different mechanism would take possibly the same amount of time as for the Dantzig-Wolf decomposition, two different approaches were proposed. The first one used a heuristic approach and restructured the Martins' model (2018) in a way that it would reduce the amount of free decision variables and combinations to test. This approach was explained in detail in section 4.2. The second one consisted of introducing new constraints to Martins' model (2018) based on characteristics of the optimal solution that could be expected a priori, and thus reducing the number of nodes the solver had to go through in the branch and bound algorithm, while improving the computational time. This alternative method was described in section 4.3. Both these new models using alternative formulations were then validated for the illustrative example and for the real case study, achieving a better final solution in a shorter computational time than Martins' original model.

Regarding the second approach, though this would not be the ideal way to approach the problem, one can conclude that the goal of this dissertation was achieved, as it was shown that it is possible to improve Martins' solution. Its results and discoveries will be very useful for next researchers who approach this subject. Moreover, it is important to mention that the way the additional constraints were generated can be easily adapted to other maintenance scheduling problems, and thus they serve as a basis for further improvements in other problems.

As for the first approach, the results were much more satisfactory, there was a significant improvement in the two main aspects of this dissertation, especially regarding the computational time. As for the optimality of the solution, diverging from the Martins' model (2018) and the previous approach where it is known that the solutions are not optimal, here one cannot know for sure, as it is a heuristic method. Nevertheless, it is known that for the present problem, there has not been found a solution better than the one obtained with such heuristic. Moreover, for the illustrative example, the optimal solution was achieved using the proposed heuristic approach.

5.2. Limitations

There are a few limitations that should be discussed in this work. The three conducted approaches presented limitations of their own. One of them, though could, actually, improve the optimality gap and computational time, is far from ideal. Other has some limitations regarding programming, which made it impossible to be validated. And the last one has no way to prove its optimality.

One important limitation with the approach of introducing more constraints (discussed in more detail in section 4.3) is the fact that it relies completely on the experience of the maintenance planners and knowledge, expectations and assumptions of the programmer. This means that its adaptation to other problems, different from the one studied in this paper, would require some testing and evaluation, prior to its implementation, in other words, the solution found would not be very useful for different problems with different characteristics. Nevertheless, the spirit and arguments behind the creation of additional constraints can be adapted to other problems.

Another limitation of the final approach is that there are options that are disregarded which could represent a better solution than the one found. It is known that for the illustrative example the solution is optimal, because the same optimal value as the Martins' model is achieved, and in less computational time. But for the case study there is no way to know if this is true. Similarly, there is no way to know that this is not true, even if the optimality gap is really close to zero. It is only known that the new solution found is better than the one found by Martins (2018), and it is largely based on it. What would be ideal is to have a solution that relies only on the constraints and restrictions one can identify prior to the knowledge of any solution. These are mainly the restrictions in the Martins' model.

Although in the present document the Dantzig-Wolfe decomposition algorithm was first perceived as a solution to this problem, it presented its own limitations, which were discussed in section 4.1. They were mainly related to the fact that the problem had binary decision variables and that the linking/associated constraint did not vary in the same way as the decision variables. Due to these issues, it was not possible to validate this approach to the bus maintenance scheduling problem formulated by Martins (2018).

The heuristic approach presents a big limitation, which is to know whether the solution found is optimal. Contrary to all the other approaches (except for the restrictions' one) there is no mechanism that guarantees the optimality, or not, of the solution. In the Martins' model it can be assured by the optimality gap, which is also why it is known that the one found in 2018 is not optimal. In the Dantzig-Wolfe decomposition there is Phase 2 to assure optimality. In the restrictions' approach, it can be assured through the optimality gap, one just needs to be aware that not all options were explored. But for the heuristic approach there is no mechanism proving the optimality of the solution. It is only possible to assure that the solutions found for each bus, within the spaces available for each one, were optimal.

5.3. Further Research

There are several ways in which further research on this topic can be pursued. The model developed in the first approach can potentially be modified in order to be feasible when working with problems with binary variables. This can be achieved by disregarding the weights approach and by developing a new mechanism to generate parts of every proposal without compromising its integer nature. A possibility would be to develop a mechanism which can grab any variable from any proposal, while guaranteeing that every restriction from the subproblem is fulfilled. There is a chance that the pricing data to evaluate which variables need to change can still be used, but in case it does not fully work, an option where the slack of the linear control is evaluated, and the "Pricing" value assigned to the variables whose slack is zero, could be a potential option.

Regarding the problem with changing variables, a relation between the constraint and the decision variable x_{bmtw} should be studied in order to adapt the constraint, or even, if needed, the variable to achieve a more straightforward adaptation of the variables needed and a much better computational time. It is possible to use an adaptation of the Dantzig-Wolfe Decomposition in this problem, though there is no evidence that supports the claim that this new model will be more efficient than the one developed by Martins (2018).

Another way to continue improving the solution of this problem would be to try other decomposition types, either within the Dantzig-Wolfe technique, decomposing by bus or worker as discussed before, or a new mechanism of parallel solving. Trying another mechanism might be the best option. After a dead end was reached in the Dantzig-Wolfe algorithm, other ones were considered, though not fully pursued. For example, the Cutting Plane Algorithm (Colombani & Heipcke, 2011) seems to be a well-suited choice for this problem as well. This algorithm identifies the violated inequalities and adds them as cuts to the problem, in a Cut-and-Branch kind of way. Moreover, it seems to avoid the problems encountered in the computational experiments described in this dissertation, though other ones might arise. It also might be interesting to explore other parallel solving mechanisms.

Regarding the second approach to the problem, other directions for further research would be adding restrictions to the reformulated model. There are obviously a lot of possibilities here, some of the most general ones that will eliminate most nodes have already been applied, but more specific ones can be added. However, as more specific they are, it might not only guide the problem to a solution that is already known to be a good one, but the chances of the model to improve the solution might be extremely low. There are other limitations to this approach as stated in the previous section. Improvements to this approach should be pursued if the research is related with this exact same case study. Otherwise, it might still work, but new restrictions will have to be found related specifically to the characteristics of the solution to the problem under analysis.

Regarding the heuristic approach, it is imperative that a mechanism to verify the optimality of the solution is implemented. An idea would be to keep trying different types of iterative approaches to optimize the previous result.

Although these experiments were conducted only for the bus maintenance scheduling problem, formulated and solved by Martins (2018), it should be highlighted that that case study only referred to one depot. The bus operating company operates other depots in the Lisbon area, and an implementation of the parallel solving mechanisms, including the Dantzig-Wolfe, could be more advantageous and less complex in terms of adaptations. An interesting way to decompose the problem would be by depot and the global constraints could represent bus interactions and personnel exchanges between depots. In this way, the binary problems would be solved within the sub-models avoiding one of the limitations stated above. In terms of computational time, this would be a much more complex problem, with many more variables and it would be more difficult to get a good solution within a shorter amount of time.

6. References

Adonyi, R., Heckl, I. & Olti, F., 2013. Scheduling of bus maintenance by the P-graph methodology. *Optimization and Engineering,* Volume 14(4), pp. 565-574.

Bazargan, M., 2015. An optimization approach to aircraft dispatching strategy with maintenance cost – A case study. *Journal of Air Transport Management,* Volume 42, pp. 10-14.

Colombani, Y. & Heipcke S., 2011. Multiple models and parallel solving with Mosel. *FICO™ Xpress Optimization Suite whitepaper.*

Dantzig, G.B. & Wolfe P., 1959. Decomposition Principle for Linear Programs. *The Rand Corporation, Santa Monica, California*.

Doganay, K. & Bohlin, M., 2010. Maintenance plan optimization for a train fleet. *WIT Transactions on The Built Environment,* Volume 114(2), pp. 349-358.

European Commission, 2011. 'Horizon 2020 Work Programme 2016 - 2017: 11. Smart, green and integrated transport', 2017(July 2016), p. 129. Available at: https://ec.europa.eu/programmes/horizon2020/en/h2020-section/smart-green-and-integrated-transport.

European Commission, 2016. Horizon 2020 Work Programme 2016 - 2017: 11. Smart, green and integrated transport', 2017(July 2016), p. 129. Availa.

Haghani, A., Banihashemi, M. & Chiang, K., 2003. A comparative analysis of bus transit vehicle scheduling models. *Transportation Research Part B: Methodological,* Volume 37(4), pp. 301-322.

Haghani, A. & Shafahi, Y., 2002. Bus maintenance systems and maintenance scheduling: model formulations and solutions. *Transportation Research Part A: Policy and Practice,* Volume 36(5), pp. 453-482.

Kamlu, S. & Laxmi, V., 2016. Implementation of Fuzzy Model for Maintenance Scheduling of Vehicles Based on Monte Carlo Simulation and Geographical Information System. *IETE Journal of Research,* Volume 63(2), pp. 225-237.

Martins, R., 2018. Planning maintenance in bus operating companies: Carris Case Study. *Master Thesis in Mechanical Engineering, IST.*

Öztop, H., Eliiyi, U., Eliiyi, D. & Kandiller, L., 2017. A Bus Crew Scheduling Problem with Eligibility Constraints and Time Limitations. *Transportation Research Procedia,* Volume 22, pp. 222-231.

Pour, S. et al., 2017. A hybrid Constraint Programming/Mixed Integer Programming framework for the preventive signaling maintenance crew scheduling problem. *European Journal of Operational Research.*

Sriram, C. & Haghani, A., 2003. An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A: Policy and Practice,* Volume 37(1), pp. 29-48.

Tebboth, J. R., 2001. A Computational Study of the Dantzig-Wolfe Decomposition. *Thesis submitted for the degree of Doctor of Philosophy in the University of Buckingham*.

Zhou, R., Fox, B., Lee, H. & Nee, A., 2004. Bus maintenance scheduling using multi-agent systems. *Engineering Applications of Artificial Intelligence,* Volume 17(6), pp. 623-630.

FICO® Xpress Optimization Suite, Educational license, http://www.fico.com/en/products/fico-xpress-optimization-suite

7. Appendix

A1 – CARRIS Case Study information

1 - The buses that will perform preventive maintenance

b	Bus name	Bus Type	Bus model	Maintenance Type
1	4238	Mercedes Benz OC 500	Standard	R5 C2
2	1629	Volvo B7L	Standard	R4
3	1706	Volvo B7R	Standard	R3
4	1757	Volvo B7R MKIII	Standard	R5 C2
5	4248	Mercedes Benz OC 500	Standard	R3
6	1705	Volvo B7R	Standard	R3
7	4221	Mercedes Benz OC 500	Standard	R4
8	1707	Volvo B7R	Standard	R4
9	1758	Volvo B7R MKIII	Standard	R4
10	1748	Volvo B7R MKIII	Standard	R4
11	4245	Mercedes Benz OC 500	Standard	R3
12	4264	Mercedes Benz OC 500	Standard	R5 C2
13	4625	Mercedes Benz Citaro G	Articulated	R4

2 - Maintenance tasks performed by preventive maintenance team at the Musgueira facility

Maintenance activities	Period	Performed tasks
R3	Every 15.000 km	The tasks are limited to checks (fluid levels, leaks, gaps, wear, condition) to the various systems, usually without disassembly.
R4	Every 30.000 km	In addition to that considered in R3, it includes the replacement of engine oil, filters (air and fuel), lubrication and brake test
R5	Every 60.000 km	In addition to that considered in R4, it also involves the replacement of oil and gearbox, differential and the hydraulic steering circuit, tuning of engine valves, the pneumatic brake system dehumidifier and the drive belt of the air conditioning.
R5C2	Every 120.000 km	In addition to that considered in R5, it also considered the replacement of the alternator belts, coolant system liquid and wheel hub verification (with disassembly).

3 - Amount of work for preventive inspections per vehicle type, worker competence, and type of maintenance

v_type_v (model)	c_type_c	ma_type_m			
		R3	R4	R5	R5 C2
Mercedes Benz OC 500 (standard)	Mechanic	4	4	8	16
	Lubricator	1	1	4	4
	Electrician	1	1	2	2
	Bodywork Mechanic	2	2	2	2
	total (h / ma_type)	**9**	**9**	**18**	**26**
Mercedes Benz Citaro G ("Articulated")	Mechanic	Not applicate	4	8	14
	Lubricator		1	4	4
	Electrician		1	2	2
	Bodywork Mechanic		2	2	2
	total (h / ma_type)	**0**	**8**	**16**	**22**
Volvo B7L (standard)	Mechanic	4	6	6	22
	Lubricator	1	1	4	4
	Electrician	1	1	2	2
	Bodywork Mechanic	2	2	2	2
	total (h / ma_type)	**8**	**10**	**14**	**30**
Volvo B7R (standard)	Mechanic	2	4	8	14
	Lubricator	1	1	4	4
	Electrician	2	2	2	2
	Bodywork Mechanic	1	1	2	2
	total (h / ma_type)	**6**	**8**	**16**	**22**
Volvo B7R MKIII (standard)	Mechanic	Not applicate	4	12	14
	Lubricator		1	4	4
	Electrician		1	2	2
	Bodywork Mechanic		2	2	2
	total (h / ma_type)	**0**	**8**	**20**	**22**

4 - Sets of the mathematical model

Sets	**Values**
B	{1, …, 13}
M	{1, …, 4}
D	{1, …, 5}
T	{1, …, 120}
V	{1, …, 5}
W	{1, …, 6}
C	{1, …, 4}

5 - Number of work hours required for the different bus and competence units

Bus unit	1	2	3	4	5	6	7	8	9	10	11	12	13
Bus name	4238	1629	1706	1757	4248	1705	4221	1707	1758	1748	4245	4264	4625
v_typev	OC 500	B7L	B7R	B7R III	OC 500	B7R	OC 500	B7R	B7R III	B7R III	OC 500	OC500	Articulated
ma_typem	R5 C2	R4	R3	R5 C2	R3	R3	R4	R4	R4	R4	R3	R5 C2	R4
c_typec	The amount of work hours G_{bmvc}												
Mec	16	6	2	14	4	2	4	4	4	4	4	16	4
Lub	4	1	1	4	1	1	1	1	1	1	1	4	1
Elect	2	1	1	2	1	1	1	1	1	1	1	2	1
Bw	2	2	2	2	2	2	2	2	2	2	2	2	2

6 - Information about unavailable work window for the 5-day planning period

S_b / Bus unit (b)	1st operation shift without P. Maint.		2nd operation shift without P. Maint.		3rd operation shift without P. Maint.		4th operation shift without P. Maint.		5th operation shift without P. Maint.		6th operation shift without P. Maint.	
	$ta1_b$	$ta2_b$	$tb1_b$	$tb2_b$	$tc1_b$	$tc2_b$	$td1_b$	$td2_b$	$te1_b$	$te2_b$	$tf1_b$	$tf2_b$
1	1	9	18	33	42	57	66	81	90	105	114	120
(…)	1	9	18	33	42	57	66	81	90	105	114	120
13	1	9	18	33	42	57	66	81	90	105	114	120

7 - Information about preventive maintenance crew

w	w_typew	cDw	NWtc
1	Mechanic_1	40	3
2	Mechanic_2	40	
3	Mechanic_3	40	
4	Lubricator	35	1
5	Electrician	50	1
6	Bodywork Mechanic	50	1

8 - Constants of the mathematical model

Constants	Description	Units	Values
cU	Bus unavailability costs	Monetary units	100
NT	Total number of hours	Working hours	120
ND	Number of days	Working days	5
NW	Number of workers	-	6
NML	Number of maintenance lines	-	7
SL	Number of special lines (Bus type Articulated)	-	1

A2 - Technical planning of the case study of CARRIS (Martins)

Day 1	10	11	12	13	14	15	16	17
mec1								
mec2								
mec3								
lub								
elect								
bw_mec								

Day 2	34	35	36	37	38	39	40	41
mec1								
mec2								
mec3								
lub								
elect								
bw_mec								

Day 3	58	59	60	61	62	63	64	65
mec1								
mec2								
mec3								
lub								
elect								
bw_mec								

Day 4	82	83	84	85	86	87	88	89
mec1								
mec2								
mec3								
lub								
elect								
bw_mec								

Day 5	106	107	108	109	110	111	112	113
mec1								
mec2								
mec3								
lub								
elect								
bw_mec								

Bus 1
Bus 2
Bus 3
Bus 4
Bus 5
Bus 6
Bus 7
Bus 8
Bus 9
Bus 10
Bus 11
Bus 12
Bus 13

A3 - Technical planning of the case study of CARRIS (Martins 260)

Day 1	10	11	12	13	14	15	16	17	Day 2	34	35	36	37	38	39	40	41
mec1									mec1								
mec2									mec2								
mec3									mec3								
lub									lub								
elect									elect								
bw_mec									bw_mec								

Day 3	58	59	60	61	62	63	64	65	Day 4	82	83	84	85	86	87	88	89
mec1									mec1								
mec2									mec2								
mec3									mec3								
lub									lub								
elect									elect								
bw_mec									bw_mec								

Day 5	106	107	108	109	110	111	112	113
mec1								
mec2								
mec3								
lub								
elect								
bw_mec								

Bus 1
Bus 2
Bus 3
Bus 4
Bus 5
Bus 6
Bus 7
Bus 8
Bus 9
Bus 10
Bus 11
Bus 12
Bus 13

A4 - Technical planning of the case study of CARRIS (Martins 275)

Day 1	10	11	12	13	14	15	16	17	Day 2	34	35	36	37	38	39	40	41
mec1									mec1								
mec2									mec2								
mec3									mec3								
lub									lub								
elect									elect								
bw_mec									bw_mec								

Day 3	58	59	60	61	62	63	64	65	Day 4	82	83	84	85	86	87	88	89
mec1									mec1								
mec2									mec2								
mec3									mec3								
lub									lub								
elect									elect								
bw_mec									bw_mec								

Day 5	106	107	108	109	110	111	112	113
mec1								
mec2								
mec3								
lub								
elect								
bw_mec								

Bus 1		Bus 8	
Bus 2		Bus 9	
Bus 3		Bus 10	
Bus 4		Bus 11	
Bus 5		Bus 12	
Bus 6		Bus 13	
Bus 7			