A comparative evaluation of two modelling approaches to the personnel scheduling problem: a case study in the telecommunication sector

António Xavier da Costa Monteiro Salgado

Master's Dissertation

Supervisor at FEUP: Vera Lucia Miguéis Oliveira e Silva

U. PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Master's in Industrial Engineering and Management

2023-06-26

Abstract

One critical component of workforce management that plays a vital role in the success and efficiency of any organization is appropriate scheduling. Optimizing this task offers several benefits. Firstly, it helps to increase productivity by ensuring that the workforce is adequately staffed, avoiding over-staffing or under-staffing situations. By accurately forecasting workload and scheduling accordingly, organizations can optimize their resources and reduce costs, whilst improving their service levels and increasing sales. Secondly, appropriate scheduling plays a crucial role in compliance with labor laws and regulations. It helps organizations adhere to rules related to maximum working hours and rest periods. Compliance not only protects employees' rights but also mitigates legal risks and potential financial penalties for the organization. Finally, by considering employees' preferences and skills, companies can ensure equitable distribution of shifts and opportunities, contributing to a positive work environment.

This thesis aims to contribute to the existing body of literature by: (1) being the first work to implement a variant of a genetic algorithm in personnel scheduling; (2) performing a comparative evaluation on some of the methodologies that can be used to solve this problem; (3) understanding the trade-off between collaborators' satisfaction – by meeting all of their needs – and the performance of the obtained schedules, since the more demanding employees are, the fewer the possible scheduling alternatives are and the larger the impact on the observed measures of efficiency is; (4) being one the few works that implements the proposed solution in a practical setting.

The conclusions drawn from this work are the following: (1) the proposed genetic algorithm exhibits near-optimal solutions in a reduced time; (2) the genetic algorithm is better suited for larger stores; (3) companies should try to meet collaborators' requirements, as it does not appear to have a significant impact on the created schedules; (4) the application of the proposed solution in the telecommunications sector exhibited increases in productivity levels by 22% and a decrease in the average service time by 9%, with no statistically significant effect on the collaborators' sales efficacy. Feedback collected from the managers responsible for this task also highlighted improvements on the time and perceived effort to create the schedules for each worker.

Resumo

Um ponto crítico na gestão de recursos humanos, que desempenha um papel vital no sucesso e eficiência de qualquer organização, é a criação adequada de horários de trabalho. Esforços para otimizar esta tarefa oferecem múltiplos benefícios. Em primeiro lugar, ao garantir que a força de trabalho está adequadamente dimensionada, é possivel evitar situações de sobre ou sub-dimensionamento. Deste modo, as organizações podem otimizar recursos e reduzir custos, ao mesmo que tempo que melhoram o seu nível de serviço e potenciam mais vendas. Em segundo lugar, a criação correta de horários desempenha um papel crucial na conformidade com leis e regulamentos laborais. É possivel assim garantir não só a proteção dos direitos dos funcionários, como também reduzir os riscos legais e possíveis penalizações financeiras para a organização. Por último, ao considerar as preferências e skills dos funcionários, as empresas podem garantir a distribuição equitativa de turnos e oportunidades, contribuindo para um melhor ambiente de trabalho.

A seguinte tese procura contribuir para a literatura atual ao: (1) ser o primeiro trabalho a implementar uma variante do algoritmo genético no contexto descrito acima; (2) realizar uma avaliação comparativa das várias metodologias que podem ser utilizadas; (3) compreender o trade-off entre a satisfação dos trabalhadores – ao satisfazer todas as suas necessidades – e a qualidade dos horários criados; (4) ser um dos únicos trabalhos que implementa a solução proposta num contexto prático.

As conclusões do presente trabalho são as seguintes: (1) o algoritmo genético é capaz de apresentar soluções quase ótimas num reduzido espaço de tempo; (2) o algoritmo genético é mais indicado para lojas com um elevado número de colaboradores; (3) as empresas devem procurar satisfazer os requisitos dos colaboradores, dado que o seu efeito na qualidade dos horários criados é bastante reduzido; (4) a implementação da solução numa empresa de telecomunicações implicou aumentos nos níveis de produtividade em 22% e reduções no tempo médio de serviço em 9%, ao mesmo tempo que foi possível preservar a capacidade de venda dos colaboradores. O feedback recebido por parte dos gerentes responsáveis pela geração dos horários aponta para reduções significativas no tempo e esforço necessário para criar o horário de cada colaborador.

Acknowledgments

This work is the hallmark of a major chapter of my life that I will always cherish. On the one hand, I must confess I feel slightly nostalgic and emotional as I recognize that I am unlikely to step into a classroom again, have a lunch break at the university canteen with my friends or simply enjoy the university lifestyle. On the other, I look forward to what comes ahead, which I am sure will be just as interesting and challenging.

I would like to take a moment to appreciate everyone who has contributed to my success. First, and not in any particular order, I would like to thank my team at LTPLabs - Miguel, Paulo and Pedro - who have pushed me to develop the best output possible and provided me with many memorable moments.

Second, I would like to thank my supervisor Vera Silva for her guidance, support, and invaluable feedback throughout my dissertation. Additionally, a especial thank you is in order to all of my professors who have taught me so much, including and not limited to Bernardo Almada-Lobo, Henriqueta Nóvoa, José Faria, José Machado and Pedro Amorim.

Lastly, I would like to dedicate this work to my parents and brothers, whose love, patience, and sacrifices have been the driving force behind my success.

Thank you all for making this achievement possible.

"You can use all the quantitative data you can get, but you still have to distrust it and use your own intelligence and judgment"

Alvin Toffler

Contents

Acronyms and Symbols

AC	Absolute Count
ACD	Absolute Complete Deviation
AD	Absolute Deviation
AHP	Analytical Hierarchical Process
AST	Average Service Time
AWT	Average Waiting Time
BRKGA	Biased Random-Key Genetic Algorithm
BRKGA-GD	Gender-Defining Biased Random-Key Genetic Algorithm
BRKGA-MP	Multi-Parent Biased Random-Key Genetic Algorithm
CBC	COIN-OR Branch and Cut
CSV	Comma-Separated Values
DES	Discrete Event Simulation
DiD	Difference-in-Differences
FPCI	Fundamental Problem of Causal Inference
FPSI	Fundamental Problem of Statistical Inference
FTE	Full-Time Employee
GA	Genetic Algorithm
HHI	Herfindahl–Hirschman index
KPI	Key Performance Indicator
LP	Linear Programming
LU	Labor Utilization
MILP	Mixed Integer Linear Programming
MIP	Mixed-Integer Programming
NWB	Number Working Blocks
NPV	Net Present Value
POM	Production and Operations Management
RC	Relative Count
RCD	Relative Complete Deviation
RKGA	Random-Key Genetic Algorithm
SC	Set Covering
SP	Set Partitioning

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides an initial overview of the thesis structure: it starts with an introduction of the problem being addressed. It then proceeds with a short remark about the company on which the case study was built upon. Afterwards, the main objectives of this work are stated. Finally, more detail is provided on both the timeline associated with the project and the structure of the remainder of the dissertation.

1.1 Motivation

In the past few decades, personnel scheduling[1] - to be understood as "determining which staff should cover which shifts so that the demand for manpower is met at all times, taking into account organisational and legal rules" (Baskaran, 2016) - has been subject to intense study (Bergh et al., 2013) in areas as diverse as Health Care (Ngoo et al., 2008; Azaiez and Al Sharif, 2005), Airlines (Chu, 2007) and Retail (Özgür Kabak et al., 2008; Álvarez et al., 2020). This trend is driven by the idea, within service companies, that an efficient scheduling process can help maintain competitive levels (Bard et al., 2003). In fact, improving the allocation of work schedules ends up being the least costly way in many circumstances to deal with variable demand (Oke, 2013). However, this increased focus is being driven not by only by economic factors, but also by a need to maximize customer service, minimize waiting times (Allon et al., 2011) and optimize sales efficacy (Lu et al., 2013). Consequently, the ability to cut down on even a few percentage points on total labor needs while preserving existing customer service levels can represent immense savings for a company. As put by Valouxis et al. (2012), personnel scheduling is a rare situation in which good work schedules can benefit not only the employer, but also the employee.

1.2 The Company - Delta Corporations

This thesis came to fruition in the context of a consulting project with one of the largest Portuguese telecommunication providers, hereafter name "Delta Corporations". The industry, with the size of

[1]This problem can also be referred as shift scheduling, personnel shift scheduling, staff scheduling or rostering.

about 1.3 B€ ANACOM (2023), provides multiple services such as fixed and unlimited broadband and television and mobile subscriptions. It is a highly concentrated market, with a 4-firm concentration ratio of 99% and a Herfindahl–Hirschman index (HHI) of 3245[2].

Delta has multiple stores spread across the country that operate as touch points where clients can pay their bills, update their subscriptions, acquire new services, demand technical assistance or solve any other issue. Naturally, clients' arrival rate is not the same across time or between stores, with peaks and valleys occurring at regular periods of the day. In other words, the demand for collaborators at a store is not homogeneous throughout the days, weeks, and months. Consequently, it is ideal to have a varying sized workforce that matches the observed customer needs. The company has thus hired a third-party company to whom it specifies, every month, the necessary workforce for each one of its stores. It is then up to this provider to ensure that the number of full-time employees (FTEs) matches the order previously indicated by Delta. One should note that this order must be made 3 months in advance so that the firm has enough time to go through the contracting process.

While the solution for this problem appears to be pretty straightforward for small-sized teams, it is actually an NP-hard problem (Brucker et al., 2011). The consulting firm was thus hired to develop an end-to-end platform that would optimize this step and automatically indicate the monthly schedule for each collaborator, taking into consideration not only legal constraints, but also the preferences of the company and of the workers. This improvement in the company's operations would also contribute towards a better client service.

1.3 Goals

Prior to the consulting firm's intervention, the process of personnel scheduling was done manually by the manager of each store. This solution had large room for improvement for multiple reasons: firstly, the proposed schedules may be far from optimal; secondly, it could take several hours to multiple days to build a feasible schedule, depending on the number of collaborators and on their imposed restrictions; thirdly, there was not any standardization of processes across stores with regards to what particular constraints had to be met or what their order of importance was; fourthly, the entire process was prone to human error. This last aspect is quite relevant, since the lack of compliance with legislative procedures can have monetary penalties that, in some circumstances, are quite significant (Código do Trabalho, Chapter II, 2009). Consequently, the main objective of the consulting project was to develop a tool that would be able to tackle these issues adequately - optimization, duration, standardization and automatization.

Additionally, this thesis aims to contribute to the existing body of literature by: (1) being the first work to implement the Multi-Parent Biased Random-Key Genetic Algorithm (BRKGA-MP) in personnel scheduling; (2) performing a comparative evaluation on some of the methodologies that can be used to solve this problem (including BRKGA-MP); (3) understanding the trade-off

[2]The standards of US antitrust agencies define a market as highly concentrated if the index is above 1800 (Bellaflamme and Peitz, 2015)

between collaborators' satisfaction – by meeting all of their needs – and the performance of the obtained schedules, since the more demanding employees are, the fewer the possible scheduling alternatives are and the larger the impact on the observed measures of efficiency is; (4) being one the few works that implement the proposed solution in a practical setting. This work resulted in the submission of a paper to the journal of Productions and Operation Management (POM).

1.4 Dissertation Structure

This dissertation is divided into six different chapters. In Chapter 2, a literature review on personnel scheduling and its several implementations is presented to build a strong foundation from which all other chapters follow. Chapter 3 provides more detail into the methodology applied. The mathematical formulation for the problem is detailed in Chapter 4. Results are shown in Chapters 5 and 6: the former focuses on the optimization and evaluation of BRKGA-MP, while the latter discusses the practical implications of the solution in the company. Chapter 7 is a summary and a reflection on the findings of this thesis.

Chapter 2

Review of Contextual and Related Literature

This chapter seeks to explain key concepts that are used throughout the present document. It starts with some important labor allocation concepts that permeate the remainder literature review. Afterwards, some detail is presented on the process of personnel scheduling. Subsequently, a greater emphasis is given to both Mathematical Programming and Metaheuristics as relevant approaches to solve the aforementioned problem. If one is not familiar with these concepts, it is strongly advised to go through Appendix A for a theoretical introduction.

2.1 Labor Allocation

According to Ernst et al. (2004), the process of labor allocation is essentially comprised of two main steps. First of all, it is necessary to forecast demand based on historical data and to convert these needs into staffing levels. Secondly, a solution that converts these staffing requirements into the workers' schedules is developed. The authors also highlight that mathematical models and algorithms are typically very context-specific and hard to generalize, since requirements vary across industries.

Bhulai et al. (2008) offers a similar but more comprehensive view of the process, comprised of four key stages: (1) workload prediction, (2) staffing, (3) shift scheduling, and (4) rostering. Workload prediction is concerned with the prediction of the future amount of work a company will have to handle. Staffing translates this amount of work into the number of required collaborators such that a pre-specified service level is met[1]. Shift scheduling then generates shifts such that satisfy these staffing level. Finally, rostering refers to the pairing of shifts into rosters and the assignment of employees to the rosters. In this thesis, an integrated solution to steps (3) and (4) is proposed. One of the big disadvantages of treating each one of these steps individually relates to the fact that some iterations may be required to obtain a feasible solution. In other words, a more holistic

[1]Other methodologies can be considered, such as analyzing the trade-off of shortage costs and surplus costs – staff requirements should be increased as long as the marginal savings are superior to the marginal costs (Baker, 1976)

framework necessarily generates a final solution that meets both shift and personnel constraints. Additionally, Caprara et al. (2001) and Gamache and Soumis (2001) highlight improvements over previous solutions that treated these stages separately.

2.2 Personnel Scheduling

Shifting the focus to the third and fourth steps previously mentioned, it becomes apparent that a significant body of literature dedicated to the subjects of shift scheduling and rostering already exists. In fact, several general survey papers in these areas have been developed (Bechtold et al., 1991; Alfares, 2004; Ernst et al., 2004; Bergh et al., 2013; Özder et al., 2020). There are even studies on particular fields, which include surveys on airline crew scheduling (Gamache and Soumis, 2001; Kohl and Karisch, 2004; Deveci and Demirel, 2018) and nurse scheduling (Bradley and Martin, 1991; Burke et al., 2004; Ngoo et al., 2022). Their work reveals that the advantages of having a meticulously designed schedule are noteworthy and apparent. According to Ernst et al. (2004), organizations often require decision support tools to assist in creating schedules that ensure the right workers are available at the appropriate time and cost, while also promoting high levels of worker job satisfaction. Topaloglu and Ozkarahan (2004) emphasized that a high-quality schedule can result in a more content and, consequently, more productive workforce.

Ernst et al. (2004) further argued that as the modern workplace becomes more intricate and individual-focused, rather than centered on groups or teams, roster solutions will likely need to accommodate individual preferences. This implies that in order to develop well-designed work schedules, it is crucial to identify and consider each worker’s preferences for different shifts, thereby increasing the personalization of rosters. Additionally, there is some evidence suggesting that employees well-being is negatively correlated with turnover and positively correlated with job performance (Page and Vella-Brodrick, 2009). According to Cascio (2015), the cost of losing an employee can range between 1.5 and 2.5 times the departing employee’s annual salary. Therefore, even from a profit-maximizing perspective, it is optimal to make efforts to satisfy the requirements of workers.

2.2.1 Classification Methods

Several classification methods have been proposed for personnel scheduling problems, starting with Baker in 1976 - essentially, two categories can be defined: shift scheduling and days off scheduling. While in shift scheduling one is focused on a single working day, in days off scheduling there is a mismatch between the employee’s work week and the service facility’s operating week that must be taken into account. The most common version of this problem – which is present in the case study - involves 5-day work weeks and 7-days a week operation. The current thesis is focused on days off scheduling.

Personnel scheduling problems can also be categorized into different types based on the nature of the work and the factors influencing the scheduling process (Causmaecker et al., 2004):

1. permanence centred planning;
2. fluctuation centred planning;
3. mobility centred planning;
4. project centred planning.

The number of personnel needed, for example, in police stations and hospitals, is usually defined in advance. Corresponding personnel scheduling problems are called permanence centred. In contrast, in warehouses or distribution centers, as well as in call centers, fast food restaurants and postal services, personnel planning is based on fluctuating demand. Mobility centred planning occurs when duties involve transportation. Transportation companies such as airlines and railway companies face such planning problems. Project centred planning arises in companies which divide their work into projects. Typical examples include software development and consultancy.

Other classification methods are based on the solution method applied (Bechtold et al., 1991; Alfares, 2004), type of problem or application area (Ernst et al., 2004). In Bergh et al. (2013), three big groups where defined based on the solution method considered: (1) Mathematical Programming, (2) Metaheursitics and (3) Simulation. From (1), the Set Covering (SC) formulation presented by Dantzig (1954) is still the basis of many researchers that then adapt it to their needs[2]. From (2), most solutions are based on either Tabu Search or Genetic Algorithms (GAs). From (3), particular emphasis in given to Discrete Event Simulation (DES). The present document focuses both on Mathematical Programming and Metaheuristics.

On the one hand, Ernst et al. (2004) points out that Mathematical Programming solutions typically achieve the lowest cost solution. However, these solutions also present some drawbacks: namely, difficulties in expressing the constraints and objectives of the problem, and large efforts are required to properly implement the solution obtained. On the other hand, the popularity of Metaheuristics is related to their robustness, relative ease of implementation, and ability to deal with complex objectives (Zolfaghari et al., 2010). Nonetheless, a major drawback of these methods (including GA) is that they can neither provably produce optimal solutions nor provably reduce the search space (Bergh et al., 2013). Moreover, as most personnel scheduling problems are highly constrained, the feasible regions of the solution space can be disconnected (i.e. separated by an infeasible area) (Baskaran, 2016) - Meta-heuristics generally have difficulties in dealing with such situations, as they tend to get stuck in local optima (Burke et al., 2008).

In any case, all implementations have to deal with a given set of requirements, whilst trying to maximize or minimize some goal. The following subsections focus precisely on these themes.

2.2.2 Constraints

In the scheduling literature, constraints have been classified under multiple categories. Qu et al. (2009) highlights the distinction between soft and hard constraints. Hard constraints are necessarily enforced by the model and are typically associated with labour requirements that must be

[2]More detail on other formulations is present further in the Chapter.

complied with. They usually include restrictions such as maximum number of consecutive working days and maximum number of working days in a week or month[3] (Azaiez and Al Sharif, 2005). Soft constraints are desirable and are typically included in the model as penalties for any deviation from their ideal values. They includes restrictions such as appropriate balancing between morning, evening and night shifts (Azaiez and Al Sharif, 2005) and complete weekends (Burke et al., 2013). Miller and Pierskalla (1976) consider a similar grouping, which they label as feasibility set (hard) and non-binding constraints (soft). Additionally, they make the distinction between non-binding schedule pattern constraints and non-binding staffing level constraints. Pattern constraints deal with the sequence of the type of shifts that are performed by each worker, while schedule staffing level constraints are concerned with the ability to cover demand.

Bergh et al. (2013) does not focus on feasibility, but rather consider three different categories of constraints: (1) coverage, (2) time-related, (3) fairness and balance. (1) is concerned with having enough workers to meet demand in each time period. Papers consider this constraint to be hard or soft depending primarily on whether the number of available workers is assumed to be fixed or not. This decision is in turn dependant on the nature of the work being conducted. For instance, in hospitals, police stations or other related fields, it is crucial to make sure that all demand is being satisfied. This is probably not going to be the case in most service industries, such as retail. (2) deals with constraints that have an impact on the number of possible shift patterns that exist - Table B.1 of Appendix B details some the most relevant constraints considered. (3) tries to make sure that the workload is distributed evenly among all workers.

2.2.3 Objective Function

The objective function is used to measure the success of the created schedules. Typically, it can be divided into two big categories, based on whether the workforce is assumed to be fixed or not. When one is dealing with a varying-sized workforce, the main goal is to minimize the total number of employees or total staff costs, by taking into account factors such as wages, overtime costs, outsourcing costs, travel costs or even missed production costs (Bergh et al., 2013). When the workforce is defined *a priori*, the main focus is on minimizing deviations from soft constraints. In this scenario, the focus is solely on defining the order of importance in satisfying each restriction. Monetary cost based models can incorporate these penalty factors, but run into additional complexities given the need to set a monetary price for the violation of each constraint. However, they do run into additional complexities, as they have to set a monetary price for the violation of each constraint.

For both of these problem types, due to the large number of constraints (sometimes conflicting), it is rarely possible to achieve a perfect roster (Burke et al., 2013).

[3]These are just illustrative examples to ease interpretation. It may be the case that in some specific instance they are not hard constraints.

2.2.4 Evaluation Criteria for Effective Scheduling

The biggest problem with objective functions is that the interpretation of their value is context-dependent. Several authors have thus tried to establish criteria for an effective schedule. According to Causmaecker et al. (2004), these parameters should include evaluating the schedule's quality from both individual and organizational perspectives, assessing the differences between the planned schedule and its actual execution, incorporating personnel preferences, examining the consequences of schedule modifications, and considering the cost associated with executing the schedule. Oldenkamp (1996) proposes five factors for nurse scheduling:

1. Optimality factor: represents the degree to which expertise is distributed over the different shifts;
2. Completeness factor: represents the degree to which quantitative demands are met;
3. Proportionality factor: represents the degree to which each nurse has similar morning, evening and night shifts, as well as similar weekends off;
4. Healthiness factor: represents the degree to which the welfare and health care of the staff is taken into consideration;
5. Continuity factor: represents the degree to which there is some continuity in the crew during different shifts.

The problem with these criteria is that they are merely qualitative and do not translate directly into any measurable metric. In this regard, Jacobs and Bechtold (1993) stand out by considering Labor Utilization (LU), which is defined as:

$$LU = \frac{Total\_Demand}{Total\_Supply} X100 \tag{2.1}$$

The objective was to minimize the total number of hours of labor scheduled ($Total\_Supply$) in order to satisfy forecast labor requirements ($Total\_Demand$) without under-staffing. Therefore, the closer a solution was to 100%, the more efficient the allocation process was.

2.2.5 Labor Flexibility and its Impact on Effective Scheduling

In fact, Jacobs and Bechtold (1993) developed the aforementioned metric to understand the effect that labor scheduling flexibility had on the quality of the solutions created. By labour scheduling flexibility one includes all factors that have an effect on the number of possible schedules for any collaborator. They include shift scheduling alternatives, such as: (1) shifts may or may not start at any period during the day [4] (start-time flexibility); (2) shifts may or may not have different lengths within the day (shift-length flexibility) and (3) breaks may or may not be placed within a certain time window (break-placement flexibility). Days-off scheduling flexibility alternatives include: (4) it may or may not be possible to work a varying number of working days per week and (5) it

[4] This does not include period for which the end of the shift would be beyond the end of the operating day.

may or may not be possible to have days off on any day of the week (days-off flexibility). Tour scheduling flexibility alternatives include: (6) shifts may or may not start at different times across the assigned work days of the week (start-time float); (7) shifts may or may not be of different length across the assigned work days of the week (shift-length float); (8) shits may contain breaks at different periods of the day across the assigned work days of the week (break-placement float) and (9) there may be a smaller or larger number of work days in a week (tour-length flexibility). In their study, they focused on factors (1), (2), (3), (5), (6) and (9), subject to several demand patterns[5]. The main implication of their study is that all factors can have a significant positive or negative impact on the created schedules. In particular, they highlight that: (a) break-placement flexibility - allowing for breaks to be taken within a specific time window rather than a specific time - can be particularly effective; (b) part-time employees provide greater flexibility; (c) increasing the number of possible starting periods improves labor utilization; (d) having non-consecutive days-off only improved the models slightly; (e) total demand does not have a significant impact on the schedules created - smaller sized-teams can obtain similar results to larger labor forces and (f) large fluctuations in the hourly demand have a significant effect. (a), (b) and (c) were actively implemented in the model proposed in this thesis.

2.3 Proposed Solutions

As previously mentioned, this thesis focuses on using Mathematical Programming and Metaheuristics to adress personnel scheduling. Namely, it provides a comprehensive discussion of two methods: Goal Programming and GAs. The selection of these methods is partly motivated by their previous achievements. For instance, in a recent survey conducted by Ngoo et al. (2022) regarding nurse rostering, Goal Programming solutions emerged as top performers in various competitions with benchmark datasets. Notably, these solutions secured the first place in the second international rostering competition dataset, which was conducted in 2014.

Additionally, GAs have demonstrated notable success when compared to alternative approaches in solving personnel scheduling problems, as highlighted in the study by Qu et al. (2009). Furthermore, population-based algorithms are quite able to explore search spaces as they deal with multiple candidate solutions (Behesti and Shamsuddin, 2013). Finally, GAs do not need to calculate the rate of change of the fitness function because the natural selection process is determined by the fitness of each chromosome (Goldberg, 1989). It should be noted, however, that they do tend to converge slower than other methods, with results varying largely from one run to another.

The next subsections will also make it clear that there is still room for improvement in both of these methods. The aim is thus to define state-of-the art solutions.

[5]These included varying length days, varying average requirements and varying requirement amplitudes.

2.3.1 Mathematical Programming Approaches

Scheduling can typically be modelled as a Set Covering or a Set Partitioning (SP) problem (Yan and Tu, 2002). The biggest difference resides on whether demand is to be exactly met (SP) or not (SC). SC models are the standard choice for most problems since they allow for extra coverage and are harder to originate infeasible solutions (Deveci and Demirel, 2018).

Azaiez and Al Sharif (2005) propose a Goal Programming solution for nurse scheduling. The formulation for their problem is quite similar to the one proposed in the scope of this master thesis, where the decision variable is binary, indicating whether a worker is assigned or not to a particular shift. Additionally, and unlike many other formulations, the staffing decision has already been made and it is not possible to change the number of workers. Therefore, the main objective is not necessarily to minimize costs, but rather to satisfy daily demand and the requirements of both the hospital and the nurses. Regarding the former, the solution specifies the maximum number of consecutive days of work and the percentage of night shifts, among others. Regarding the latter, requests were done on the minimum number of weekends and isolated days off and days on. However, the model does have some limitations: on the one hand, the computational time was quite long for larger instances, requiring sub grouping as a solution to deal with this problem. On the other, it did not include some relevant constraints, including days off, vacation requests and lunch breaks. The number of possible shifts was also reduced to only two possible combinations - either a morning or night shift.

Gordon and Erkut (2004) also propose a Goal Programming solution for the Edmonton Folk Festival - they decompose the problem into scheduling and rostering. Their objective function and restrictions are difficult to generalize to other problems, since the authors are focused on maximizing collaborators' satisfaction, as the entire operation is volunteers based.

Chu (2007) decomposes personnel scheduling in airports into three stages. The solution differs from the one proposed in several fronts. As previously mentioned, by dealing with shift scheduling and rostering, the process could be iterative. Another important difference is that Goal Programming models are preemptive. Finally, the nature of their restrictions is also quite simplistic - it does not take into consideration any requests from staff.

Topaloglu (2006) considers the Analytical Hierarchical Process (AHP) as a statistical tool to better assign weights to the deviations of the objective function in resident scheduling. Similarly to the problem being introduced, staffing costs do not play a relevant role. They also compare the quality of their solution to the manual approach that was being used.

Lin et al. (2012) uses information on (1) suitability, (2) compatibility and (3) fondness to create the schedules. That is: (1) tries to best match a worker's abilities to the existing tasks; (2) attempts to maximize the affinity of collaborators working at the same time and (3) makes efforts to ensure that everyone is content with the shift they are assigned.

Li et al. (2012) contributes to the existing literature by using a falling tide algorithm to create an approximated Pareto set of the problem in nurse scheduling. Instead of creating a single schedule, their solution is able to present several schedules of similar quality, from which nurses can select their favorite. Their restrictions are also quite comprehensive.

2.3.1.1 Decomposition Techniques

As emphasized in the study conducted by Azaiez and Al Sharif (2005), a major challenge associated with Mathematical Programming is the limited scalability of the proposed methods when dealing with large-scale problems. Several decomposition techniques have thus been used in the literature (Gamache and Soumis, 2001), which divide the problem into simpler sub problems that can be efficiently solved with standard Mathematical Programming approaches. They include: (1) assigning the highest-priority activities to the highest-priority employees, (2) dividing the entire roster into subgroups, (3) splitting the rostering horizon shorter time-periods (typically days), or (4) combinations of these approaches. (1) and (2) are criticized for generating schedules that are not of uniform quality; (3) does not take into account problems that may arise when scheduling for later days; (4) mitigates (but not eliminates) some of the disadvantages of the previous methods. A way to control for some of the disadvantages aforementioned would be to consider the largest groups/time-period for which restrictions are in place. In other words, if there is, for the most part, independence between allocations of workers with different skills or in different time-periods, this problem becomes less relevant. Very good solutions can be obtained nonetheless. For instance, Valouxis et al. (2012) was the winners of the First International Nurse Rostering Competition by decomposing the problem into two phases which were both solved using integer Mathematical Programming. The first phase focused on defining whether a collaborator worked or not for a particular day. Since demand coverage was essentially the only hard constraint, their formulation was quite simple. The second phase attributed a single shift type every day that a worker was in operation (given by the first phase).

2.3.1.2 Solvers

Regarding actual implementation, there are several solvers in the market that tackle Goal Programming problems. These include commercial software, such as IBM CPLEX, GUROBI and COPT, and open source solvers, including CBC Coin-OR, SCIP and HiGHS. Based on benchmark dataset tests conducted by Mittelman (2023), Gurobi has not only demonstrated the ability to solve the highest number of problems but also emerged as the fastest software among all the solutions. HiGHS, on the other hand, claimed the top position among the free solvers. It should be noted that both COPT and CPLEX were not included in this benchmark, at their request. In any case, there are many other instances in the literature that highlight GUROBI's dominance over other commercial software, including Luppold et al. (2018) who compares lpSolve, IBM CPLEX and GUROBI and Hvattum et al. (2012) who focuses on XPRESS, CPLEX and GUROBI. Naturally, it should be highlighted that the specifications of each problem make some solvers more suited than others (Anand et al., 2017).

In this thesis, both CBC Coin-OR and GUROBI were considered. On the one hand, since the final solution had to be sent to the client, who does not have access to commercial software, an open-source solution, such as CBC Coin OR, necessarily had to implemented. In particular, CBC (1) performs well when compared to other free software, (2) is quite flexible for additional customization and (3) is already integrated with the PULP package in python. According to Ngoo et al. (2022) in their survey on nurse scheduling, CBC was also the most used open-source software. On the other hand, it is interesting to compare the performances between the fastest solver and CBC with each other and with the GA proposed. This practice is not novel in the literature, with Hvattum et al. (2012) comparing the performance of CPLEX, XPRESS and GUROBI with Tabu Search and Schuster et al. (2020) comparing GUROBI and SYMPHONY with simulated annealing in conservation planning problems for several instances of the Davoine et al. (2003) test set.

2.3.2 Metaheuristic Approaches

In their study, Puente et al. (2009) introduces a GA implementation that incorporates several modifications relative to a standard approach. Specifically, they use roulette wheel selection, which increases the chances of individuals with higher fitness being selected for reproduction. Additionally, they employ crossover and mutation operators tailored to their specific problem.

Frey et al. (2009) also employed a GA for emergency shifts for doctors. Although their approach yielded noteworthy results, there is potential for further improvement. Notably, their solution lacked crossover operators and there appeared to be no systematic exploration of the optimal model parameter.

The paper from Zolfaghari et al. (2010) bears significant resemblance to the problem being addressed. Firstly, they introduce a GA and compare it with a Goal Programming model (branch-and-bound). Secondly, their study focuses on the retail sector, which shares many similarities with the current case study, including similar constraints. Thirdly, they conduct a sensitivity analysis on certain parameters of their GA. In particular, they highlight that for small-sized problems (17 employees), a small population size (#employees/6 or #employees/3) and mutation rate (0.3 or 0.4) are the best. However, their solution overlooks important elements: (1) they do not consider days-off or vacation requests; (2) they assume a linear penalty for under-staffing, which is unrealistic as larger deviations should carry a stronger impact; (3) their GA employs a fixed number of iterations without improvement as the stopping criterion, which is not aligned with most organizations' requirements where time plays a crucial role; (4) their crossover operator randomly selects parents from the initial population instead of an elite and non-elite set; and (5) their crossover operator is based on a one-point crossover. Lastly, (6) as significant advancements have been made in Mathematical Programming solvers in the past 10 years, their previous findings - that the GA is on par with or sometimes superior to the branch-and-bound method - may no longer hold true.

Aickelin and Dowsland (2000) show that a canonical implementation of the GA is insufficient to address their nurse scheduling problem effectively. In fact, only 2 out of 1040 runs were able to reach an optimal solution. The underlying issue is that following the building block hypothesis, which suggests that assembling effective partial solutions leads to successful overall solutions, does not guarantee constraint consistency (Aickelin and White, 2004). They also highlight that repair functions are not suitable for problems with a fixed number of employees[6]. By incorporating problem-specific information, such as co-evolving sub-populations and a modified fitness function combined with a hill-climbing strategy, reasonable solutions can be obtained quickly. However, the quality of the solutions obtained is not always consistent and lack of robustness is observed to small changes in problem specification. Aickelin and Dowsland (2004) then opted for a different strategy by letting the genetic operators act on an unconstrained solution space that are afterwards converted into a schedule with a decoder. They particularly investigate the effectiveness of different crossover operators. Later, Bai et al. (2010) developed a hybrid algorithm based on a GA with Stochastic Ranking and Simulated-Annealing hyper-heuristic, obtaining even better results than the indirect GA aforementioned.

Burke et al. (2008) introduced a hybrid approach combining Heuristic Ordering and Variable Neighborhood Search to tackle the nurse rostering problem. This approach yielded improved outcomes for small-scale instances compared to a widely-used GA implemented in numerous hospitals. The heuristic ordering phase involved assessing the difficulty of each shift, which were then assigned to the nurses who had the least impact on the penalty function. The shifts were allocated from most challenging to least troublesome. The Variable Neighborhood Search used a Variable Neighborhood Descent technique, considering two different neighborhoods.

Something that became clear from the analyses of most papers on GAs were the differences in (1) how hard constraints are included in the model; (2) how the solution is encoded and (3) what crossover operators achieve the best solutions. The following subsections thus delve into each one of these themes in order to determine the optimal approach.

2.3.2.1 Hard Constraints in Genetic Algorithms

Various techniques have been devised over time to address the lack of predefined hard constraint handling in GAs (Aickelin and White, 2004), as outlined in Michalewicz's survey. The survey emphasizes several solutions, with particular attention given to (1) penalty functions, (2) rejection of infeasible individuals, and (3) repair algorithms. From (1), both static penalties and dynamic penalties can be considered - while in the former penalties are only a function of the degree of the violation of constraints, the latter are also a function of the generation number. In the method proposed by Joines and Houck (1994), the evaluation function is given by:

$$eval(X) = f(X) + (C \cdot g)^{\alpha} \cdot \sum_{j=1}^{m} f_j^{\beta}(X) \tag{2.2}$$

Where $X = (x_1, \ldots, x_n) \in \mathbb{R}^n$, C, α and β are constants and g is the generation number. This expression is the sum of the value of a feasible individual ($f(X)$) and the penalty function ($\sum_{j=1}^{m} f_j^{\beta}(X)$) multiplied by a penalty coefficient ($(C \cdot g)^{\alpha}$). The penalties on unfeasible solutions are increasingly larger due to the $(C \cdot g)^{\alpha}$ component.

[6]This is the case for the problem at hand.

As the name suggests, (2) eliminates all infeasible solutions without further consideration. Michalewicz (1995) demonstrated that this approach yielded a poorer performance when compared to (1). The limitation of this method lies in its assumption that unfeasible solutions only offer irrelevant information - however, "GAs optimize by combining partial information from the total population" (Richardson et al., 1989). (3) focuses on methods that are able to determine a feasible point from an unfeasible solution. This feasible point can be used only for evaluation purposes or can also replace the unfeasible individual. In their work, Aickelin and White (2004) discuss the drawbacks of this approach, highlighting the challenges involved in formulating techniques that convert an invalid solution into a feasible one. Furthermore, they note that such a technique may require significant computational time to accomplish its objective. In the present case, it was not possible to identify a fast repair function.

2.3.2.2 Encoding in Genetic Algorithms

Regarding encoding of GAs, Cheang et al. (2003) defined three representations for nurse scheduling problems. These are the (1) nurse-day view, (2) the nurse-task view and the (3) nurse-pattern view. (1) and (2) are similar to each other and provide direct representations of the rosters, representing the assigned shifts or tasks for nurses during the scheduling period, respectively. (3) diverges from these perspectives by indirectly representing the nurse's schedule within the structure of the data representation.

Given the straightforwardness of the nurse-day view (there are no tasks for the problem at hand), an adaptation of this representation will be considered.

2.3.2.3 Crossover Operators in Genetic Algorithms and the Multi-Parent Biased Random-Key Genetic Algorithm

Maenhout and Vanhoucke (2007) compare multiple crossover operators from the literature for nurse scheduling. They identify that uniform crossover operators tend to perform the best in two distinct instances. From this conclusion, a further extension of this operator is proposed with the BRKGA-MP (Lucena et al., 2014). Four main new features are introduced compared to a canonical GA implementation with this crossover. Firstly, more than two parents can be considered for crossover. This allows for a deeper exploration of the search space. No paper reviewed considered this possibility. Secondly, the probability of selecting a gene is higher for parents with better fitness values. This bias towards fitter parents aims to enhance the exploitation of good solutions in the crossover operation. None of the papers analyzed have investigated or examined this particular aspect. Thirdly, the probability of being selected as a parent for crossover is higher for parents with better fitness values. This bias further promotes the inheritance of advantageous genetic material from parents with superior fitness and has been applied at least by Puente et al. (2009) and Tsai and Li (2009). Fourthly, the algorithm involves multiple populations evolving simultaneously, with exchanges occurring periodically. This multi-population approach allows for diversification and information sharing between populations, potentially facilitating a more comprehensive search of the solution space. At least Aickelin and Dowsland (2000) have used this concept.

No specific implementation of this algorithm has been documented in the literature for personnel scheduling. However, given the success of similar approaches in other optimization problems, one can expect the proposed implementation to exhibit a good performance. In fact, both BRKGA and BRKGA-MP have been used with quite a lot of success in complex problems such as vehicle routing (Andrade et al., 2013; Rochman et al., 2017), complex network design (Andrade et al., 2015), clustering (Andrade et al., 2014) and flowshop scheduling (Andrade et al., 2019).

Chapter 3

Problem Contextualization

Inspired by Bhulai et al. (2008) (see Section 2.1 for more details), the proposed solution is comprised of two key steps, which are then further sub-divided into additional elements: firstly, determine the optimal number of staff members; secondly, allocate the available staff to maximize the company's response to the expected hourly demand. The first step starts by clustering all stores according to the average historical daily traffic, the average service time and the average store headcount, from which 4 main groups emerge, as is depicted in Figure 3.1:

1. Stores with high-traffic, short service time and high headcount;
2. Stores with low to moderate traffic, short service time and low headcount;
3. Stores with low to moderate traffic, high service time and low headcount;
4. Street stores[1].

Afterwards, monthly predictions of the expected number of entrants are generated with a machine learning model at the cluster level - these predictions are then disaggregated to the store level. As previously mentioned, computations are performed 3 months in advance. Finally, the optimal number of staff members is determined based on ideal efficiency levels, which in the context of this thesis is defined as $\frac{\text{Served Clients}}{\text{\# Full-Time Employees (FTEs)}}$. If both efficiency levels and the number of served clients are known, then the necessary workforce can be easily inferred.

The ideal efficiency levels were computed based on the company's historical performance. Firstly, all observations that do not match the required service level or are invalid were eliminated, specified by: Average Waiting Time (AWT) inferior to 5 minutes, ratio of $\frac{ServedClients}{Entrants} > 0.86$ and ratio of $\frac{Tickets}{Interactions} < 1.15$. The concept underlying this methodology is to conduct "internal benchmarking" among all stores and set efficiency targets comparable to the levels achieved during the stores' peak performance. From these requirements, approximately 81% of all observations are discarded. Subsequently, several ambition levels can be specified that are increasingly more demanding, by considering only the top X% of these data points. Two particular ambition levels are the intermediate and top, corresponding to the best 37.5% and 25% observations, respectively.

[1]This category corresponds to open space stores that have unique opening and closing hours, demand profiles and type of workers. In fact, the need for this category was highlighted by the company.

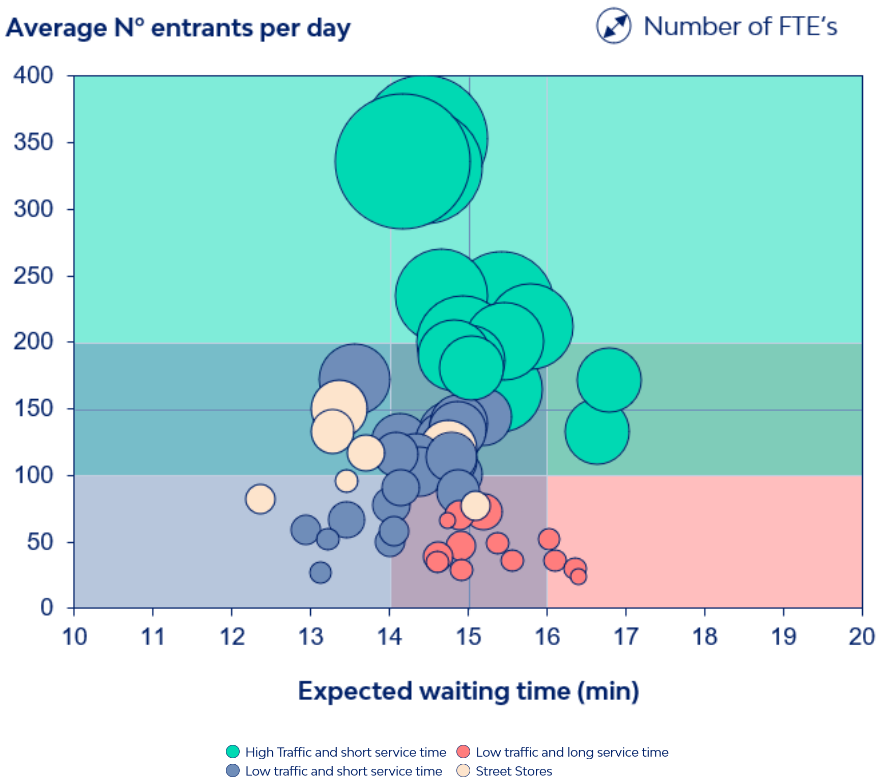

Figure 3.1: Stores Clustering

The second step starts with the definition of the necessary number of FTEs in each hour. This computation depends not only on demand coming in from clients, but also on other back office tasks that have to be done. The direct demand from clients is derived from a disaggregation of the monthly forecast into hourly blocks, which is the result of a projection of the weight of each day in the month and the weight of each hour within the day. The weight of each day varies according to the period of the month (beginning, middle and end), the day of the week and the cluster. The weight of each hour depends on the hour, the day type (weekday, Saturday and Sunday) and the cluster. Figure 3.2 presents an example of such a distribution[2]. From the analysis of the figure, one can draw several conclusions: for instance, Mondays have more traffic than Saturdays and the first hours of every day are particularly traffic heavy. Finally, back-office tasks are allocated in an attempt to avoid large hourly fluctuations, as is depicted in Figure 3.3.

[2]Example without distinction between clusters.

Projection of weight of day in month (100% = average day)
ML Model with 3 factors (day of week and month, cluster)

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Day 1-10	111%	105%	106%	102%	105%	95%	94%
Day 11-20	109%	104%	98%	94%	92%	88%	83%
Day 21-31	112%	104%	102%	98%	98%	97%	92%

Projection of weight of hour in day (100% = average hour)
Average profile of 3 months of the data, grouped by cluster

	Weekday	Sat	Sun
9h	46%	48%	30%
10h	147%	103%	92%
11h	124%	93%	87%
12h	90%	71%	67%
13h	77%	57%	52%
14h	85%	64%	64%
15h	94%	75%	74%
16h	83%	67%	66%
17h	71%	58%	56%
18h	61%	49%	46%
19h	53%	38%	34%
20h	42%	30%	27%
21h	32%	24%	19%
22h	19%	16%	13%
23h	14%	12%	9%

Figure 3.2: Traffic Disaggregation

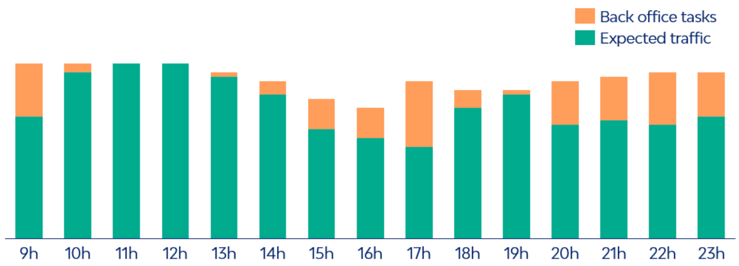

Figure 3.3: Balancing of Traffic and Back Office Tasks

Given the number of FTEs hired for each month and the total hourly demand, it is possible to feed both inputs into a shift scheduling model that specifies the day-to-day schedule of each collaborator. This thesis focuses on this last step, as is illustrated in Figure 3.4.

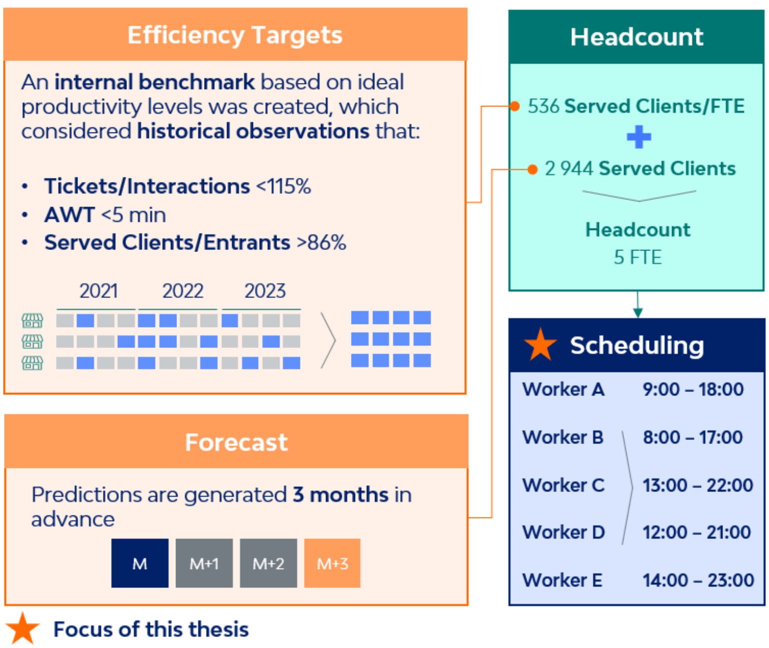

Figure 3.4: Thesis Outline

Chapter 4

Mathematical Formulation

This chapter outlines the mathematical formulation of the problem, delving into the four core stages of the process: problem statement, decision variables, constraints and objective function. The problem statement section details the context of the problem under analysis. Afterwards, both the variables under control and the restrictions that the model must obey to are outlined. Finally, the mathematical goal of the problem is stated. If one is interested on a conceptual introduction to Goal Programming, Section A.1 provides more detail.

4.1 Problem Statement

For any scheduling problem, there are essentially two forces at play: on the one hand, there exists a certain observed demand that should be met; on the other, there is a certain supply of a given resource to satisfy that demand. Ideally, supply will outweigh demand, such that no needs are left unmet. Additionally, this need is spread over a given period of time, which is typically then sub-divided into blocks or fragments. For the problem being tackled, this demand was represented by the number of workers needed at each 30-minute block for a whole month, and the supply was represented by the number of workers available for that month. Since most restrictions for the problem have a weekly nature, the mathematical formulation is also built around a whole week – the model then needs to be run 4/5 times to cover an entire month. One should note that decomposition techniques are quite standard in the literature: a deeper analysis is present in Section 2.3.1.1. One of the biggest disadvantages of this approach relates to the fact that dividing the rostering horizon into smaller time-periods - as per the current case - simply does not consider a holistic view of the problem. First of all, it can be argued that this will always be technically true, independent of the time horizon considered. In other words, scheduling for the next few days is not as complete as scheduling for the next weeks, which in turn is less complete than scheduling for the next few years. Naturally, the problem becomes less relevant the longer the time horizon considered. Additionally, Valouxis et al. (2012) highlighted that a planning horizon of 7 days lead to an optimal problem size for their nurse rostering challenge, which has a similar number of decision variables as the current problem.

4.2 Decision Variables

An intuitive representation of the decision variable for the problem follows as $W_{ijk} \in \{0,1\}$, with $i = 1,\dots,I$ number of workers, $j = 1,\dots,7$ number of days and $k = 1,\dots,M$ number of working blocks. Essentially, a worker is either working or not at each 30-minute block in a particular day. However, this representation is not computationally efficient. An illustrative example may provide some clarity: let us consider I workers and a working period between 09:00 and 21:00. In a week, there are thus $2^{7\times 24} = 3.7\times 10^{50}$ unique possible combinations for each worker in a week, and a total of $(3.7\times 10^{50})^I$ total unique combinations[1] for the problem as a whole. From this perspective, it becomes clear that the solution space for this problem grows exponentially as the number of workers increases. Consequently, an alternative representation is needed. To this effect, labor laws have pre-defined a set of rules that any schedule for a day must follow and that reduce the number of possible alternatives: essentially, each worker must work for a period of 8 hours that is complemented by an 1 hour lunch-break taking place anytime between 3 and 5 hours of work completed[2]. Thus, for the same example aforementioned, the total number of unique combinations for each worker and as a whole drastically reduces to $(7\times 5)^7 = 6.43\times 10^{10}$ and $(6.43\times 10^{10})^I$ in a week, respectively[3]. In essence, the introduction of labor laws replaces $k = 1,\dots,M$ number of working blocks by $k = 1,\dots,M$ number of admissible shift patterns. To be more precise, part-time periods of 4 working hours per day can also be considered but are specified in advance. In these cases, the number of possible shift patterns is slightly higher. Finally, workers with breastfeeding responsibilities have also been included in the model, whose shifts are characterized by an additional break of 2 hours within the 9-hour workday.

4.3 Constraints

This section presents the formulation of the problem described in the previous section. Firstly, some hard constraints are already met given how the decision variables have been defined. In fact, all restrictions related to the number of working hours and lunch break do not need to be explicitly defined in the mathematical formulation. The remainder of the hard constraints go as follows:

$$\sum_{j=1}^{7}\sum_{k=1}^{M} W_{ijk} \leq 5, \quad \forall i \in I \tag{4.1}$$

$$\sum_{k=1}^{M} W_{ijk} \leq 1 \quad \forall i \in I, j \in \{1,\dots,7\} \tag{4.2}$$

[1] for clarity purposes: 7 (days a week) $\times$ (12×2) (working blocks).

[2] Article 203º of the Portuguese Labor Law - point 1 establishes the maximum number of working hours per day to 8 and article 213º - point 1 defines that any working period should be interrupted by a break of 1 to 2 hours of duration, if the number of working hours is superior to 5.

[3] for clarity purposes: there are 7 starting hours (09:00, 09:30, 10:30,11:00, 11:30 or 12:00) and 5 possible lunch break hours for each starting hour (3, 3.5, 4, 4.5 or 5 hours later).

$$\sum_{k=1}^{l} W_{ijk} = 0 \quad \forall i \in I, j \in \{1,\ldots,7\} \tag{4.3}$$

$$\sum_{k=u}^{M} W_{ijk} = 0 \quad \forall i \in I, j \in \{1,\ldots,7\} \tag{4.4}$$

$$\sum_{i=1}^{N} \sum_{k=l+1}^{u-1} W_{ijk} \geq 1 \quad \forall i \in I \tag{4.5}$$

$$\sum_{k=1}^{M} W_{ijk} = 0 \quad \forall i \in I, j \in Vacation_{Days} \tag{4.6}$$

$$\sum_{k=b}^{k} W_{ijk} = 0 \quad \forall i \in I, j \in \{1,\ldots,7\} \tag{4.7}$$

$$W_{ijk} \in \{0,1\} \tag{4.8}$$

Constraint (4.1) assures that a worker must enjoy at least two rest days within a week. Even though labor laws only explicitly require one day-off per week[4], they also establish a maximum of 40 hours of work per week[5]. Since each schedule is comprised of 8 hours of work, each team member may only work a maximum of 5 days[6]. Constraint (4.2) assures that each worker may only be assigned one schedule in a day. Constraints (4.3) and (4.4) guarantee that all schedules that include working hours before opening time or after closing time are not possible to be chosen[7]. Constraint (4.5) guarantees that at least one worker must always present between opening and closing hours. Constraints (4.6) and (4.7) relate to requests set by the staff: the former guarantees that a worker get his assigned vacation days; the latter makes sure that only the desired working periods can be chosen. Constraint (4.8) specifies that the decision variable is binary.

The soft constraints go as follows, considering that $a_{il} = 1$ if the worker is active in block l for schedule i and 0 otherwise, and that L_{jl} represents the load needed for day j at the l block of time:

$$\sum_{i=1}^{I} \sum_{k=1}^{M} a_{il} W_{ijk} + d_1^- \geq L_{jl} \quad \forall j \in \{1,\ldots,7\}, l \in L \tag{4.9}$$

$$\sum_{i=1}^{I} \sum_{k=1}^{M} a_{il} W_{ijk} + d_2^- \geq L_{jl} - 1 \quad \forall j \in \{1,\ldots,7\}, l \in L \tag{4.10}$$

$$\sum_{i=1}^{I} \sum_{k=1}^{M} a_{il} W_{ijk} + d_3^- \geq L_{jl} - 2 \quad \forall j \in \{1,\ldots,7\}, l \in L \tag{4.11}$$

[4] Article 232º of the Portuguese Labor Law - point 1.
[5] Article 203º of the Portuguese Labor Law - point 1.
[6] This is only the case for full-time employees.
[7] l and u establish the lower and upper bound, respectively.

$$\ldots \tag{4.12}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{M} a_{il}W_{ijk} + d_{y+1}^{-} \geq L_{jl} - Y \quad \forall j \in \{1,\ldots,7\}, l \in L \tag{4.13}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{M} a_{il}W_{ijk} - d_{1}^{+} \leq L_{jl} + 1 \quad \forall j \in \{1,\ldots,7\}, l \in L \tag{4.14}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{M} a_{il}W_{ijk} - d_{2}^{+} \leq L_{jl} + 2 \quad \forall j \in \{1,\ldots,7\}, l \in L \tag{4.15}$$

$$\ldots \tag{4.16}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{M} a_{il}W_{ijk} - d_{k}^{+} \leq L_{jl} + U \quad \forall j \in \{1,\ldots,7\}, l \in L \tag{4.17}$$

$$\sum_{j=1}^{7}\sum_{k=1}^{O} W_{ijk} - \sum_{j=1}^{7}\sum_{k=O+1}^{M} W_{ijk} - d_{k+1}^{+} + d_{y+2}^{-} = 0 \quad \forall i \in I \tag{4.18}$$

$$\sum_{i=1}^{S_1}\sum_{k=l}^{l+5} W_{ijk} + d_{y+3}^{-} \geq 1 \quad \forall j \in \{1,\ldots,7\} \tag{4.19}$$

$$\sum_{i=1}^{S_1}\sum_{k=u-5}^{u-1} W_{ijk} + d_{y+4}^{-} \geq 1 \quad \forall j \in \{1,\ldots,7\} \tag{4.20}$$

$$\frac{\sum_{i=S1+1}^{S2}\sum_{k=1}^{M} a_{il}W_{ijk} + 1}{\sum_{i=1}^{S4}\sum_{k=1}^{M} a_{il}W_{ijk} + 1} - d_{k+2}^{+} + d_{y+5}^{-} = R \quad \forall j \in \{1,\ldots,7\}, l \in L \tag{4.21}$$

$$\sum_{i=S3+1}^{S4}\sum_{k=1}^{M} a_{il}W_{ijk} + d_{y+6}^{-} \geq 1 \quad \forall j \in \{1,\ldots,7\}, l \in L \tag{4.22}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{M} a_{il}W_{ijk} + d_{y+7}^{-} \geq 1 \quad \forall j \in \{1,\ldots,7\}, l = (Opening + 4) \tag{4.23}$$

$$\sum_{i=1}^{I}\sum_{k=1}^{M} a_{il}W_{ijk} + d_{y+8}^{-} \geq 1 \quad \forall j \in \{1,\ldots,7\}, l = (Closing - 4) \tag{4.24}$$

Constraint (4.9) tries to assure that existing needs are met by the workers that are available - this is the basic idea behind Set Covering (SC) models. Constraints (4.10) through (4.13) introduce increasingly larger penalties for situations of under allocation and constraints (4.14) through (4.17) fulfill a similar objective for over allocation. The rationale behind the first set of constraints is to evenly distribute under allocation across all hours rather than concentrating it in a few specific time blocks. For example, it is preferable to have five hours with a shortage of one person rather than one hour with a shortage of five people. A similar explanation can be applied to constraints (4.14) through (4.17). Constraint (4.18) tries to assure a balanced allocation between morning (going from 1 to O) and night (going from O+1 to M) schedules[8] in order to guarantee a fair distribution of shifts types. Constraints (4.19) and (4.20) try to guarantee that at opening (schedules P to $P+4$) and closing hours (schedules G to $G+4$), at least one member from management is present, respectively. Ideally, only individuals belonging to this category are qualified to start the store's operations. Constraint (4.21) tries to ensure that the ratio between top performers and rookies does not deviate from a target level in order to create a homogeneous roster of skills at every moment. One should notice that one is added to both the numerator and denominator to avoid mathematical impossibilities. The purpose of constraint (4.22) is to ensure that rookies are always accompanied at the store, preventing them from being alone.

To ensure that no individual is left alone at the store for extended periods, it is important to avoid having someone unaccompanied for more than two hours. One approach to achieve this is by examining all four consecutive blocks of 30 minutes and verifying if the sum of active workers is equal to or greater than 4. However, this method can be computationally demanding. As a result, a simplified check is performed two hours after opening and two hours before closing, as indicated by constraints (4.23) and (4.24).

4.4 Objective Function

The objective function is as follows:

$$\sum_{j=1}^{7}\sum_{l=1}^{L}\left(\frac{\sum_{i=1}^{I}\sum_{k=1}^{M} W_{ijk}}{L_{jl+1}}\right)+\sum_{b=1}^{B} w_b^+ d_b^+ + \sum_{c=1}^{C} w_c^- d_c^- \tag{4.25}$$

It is essentially comprised of two elements: the first component highlights the need to maximize allocations where needs are scarcer; the second component is associated with the relative weights that are assigned to the positive and negative deviation variables. These weights were assigned so that the soft constraints are minimized according to the following order of importance (from most important to least):

1. Minimize under-allocations;
2. Avoid rookies being alone at the store;
3. Avoid periods where a worker is alone for more than 2 hours;
4. Guarantee that someone from management is present at opening and closing hours;
5. Minimize over-allocations;
6. Guarantee balance between morning and night shifts of workers;
7. Guarantee balance between top performers and rookies at every hour.

[8]Considering that the schedules are ordered by entering time.

Whilst interesting, the AHP described by Topaloglu (2006) (and detailed in Section 2.3.1) proved not to be necessary, as the company already had a clear perspective on the relative importance of each criteria.

Chapter 5

Comparative Evaluation of Modelling Approaches

This chapter compares the performance of the solutions generated by two solvers based on the weighted goal programming model - detailed in the previous section - and those generated by the BRKGA-MP algorithm. This comparison has two main purposes: on the one hand, it tries to capture how all solutions perform under time-constraints; on the other, it evaluates how well the heuristic developed matches the solvers. As the late Herbert Simon pointed out, "satisficing" is much more prevalent than optimizing in actual practice. In coining the term "satisficing" as a combination of the words satisfactory and optimizing, Simon was describing the tendency of managers to seek a solution that is "good enough" for the problem at hand (Hillier and Lieberman, 2014). Therefore, due to time limitations, it may be more suitable to go with the BRKGA-MP solution, which can generate feasible solutions, even if they are sub-optimal.

The objective of this section is precisely to test this hypothesis. Section 5.1 details the indicators considered to assess the performance of the solutions. Section 5.2 provides a description of the instances used. In Section 5.3, the GA is optimized and in Section 5.4 the results are compared.

5.1 Performance Indicators

Regarding the metrics of success to be used, the most intuitive approach would be to simply compare the objective function values obtained, where a smaller value would correspond to a better solution. Naturally, this methodology is only valid if the penalties on the soft constraints are exactly the same in both solutions. However, it becomes quite difficult to grasp the managerial implications associated with differences in objective function values – for instance, what is the net effect of a 10-point difference on the schedules of the workers?

Consequently, some other metric of success is in need. In this regard, the literature has defined some qualitative and quantitative criteria that can serve as a starting point (further detailed in Section 2.2.4). In particular, Labor Utilization is an interesting metric that assesses the effectiveness of the schedules created. However, it cannot be directly used in the case study, since the number of hours of labor scheduled are fixed. In any case, the principle behind this metric - schedules that are capable of avoiding idle time are better - is quite valid. Therefore, the following 5 metric of success, hereby detailed, try to capture this idea:

1. Number of hours under-allocated or number of hours over-allocated is named Absolute Deviation (AD): in a perfect scenario where the demand for workers is the same as the existing supply, the total number of hours under-allocated or over-allocated would be zero, where under-allocation is assumed to be observed when the difference between demand and supply is greater than 0.5. In this case, there would be a perfect match between necessities and available resources. However, this is nearly impossible to obtain in practice, given the fluctuations of needs that exist over time. Additionally, the following relationship must be always observed:

$$Total\_Demand + Over\_allocation = Total\_Supply + Under\_Allocation \tag{5.1}$$

   Consequently, for each additional hour that is under-allocated, there will be a corresponding hour that is over-allocated – these metrics are thus indicators of the intensity of mismatch.

2. The following difference is named Absolute Complete Deviation (ACD):

$$ACD = Over\_allocation - max(0, Total\_Supply - Total\_Demand) \tag{5.2}$$

   This metric provides a better framing into the quality of the schedule created, since it takes into account the relationship between supply and demand. If, for instance, demand is equal to 1000 hours and supply to 1500 hours, the model will necessarily have at least 500 hours over-allocated, according to the definition in Equation (5.1). Consequently, the AD may portray the idea that the model is being unsatisfactory when that is actually not the case.

3. The following ratio is named Relative Complete Deviation (RCD):

$$RCD(\%) = \frac{ACD}{Total\_Supply} \times 100 \tag{5.3}$$

   The objective of this measure is to allow for the comparison between different instances. The interpretation is the following: an RD of 10% implies that 10% of all available hours are not being perfectly allocated. Note that this does not imply that they are not being optimally allocated;

4. Number of working blocks under-allocated is named Absolute Count (AC): This is an alternative measure that does not capture intensity, but rather takes a binary value of 1 if there is a block under-allocated.

5. The following ratio is named Relative Count (RC):

$$RC(\%) = \frac{AC}{Number\_Working\_Blocks(NWB)} \times 100 \tag{5.4}$$

Where NWB is the sum of all working blocks of the period under analysis. For instance, if a store is open from 9:00h to 21:00h, the NWB in a week is equal to $(21-9)(hours) \times 2(30\ minblocks) \times 7(days)$, which is equal to 168. Again, the objective is to make comparisons fairer. The interpretation could be the following: an RC of 10% implies that 10% of all working blocks are likely to be understaffed, with higher waiting times and a poorer service. For all measures, one seeks to minimize them.

5.2 Instances

To evaluate the models' quality, several instances were generated, as is shown in Table 5.1. The goal was to ensure diversity in factors such as headcount[1], total demand and supply and their relationship (sometimes excess demand, sometimes excess supply) and opening and closing hours. For additional information on each worker - including contract type and hierarchical classification - one can refer to Table B.2 of Appendix B.

Table 5.1: Instances Main Information

Instance #	Total Demand (hours)	Total Supply (hours)	Opening and Closing Hours
Instance 4	144	160	[09:00-22:00]
Instance 6	203	220	[09:00-22:00]
Instance 8	362	300	[08:00-23:00]
Instance 10	371	360	[10:00-23:00]
Instance 12	416	460	[10:00-21:00]
Instance 14	672	560	[08:00-22:00]
Instance 20	786	750	[08:00-21:00]
Instance 25	895	930	[08:00-21:00]
Instance 30	1009	1140	[08:00-23:00]
Instance 35	1236	1290	[10:00-23:00]
Instance 50	1835	1880	[09:30-22:00]

With the defined metrics of performance and instances, all that is left is to determine the optimal parameters of the BRKGA-MP algorithm. It will then be possible to compare all solutions.

5.3 Multi-Parent Biased Random-Key Genetic Algorithm

If the reader is not familiar with GAs and BRKGA-MP, it is strongly recommended to go to Section A.2 for more detail. In any case, Figure 5.1 depicts a summary of the main innovative features of the algorithm.

[1] The number of the instance represents headcount.

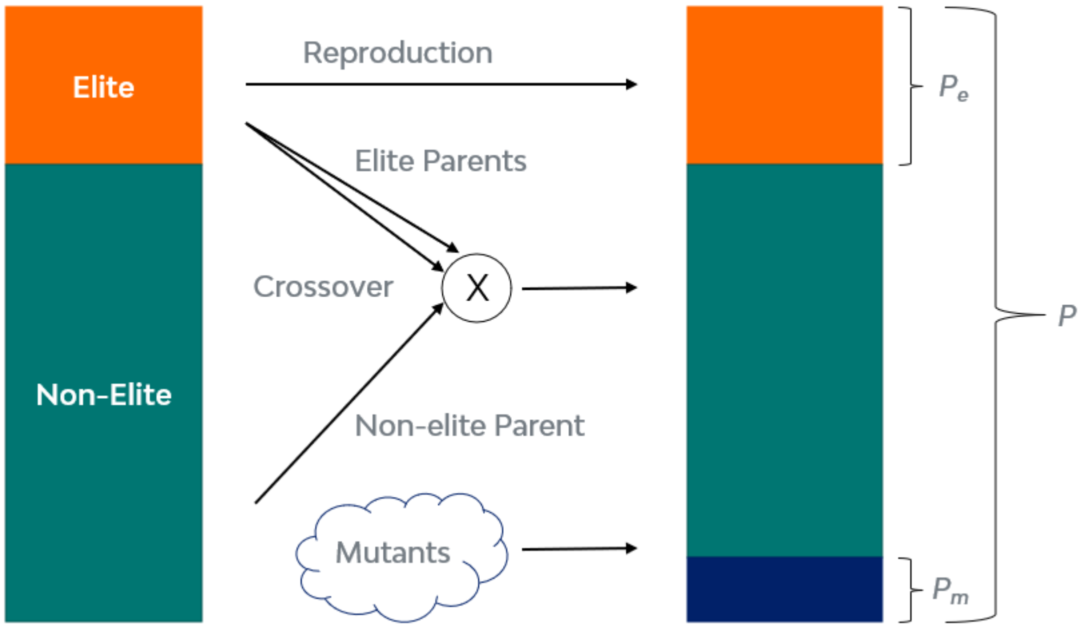

Figure 5.1: BRKGA-MP Population Evolution Scheme (adapted from Andrade et al. (2019))

At its core, the algorithm operates by assigning random numbers between 0 and 1 to the genes of each individual. These random numbers are then decoded to generate a solution for the given problem. In each generation, the population is divided into two sets: the elite set and the non-elite set. The elite set, which contains the best individuals, is automatically carried over to the next generation. The remaining individuals are either produced through uniform crossover or generated as "mutants" (entirely new individuals). Unlike traditional crossover methods that involve only two parents, this algorithm allows for the selection of multiple parents. Finally, more than one population can be evolving at the same time.

To implement the BRKGA-MP, the algorithm proposed by Andrade et al. (2021) was the starting point. What follows is a customization of the algorithm to the personnel scheduling problem.

5.3.1 Hard constraints

Hard constraints were introduced by applying the method proposed by Joines and Houck (1994) detailed in Section 2.3.2.1. In essence, this formulation introduces dynamic penalties such that the penalty from not satisfying all hard constraints increases with the number of generations.

5.3.2 Representation and Decoder

Each chromosome that encodes a solution is composed of $N \times 8$ genes, where N represents headcount. The decoder maps this vector of random keys into a solution for the problem: the first position will give information on the days a collaborator is working; the remaining positions indicate the schedule associated with each day. Figure 5.2 presents an example for a full-time worker in a random week.

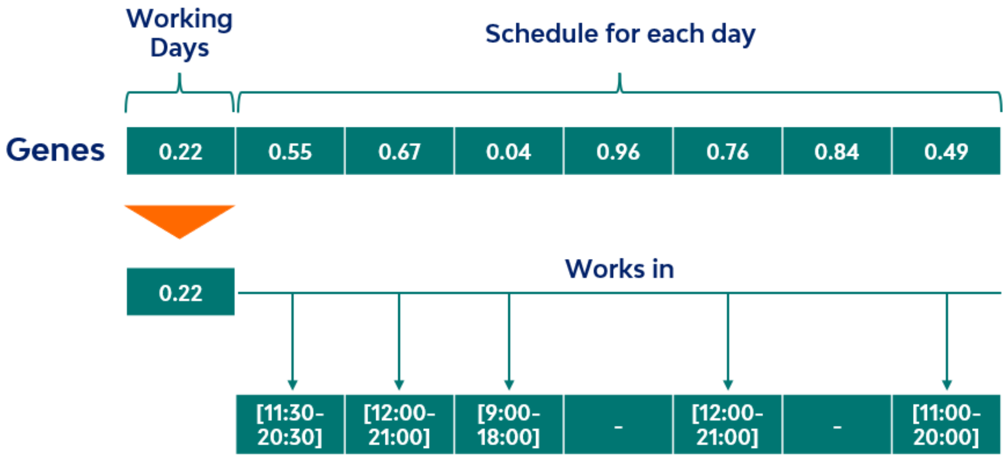

Figure 5.2: BRKGA-MP Decoder Scheme

In this specific instance, if the first allele is 0.22, it means that a worker will be scheduled to work on days 1, 2, 3, 5, and 7. The remaining alleles correspond to specific time schedules: for example, 0.55 corresponds to the schedule 11:30-20:30, 0.67 corresponds to 12:00-21:00, and so forth.

5.3.3 Crossover

The crossover operator used in the BRKGA-MP algorithm is a parameterized uniform crossover, where each allele is selected from one of the parents using roulette wheel selection. In the standard implementation, all genes undergo this operation. However, in this particular problem, it is specified that the first gene of each worker cannot be exchanged. The rationale behind this constraint is to preserve the specific working days assigned to each worker.

5.3.4 Parameter Optimization

After completely defining the algorithm, it is essential to tune the parameters of the method to obtain the best solution possible. There are a total of 11 parameters to analyse, which are:

1. Population size P;
2. Number of independent populations N;
3. Elite set percentage $P_e(\%)$;
4. Mutant set percentage $P_m(\%)$;
5. Number of parents and elite parents in the crossover π_t/π_e;

6. Exchange interval l;
7. Number of exchange individuals i;
8. Bias function ϕ;
9. Constant C (penalty function);
10. Constant α (penalty function).
11. Constant β (penalty function).

The starting values for all parameters were based either on previous works, including the Set Covering problem described in Lucena et al. (2014) (very similar to the current problem) and the test cases mentioned in Joines and Houck (1994) or initial experimentation. Table 5.2[2] summarizes all possible choices.

Table 5.2: Parameters of BRKGA-MP

Trial	Level 1	Level 2	Level 3
P	$5S$	$10S$	$15S$
N	2	3	4
$P_e(\%)$	10	15	20
$P_m(\%)$	40	50	60
π_t/π_e	3/1	3/2	6/4
l	50	100	150
i	1	2	3
ϕ	Log Inverse	Quadratic	Exponential
C	20	30	40
α	0.2	0.3	0.4
β	1	2	3

To find the best combination of scenarios, one makes use of the Taguchi robust design method (Roy, 2001), where the response variable is given by the *RCD*. This method was chosen considering its simplicity and capability to deal with a lot of factors with many levels. Additionally, the focus is on the main factors and not necessarily on the possible interactions. Unlike the full factorial design that would require a minimum of $11^3 = 1331$ observations, the Taguchi scheme specified in Table B.3 of Appendix B depends solely on 27 observations. Moreover, this method has already been applied to similar GA implementations (Sadjadi et al., 2011).

For each trial, all instances were replicated three times to avoid random error. In the context of the business setting proposed, a time limit of 60 minutes was imposed by Delta to generate the schedule for each month. Consequently, a weekly schedule would require, at most, approximately 15 minutes. All results were computed on a laptop with the following specifications: 12^{th} Gen Intel® Core™ i7-1265U 1.80 GHz, 16 GB of ram and Windows 11 Pro.

The results were transformed into S/N ratios using Equation 5.5, since the performance measure to the proposed problem is of 'the smaller the better' type.

[2] S represents the number of sets of the Set Covering problem.

$$S/N = -10 \times \log \sum_{k=1}^{K} \frac{RCD_k^2}{k} \tag{5.5}$$

Table 5.3 presents the S/N ratio values for each level of the parameters. The table also includes the ranking of the parameters based on delta statistics, which indicate the relative magnitude of their effects. In this context, the parameter level with the highest S/N ratio is chosen for each parameter.

Table 5.3: Orthogonal Array $L_{27}(3^{13})$

Level	P	N	$P_e(\%)$	$P_m(\%)$	π_t/π_e	l	i	ϕ	C	α	β
1	-27.90	-27.97	-27.58	-27.81	-28.13	-28.09	-28.15	-27.90	-28.25	**-27.84**	**-27.81**
2	**-27.41**	-27.90	-28.03	-28.09	**-27.83**	**-27.97**	-27.89	**-27.75**	**-27.64**	-27.96	-27.95
3	-28.24	**-27.81**	**-27.38**	**-27.76**	-27.91	-28.08	**-27.75**	-27.88	-27.96	-28.00	-27.89
Δ	0.82	0.16	0.66	0.34	0.31	0.13	0.40	0.15	0.62	0.16	0.14
Rank	1	8	2	5	6	11	4	9	3	7	10

An Analysis of Variance was performed to assess the statistical significance of the factors, as presented in Table B.4 of Appendix B. The findings indicate that none of the factors exhibit a statistically significant influence on the robustness of the Genetic Algorithm proposed.

5.4 Solutions Quality

With the conclusions taken from the previous section, the following parameters were assumed:

1. Population size $P = 10N$;
2. Number of independent populations $N = 4$;
3. Elite set percentage $P_e(\%) = 20$;
4. Mutant set percentage $P_m(\%) = 60$;
5. Number of parents and elite parents in the crossover $\pi_t/\pi_e = 3/2$;
6. Exchange interval $l = 100$;
7. Number of exchange individuals $i = 3$;
8. Bias function $\phi = Quadratic$;
9. Constant $C = 30$ (penalty function);
10. Constant $\alpha = 0.3$ (penalty function);
11. Constant $\beta = 1$ (penalty function).

To assess the performance of the BRKGA-MP algorithm, each instance was compared against the best solution obtained with the GUROBI solver - either with an integer solution that is equal to the LP relaxation of the problem or by exploring all possible nodes[3]. Again, the solution ran for a total of 15 minutes. The results are presented in Tables 5.4 and 5.5:

[3] The model was also stopped after 3 hours of running time.

Table 5.4: Instance Results for GUROBI Versus BKRGA-MP

Store	GUROBI					BRKGA-MP				
	AD	ACD	RCD	AC	RC	AD (Δ)	ACD (Δ)	RCD (Δ)	AC (Δ)	RC (Δ)
Instance 4	55	22	6.88%	17	18.68%	61 (+6)	28 (+6)	8.75% (+1.88 p.p.)	21 (+4)	23.08% (+4.40 p.p.)
Instance 6	75	42	9.55%	15	16.48%	84 (+9)	51 (+9)	11.59% (+2.05 p.p.)	14 (-1)	15.38% (-1.10 p.p.)
Instance 8	77	77	12.83%	26	24.76%	88 (+11)	88 (+11)	14.67% (+1.83 p.p.)	30 (+4)	28.57% (+3.81 p.p.)
Instance 10	67	67	9.31%	23	25.27%	76 (+9)	76 (+9)	10.56% (+1.25 p.p.)	25 (+2)	27.47% (+2.20 p.p.)
Instance 12	194	106	11.52%	11	14.29%	212 (+18)	124 (+18)	13.48% (+1.96 p.p.)	13 (+2)	16.88% (+2.60 p.p.)
Instance 14	155	155	13.84%	31	31.63%	155 (0)	155 (0)	13.84% (0 p.p.)	31 (0)	31.63% (0.00 p.p.)
Instance 20	203	203	13.50%	19	22.62%	276 (+73)	276 (+73)	18.40% (+4.90 p.p.)	22 (+3)	26.19% (+3.57 p.p.)
Instance 25	231	161	8.66%	15	16.48%	255 (+24)	185 (+24)	9.95% (+1.29 p.p.)	19 (+4)	20.88% (+4.40 p.p.)
Instance 30	432	171	7.50%	21	20.00%	455 (+23)	194 (+23)	8.51% (+1.01 p.p.)	20 (-1)	19.05% (-0.95 p.p.)
Instance 35	338	230	8.91%	20	21.98%	412 (+74)	304 (+74)	11.78% (+2.87 p.p.)	23 (+3)	25.27% (+3.30 p.p.)
Instance 50	322	231	6.14%	17	19.43%	355 (+33)	264 (+33)	7.02% (+0.88 p.p.)	19 (+2)	21.71% (+2.29 p.p.)

Table 5.5: Average Results for GUROBI Versus BRKGA-MP

GUROBI		BRKGA-MP	
RCD	RC	RCD (Δ)	RC (Δ)
9.88%	21.06%	11.69% (+1.81 p.p.)	23.28% (+2.23 p.p.)

Based on the analysis of these tables, it can be inferred that the BRKGA-MP implementation exhibits a slightly lower performance compared to the best available solution, with a difference of 1.81 percentage points in the RCD. Interestingly, the BRKGA-MP is capable of achieving a similar solution to GUROBI in instance 14. Therefore, if the time advantages offered by the BRKGA-MP are substantial, it might be advantageous to consider this implementation. The subsequent subsection examines this hypothesis.

5.4.1 Solutions Quality under Time Restrictions

As it has been described in previous sections, meta-heuristics gained popularity in the 1980s and 1990s as an alternative approach to optimization due to their ability to solve large and complex problems in a timely manner. The objective of this subsection is to infer the point at which the BRKGA-MP is more efficient relative to both solvers. The solution proposed consists in starting with 3 full-time collaborators (minimum number) and determining the moment where each approach produces a feasible solution. Collaborators are then added one by one, and the total demand is adjusted to take this aspect into consideration. Since the BRKGA-MP algorithm has a stochastic component, the average of 5 distinct runs was assumed. Figure 5.3 describes an example of such a situation and Table 5.6 presents the average results.

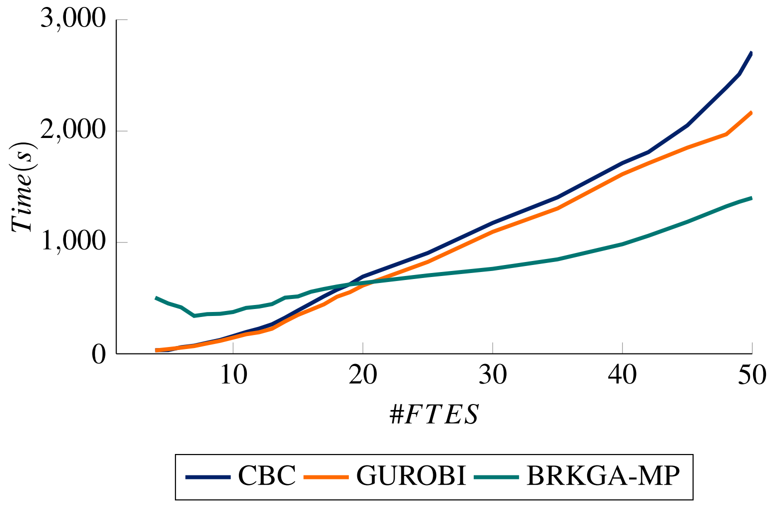

Figure 5.3: Time Efforts to Create a Feasible Solution

Table 5.6: Average Results of Feasible Solutions

CBC		GUROBI		BRKGA-MP	
RCD	RC	RCD	RC	RCD	RC
19.56%	30.24%	**18.64%**	**30.65%**	21.52%	36.56%

As shown in the figure, when the number of collaborators is low, the solvers clearly outperform the BRKGA-MP. This is not surprising, since the models being created are quite small. However, as this number starts to increase, the time it takes for the solvers to generate a feasible solution increases exponentially. In contrast, the BRKGA-MP solution initially reduces the solution time until 6 workers, after which it gradually increases in a more subtle manner. This pattern could be attributed to the initial difficulty faced by the model in finding a solution when the number of collaborators is very reduced, akin to searching for a "needle in a haystack". At 20 workers, the GA actually becomes faster. It should be noted that the quality of the solutions generated tends to be worse for the BRKGA-MP implementation. To sum up, it is reasonable to conclude that the BRKGA-MP algorithm is an interesting solution for very large stores where the creation of schedules is time-sensitive.

Chapter 6

Case Study Results

While the pursuit of knowledge has inherent worth and significance, it is considered by many - as is the case for instrumentalism (Dewey, 2001) - as a means to achieve specific ends or goals. As pointed by Bergh et al. (2013) and Özder et al. (2020), most research never gets implemented in practice, and when it does, authors provide very few details on the process of implementation or on the results of such implementation. This chapter thus focuses on the practical impact of implementing the proposed solution in the company. Section 6.1 provides a description of the instances used. In Section 6.2, the results generated are compared against the manual approach that was previously employed by the company. In Section 6.3, the individual effect of each soft constraint is analysed. Section 6.4 details the measures used to assess the success of the implementation. Section 6.5 then employs these measures to statistically verify the impact of said implementation. Finally, Section 6.6 presents the tool that was developed and is currently in use by the company.

6.1 Instances

A total of 15 stores were used as a pilot group to validate and assess the performance of the solution developed for the month of April. These cover a broad array of scenarios, ranging from low to high number of collaborators, low to high traffic, different opening and closing hours, and even low to high ratio of top-performers to rookies. The main features of each store are detailed in Table 6.1.

The number of FTEs, which is presented in the third column, that are ordered and paid by the company each month is calculated with the following formula:

$$\#FTEs = \frac{\sum_{a=1}^{A}(30 - \text{days\_off}_a) \times \text{daily\_working\_hours}_a}{172.67 \times A} \tag{6.1}$$

Where $a = 1, \ldots, A$ represents total headcount. The number 172.67 represents the number of working hours per employee per month that works full-time. This value is obtained by:

$$Hours/FTE/Month = \frac{Number\_working\_days}{12\ (months)} \times daily\_working\_hours \tag{6.2}$$

Table 6.1: Stores Main Information

Store #	Total Demand (hours)	Total Supply (hours) [FTEs]	Opening and Closing Hours
Store 1	941	1048 [6.1]	[09:30-23:00]
Store 2	983	864 [5.0]	[09:00-21:00]
Store 3	1071	984 [5.7]	[10:00-23:00]
Store 4	1049	966 [5.6]	[09:30-23:00]
Store 5	1525	1546 [8.9]	[09:30-00:00]
Store 6	1667	1657 [9.6]	[10:00-23:00]
Store 7	829	808 [4.7]	[10:00-21:00]
Store 8	1033	928 [5.2]	[09:30-22:00]
Store 9	1289	1352 [7.8]	[10:00-23:00]
Store 10	1279	1248 [7.2]	[10:00-23:00]
Store 11	1090	1040 [6.0]	[10:00-23:00]
Store 12	1245	1280 [7.4]	[09:00-23:00]
Store 13	1107	1048 [6.1]	[10:00-23:00]
Store 14	948	912 [6.3]	[10:00-23:00]
Store 15	1004	936 [5.4]	[10:00-23:00]

$$172.67 = \frac{259}{12} \times 8 \tag{6.3}$$

The number of working days is calculated taking into consideration 22 paid vacation days per year and 2 rest days per week. A more detailed description of each store can be found in Table B.5 of Appendix B.

6.2 Manual Versus Automatic Solutions

Since the solvers provided better and quicker solutions for the instances in Chapter 5, where the number of collaborators is similar to those provided by the company, the remaining sections focus on this solution. In particular, they detail the results of the CBC solver, which was used by the company.

In order to grasp the quality of the model developed, the company provided the detailed manual schedules of each collaborator for all stores under the pilot for April. These schedules were generated by the managers of each store, who have manually performed this task since they were hired. They did not have any analytical insights into the forecasted hourly demand, relying solely on empirical knowledge.

According to feedback that was collected, it takes them anywhere from several hours to multiple days to create a feasible solution, depending on the number of collaborators and their requests. Table 6.2 portrays the comparative results for all 15 instances and Table 6.3 details the average metrics for the RCD and RC[1]. The CBC solver ran for a maximum of 1 hour, as specified by the company's needs.

[1]Only these two metrics were considered, since comparison between instances is adequate. That is not the case for the remaining metrics.

Table 6.2: Instance Results for Manual Versus Automatic Solutions

Store	Manual Solution					Optimized Solution				
	AD	ACD	RCD	AC	RC	AD (Δ)	ACD (Δ)	RCD (Δ)	AC (Δ)	RC (Δ)
Store 1	183	76	7.25%	103	12.72%	162 (-21)	55 (-21)	5.21% (-2.04 p.p.)	54 (-50)	6.60% (-6.11 p.p.)
Store 2	72	72	8.31%	84	11.67%	37 (-35)	37 (-35)	4.27% (-4.05 p.p.)	75 (-9)	10.42% (-1.25 p.p.)
Store 3	89	89	9.04%	136	17.37%	66 (-23)	66 (-23)	6.68% (-2.36 p.p.)	151 (+16)	19.36% (+1.99 p.p.)
Store 4	162	162	16.74%	180	22.22%	49 (-113)	49 (-113)	5.04% (-11.71 p.p.)	144 (-36)	17.78% (-4.44 p.p.)
Store 5	152	141	9.20%	124	14.25%	103 (-49)	93 (-49)	6.03% (-3.17 p.p.)	95 (-29)	10.92% (-3.33 p.p.)
Store 6	159	159	9.57%	145	18.59%	85 (-74)	85 (-74)	6.98% (-4.44 p.p.)	119 (-26)	15.26% (-3.33 p.p.)
Store 7	126	126	15.63%	124	18.80%	101 (-25)	101 (-25)	12.54% (-3.10 p.p.)	107 (-17)	16.22% (-2.58 p.p.)
Store 8	81	81	8.68%	160	23.12%	45 (-36)	45 (-36)	4.83% (-3.85 p.p.)	181 (+22)	26.23% (+3.12 p.p.)
Store 9	154	91	6.71%	124	15.83%	136 (-19)	72 (-19)	5.34% (-1.37 p.p.)	74 (-50)	9.49% (-6.35 p.p.)
Store 10	190	190	15.23%	147	18.85%	83 (-107)	83 (-107)	6.66% (-8.57 p.p.)	116 (-31)	14.87% (-3.97 p.p.)
Store 11	119	119	11.46%	124	15.90%	77 (-42)	77 (-42)	7.42% (-4.03 p.p.)	140 (+16)	17.95% (+2.05 p.p.)
Store 12	185	150	11.70%	139	16.49%	153 (-32)	118 (-32)	9.21% (-2.49 p.p.)	145 (+6)	17.20% (+0.71 p.p.)
Store 13	126	126	11.98%	153	19.55%	84 (-42)	84 (-42)	8.01% (-3.97 p.p.)	151 (-2)	19.36% (-0.19 p.p.)
Store 14	102	102	11.15%	96	12.31%	63 (-38)	63 (-38)	6.95% (-4.20 p.p.)	139 (-43)	17.82% (-5.51 p.p.)
Store 15	120	120	12.84%	136	17.37%	67 (-53)	67 (-53)	7.15% (-5.69 p.p.)	138 (+3)	17.69% (+0.32 p.p.)

Table 6.3: Average Results for Manual Versus Automatic Solutions

Manual Solution		Optimized Solution	
RCD	RC	RCD (Δ)	RC (Δ)
11.03%	17.37%	6.70% (-4.34 p.p.)	15.44% (-1.93 p.p.)

As it can be apprehended from Table 6.2, the model outperforms the manual solution in most metrics for all instances. In fact, the AD, ACD and RCD metrics are inferior in all cases, while the AC and RC are inferior in 10 out of the 15 stores. Again, these results are compatible with the constraints set by the model: it attempts to minimize large differences between supply and demand across all working blocks, such that they may be more hours in deficit - larger AC and RC - but with a much smaller intensity - smaller AD, ACD and RCD. The average results are also quite positive, with a decrease in 4.34 percentage points in the RCD and 1.93 percentage points in the RC. This implies a 39.29% reduction in the total supply that is not perfectly allocated and a 11.09% decrease in the number of hours under stress.

Moreover, the quality of both the automatic and manual solutions seems to depend not only on the relationship between supply and demand, but also on the supply itself. In other words, if the supply is larger than existing demand, it becomes easier to create better schedules. Similarly, if the number of collaborators is reduced, there are less options to explore and it will be more difficult to obtain interesting solutions. This seems to be the case for stores 7 and 14, where the RCD takes the largest values of 12.54% and 11.15%. Finally, the composition of the workforce available to each store also seems to have an important effect, with part-time workers providing a lot more flexibility than full-time employees, since the number of possible schedules is far larger.

The differences between the automatic and manual solutions also vary significantly among instances, and this can be attributed to several factors. For example, a larger workforce is more difficult to manage as there are exponentially more scheduling possibilities with each additional member. This seems to be the case for store 6, for instance. Additionally, there may be differences regarding years of experience, capabilities or effort placed by each manager in the creation of the schedules, leading to heterogeneity in the quality of the solutions generated. Finally, it was discovered through discussions with the company that not only do managers consider different constraints, but their relative importance also differs. For instance, some managers consider it extremely important to never have a collaborator alone for more than two hours, while others prioritize different periods or even find it to be an irrelevant requirement. As a result, schedules are not of uniform quality.

Finally, feedback collected from the managers responsible for this task also highlighted improvements on the time and perceived effort to create the schedules for each worker.

6.3 Soft Constraints Analysis

Besides the performance improvements over the incumbent solution, it is also interesting to understand the impact that each soft constraint, detailed in Chapter 4, has on the model. In other words, this section attempts to provide insights into the deterioration of the model's quality as a result of demands set by the company and its users. According to Bergh et al. (2013), this type of analysis has not yet been thoroughly researched.

In a classical linear programming model, several post-optimality tools can be used: (1) sensitivity analysis, (2) parametric linear programming and (3) shadow pricing (Hillier and Lieberman, 2014). From (1) it is possible to determine the most sensitive parameters - those whose values cannot be changed, without changing the optimal solution; (2) involves the systematic study of how the optimal solution changes as many of the parameters change simultaneously over a certain range. It can be seen as an extension of the sensitivity analysis that checks the effect of correlated parameters; (3) provides information about the impact of relaxing or tightening constraints on the objective function. A positive shadow price indicates that relaxing the constraint would improve the objective, while a negative shadow price suggests that tightening the constraint would be beneficial. All of these analyses focus on changes of the parameters and not on the removal of the constraints themselves, since they are assumed to be hard restrictions - they must necessarily be enforced. In the context of a Mixed-Integer Goal Programming model that is no longer true, since soft constraints admit deviation variables to the objective function and do not have be strictly satisfied.

One can thus consider two types of analyses: on the one hand, to make use of the post-optimality tools aforementioned; on the other, to remove each soft constraint and their combination in the model. The first analysis is not interesting, since constraints are either legally imposed and their parameters simply cannot be changed or are not appealing from a business perspective. For instance, one of the soft constraints is focused on ensuring that at least one person from management is at the store during opening hours. This constraint reflects a series of logistical considerations - the manager has the keys to open the store, there is an operational meeting just before opening, the cashier has to be properly setup, etc... Studying the impact of having two, three or more managers doesn't bring any added value. The same rationale could be expanded to other constraints. On the other hand, if the company were to set up the adequate processes so that no manager had to be present, the model's ability to generate better solutions could be impacted. Thus, the remaining of this section deep-dives into the second type of analysis described.

To study all possible interactions, a Design-of-Experiments inspired approach based on a full factorial design (Montgomery, 2001) was defined. The use of such a tool in studying parameters of schedules has already been described in the literature by Jacobs and Bechtold (1993). In this context, 5 constraints that can be removed without impacting legislative barriers or the ability of the model to produce interesting solutions were considered. They are: (A) avoid rookies being alone at the store; (B) avoid periods where a worker is alone for more than 2 hours; (C) guarantee that someone from management is present at opening and closing hours; (D) balance between morning and night shifts of workers and (E) balance between top performers and rookies at all moments. The design matrix thus entails 2^5 possible combinations for a single replication. As each weekly run takes approximately 15 minutes, it would be extremely cumbersome to perform this analysis for the 15 stores available, as it would imply a total of 32x15 = 480 runs or 5 days. Consequently, the analysis was conducted on three stores considered to be representative of the entire set - stores 1, 5 and 7. Another important aspect relates to the dependent variable to be analysed. In this case, one needs to consider an adimensional variable, so that all replications are comparable. Additionally, the focus is on understanding the incremental impact of each constraint and not necessarily on its absolute importance. Regarding the former aspect, both the RCD and

RC satisfy this criteria. The latter is satisfied by considering the difference of the output of these measures relative to the baseline (all constraints incorporated). The results of each replication and the design matrix, which is based on the Yates order, can be visualized in Table 6.4 and Table B.6 of Appendix B, respectively.

Table 6.4: Results of Design of Experiments

Factor	Store 1		Store 5		Store 7	
	δ RCD (p.p.)	δ RC (p.p.)	δ RCD (p.p.)	δ RC (p.p.)	δ RCD (p.p.)	δ RC (p.p.)
(1)	-1.55	-1.11	-1.38	-2.07	-0.74	-0.98
A	-1.23	-0.74	-1.02	-1.49	-0.57	-0.76
B	-1.44	-0.86	-0.94	-1.26	-0.67	-0.83
AB	-0.77	-0.68	-0.70	-0.92	-0.25	-0.45
C	-1.46	-0.86	-1.13	-1.55	-0.74	-0.98
AC	-1.18	-0.62	-0.79	-1.26	-0.57	-0.76
BC	-0.76	-0.74	-0.44	-1.09	-0.67	-0.83
ABC	-0.51	-0.25	-0.30	-0.40	-0.25	-0.45
D	-1.33	-0.99	-1.30	-1.61	-0.74	-0.91
AD	-1.15	-0.68	-0.95	-1.26	-0.50	-0.76
BD	-1.37	-0.99	-0.76	-1.03	-0.67	-0.83
ABD	-0.67	-0.68	-0.62	-0.75	-0.25	-0.45
CD	-1.13	-0.80	-1.04	-1.32	-0.74	-0.98
ACD	-0.51	-0.56	-0.69	-1.15	-0.36	-0.53
BCD	-1.02	-0.62	-0.35	-0.80	-0.67	-0.83
ABCD	-0.24	-0.12	-0.10	-0.23	-0.25	-0.45
E	-1.48	-0.93	-1.23	-1.49	-0.59	-0.93
AE	-1.19	-0.74	-0.91	-1.15	-0.49	-0.76
BE	-1.37	-0.80	-0.93	-0.92	-0.50	-0.76
ABE	-0.71	-0.62	-0.39	-0.69	-0.06	-0.30
CE	-1.28	-0.80	-1.10	-1.38	-0.50	-0.76
ACE	-1.07	-0.48	-0.57	-0.92	-0.22	-0.53
BCE	-0.57	-0.56	-0.18	-0.86	-0.35	-0.68
ABDE	-0.41	-0.19	-0.06	-0.17	-0.25	-0.61
DE	-1.25	-0.80	-1.14	-1.49	-0.47	-0.76
ADE	-0.94	-0.56	-0.85	-1.03	-0.23	-0.68
BDE	-0.66	-0.62	-0.55	-0.86	-0.31	-0.76
ABDE	-0.31	-0.56	-0.46	-0.40	-0.15	-0.38
CDE	-0.77	-0.74	-0.43	-1.15	-0.50	-0.91
ACDE	-0.39	-0.56	-0.42	-0.92	-0.19	-0.68
BCDE	-0.47	-0.43	-0.27	-0.52	-0.04	-0.15
ABCDE	0.00	0.00	0.00	0.00	0.00	0.00

In order to provide more clarity into the different metric, some examples have been provided. For instance, the value of -1.55 p.p. of the δRCD for factor (1) at Store 1 means that, when no soft constraint is present, there is a reduction of 1.55 percentage points relative to all soft constraints being present - that is, we go from an RCD of 5.21% (Table 6.2) to an RCD of 3.66%. This implies that 1.55% of all supply is better allocated when there is no concern about any soft constraint. A

δRC of -0.52 p.p. for factor BCDE at Store 1 implies that, when all but soft constraint A are present, there is a 0.52 percentage point reduction of the *RC* relative to the case where all soft constraints are satisfied. This implies that 0.52% of all working blocks are no longer under stress.

In order to statistically validate each factor and their interaction, Table 6.5 presents an analysis of variance for the full model of the δRCD, including all main factors and possible interactions.

Table 6.5: Results of Analysis of Variance for Soft Constraints and Interactions on Relative Complete Deviation

Factor	Degrees of Freedom	Sum of Squares	Mean Squares	F Value	P-value
A	1	2.43	2.43	26.48	2.74×10^{-6} ***
B	1	3.12	3.12	33.96	1.96×10^{-7} ***
AB	1	0.02	0.02	0.187	0.67
C	1	1.72	1.72	18.69	5.49×10^{-5} ***
AC	1	0.01	0.01	0.06	0.81
BC	1	0.04	0.04	0.41	0.53
ABC	1	0.05	0.05	0.50	0.48
D	1	0.71	0.71	7.73	7.13×10^{-3} **
AD	1	1	0.01	0.01	0.01
BD	1	0.01	0.01	0.14	0.71
ABD	1	1.20×10^{-3}	1.20×10^{-3}	0.01	0.91
CD	1	0.01	0.01	0.19	0.66
ACD	1	0.06	0.06	0.61	0.44
BCD	1	0.07	0.07	0.71	0.40
ABCD	1	0.04	0.04	0.44	0.51
E	1	1.09	1.09	11.86	1.02×10^{-3} **
AE	1	0.04	0.04	0.44	0.51
BE	1	0.01	0.01	0.14	0.71
ABE	1	0.01	0.01	0.11	0.74
CE	1	0.02	0.02	0.24	0.63
ACE	1	0.01	0.01	0.15	0.70
BCE	1	0.01	0.01	0.11	0.74
ABDE	1	0.03	0.03	0.03	0.85
DE	1	0.09	0.09	0.99	0.32
ADE	1	0.04	0.04	0.40	0.53
BDE	1	5.50×10^{-3}	5.50×10^{-3}	0.06	0.82
ABDE	1	3.10×10^{-3}	3.10×10^{-3}	0.03	0.85
CDE	1	3.35×10^{-4}	3.35×10^{-4}	3.20×10^{-3}	0.96
ACDE	1	1.40×10^{-3}	1.40×10^{-3}	0.02	0.90
BCDE	1	2.60×10^{-3}	2.60×10^{-3}	0.03	0.87
ABCDE	1	0.04	0.04	0.38	0.54
Residuals	64	5.89	0.09	-	-

$Notes: *p < 0.025, **p < 0.005, ***p < 0.0005$

All main factors are statistically significant - it does not appear to be the case for the interaction terms. In other words, all soft constraints appear to have a negative statistically significant impact on the solution obtained. Finally, it is also possible to build a regression model to estimate the precise impact of each factor. This type of analysis has been suggested for future research in a similar framework by Jacobs and Bechtold (1993). Since the interaction terms were not statistically significant, two models can be considered:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_A x_A + \hat{\beta}_B x_B + \hat{\beta}_C x_C + \hat{\beta}_D x_D + \hat{\beta}_E x_E \tag{6.4}$$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_A x_A + \hat{\beta}_B x_B + \hat{\beta}_{AB} * x_A x_B + \ldots + \hat{\beta}_{ABCDE} x_A x_B x_C x_D x_E \tag{6.5}$$

The estimates for the reduced and full model are present in Equations 6.6 and D.1 of Appendix D, respectively. A partial F-test was conducted between the two models with an F-value of 0.2527 ($p - value = 0.99$), pointing to the same conclusion of the analysis of variance - the interaction terms do not have a significant explanatory power.

$$\hat{y} = -1.34 + 0.32x_A + 0.36x_B + 0.27x_C + 0.17x_D + 0.22x_E \tag{6.6}$$

It is thus possible to conclude that factor B has the largest negative impact on the solution, with an increase in 0.36 percentage points in the RCD, while factor E has the smallest one. In any case, from a managerial point of view, the impact of any factor is quite reduced. The company should feel comfortable with all of the requirements that have been set. A similar analysis is present in Table B.7 of Appendix B and Equations D.2 and D.3 of Appendix D for the case of δRC, where the results support maintaining the current restriction set - no main factor or interaction is statistically significant.

6.4 Performance Measures

Even though the metrics detailed in Section 5.1 are a practical way to assess how well a given schedule will likely perform, the company also keeps track of each client's data, including information on time of arrival, waiting time and service time, from which several Key Performance Indicators (KPI's) can be defined. These include:

1. Productivity, given by:
$$Productivity = \frac{Entrants}{FTEs} \tag{6.7}$$
Where the number of entrants and number of FTEs correspond to the number of people that enter the store to be attended and the number of employees present at the store, respectively.

2. Efficiency, given by:
$$Efficiency = \frac{Served}{FTEs} \tag{6.8}$$
Where the numerator is given by the number of clients that are actually attended, since some clients may not be willing to wait and thus exit the store.

3. Service Level (%), given by:

$$Service\ Level(\%) = \frac{Served}{Entrants} \times 100 \tag{6.9}$$

4. Average Waiting Time (AWT), given by the total number of seconds that each client, on average, has to stay in line waiting to be served.

5. Average Service Time (AST), given by the total number of seconds that each client, on average, takes to be served.

6. Commercial Efficacy (CE), given by:

$$CE = \frac{Equipments\ Sold}{Entrants} \tag{6.10}$$

Where the number of equipments sold takes only into consideration clients that did not have any planned order.

With these indicators, the impact of the proposed solution becomes even more tractable.

6.5 Impact on the Company

There are several research designs available to estimate the impact of a treatment on an outcome variable - the reader is advised to go through Section A.3 of Appendix A for an introduction to this topic. Difference-in-Differences (DiD) stands out as the most standard approach for problems with longitudinal data where the number of treatment groups is large and the number of pre- and post- treatment data points are reduced (Huntington 2022). This is precisely the case for the problem at hand, where 15 stores were subject to the methodology described in Chapter 3. The company also provided information about 15 other stores that were not subject to any treatment but shared similarities with the pilot group. Data is available for the months of February, March, April and May for all stores. The package "did" (Callaway and Sant'Anna, 2021) allows for a simple implementation of DiD with multiple time periods in R. The analysis was conducted for all metrics detailed in Section 6.4. Table 6.6 presents the results.

A small note should be given on these indicators. As per the description in Chapter 3, the entire methodology was exclusively applied to the pilot stores. This implies that the observed changes cannot be completely attributed to the scheduling process solely. In particular, there were substantial decreases in the headcount of each store, due to the increased productivity levels that started being required from the collaborators.

In any case, the interpretation is the following: when the event time column takes a null or positive value, the estimate corresponds to the impact of treatment X periods after treatment. When this value is negative, one is trying to validate the parallel trends assumption, by checking the impact of treatment X periods prior to its application. Consequently, the expectation is for this estimate to not be statistically significant.

Table 6.6: Dynamic Effect of DiD

Measure	Event Time	Estimate	P-value
Productivity	-1	4.42	0.30
	0	85.66	7.90×10^{-8} ***
	1	96.86	1.21×10^{-4} ***
Efficiency	-1	7.28	0.20
	0	120.05	3.79×10^{-9} ***
	1	139.82	4.91×10^{-5} ***
Service Level	-1	-2.20×10^{-3}	0.39
	0	-1.91×10^{-2}	0.03
	1	-3.06×10^{-2}	3.64×10^{-3} **
AWT	-1	12.19	0.32
	0	86.81	4.00×10^{-3} **
	1	134.00	3.66×10^{-3} **
AST	-1	17.25	0.29
	0	-61.56	4.94×10^{-3} **
	1	-80.40	1.39×10^{-2} *
CE	-1	1.57×10^{-2}	0.03
	0	7.60×10^{-3}	0.18
	1	3.80×10^{-3}	0.34

$Notes: *p < 0.025, **p < 0.005, ***p < 0.0005$

Several conclusions can be drawn: firstly, statistically significant increases in the productivity and efficiency of workers of about 22% and 26%, respectively; secondly, statistically significant reductions in the service level (only for May) of about 4%; thirdly, statistically significant increases in the AWT and decreases in the AST of about 46% and 9%, respectively; finally, there was no statistically significant evidence indicating changes in the commercial efficacy of collaborators. To sum up, a reduction in the number of FTEs implied necessarily increases in productivity and efficiency. In addition, customers had to wait more time to be served, with some simply quitting - leading to an increase in the average waiting time and decrease in the service level. In turn, workers felt an increased pressure to be quicker, which resulted in a decreased average service time. Finally, collaborators were still able to sell the companies' products with the same efficacy.

6.6 Tool

To allow each store manager to autonomously create the schedule for his collaborators, a flexible and intuitive tool in Excel, with a connection to an executable coded in python, was developed. The decision to use Excel was driven by its simplicity, ease of use, and familiarity among the managers. As it is illustrated in Figure 6.1, the home page is essentially composed of three main blocks: inputs, calculation of total hourly demand and schedule optimization.

Figure 6.1: Cover of Tool

6.6.1 Inputs

The first block deals with all the relevant inputs that are necessary for the model to run. It starts with the definition of some relevant store parameters, including the ambition level to consider (as has been detailed in Chapter 3), the expected number of monthly entrants, the opening and closing hours at each day of the week and the time necessary to clean the store prior to opening every day - Figure 6.2 shows precisely this. It continues with the definition of all the necessary information about each collaborator, as it can be seen in Figures C.1 and C.2 of Appendix C. The tool is flexible enough to allow for any number of employees, days-off and/or holiday periods. These elements were consistently requested by the organizations that Causmaecker et al. (2004) visited in his research. A brief summary of each parameter is detailed in the Table B.8 of Appendix B. The back-office tasks for the entire month can be easily modified, as shown in Figure 6.3.

Figure 6.2: Store Parameters

Mapeamento de tarefas indiretas

Tarefa	Tipificação	Duração (min)	Pessoas	Frequência	Hora	Dia da Semana	Dia da Semana 2	Dia do Mês	Funções	Prescrita?
Reunião IOS Daily Abertura	Planeamento	15	1	Diária	10:00	-	-	-	Todos	Prescrito
Reunião IOS Daily 14 h	Planeamento	15	1	Diária	14:00	-	-	-	Todos	Prescrito
Declaração de vendas/Ac. Vendas Pend	Processo Administrativo	10	1	Diária	10:00	-	-	-	AL	Prescrito
CAT envio/recepção	Processo Logístico	15	1	Diária	21:00	-	-	-	AL	Prescrito
Tratamento Material Danificado	Processo Logístico	20	1	Diária	13:00	-	-	-	AL	Prescrito
Envio SPV/Logic	Processo Logístico	20	1	Diária	21:30	-	-	-	AL	Prescrito
Tratamento de PLI	Processo Logístico	20	1	Diária	12:00	-	-	-	GL/CE	Prescrito
IOP	Planeamento	45	Todos	Semanal (1x)	20:00	Segunda	-	-	GL/AL	Prescrito

Figure 6.3: Indirect Tasks Parameters

This flexibility allows the manager to set criteria such as task duration, number of required employees and frequency (daily, weekly (1X or 2X), or monthly). Converting these tasks into the number of FTEs required for each 30-minute work block is simple. If the task duration is equal to or less than 30 minutes, the number of necessary employees is multiplied by the duration of the task and divided by 30. However, if the task duration exceeds 30 minutes, the remaining time is shifted to the next work block, up to a maximum of 60 minutes. This process continues from one 30-minute block to another until the task's total duration is accounted for in the model.

6.6.2 Hourly demand

To determine the expected demand for each 30-minute interval based on the number of monthly entrants, the manager can press the "Geração de Carga" button in the home page. The final result is illustrated in Figure 6.4 - for instance, only 0.5 FTEs are needed at 9:30h on June the 2^{nd}, while 3.7 FTEs are required at 10:00h.

Total Colaboradores	9:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00	14:30
01-06-23	0.4	3.5	3.5	3.1	3.1	3.1	3.1	2.7	2.7	3.5	3.5
02-06-23	0.5	3.7	3.7	3.2	3.2	2.8	2.8	3.3	3.3	3.7	3.7
03-06-23	0.4	3.2	3.2	3.0	3.0	3.4	3.4	2.1	2.1	3.0	3.0
04-06-23	0.2	2.5	2.5	2.7	2.7	2.6	2.6	2.1	2.1	2.8	2.8
05-06-23	0.4	3.4	3.4	3.4	3.4	3.6	3.6	4.0	4.0	3.7	3.7
06-06-23	0.4	3.8	3.8	3.5	3.5	3.8	3.8	3.9	3.9	4.2	4.2
07-06-23	0.4	4.0	4.0	4.5	4.5	3.3	3.3	4.7	4.7	3.9	3.9
08-06-23	0.4	3.5	3.5	3.0	3.0	4.4	4.4	3.3	3.3	4.1	4.1
09-06-23	0.4	3.8	3.8	3.1	3.1	3.0	3.0	4.1	4.1	4.1	4.1
10-06-23	0.3	2.7	2.7	2.7	2.7	4.6	4.6	1.7	1.7	2.5	2.5
11-06-23	0.2	2.4	2.4	2.7	2.7	3.4	3.4	2.4	2.4	2.9	2.9
12-06-23	0.4	3.2	3.2	3.3	3.3	4.0	4.0	4.5	4.5	3.7	3.7
13-06-23	0.4	3.4	3.4	3.1	3.1	3.1	3.1	3.1	3.1	3.6	3.6
14-06-23	0.4	3.9	3.9	4.3	4.3	3.2	3.2	4.4	4.4	3.8	3.8
15-06-23	0.4	3.5	3.5	2.8	2.8	5.9	5.9	4.0	4.0	4.7	4.7
16-06-23	0.4	4.3	4.3	2.9	2.9	3.6	3.6	6.4	6.4	5.2	5.2
17-06-23	0.3	3.0	3.0	3.1	3.1	7.0	7.0	1.9	1.9	2.7	2.7
18-06-23	0.2	2.4	2.4	2.7	2.7	3.5	3.5	2.4	2.4	2.9	2.9
19-06-23	0.5	3.5	3.5	3.7	3.7	4.8	4.8	5.6	5.6	4.2	4.2
20-06-23	0.4	3.8	3.8	3.4	3.4	3.7	3.7	3.7	3.7	4.1	4.1
21-06-23	0.4	4.0	4.0	4.2	4.2	3.2	3.2	4.2	4.2	3.9	3.9
22-06-23	0.4	3.3	3.3	2.8	2.8	3.7	3.7	2.9	2.9	3.6	3.6
23-06-23	0.4	3.6	3.6	3.0	3.0	2.8	2.8	3.7	3.7	3.8	3.8
24-06-23	0.2	2.1	2.1	2.1	2.1	3.2	3.2	1.4	1.4	2.0	2.0
25-06-23	0.2	2.8	2.8	3.0	3.0	3.3	3.3	2.5	2.5	3.2	3.2
26-06-23	0.6	4.2	4.2	4.1	4.1	4.5	4.5	5.0	5.0	4.5	4.5
27-06-23	0.6	4.6	4.6	4.2	4.2	3.9	3.9	3.9	3.9	4.6	4.6
28-06-23	0.6	5.1	5.1	5.3	5.3	4.0	4.0	5.1	5.1	4.9	4.9
29-06-23	0.5	3.8	3.8	3.2	3.2	4.4	4.4	3.4	3.4	4.2	4.2
30-06-23	0.4	3.4	3.4	2.8	2.8	2.6	2.6	3.6	3.6	3.6	3.6
01-07-23	0.0	4.1	4.1	4.2	4.2	3.1	3.1	3.8	3.8	3.9	3.9

Total Colaboradores	15:00	15:30	16:00	16:30	17:00	17:30	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30
01-06-23	3.9	3.9	3.7	3.7	3.1	3.1	2.8	2.8	2.8	2.8	1.8	1.8	1.4	1.4	0.7	0.7
02-06-23	4.3	4.3	4.0	4.0	3.2	3.2	2.6	2.6	3.0	3.0	1.8	1.8	1.9	1.9	0.4	0.4
03-06-23	3.5	3.5	3.3	3.3	2.9	2.9	3.4	3.4	2.5	2.5	2.0	2.0	1.7	1.7	1.0	1.0
04-06-23	4.3	4.3	4.0	4.0	3.1	3.1	2.5	2.5	3.1	3.1	1.7	1.7	1.0	1.0	0.0	0.0
05-06-23	4.6	4.6	3.5	3.5	3.4	3.4	3.3	3.3	2.5	2.5	2.4	2.4	2.0	2.0	0.3	0.3
06-06-23	4.4	4.4	6.0	6.0	3.4	3.4	4.0	4.0	3.3	3.3	2.4	2.4	1.9	1.9	0.5	0.5
07-06-23	4.4	4.4	5.3	5.3	4.2	4.2	3.5	3.5	3.0	3.0	1.4	1.4	2.0	2.0	0.5	0.5
08-06-23	4.4	4.4	4.4	4.4	3.8	3.8	3.2	3.2	3.7	3.7	2.0	2.0	2.2	2.2	1.6	1.6
09-06-23	4.8	4.8	4.7	4.7	3.7	3.7	2.7	2.7	3.6	3.6	1.9	1.9	2.8	2.8	0.4	0.4
10-06-23	2.9	2.9	2.8	2.8	2.4	2.4	4.5	4.5	3.1	3.1	2.8	2.8	2.4	2.4	1.8	1.8
11-06-23	4.9	4.9	4.4	4.4	3.3	3.3	2.4	2.4	5.0	5.0	2.6	2.6	1.7	1.7	0.0	0.0
12-06-23	4.8	4.8	3.4	3.4	3.6	3.6	3.5	3.5	2.4	2.4	2.6	2.6	2.3	2.3	0.3	0.3
13-06-23	3.8	3.8	4.7	4.7	2.9	2.9	3.2	3.2	2.7	2.7	2.0	2.0	1.5	1.5	0.4	0.4
14-06-23	4.3	4.3	5.0	5.0	4.0	4.0	3.4	3.4	3.0	3.0	1.4	1.4	1.9	1.9	0.4	0.4
15-06-23	4.9	4.9	5.3	5.3	4.7	4.7	3.7	3.7	4.9	4.9	2.2	2.2	3.2	3.2	2.6	2.6
16-06-23	6.2	6.2	6.5	6.5	4.9	4.9	2.7	2.7	5.2	5.2	2.2	2.2	5.0	5.0	0.5	0.5
17-06-23	3.2	3.2	3.1	3.1	2.6	2.6	6.9	6.9	4.5	4.5	4.2	4.2	3.6	3.6	3.0	3.0
18-06-23	4.9	4.9	4.5	4.5	3.3	3.3	2.4	2.4	5.2	5.2	2.7	2.7	1.8	1.8	0.0	0.0
19-06-23	5.6	5.6	3.7	3.7	4.2	4.2	4.1	4.1	2.6	2.6	3.2	3.2	2.9	2.9	0.4	0.4
20-06-23	4.3	4.3	5.7	5.7	3.3	3.3	3.9	3.9	3.2	3.2	2.3	2.3	1.8	1.8	0.5	0.5
21-06-23	4.4	4.4	4.9	4.9	3.9	3.9	3.4	3.4	3.0	3.0	1.5	1.5	1.8	1.8	0.4	0.4
22-06-23	4.0	4.0	3.9	3.9	3.4	3.4	2.8	2.8	3.2	3.2	1.8	1.8	1.9	1.9	1.2	1.2
23-06-23	4.4	4.4	4.3	4.3	3.4	3.4	2.5	2.5	3.3	3.3	1.8	1.8	2.4	2.4	0.4	0.4
24-06-23	2.3	2.3	2.2	2.2	1.9	1.9	3.1	3.1	2.2	2.2	1.9	1.9	1.6	1.6	1.2	1.2
25-06-23	5.1	5.1	4.7	4.7	3.6	3.6	2.8	2.8	4.2	4.2	2.2	2.2	1.4	1.4	0.0	0.0
26-06-23	5.7	5.7	4.3	4.3	4.2	4.2	4.1	4.1	3.0	3.0	3.0	3.0	2.5	2.5	0.4	0.4
27-06-23	5.1	5.1	5.7	5.7	3.9	3.9	4.1	4.1	3.5	3.5	2.5	2.5	1.8	1.8	0.6	0.6
28-06-23	5.6	5.6	6.1	6.1	4.9	4.9	4.2	4.2	3.8	3.8	2.0	2.0	2.2	2.2	0.6	0.6
29-06-23	4.6	4.6	4.6	4.6	3.9	3.9	3.3	3.3	3.8	3.8	2.1	2.1	2.2	2.2	1.5	1.5
30-06-23	4.2	4.2	4.1	4.1	3.3	3.3	2.4	2.4	3.1	3.1	1.7	1.7	2.4	2.4	0.4	0.4
01-07-23	4.5	4.5	4.7	4.7	3.8	3.8	3.3	3.3	2.8	2.8	1.7	1.7	1.5	1.5	0.4	0.4

Figure 6.4: Total Monthly Demand

The manager can also have a more detailed view of the store's necessities on each day in Figure C.3 of Appendix C.

Lastly, the manager can suggest modifications to the anticipated number of entrants for each interval, indicated by the yellow cells in Figure C.4 of Appendix C. This feature provides additional flexibility as there may be events such as a moving holiday, fairs, or rallies that the model will not factor in.

6.6.3 Schedule Optimization

Similarly to the previous task, the manager can generate the worker schedules automatically by clicking the "Geração de Escalas" button at the home page. The resulting schedule can be viewed on a monthly, weekly, or even daily basis, as shown in Figure 6.5 and Figures C.5 and C.6 of Appendix C.

Total Colaboradores	9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00
01-06-23	1.0	1.6	-1.4	-0.5	-0.1	-0.1	-0.7	-1.1	-1.4	0.3	-1.0
02-06-23	1.0	1.5	-2.5	-1.2	-0.2	-2.2	-4.4	-1.8	-1.9	-0.8	-1.2
03-06-23	1.0	0.6	-3.1	-2.2	-2.0	-1.0	-2.0	-1.4	-0.8	-0.1	-1.5
04-06-23	1.0	0.8	-1.4	-0.5	-0.7	-0.7	-1.3	-0.6	-0.8	-0.1	-1.3
05-06-23	1.0	1.6	-2.3	-1.4	-1.4	-1.4	-1.3	-1.6	-2.6	-1.0	-1.2
06-06-23	1.0	1.6	-2.6	-1.3	-0.5	-0.5	-0.5	-0.8	-1.5	-0.9	-0.7
07-06-23	1.0	1.6	-0.8	0.0	-0.5	-0.5	-1.9	-0.3	-2.4	-0.7	-0.4
08-06-23	1.0	1.6	-1.3	-0.5	0.0	0.0	-1.0	-1.4	-1.0	-0.3	-0.6
09-06-23	1.0	1.6	-2.7	-1.3	-0.1	-0.1	-1.6	0.0	-1.8	-0.6	-0.6
10-06-23	1.0	0.7	-2.5	-1.7	-1.7	-0.7	-3.3	-2.6	-0.4	-0.7	-1.0
11-06-23	1.0	0.8	-2.3	-1.4	-0.7	-0.7	-2.1	-1.4	-1.1	-0.4	-1.4
12-06-23	1.0	1.6	-2.1	-1.2	-1.3	-1.3	-1.6	-2.0	-3.2	-1.5	-1.2
13-06-23	1.0	1.6	-3.3	-1.9	-1.1	-1.1	-1.8	-2.1	-2.8	-1.1	-1.1
14-06-23	1.0	1.6	-1.8	-0.9	-1.3	-1.3	-2.8	-1.2	-3.1	-1.4	-1.3
15-06-23	1.0	1.6	-2.3	-1.5	-0.8	-0.8	-3.6	-3.9	-2.6	-1.0	-1.2
16-06-23	1.0	1.6	-4.1	-2.8	-0.9	-0.9	-4.2	-2.6	-4.1	-2.9	-1.7
17-06-23	1.0	0.7	-2.8	-2.0	-2.1	-2.1	-3.0	-3.0	-1.6	-0.9	-2.2
18-06-23	1.0	0.8	-2.3	-1.4	-1.7	-1.7	-3.1	-2.5	-2.1	-1.4	-1.4
19-06-23	1.0	1.5	-1.3	-0.5	-0.7	-0.7	-1.4	-1.8	-3.2	-1.6	-0.7
20-06-23	1.0	1.6	-2.6	-1.3	-0.4	-0.4	-1.3	-1.7	-2.4	-0.7	-1.6
21-06-23	1.0	1.6	-1.8	-1.0	-1.2	-1.2	-2.8	-1.2	-2.9	-1.2	-1.4
22-06-23	1.0	1.6	-2.1	-1.3	-0.8	-0.8	-2.4	-2.7	-2.6	-0.9	-1.1
23-06-23	1.0	1.6	-3.5	-2.1	-1.0	-1.0	-2.5	-0.8	-2.4	-1.2	-1.3
24-06-23	1.0	0.8	-2.0	-1.1	-1.1	-1.1	-2.8	-2.2	-1.0	-0.4	-1.5
25-06-23	1.0	0.8	-2.7	-1.8	-1.0	-1.0	-1.9	-1.3	-1.2	-0.5	-1.7
26-06-23	1.0	1.4	-2.1	-1.2	-1.1	-1.1	-1.2	-1.5	-2.6	-1.0	-1.0
27-06-23	1.0	1.4	-3.4	-2.1	-1.2	-1.2	-1.6	-2.9	-3.6	-0.9	-1.1
28-06-23	1.0	1.4	-2.0	-1.1	-1.3	-1.3	-2.7	-2.0	-3.7	-1.1	-1.4
29-06-23	1.0	1.5	-2.6	-1.8	-1.2	-1.2	-2.1	-2.4	-2.1	-0.4	-0.7
30-06-23	1.0	1.6	-2.3	-0.9	0.2	0.2	-2.3	-0.6	-2.2	-1.1	-1.1

Total Colaboradores	14:30	15:00	15:30	16:00	16:30	17:00	17:30	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30
01-06-23	-0.5	-0.9	-0.9	0.3	0.3	0.9	-0.1	-0.8	-0.8	-1.8	-1.8	-0.8	-0.8	-0.9	-1.1	0.3	0.3
02-06-23	-0.7	-1.3	-1.3	0.0	0.0	0.8	-0.2	-0.6	-0.6	-2.0	-2.0	-2.8	-1.8	-1.4	-1.6	0.6	0.6
03-06-23	-1.0	-1.5	-1.5	-1.3	-1.3	-1.9	-1.9	-1.4	-1.4	-0.5	-2.5	-2.0	-1.0	-1.2	-1.3	0.0	0.0
04-06-23	-0.8	-1.3	-1.3	-2.0	-2.0	-1.1	-1.1	-1.5	-1.5	-2.1	-2.1	-0.7	-0.7	-1.0	-0.6		
05-06-23	-0.7	-0.6	-0.6	-0.5	-0.5	-0.4	-0.8	-0.3	-1.3	-0.5	-0.5	-2.4	-1.4	-1.5	-1.6	0.7	0.7
06-06-23	-0.2	-1.4	-0.4	-1.0	-1.0	-0.4	-0.4	-1.0	-2.0	-1.3	-1.3	-1.4	-1.4	-1.4	-1.6	0.5	0.5
07-06-23	-0.9	-1.4	-0.4	-0.3	-0.3	-0.2	-1.2	-1.5	-1.5	-2.0	-2.0	-0.4	-0.4	-1.5	-1.7	0.5	0.5
08-06-23	-0.1	-0.4	-0.4	-0.4	-0.4	-0.8	-0.8	-0.2	-0.2	-1.7	-1.7	0.0	0.0	-1.7	-1.9	-0.6	-0.6
09-06-23	-0.1	-0.8	-0.8	-0.7	-0.7	0.3	-0.7	-0.7	-0.7	-1.6	-1.6	-1.9	-0.9	-2.3	-2.4	0.6	0.6
10-06-23	-0.5	-0.9	0.1	-0.8	-0.8	-1.4	-1.4	-2.5	-2.5	-1.1	-3.1	-2.8	-1.8	-1.9	-2.0	-0.8	-0.8
11-06-23	-0.9	-1.9	-1.9	-1.4	-1.4	-1.3	-1.3	-0.4	-0.4	-3.0	-3.0	-1.6	-1.6	-1.7	-1.4		
12-06-23	-0.7	-0.8	-0.8	0.6	-0.4	-0.6	-0.6	-1.5	-1.5	-0.4	-0.4	-2.6	-1.6	-1.8	-2.0	0.7	0.7
13-06-23	-1.6	-1.8	-0.8	-1.7	-1.7	-0.9	-0.9	-1.2	-2.2	-1.7	-1.7	-2.0	-2.0	-1.0	-1.2	0.6	0.6
14-06-23	-0.8	-1.3	-1.3	-1.0	-1.0	-1.0	-1.0	-0.4	-1.4	-2.0	-2.0	-0.4	-0.4	-1.4	-1.5	0.6	0.6
15-06-23	-1.7	-1.9	-0.9	-1.3	-1.3	-1.7	-1.7	-1.7	-2.7	-2.9	-2.9	-0.2	-0.2	-2.7	-2.8	-1.6	-1.6
16-06-23	-1.2	-1.2	-1.2	-1.5	-1.5	-0.9	-0.9	-0.7	-1.7	-2.2	-2.2	-1.2	-0.2	-2.5	-2.7	0.5	0.5
17-06-23	-1.7	-1.2	-1.2	-1.1	-1.1	-1.6	-1.6	-2.9	-2.9	-3.5	-4.2	-4.2	-3.2	-3.1	-3.3	-2.0	-2.0
18-06-23	-0.9	-2.9	-2.9	-2.5	-2.5	-2.3	-2.3	-1.4	-1.4	-4.2	-4.2	-1.7	-1.7	-1.8	-1.4		
19-06-23	-1.2	-1.6	-0.6	0.3	0.3	-0.2	-0.2	-0.1	-1.1	-0.6	-0.6	-3.2	-2.2	-2.4	-2.5	0.6	0.6
20-06-23	-1.1	-1.3	-1.3	-1.7	-1.7	0.7	-0.3	-1.9	-1.9	-2.2	-2.2	-2.3	-2.3	-1.3	-1.5	0.5	0.5
21-06-23	-0.9	-1.4	-1.4	-0.9	-0.9	-0.9	-0.9	-0.4	-1.4	-2.0	-2.0	-0.5	-0.5	-1.3	-1.4	0.6	0.6
22-06-23	-1.6	-2.0	-1.0	-0.9	-0.9	-0.4	-1.4	-1.8	-1.8	-2.2	-2.2	-0.8	-0.8	-1.4	-1.5	-0.2	-0.2
23-06-23	-0.8	-0.4	-0.4	-0.3	-0.3	-0.4	-0.4	-0.5	-1.5	-1.3	-1.3	-1.8	-0.8	-1.9	-2.1	0.6	0.6
24-06-23	-1.0	-0.3	-0.3	-0.2	-0.2	-0.9	-0.9	-2.1	-2.1	-1.2	-3.2	-2.9	-0.9	-1.1	-1.3	-0.2	-0.2
25-06-23	-1.2	-2.1	-2.1	-1.7	-1.7	-1.6	-1.6	-0.8	-0.8	-2.2	-2.2	-1.2	-1.2	-1.4	-1.0		
26-06-23	-1.5	-1.7	-0.7	-0.3	-1.3	-0.2	-0.2	-1.1	-2.1	-2.0	-1.0	-3.0	-2.0	-2.0	-2.1	0.6	0.6
27-06-23	-1.6	-2.1	-1.1	-1.7	-1.7	-1.9	-1.9	-2.1	-3.1	-2.5	-2.5	-2.5	-2.5	-1.3	-1.5	0.4	0.4
28-06-23	-0.9	-1.6	-1.6	-1.1	-1.1	-0.9	-1.9	-2.2	-2.2	-2.8	-2.8	-1.0	-1.0	-1.7	-1.8	0.4	0.4
29-06-23	-1.2	-1.6	-0.6	-0.6	-0.6	-0.9	-0.9	-1.3	-2.3	-1.8	-1.8	-0.1	-0.1	-1.7	-1.9	-0.5	-0.5
30-06-23	-0.6	-1.2	-1.2	-0.1	-1.1	-0.3	-0.3	-1.4	-1.4	-2.1	-2.1	-2.7	-1.7	-1.9	-2.0	0.6	0.6

Figure 6.5: Visualization of Monthly Results

While Figure 6.5 simply depicts the difference between supply and demand at each interval – with red coloring representing situations of under-allocation and green coloring over-allocation- Figures C.5 and C.6 present detail at the level of each collaborator. It is also possible to propose changes to the schedules of each worker in the cells depicted in yellow, as per Figure 6.6. Finally, the manager has the possibility to export the schedule into a Comma-Separated Values (CSV) file that can be read in a platform used by the company for workforce management. Overall, the scheduling system is designed to ensure that tasks are allocated efficiently and fairly among collaborators, and that the manager has the tools necessary to make adjustments as needed.

					Ajuste	Ajuste	Ajuste	Final	Final	Final
Recurso	Dia	Hora entrada	Hora refeição	Hora saida	Alteração entrada	Alteração refeição	Alteração saida	Entrada final	Refeição final	Saida final
Diana	01-06-23	9:00	12:30	18:00				9:00	12:30	18:00
Diana	02-06-23	9:00	12:30	18:00				9:00	12:30	18:00
Diana	03-06-23	Folga	Folga	Folga				Folga	Folga	Folga
Diana	04-06-23	Folga	Folga	Folga				Folga	Folga	Folga
Diana	05-06-23	9:00	12:30	18:00				9:00	12:30	18:00
Diana	06-06-23	9:00	12:30	18:00				9:00	12:30	18:00
Diana	07-06-23	9:00	12:30	18:00				9:00	12:30	18:00
Diana	08-06-23	9:00	12:30	18:00				9:00	12:30	18:00
Diana	09-06-23	9:00	12:30	18:00				9:00	12:30	18:00
Diana	10-06-23	Folga	Folga	Folga				Folga	Folga	Folga
Diana	11-06-23	Folga	Folga	Folga				Folga	Folga	Folga
Diana	12-06-23	9:00	12:30	18:00				9:00	12:30	18:00
Diana	13-06-23	9:00	12:30	18:00				9:00	12:30	18:00
Diana	14-06-23	9:00	12:30	18:00				9:00	12:30	18:00
Diana	15-06-23	9:00	12:30	18:00				9:00	12:30	18:00

Figure 6.6: Daily Schedule of Collaborators

Chapter 7

Conclusion and Future Work

This closing chapter reviews the work developed during the elaboration of this thesis and concludes with suggestions for possible developments in the field and in the work developed.

7.1 Critical Analysis

The conclusions drawn from this work are the following: (1) the proposed Genetic Algorithm exhibits near-optimal solutions in a reduced time; (2) the Genetic Algorithm is better suited for larger stores; (3) companies should try to meet collaborators' requirements, as it does not appear to have a significant impact on the created schedules. If only one option was available, the BRKGA-MP algorithm solves a wider range of problems with interesting results, as it always generates feasible solutions within a short period of time. However, with an increasing trend towards convenience and smaller stores (Deloitte, 2020), the latter solution will become less and less appealing.

There are some limitations to the work developed. First of all, any mathematical formulation is specific to the realities of a particular context. Other organizations may have different requirements which imply different constraints. Consequently, it is difficult to generalize the results obtained in the case study. In any case, the improvements over the incumbent solution were quite substantial and one can suspect that it would be the case that it would be the case in many other industries.

Secondly, the model had a weekly time frame, which implied that it had to be ran multiple times to produce the required output for the entire month. While this approach reduces each individual problem's total dimension substantially, it makes it difficult to satisfy requirements that are of a monthly nature. For example, the company specified that 8 days-off are the standard for a 30 day month, while 9 days should be considered for a 31 day month. Since the model has to be ran 5 times for practically every month and two days-off are given to employees per week, this implies that anywhere between 8 and 10 days may be given. Whilst some post-processing attempts were conducted, they never completely solved the problem, as all of them could lead to infeasible solutions. Consequently, some post-analysis had to be done by the managers to ensure all days-off were correctly allocated.

Thirdly, the allocation of the indirect tasks was done prior to the creation of the schedules, by normalizing total demand across the day. However, any solution generated may have idleness in some periods and under-allocation in others. It would thus be optimal to recursively generate new solutions that would tweak already existing ones by taking these periods into consideration or by changing the allocation of the indirect tasks.

Fourthly, as detailed in Chapter 4, exponential penalties on deviations from the observed demand were introduced in the model in a particular way as to maintain linearity. Despite being unlikely (Deveci and Demirel, 2018), it is possible that a non-linear formulation of the problem would perform better.

Last but not least, the model is also quite fragile to the initial stages of the methodology. Any substantial errors in the prediction of traffic can lead to schedules that are worse than a standard manual solution.

7.2 Future Research

Something that became apparent throughout the development of the current work is that there have only been a few attempts to uncover the best solution for different problem classes. For instance, Ngoo et al. (2022) presented the best state-of-the-art methods for different benchmark datasets on nurse rostering. This made the choice of the appropriate methodology to follow quite difficult and demanding. In this sense, there are clear opportunities for future research in this area. Additional benchmark datasets for different problem types as the ones detailed in Brucker et al. (2011) could be constructed, allowing for comparisons to be made on the same data. It would then be possible to have a clearer picture on the methodologies that work best under each instance.

Additionally, despite the growing complexity of the modeling approaches to the personnel scheduling problem, there is still significant room for research that focuses on the customer. Essentially, existing research assumes that all demand is the same, independent of the period of the day or month. Actually, this is not entirely true in many areas, including the example that has been explored. For instance, clients in the morning are typically older and come mostly to the store to pay their bills, whereas clients at the end of the day are much more likely to buy products or acquire new services. If one had to prioritize periods, it would thus be optimal to ensure that clients have a better customer experience during these later hours of the day, since their Net Present Value (NPV) is much higher. Any attempt to understand and include the NPV of the client at each period would prove to be extremely valuable for an organization.

This thesis also presents the development of a deterministic model that can establish individual schedules for collaborators. Nonetheless, the majority of systems function within dynamic and uncertain settings, wherein unforeseen events frequently cause disruptions in the schedule. Consequently, it is inevitable that organizations will eventually encounter the challenge of rescheduling, requiring the adjustment of activity schedules. Whilst some research has already been conducted on this topic (Chiaramonte, 2008; Wickert et al., 2019), it is still an emerging area of research (Ngoo et al., 2022). Therefore, rescheduling emerges as a significant and intriguing subject that warrants further exploration and investigation. In situations for which this flexibility is non-existent, two alternatives exist: (1) simply test the robustness of the solutions generated to stochastic factors, such as changes in demand or workers availability; (2) incorporate some of these uncertainties into the model, as is done in Yan and Tu (2002) with stochastic demand.

Furthermore, it would be interesting to compare the proposed methods against other solutions that have been developed in the literature, by reformulating them into this particular problem. This task would provide additional confirmation on the quality of the proposed methods.

More and more research is also focused on Matheuristic approaches that combine mathematical optimization and meta-heuristics simultaneously. Ngoo et al. (2022) highlight in their survey that Matheuristics were as competitive if not better than other approaches in benchmark datasets. It could be thus be interesting to test the quality of hybrids betwwen the integer programming solution proposed and other methods such as constraint programming (Rahimian et al., 2017).

Finally, demand is always modeled as a time-bounded phenomena that doesn't extend across multiple periods. In the current representation, if the demand exceeds the available supply within a specific time period, it is assumed that all unsatisfied clients will relinquish their requests entirely. However, in reality, some clients may choose to walk away, while others are willing to wait for their needs to be fulfilled. Consequently, the actual demand for the subsequent time period may surpass what was initially assumed. Once again, one possible approach to address this issue would involve recursively generating new schedules that adjust the anticipated real demand for each time period.

Bibliography

Abadie, A. and Imbens, G. W. (2016). Matching on the Estimated Propensity Score. *Econometrica*, 84(2), 781-807. https://doi.org/10.3982/ECTA11293.

Aickelin, U. and Dowsland, K. (2000). Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem. *Journal of Scheduling,*, 3(3), 139-153. https://doi.org/10.48550/arXiv.0802.2001.

Aickelin, U. and Dowsland, K. (2004). An indirect genetic algorithm for a nursescheduling problem. *Computers & Operations Research*, 31, 761-778. https://doi.org/10.1016/S0305-0548(03)00034-0.

Aickelin, U. and White, P. (2004). Building better nurse scheduling algorithms. *Annals of Operations Research*, 128, 159-177. https://doi.org/10.1023/B:ANOR.0000019103.31340.a6.

Akella, M., Gupta, S., and Sarkar, A. (1977). Branch and Price Column Generation for Solving Huge Integer Programs. In M. Akella, S. Gupta, and A. Sarkar (Eds.), *Branch and Price Column Generation for Solving Huge Integer Programs.* University at Buffalo. https://tinyurl.com/4hcwhhvp.

Alfares, H. K. (2004). Survey, Categorization, and Comparison of Recent Tour Scheduling Literature. *Annals of Operations Research*, 127, 145-175. https://doi.org/10.1023/B:ANOR.0000019088.98647.e2.

Allon, G., Federgruen, A., and Pierson, M. (2011). How Much Is a Reduction of Your Customers' Wait Worth? An Empirical Study of the Fast-Food Drive-Thru Industry Based on Structural Estimation Methods. *Manufacturing & Service Operations Management*, 13(4), 489-507. https://doi.org/10.1016/j.ijpe.2011.09.014.

ANACOM (2023). Factos & Números - 1.º trimestre 2023. https://www.anacom.pt/render.jsp?contentId=1746135.

Anand, R., Aggarwal, D., and Kumar, V. (2017). A comparative analysis of optimization solvers. *Journal of Statistics and Management Systems*, 20(4), 623-635. https://doi.org/10.1080/09720510.2017.1395182.

Andrade, C. E., Miyazawa, F. K., and Resende, M. G. C. (2013). Evolutionary algorithm for the k-interconnected multi-depot multi-traveling salesmen problem. *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, 1, 463-470. https://doi.org/10.1145/2463372.2463434.

Andrade, C. E., Resende, M. G. C., Karloff, H. J., and Miyazawa, F. K. (2014). Evolutionary algorithms for overlapping correlation clustering. *Proceedings of the 16th Conference on Genetic and Evolutionary Computation*, 1, 404-412. https://10.1145/2576768.2598284.

Andrade, C. E., Resende, M. G. C., Zhang, W., Sinha, R. K., Reichmann, K. C., Doverspike, R. D., and Miyazawa, F. K. (2015). A biased random-key genetic algorithm for wireless backhaul network design. *Applied Soft Computing*, 33, 150-169. https://doi.org/10.1016/j.asoc.2015.04.016.

Andrade, C. E., Silva, T., and Pessoa, L. S. (2019). Minimizing flowtime in a flowshop scheduling problem with a biased random-key genetic algorithm. *Expert Systems with Applications*, 128, 67-80. https://doi.org/10. 1016/j.eswa.2019.03.007.

Andrade, C. E., Toso, R. F., Gonçalves, J. F., and Resende, M. G. C. (2021). The Multi-Parent Biased Random-key Genetic Algorithm with Implicit Path Relinking. *European Journal of Operational Research*, 289(1), 17-30. https://doi.org/10.1016/j.ejor.2019.11.037.

Angrist, J. D. and Pischke, J.-S. (2009). *Mostly Harmless Econometrics: An Empiricistís Companion* (1st ed.). Princeton University Press. https://press.princeton.edu/books/paperback/9780691120355/mostly-harmless-econometrics.

Azaiez, M. and Al Sharif, S. (2005). A 0-1 goal programming model for nurse scheduling. *Computers Operations Research*, 32(3), 491-507. https://doi.org/10.1016/S0305-0548(03)00249-1.

Bai, R., Burke, E. K., Kendall, G., Li, J., and McCollum, B. (2010). A Hybrid Evolutionary Approach to the Nurse Rostering Problem. *IEEE Transactions on Evolutionary Computation*, 14(4), 580-590. https://doi.org/10.1109/TEVC.2009.2033583.

Baker, K. R. (1976). Workforce Allocation in Cyclical Scheduling Problems: A Survey. *Operational Research Quarterly*, 27(1), 155-167. https://doi.org/10.2307/3009134.

Bard, J. F., Binici, C., and deSilva, A. H. (2003). Staff scheduling at the United States Postal Service. *Computers & Operations Research*, 30(5), 745-771. https://doi.org/10.1016/S0305-0548(02)00048-5:745–771.

Baskaran, G. (2016). *A Domain Transformation Approach for Addressing Staff Scheduling Problems* (Publication No. 31249) [Doctoral dissertation, University of Nottingham]. Nottingham eTheses. https://eprints.nottingham.ac.uk/id/eprint/31249.

Bean, J. (1994). Genetic Algorithms and Random Keys for Sequencing and Optimization. *ORSA Journal of Computing*, 6, 154-160. http://dx.doi.org/10.1287/ijoc.6.2.154.

Bechtold, S. E., Brusco, M. J., and Showalter, M. J. (1991). A Comparative Evaluation of Labor Tour Scheduling Methods. *Decision Sciences*, 22(4), 683-699. https://doi.org/10.1111/j.1540-5915.1991.tb00359.x.

Behesti, Z. and Shamsuddin, S. M. (2013). A review of population-based meta-heuristic algorithm. *International Journal of Advances in Soft Computing and its Applications*, 5(1), 1-35. https://tinyurl.com/nead23cr.

Bellaflamme, P. and Peitz, M. (2015). *Industrial Organization Markets and Strategies* (2nd digital ed., pp. 34-36). Cambridge University Press. https://tinyurl.com/aw9jz84u.

Bergh, J. V., Beliën, J., Bruecker, P. D., Demeulemeester, E., and Boeck, L. D. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research*, 226, 367-385. https://doi.org/10.1016/j.ejor.2012.11.029.

Bhulai, S., Koole, G., and Pot, A. (2008). Simple Methods for Shift Scheduling in Multiskill Call Centers. *Manufacturing Service Operations Management*, 10, 411-420. https://doi.org/10.1287/msom.1070.0172.

Bradley, D. J. and Martin, J. B. (1991). Continuous personnel scheduling algorithms: a literature review. *Journal of the Society for Health Systems*, 2(2), 8-23. https://pubmed.ncbi.nlm.nih.gov/1760547/.

Bresina, J. (1994). Heuristic-biased stochastic sampling. *In Proceedings of the thirteenth national conference on Artificial Intelligence*, 1, 271-278. https://cdn.aaai.org/AAAI/1996/AAAI96-041.pdf.

Brucker, P., Qu, R., and Burke, E. (2011). Personnel scheduling: Models and complexity. *European Journal of Operational Research*, 210(3), 467-473. https://doi.org/10.1016/j.ejor.2010.11.017.

Burke, E. K., Causmaecker, P. D., and Berghe, G. V. (2004). The State of the Art of Nurse Rostering. *Journal of Scheduling*, 7, 441-499. https://doi.org/10.1023/B:JOSH.0000046076.75950.0b.

Burke, E. K., Curtois, T., Post, G., Qu, R., and Veltman, B. (2008). A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. *European Journal of Operational Research*, 188(2), 330-341. https://doi.org/10.1016/j.ejor.2007.04.030.

Burke, E. K., Curtois, T., Qu, R., and Berghe, G. V. (2013). A Time Predefined Variable Depth Search for Nurse Rostering. *Informs Journal on Computing*, 25(3), 144-159. https://doi.org/10.1287/ijoc.1120.0510.

Callaway, B. and Sant'Anna, P. H. (2021). Difference-in-Differences with Multiple Time Periods. *Journal of Econometrics*, 225(2), 200-230. https://doi.org/10.1016/j.jeconom.2020.12.001.

Caprara, A., Monaci, M., and Toth, P. (2001). *A Global Method for Crew Planning in Railway Applications* (1st digital ed., pp. 17-36). Springer Berlin Heidelberg. https://www.springer.com/series/300.

Cascio, W. (2015). *Managing Human Resources: Productivity, Quality of Work Life, Profits* (10th digital ed.). McGraw Hill. https://tinyurl.com/5n88y7fh.

Causmaecker, P. D., Demeester, P., Berghe, G. V., and Verbeke, B. (2004). Analysis of real-world personnel scheduling problems. *Practice and Theory of Automated Timetabling*, 183-198. https://patatconference.org/patat2004/proceedings/183.pdf.

Charnes, A. and Cooper, W. (1977). Goal programming and multiple objective optimizations: Part 1. *European Journal of Operational Research*, 1(1), 39-54. https://doi.org/10.1016/S0377-2217(77)81007-2.

Cheang, B., Li, H., Lim, A., and Rodrigues, B. (2003). Nurse rostering problems—a bibliographic survey. European Journal of Operational Research, 151(3), 447-460. https://doi.org/10.1016/S0377-2217(03)00021-3.

Chiaramonte, M. V. (2008). *Competitive Nurse Rostering And Rerostering* (Publication No. ADA476319) [Doctoral dissertation, University of Arizona]. Defense Technical Information Center. https://apps.dtic.mil/sti/pdfs/ADA476319.pdf.

Chu, S. (2007). Generating, scheduling and rostering of shift crew-duties: Applications at the Hong Kong International Airport. *European Journal of Operational Research*, 177, 1764-1778. https://10.1016/j.ejor.2005.10.008.

Cunningham, S. (2021). *Causal Inference The Mixtape* (1st ed.). Yale Press. https://mixtape.scunning.com/.

Código do Trabalho, Chapter II (2009). https://www.pgdlisboa.pt/leis.

Dantzig, G. B. (1954). Letter to the Editor—A Comment on Edie's "Traffic Delays at Toll Booths". *Journal of the Operations Research Society of America*, 2(3), 183-198. https://doi.org/10.1287/opre.2.3.339.

Dantzig, G. B. and Wolfe, P. (1998). The Decomposition Algorithm for Linear Programs. *Econometrica*, 9(4), 767-778. https://doi.org/10.2307/1911818.

Davoine, T., Hammer, P. L., and Vizvári, B. (2003). A Heuristic for Boolean Optimization Problems. *Journal of Heuristics*, 9, 229-247. https://doi.org/10.1023/A:1023717307746.

Deloitte (2020). Convenience - Anything, Anytime, Anywhere. https://tinyurl.com/k55dc349.

Deveci, M. and Demirel, N. (2018). A survey of the literature on airline crew scheduling. *Engineering Applications of Artificial Intelligence*, 74, 54-69. https://doi.org/10.1016/j.engappai.2018.05.008.

Dewey, J. (2001). *Democracy and Education* (1st digital ed.). Pennsylvania State University Electronic Classics Series. https://www.gutenberg.org/files/852/852-h/852-h.htm.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1), 3-27. https://doi.org/10.1016/S0377-2217(03)00095-X.

Frey, L., Hanne, T., and Domberger, R. (2009). Optimizing staff rosters for emergency shifts for doctors. *2009 IEEE Congress on Evolutionary Computation*, 1, 2540-2546. https://doi.org/10.1109/CEC.2009.4983260.

Gamache, M. and Soumis, F. (2001). *A Method for Optimally Solving the Rostering Problem* (pp. 17-36). Springer US. In Gang Yu (Ed.). https://www.springer.com/series/6161.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (1st digital ed.). Addison-Wesley Professional. https://tinyurl.com/yckhnc2f.

Gonçalves, J. F. and Resende, M. G. C. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Jornal of Heuristics*, 17, 487-525. https://doi.org/10.1007/s10732-010-9143-1.

Gordon, L. and Erkut, E. (2004). Improving Volunteer Scheduling for the Edmonton Folk Festival. *Interfaces*, 34(5), 367-376. https://www.jstor.org/stable/25062936.

Hillier, F. S. and Lieberman, G. J. (2014). *Introduction to Operations Research* (10th digital ed., pp. 46-48). McGraw-Hill Education. https://tinyurl.com/k73pwstn.

Hvattum, L. M., Løkketangen, A., and Glover, F. (2012). Comparisons of Commercial MIP Solvers and an Adaptive Memory (Tabu Search) Procedure for a Class of 0-1 Integer Programming Problems. *Algorithmic Operations Research*, 7, 12-21. https://tinyurl.com/32bmds8y.

Jacobs, L. W. and Bechtold, S. (1993). Labor Utilization Effects of Labor Scheduling Flexibility Alternatives in a Tour Scheduling Environment. *Decision Sciences*, 24(1), 148-166. https://doi.org/10.1111/j.1540-5915.1993.tb00467.x.

Joines, J. and Houck, C. (1994). On the Use of Non-Stationary Penalty Functions to Solve Constrained Optimization Problems with Genetic Algorithms. *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, 2, 579-584. https://doi.org/10.1109/ICEC.1994.349995.

Kohl, N. and Karisch, S. (2004). Airline Crew Rostering: Problem Types, Modeling, and Optimization. *Annals of Operations Research*, 127(1), 223-257. https://doi:10.1023/B:ANOR.0000019091.54417.ca.

Li, J., Burke, E. K., Curtois, T., Petrovic, S., and Qu, R. (2012). The falling tide algorithm: A new multi-objective approach for complex workforce scheduling. *Omega*, 40(3), 283-293. https://doi.org/10.1016/j.omega.2011.05.004.

Lin, H.-T., Chen, Y.-T., Chou, T.-Y., and Liao, Y.-C. (2012). Crew rostering with multiple goals: An empirical study. *Computers & Industrial Engineering*, 63(2), 484-493. https://doi.org/10.1016/j.cie.2012.04.013.

Lu, Y., Musalem, A., Olivares, M., and Schilkrut, A. (2013). Measuring the Effect of Queues on Customer Purchases. *Management Science*, 59(8), 1743-1763. https://doi.org/10.1287/mnsc.1120.1686.

Lucena, M. L., Andrade, C. E., Resende, M. C. G., and Miyazawa, F. K. (2014). Some extensions of biased random-key genetic algorithms. *Proceedings of the 46th Brazilian Symposium of Operational Research*, 1, 2469–2480. http://www.din.uem.br/sbpo/sbpo2014/pdf/arq0357.pdf.

Luppold, A., Oehlert, D., and Falk, H. (2018). Evaluating the performance of solvers for integer-linear programming. *Eingebettete Systeme E-13*, https://doi.org/10.15480/882.1839.

Maenhout, B. and Vanhoucke, M. (2007). Comparison and hybridization of crossover operators for the nurse scheduling problem. *Annals of Operations Research*, 159(1), 333-353. https://doi.org/10.1007/s10479-007-0268-z.

Michalewicz, Z. (1995). A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. In J. R. McDonnell, R. G. Reynolds, D. B. Fogel (Eds.), *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*. *IEEE Transactions on Evolutionary Computation*, 1, 135-155. The MIT Press. https://doi.org/10.7551/mitpress/2887.001.0001.

Miller, H. E. and Pierskalla, W. P. (1976). Nurse Scheduling Using Mathematical Programming. *Operations Research*, 24(5), 857-870. https://doi.org/10.1287/opre.24.5.857.

Mitchell, M. (1999). *An introduction to Genetic Algorithms* (1st ed.). MIT Press. https://mitpress.mit.edu/9780262631853/an-introduction-to-genetic-algorithms/.

Mittelman, H. (2023). The MIPLIB2017 Benchmark Instances. https://http://plato.asu.edu/ftp/milp.html.

Montgomery, D. C. (2001). *Design and Analysis of Experiments* (5th ed.). John Wiley & Sons Inc. https://tinyurl.com/nh2z6x5t.

Ngoo, C. M., Goh, S. L., Sze, S. N., Saber, N. R., Abdullah, S., and Kendall, G. (2008). An Indirect Genetic Algorithm for a Nurse Scheduling Problem. *Computers Operations Research*, 31, 761-778. https://doi.org/10.1016/S0305-0548(03)00034-0.

Ngoo, C. M., Goh, S. L., Sze, S. N., Saber, N. R., Abdullah, S., and Kendall, G. (2022). A Survey of the Nurse Rostering Solution Methodologies: The State-of-the-Art and Emerging Trends. *IEEE Acess*, 10, 56504-56524. https://doi.org/10.1109/ACCESS.2022.3177280.

Oke, A. (2013). Linking manufacturing flexibility to innovation performance in manufacturing plants. *International Journal of Production Economics*, 143(2), 242-247. https://doi.org/10.1016/j.ijpe.2011.09.014.

Oldenkamp, J. H. (1996). *Competitive Nurse Rostering And Rerostering* (Publication No. ADA476319) [Master Thesis, University of Groningen]. University of Groningen Repository. https://research.rug.nl/en/publications/quality-in-fives-on-the-analysis-operationalization-and-applicati.

Page, K. M. and Vella-Brodrick, D. A. (2009). The 'What', 'Why' and 'How' of Employee Well-Being: A New Model. *Soc Indic Res*, 90, 441-458. https://doi.org/10.1007/s11205-008-9270-3.

Puente, J., Gómez, A., Fernández, I., and Priore, P. (2009). Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms. *Computers Industrial Engineering*, 56(4), 1232-1242. https://doi.org/10.1016/j.cie.2008.07.016.

Qu, R., Burke, E. K., McCollum, B., Merlot, L. T. G., and Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, 12, 55-89. https://doi.org/10.1007/s10951-008-0077-5.

Rahimian, E., Akartunalı, K., and Levine, J. (2017). A hybrid integer and constraint programming approach to solve nurse rostering problems. *Computers Operations Research*, 82, 84-94. https://doi.org/10.1016/j.cor.2017.01.016.

Richardson, J. T., Palmer, M. R., Liepins, G., and Hilliard, M. (1989). Some guidelines for genetic algorithms with penalty functions. *Proceedings of the Third International Conference on Genetic Algorithms*, 1, 191-197. https://dl.acm.org/doi/10.5555/93126.93190.

Rochman, A. N., Prasetyo, H., and Nugroho, M. T. (2017). Biased random key genetic algorithm with insertion and gender selection for capacitated vehicle routing problem with time windows. *AIP Conference Proceedings*, 1855(1), Article 020025. https://doi.org/10.1063/1.4985470.

Roy, R. K. (2001). *Design of Experiments Using the Taguchi Approach* (1st digital ed.). John WIley & Sons Inc. https://tinyurl.com/pdrs44ph.

Sadjadi, S. J., Soltani, R., Izadkhah, M., Saberian, F., and Darayi, M. (2011). A new nonlinear stochastic staff scheduling model. *Scientia Iranica*, 18(3), 699-710. https://doi.org/10.1016/j.scient.2011.05.017.

Sant'Anna, P. H. and Zhao, J. (2020). Doubly robust difference-in-differences estimators. *Journal of Econometrics*, 219(1), 101-122. https://doi.org/10.1016/j.jeconom.2020.06.003.

Schuster, R., Hanson, J., Strimas-Mackey, M., and Bennet, J. (2020). Exact integer linear programming solvers outperform simulated annealing for solving conservation planning problems. *PeerJ*, 27(8), Article e9258. https://doi.org/10.7717/peerj.9258.

Spears, W. and Young, K. D. (1991). On the Virtues of Parametrized Uniform Crossover. *Proceedings of the 4th International Conference on Genetic Algorithms*, 1, 225-232. https://tinyurl.com/29jxjpw2.

Topaloglu, S. (2006). A multi-objective programming model for scheduling emergency medicine residents. *Computers & Industrial Engineering*, 51(3), 275-388. https://doi.org/10.1016/j.cie.2006.08.003.

Topaloglu, S. and Ozkarahan, I. (2004). An Implicit Goal Programming Model for the Tour Scheduling Problem Considering the Employee Work Preferences. *Annals of Operations Research*, 128, 135-158. https://doi.org/10.1023/B:ANOR.0000019102.68222.df.

Tsai, C. C. and Li, S. H. A. (2009). A two-stage modeling with genetic algorithms for the nurse scheduling problem. *Expert Systems with Applications*, 36(5), 9506-9512. https://doi.org/10.1016/j.eswa.2008.11.049.

Valouxis, C., Gogos, C., Goulas, G., Alefragis, P., and Housos, E. (2012). A systematic two phase approach for the nurse rostering problem. *European Journal of Operational Research*, 219(2), 425-433. https://doi.org/10.1016/j.ejor.2011.12.042.

Whitley, D., Rana, S., and Heckendorn, R. B. (1998). The Island Model Genetic Algorithm: On Separability, Population Size and Convergence. *Journal of Computing and Information Technology*, 7(1), 125-142. https://tinyurl.com/3dh7tahj.

Wickert, T. I., Smet, P., and Vanden Berghe, G. (2019). The nurse rerostering problem: Strategies for reconstructing disrupted schedules. *Computers Operations Research*, 104, 319-337. https://doi.org/10.1016/S0377-2217(03)00021-3.

Yan, S. and Tu, Y.-P. (2002). A network model for airline cabin crew scheduling. *European Journal of Operational Research*, 140(3), 531-540. https://doi.org/10.1016/S0377-2217(01)00215-6.

Zolfaghari, S., Quan, V., El-Bouri, A., and Khashayardoust, M. (2010). Application of a genetic algorithm to staff scheduling in retail sector. *International Journal of Industrial and Systems Engineering*, 5(1), 20-47. https://doi.org/10.1504/IJISE.2010.029755.

Álvarez, E., Ferrer, J.-C., Muñoz, J. C., and Henao, C. A. (2020). Efficient shift scheduling with multiple breaks for full-time employees: A retail industry case. *Computers & Industrial Engineering*, 150, Article 106884. https://doi.org/10.1016/j.cie.2020.106884.

Özder, E. H., Özcan, E., and Eren, T. (2020). A Systematic Literature Review for Personnel Scheduling Problems. *International Journal of Information Technology & Decision Making*, 19(6), 1695-1735. https://doi.org/10.1142/S0219622020300050.

Özgür Kabak, Ülengin, F., Aktaş, E., Şule Önsel, and Topcu, Y. I. (2008). Efficient shift scheduling in the retail sector through two-stage optimization. *European Journal of Operational Research*, 184(1), 76-90. https://doi.org/10.1016/j.ejor.2006.10.039.

Appendix A

Theoretical Framework

A.1 Goal Programming

Any mathematical model of a business problem is an idealized representation of the world that's defined by two key elements: first of all, the decision maker is trying to minimize or maximize some measure of performance, which hereafter is defined as the objective function; secondly, he is not free to act on his own volition, but rather is restricted in his behavior in same regard. These restrictions are defined as constraints of the problem. Additionally, there are n related quantifiable decisions to be made, represented as decision variables (say x_1, x_2, ... , x_n), and constants that are not subject to modification, called parameters. The canonical form of a standard linear programming (LP) model (for minimization problems) is given by:

$$Minimize\ \ f = \sum_{j=1}^{n} c_j x_j \tag{A.1}$$

$$Subject\ to\ \sum_{j=1}^{n} a_{ij} x_j \geq b_i,\ for\ \ i = 1, \ldots, m \tag{A.2}$$

$$x_j \geq 0,\ for\ \ j = 1, \ldots, n \tag{A.3}$$

Mixed Integer Linear Programming models, first introduced by Dantzig and Wolfe (1998) in their seminal paper "The Decomposition Algorithm for Linear Programs", are an extension of the standard LP model, where some of the variables take integer values. Goal programming, as proposed originally by Charnes and Cooper (1977), was developed to handle multi-criteria situations within the general framework of Linear Programming. This framework implies that the decision-maker defines goals that he wants to achieve, where the objective is to find the solution that comes closest to meeting them. This is done so because, in some cases, infeasible solutions would occur if all constraints were attempted to be met. Consequently, by specifying deviation variables and including in the LP model goals as restrictions that can be satisfied with deviations, in addition to the other constraints of the original LP formulation, good solutions can actually be obtained. Weighted goal programming models are an extension of this model that include relative weights to be assigned to each positive or negative deviation variable so that different priorities can be assigned to each goal.

There are several algorithms to tackle integer linear programming problems. Branch-and-bound are the most common in software applications. The basic idea of branch-and-bound is to iteratively solve a LP relaxation of the problem and use the LP solution to guide the search for integer-feasible solutions. The first step of the algorithm is to solve the LP relaxation, which is obtained by relaxing the integer constraints on the variables. The LP relaxation is then solved, which efficiently finds a basic feasible solution that satisfies the linear constraints. If the LP solution is integer-feasible, then the algorithm terminates and returns the solution. Otherwise, the algorithm proceeds to the branching step. This involves selecting a non-integer variable in the LP solution and creating two new subproblems by branching on the variable. The variable is split into two parts, one taking the floor of its LP value and the other taking the ceiling of its LP value. The two new subproblems are created by adding a constraint that fixes the variable to either its floor or ceiling value in the respective subproblem.

In branch-and-cut, an additional cutting plane step is used to tighten the LP relaxation of the subproblems by adding valid inequalities, or "cuts", to the formulation. These cuts are generated by solving additional LPs, called separation problems, that identify facets of the feasible region that are violated by the current LP solution. The cuts are added to the formulation to eliminate the LP solution from the infeasible region defined by the cuts.

Branch-and-Price (Akella et al., 1977) is a hybrid optimization approach that combines the benefits of branch-and-bound and column generation methods to solve large-scale integer programming problems. In this method, at each node of the branch & bound tree, new columns are generated to enhance the linear programming relaxation solution. Since the number of columns in large integer programming problems can be overwhelming and handling all of them efficiently is challenging, branch-and-price excludes sets of columns from the LP relaxation. This exclusion is based on the understanding that many of these columns would have their associated variables set to zero in an optimal solution. To determine optimality, a sub-problem known as the "pricing problem" is solved. The purpose of the pricing problem is to identify columns that should be included in the basis. If such columns are found, the linear programming model is re-optimized to improve the solution. However, if no columns are identified for inclusion in the basis during the pricing problem and the LP solution fails to satisfy integrality conditions, branching occurs.

All algorithms terminates when the optimal solution is found, when the LP relaxation is proven to be infeasible, or when a termination condition is reached, such as a time or node limit. In the last case, it is possible to obtain solutions that are sub-optimal. If any algorithm terminates without finding a feasible solution, it returns the best LP solution found during the search.

A.2 Multi Parent Biased Random-Key Genetic Algorithm

The original formulation of Genetic Algorithms (GAs) is based on the theory of evolution formulated by Charles Darwin in the 19th century (Hillier and Lieberman, 2014) and was developed in the mid-sixties. Essentially, each member of a population is unique and different from every other member, such that its survivability is impacted by its phenotype (coded in the genotype). Consequently, some members will have a competitive advantage over others and are more likely to generate offspring. From generation to generation, there will thus be a tendency for a population to improve over time and become better suited to its environment. Not only can information be conveyed by inheritance, but mutations also play a critical role. In this sense, random modifications to some genes may change the features of the chromosome that a child inherits from its parents. While most of these alterations have a negative or null impact, some mutations provide quite drastic improvements. Both of these mechanisms help explain how a reproductive species evolves over time.

These concepts translate quite nicely into optimization problems. Genetic Algorithms combine survival of the fittest among string structures with a combination of structured and randomized methods for exchanging genetic information (Goldberg, 1989). Essentially, each member of a population is now a feasible solution, whose fitness is measured by the value of its objective function. In each iteration, a new population is created by pairing solutions randomly with each other. As the best solutions are more likely to be parents, the Genetic Algorithm tends to generate improving populations of trial solutions as it proceeds. Mutations occasionally occur so that certain children also can acquire features (sometimes desirable features) that are not possessed by either parent. This helps the Genetic Algorithm explore new parts of the solution space and escape local optima. Eventually, survival of the fittest should tend to lead a Genetic Algorithm to a trial solution (the best of any considered) that is at least nearly optimal.

Genetic Algorithms differ from other optimization and search procedures in some very fundamental ways (Goldberg, 1989; Mitchell, 1999), which make it an interesting and unique method. First of all, GAs require the natural parameter set of the optimization problem to be coded as a finite-length string over some finite alphabet. Therefore, they never deal with the parameters themselves. Additionally, as GAs work with several feasible solutions simultaneously, they are less likely to get stuck in local optima, in opposition to point-to-point methods, who run into the danger of locating false peaks in multimodal (multi-peaked) search spaces. Finally, and unlike many other methods, GAs use probabilistic transition rules to guide their search. In other words, they use random choice as a tool to obtain new solutions that likely leads to improvements.

Any basic Genetic Algorithm is comprised of three main key steps. First of all, it is necessary to create an initial population of feasible solutions. Secondly, there should be an iterative process that creates new populations in each trial. Naturally, this process should be biased to increase the likelihood that better solutions are chosen to become parents. Afterwards, each pair of parents should give birth to two children whose features are a random combination of the genes of its parents. Finally, only the best children and current members of the population should be retained to create the new population that is going to be subject to the same process. Finally, some stopping criteria must be defined, such as fixed number of iterations, fixed amount of CPU time or fixed number of iterations without any improvement. An illustrative example of pseudo code is presented in Table A.1.

Table A.1: Pseudo Code of the Genetic Algorithm

Pseudo code
Begin:
$g \leftarrow 0$;
Encode;
Initialize $P(g)$;
Evaluate $P(g)$
Select $P(g)$;
While (not termination condition) **do**:
Recombine $P(g)$ to yield $O(g)$;
Crossover;
Mutation;
Evaluate $O(g)$;
Select $P(g+1)$ from $P(g)$ and $O(g)$;
$g \leftarrow g+1$;
End;
Output Solution;
End;

Random-key Genetic Algorithms (RKGAs) were initially developed by Bean (1994) to target sequencing problems. In this particular context, each member of a population is now represented by a vector composed of N keys, such that each key is a random number contained in the interval [0,1[. It is afterwards necessary to develop a decoder that takes this vector of random-keys as an input and outputs a solution to the optimization problem at hand. In addition, the RKGA does not incorporate chromosome mutation. Instead, they introduce random individuals into the new population, referred to as immigrants or mutants. Translating this algorithm into the three steps previously described results in the following structure: Firstly, define an initial population of P random-key vectors, each with N keys, such that each value is generated randomly in the interval [0,1[. Secondly, at each generation, a subset of the best solutions labelled as the elite set are kept, some new solutions are generated via mating and the remainder are mutants (completely new individuals). In its original conception, mating is done by randomly selecting two members of the whole population and using a crossover operator such as uniform crossover (Spears and Young, 1991)) – for each gene, flip a biased coin to choose which parent passes its key to the child. Thus, each offspring can be tuned to be more or less likely to inherit characteristics of one of its particular parents. Finally, it is defined a stopping criterion for the optimization problem. Biased Random-key Genetic Algorithms (BRKGAs) are a variant of RKGA (Gonçalves and Resende, 2011), where one of the parents used for mating has a higher likelihood of being of a higher fitness than the other parent. While RKGA chooses randomly two elements from the entire population to generate offspring, BRKGA chooses one parent from the elite set and the other from the non-elite set. Consequently, and since the number of elite members of the population is by definition a reduced percentage of the total, elite members have a higher likelihood of being chosen and transmitting their characteristics. Thus, not only can elite members be favored in the selection process, but also in the parametrized uniform crossover by selecting a probability *pe* that is superior to 0.5. Finally, Lucena et al. (2014) presented two additional variants of BRKGA - Gender-Defining Biased Random-Key Genetic Algorithm (BRKGA-GD) and Multi-Parent Biased Random-Key Genetic Algorithm (BRKGA-MP). BRKGA-CD assigns a bit to each cromossome such that its value defines its gender. When applying crossover, two chromossomes of different gender are chosen from the elite and non-elite set. BRKGA-MP is a variant of BRKGA where more than two parents are used to generate offspring. Parameter π_t defines the number of parents for crossover, where π_e are elite parents and $\pi_t - \pi_e$ are non-elite parents. The probability of transmitting alleles from each parent to the offspring is determined by considering the parent's bias. The parent bias is established using a predetermined weighting bias function $\Phi : N \rightarrow R^+$, which operates based on the parent's rank r. They employed the bias functions originally proposed by Bresina (1994):

1. $Logarithmic : \phi(r) = \frac{1}{log(r+1)}$
2. $Linear : \phi(r) = \frac{1}{r}$
3. *Polynomial* (in d): $\phi(r) = r^{-d}$
4. $Exponential : \phi(r) = e^{-r}$
5. $Constant : \phi(r) = \frac{1}{\pi_t}$

Computational experiments highlighted that the BRKGA-MP outperformed both the BRKGA and BRKGA-GD for the set covering problem. The pseudo-code for the crossover procedure for this algorithm is presented in Table A.2.

Initially, the parents are selected and arranged based on their fitness. Subsequently, the probabilities of selecting a gene from each parent are calculated. The rank function, denoted as *rank* : $N \rightarrow N$, determines the position of a given chromosome in the sorted list of parents Q. This rank value is then passed to the bias function ϕ. It is important to note that a normalization step is necessary to compute accurate probabilities for each parent, as indicated by the weight vector.

Table A.2: Pseudo code of the BRKGA-MP Crossover Operator (in Lucena et al. (2004))

Pseudo code
Begin:
Uniformly sample π_e individuals from P_e and $\pi_t - \pi_e$ individuals from $\frac{P}{P_e}$. Let Q be the set of these individuals;
Sort Q in non-increasing order of fitness;
$total_{weight} \leftarrow 0$;
For q in Q **do**:
$weight(q) \leftarrow \phi(rank(q))$;
$total_{weight} \leftarrow total_{weight} + weight(q)$;
End;
For q in Q **do**:
$weight(q) \leftarrow \frac{weight(q)}{total_{weight}}$;
End;
Let c be an empty new chromosome;
For $i = 1$ to n **do**:
Select and individual $q \in Q$ at random, with probabilities defined by weight;
$c(i) \leftarrow q(i)$;
End;
End;

Moving on, for each gene, a parent is chosen using the widely recognized roulette wheel selection method, utilizing the probabilities calculated in the previous steps. The selected parent's allele is assigned to the gene of the newly created chromosome.

The BRKGA-MP algorithm also allows for more than one population to be evolving at the same time, with exchanges of members between them occurring at certain regular intervals. This approach is referred as the island model (Whitley et al., 1998) and is typically employed to enhance the diversity of individuals, leading to accelerated convergence and mitigating the risk of getting trapped in local optima. The following parameters thus have to be defined for each instance:

1. Population size P;
2. Number of independent populations N;
3. Elite set percentage $P_e(\%)$;
4. Mutant set percentage $P_m(\%)$;
5. Number of parents in the crossover π_t;
6. Number of elite parents in the crossover π_e;
7. Exchange interval l;
8. Number of exchange individuals i;
9. bias function ϕ;

Some of the biggest advantages of this implementation over other Genetic Algorithms are its ability to quickly converge to high-quality solutions and to be scalable to large optimization problems. However, it does require a significant amount of computational power and it can be quite sensitive to the parameters considered.

A.3 Causal Inference with Treatment and Control in Research Design

Causal inference designs are methodologies used to establish casual relationships between variables. Essentially, one is trying to measure the effect of some change, hereafter defined as treatment, on some outcome. When doing so, there are essentially two difficult problems to face: The fundamental problem of causal inference (FPCI) and the fundamental problem of statistical inference (FPSI) (Ferret, 2023). The former simply states that the effect of treatment is fundamentally unobservable, even when the sample size is infinite. This is so because it is not possible to actually know what would have been the outcome on the treated had they not receive treatment, hereafter defined as counterfactual. The most intuitive example is probably seen in the field of pharmacology. Let's consider that an adolescent named Michael is prescribed a pill for growth. A couple of months after treatment, the subject increased its height by 2 cm. What the FPCI implies is that this growth cannot necessarily be attributed to the effect of the pill on the adolescent. It might have been the case that he would have grown regardless of treatment. All of the existing methods thus try to find ways to estimate the counterfactual by using observable quantities that hopefully approximate it as well as possible. The FPSI states that even if there was an estimator that identified the effect of treatment in the population, it is not possible to observe said estimator because the whole population is not accessible. A finite sample is never representative of the entire population.

The gold standard for casual inference relies on randomized control trials, where participants are randomly assigned to treatment and it is possible to control for all cofounders (Angrist and Pischke, 2009). However, in natural experiments, treatment is typically either not randomly assigned or it is not possible to measure and control for all cofounders. In this context, the most intuitive way to get the effect of a treatment would be to measure the before- and after-treatment outcomes and simply see what changed. The biggest problem with this research design is that the outcome could have changed anyway - again, the adolescent could have grown even if he did not take the pill. To get around this problem, research designs typically bring in a control group that is always untreated. Three methodologies stand out that work particularly well in this setting: Difference-in-Differences (DiD) (Angrist and Pischke, 2009), Synthetic Control (Cunningham, 2021) and Matrix Completion (Athey et al, 2021). In DiD, the control group's change over time is used to estimate how much the treated group would have changed in the absence of treatment. To get the effect of treatment one simply subtracts this quantity by the observed change of the treated group. Figure A.1 illustrates this principle.

There are two key assumptions that DiD considers to be true - no-anticipation and parallel trends. No anticipation simply states that there is no observable treatment effect before the period of actual treatment. This assumption is particularly critical in politics with new legislative implementations, where citizens may preemptively react to the changes that will occur. The parallel trends assumption implies that the gap between treatment and control would have stayed the same, if treatment had not occurred. There is no actual way to validate these assumptions, but prior trends tests - seeing if the trends before treatment are statistically different - can be conducted to provide some statistical ground. There are several models within the DiD design, including Two-Way Fixed Effects, Doubly-Robust DiD (Sant'Anna and Zhao, 2020) and DiD with multiple time periods (Callaway and Sant'Anna, 2021). DiD with multiple time periods is a particular interesting implementation because it combines parametric linear regression with Abadie and Imbens (2016) matching procedures to generate double-robustness, while allowing for more than two time-periods. It also make it possible to include time-invariant covariates, reducing confounding. Going back to the example provided, it considers other adolescents to predict how Michael would have grown and it allows for features such as race and gender to be included in order to select those

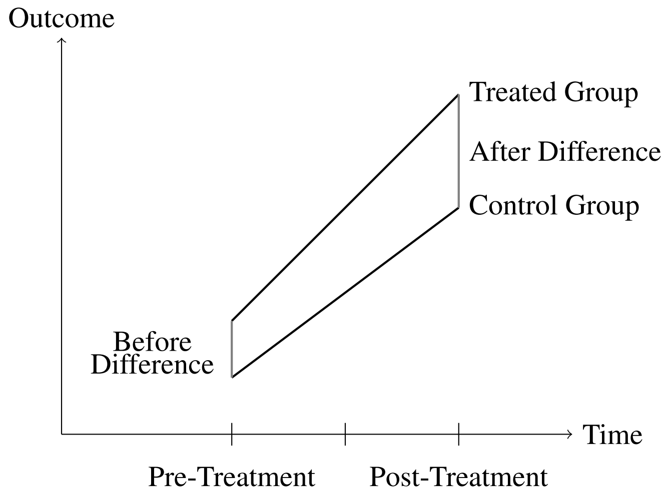

Figure A.1: Difference-in-Differences Illustration

subjects that are most similar to Michael. Synthetic Control is used in situations where a single unit received treatment, there is extensive pre-treatment data and no adequate control group. It constructs a synthetic control group by combining multiple similar units and relies on concept of exogeneity of weights, by assuming that the weights assigned to the control units in constructing the synthetic control are exogenous and not affected by treatment. This design is essentially trying to create a twin of Michael from several adolescents to predict how he would grow. Finally, Matrix completion tries to actually predict what the treated group would have been if they were untreated via regularized regressions. It attempts to predict what would have been Michael's height, had he not taken the pill.

Appendix B

Tables

Table B.1: Time-related Parameters Detailed Description

Constraint	Comment
Max # consecutive days	Restricting the maximum number of consecutive days that a worker can work for
Min # consecutive days	Restricting the minimum number of consecutive days that a worker can work for. This is to avoid day-on day-off day-on type of patterns
Max # consecutive days off	-
Min # consecutive days off	-
Min # hours	Restricting the minimum number of hours that a collaborator has to work for. Tries to avoid big discrepancies between workers
Days/shifts on/off	Allows for workers to specify certain days or shift that they are not available or are undesired
Time between assignments	Restricting a minimum number of hours between two consecutive shifts or task assignments
Patterns, consecutive shifts and sequence of shift type	Restricting the possible shift patterns that can considered
Free days after night shift	-
Max # weekends in # weeks	-
Complete (and extended) weekends	Workers prefer to work one full weekend with the knowledge they have another weekend off, rather than working two weekends for a single day
Max # consecutive weekends	-
Identical weekends	-
Time windows and deadline	Tasks are assigned such that workers have a time window to fulfill them. It is also possible specify specific deadlines for jobs
Resources related	Worker or jobs may require specific resources, which are not always available
Ratio groups of personnel	Specify ratios of types of personnel with different skills

Table B.2: Instances Detailed Information

Instance	Contract Type			Category Type			
	FTE	PTE	Maternity	Management	Top Performers	Experienced	Rookies
Instance 4	4	0	0	1	0	2	0
Instance 6	5	1	0	1	1	2	2
Instance 8	7	1	0	1	2	4	1
Instance 10	8	2	0	2	1	4	3
Instance 12	11	1	0	2	4	4	2
Instance 14	14	0	0	2	3	7	2
Instance 15	12	1	2	3	3	5	4
Instance 20	17	2	1	3	3	9	5
Instance 25	21	3	1	3	3	9	10
Instance 30	26	2	2	6	10	12	2
Instance 35	28	4	3	4	4	8	19
Instance 50	42	4	4	8	7	25	10

Table B.3: Orthogonal Array $L_{27}(3^{13})$

Trial	P	N	$P_e(\%)$	$P_m(\%)$	π_t/π_e	l	i	ϕ	C	α	β	SNR	Mean
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	1	1
3	1	1	1	1	3	3	3	3	3	3	3	1	1
4	1	2	2	2	1	1	1	2	2	2	3	1	1
5	1	2	2	2	2	2	2	3	3	3	1	1	1
6	1	2	2	2	3	3	3	1	1	1	2	1	1
7	1	3	3	3	1	1	1	3	3	3	2	1	1
8	1	3	3	3	2	2	2	1	1	1	3	1	1
9	1	3	3	3	3	3	3	2	2	2	1	1	1
10	2	1	2	3	1	2	3	1	2	3	1	1	1
11	2	1	2	3	2	3	1	2	3	1	2	1	1
12	2	1	2	3	3	1	2	3	1	2	3	1	1
13	2	2	3	1	1	2	3	2	3	1	3	1	1
14	2	2	3	1	2	3	1	3	1	2	1	1	1
15	2	2	3	1	3	1	2	1	3	2	2	1	1
16	2	3	1	2	1	2	3	3	1	2	2	1	1
17	2	3	1	2	2	3	1	1	2	3	3	1	1
18	2	3	1	2	3	1	2	2	3	1	1	1	1
19	3	1	3	2	1	3	2	1	3	2	1	1	1
20	3	1	3	2	2	1	3	2	1	3	2	1	1
21	3	1	3	2	3	2	1	3	2	1	3	1	1
22	3	2	1	3	1	3	2	2	1	3	3	1	1
23	3	2	1	3	2	1	3	3	2	1	1	1	1
24	3	2	1	3	3	2	1	1	3	2	2	1	1
25	3	3	2	1	1	3	2	3	2	1	2	1	1
26	3	3	2	1	2	1	3	1	3	2	3	1	1
27	3	3	2	1	3	2	1	2	1	3	1	1	1

Table B.4: Results of Analysis of Variance for Parameters of BRKGA-MP

Parameter	Degrees of Freedom	Sum of Squares	Mean Squares	F Value	P-value
P	2	4.53	2.26	5.39	0.07
N	2	0.06	0.03	0.06	0.94
$P_e(\%)$	2	1.49	0.74	1.77	0.28
$P_m(\%)$	2	0.47	0.23	0.56	0.61
π_t/π_e	2	0.70	0.35	0.83	0.50
l	2	0.85	0.42	1.01	0.44
i	2	0.97	0.49	1.12	0.40
ϕ	2	0.49	0.24	0.58	0.60
C	2	1.78	0.89	2.12	0.24
α	2	0.08	0.04	0.09	0.92
β	2	0.12	0.06	0.14	0.87
Residuals	4	1.68	0.42	-	-

$Notes: *p < 0.05, **p < 0.01, ***p < 0.001$

Table B.5: Stores Detailed Information

Store	Contract Type			Category Type			
	FTE	PTE	Maternity	Management	Top Performers	Experienced	Rookies
Store 1	7	2	0	2	3	3	1
Store 2	5	2	0	1	1	3	2
Store 3	6	1	0	1	2	4	0
Store 4	6	1	1	1	1	3	1
Store 5	9	0	1	2	2	3	3
Store 6	10	1	1	3	2	4	3
Store 7	6	0	0	1	1	4	0
Store 8	6	0	0	1	2	2	1
Store 9	9	0	0	3	2	3	2
Store 10	6	0	3	2	2	4	1
Store 11	7	0	0	1	3	1	2
Store 12	7	1	1	2	1	5	1
Store 13	6	0	1	2	1	3	2
Store 14	7	1	0	0	2	4	2
Store 15	5	2	0	1	2	3	1

Table B.6: Full Factorial Design Matrix

F1	F2	F3	F4	F5	Factor
-	-	-	-	-	(1)
+	-	-	-	-	A
-	+	-	-	-	B
+	+	-	-	-	AB
-	-	+	-	-	C
+	-	+	-	-	AC
-	+	+	-	-	BC
+	+	+	-	-	ABC
-	-	-	+	-	D
+	-	-	+	-	AD
-	+	-	+	-	BD
+	+	-	+	-	ABD
-	-	+	+	-	CD
+	-	+	+	-	ACD
-	+	+	+	-	BCD
+	+	+	+	-	ABCD
-	-	-	-	+	E
+	-	-	-	+	AE
-	+	-	-	+	BE
+	+	-	-	+	ABE
-	-	+	-	+	CE
+	-	+	-	+	ACE
-	+	+	-	+	BCE
+	+	+	-	+	ABCE
-	-	-	+	+	DE
+	-	-	+	+	ADE
-	+	-	+	+	BDE
+	+	-	+	+	ABDE
-	-	+	+	+	CDE
+	-	+	+	+	ACDE
-	+	+	+	+	BCDE
+	+	+	+	+	ACBDE

Table B.7: Results of Analysis of Variance for Soft Constraints and Interactions on Relative Count

Factor	Degrees of Freedom	Sum of Squares	Mean Squares	F Value	P-value
A	1	3.47	4.37	0.82	0.37
B	1	4.42	4.42	1.05	0.31
AB	1	0.35	0.35	0.08	0.78
C	1	1.83	1.83	0.43	0.51
AC	1	0.33	0.33	0.08	0.78
BC	1	0.46	0.46	0.11	0.74
ABC	1	0.38	0.38	0.09	0.76
D	1	0.92	0.92	0.22	0.64
AD	1	0.08	0.08	0.02	0.89
BD	1	0.22	0.22	0.05	0.82
ABD	1	0.18	0.18	0.04	0.84
CD	1	0.21	0.21	0.05	0.83
ACD	1	0.14	0.14	0.03	0.86
BCD	1	0.39	0.39	0.09	0.76
ABCD	1	0.16	0.16	0.04	0.85
E	1	1.41	1.41	0.33	0.57
AE	1	0.08	0.08	0.02	0.89
BE	1	0.22	0.22	0.05	0.82
ABE	1	0.15	0.15	0.04	0.85
CE	1	0.16	0.16	0.04	0.85
ACE	1	0.15	0.15	0.04	0.85
BCE	1	0.27	0.27	0.06	0.80
ABDE	1	0.17	0.17	0.04	0.84
DE	1	0.16	0.16	0.04	0.85
ADE	1	0.19	0.19	0.05	0.83
BDE	1	0.37	0.37	0.09	0.77
ABDE	1	0.17	0.17	0.04	0.84
CDE	1	0.22	0.22	0.05	0.82
ACDE	1	0.09	0.09	0.02	0.89
BCDE	1	0.27	0.27	0.06	0.80
ABCDE	1	0.23	0.23	0.06	0.81
Residuals	64	270.04	4.22	-	-

$Notes: *p < 0.05, **p < 0.01, ***p < 0.001$

Table B.8: Resources Parameters Detailed Description

Parameter	Detail
Resource	Unique identifier of each collaborator
Function	GL (store responsible), CE (management) or AL (shopkeeper). GL is not considered for headcount and works from [opening hours- opening hours + 9] on weekdays
Senior	Binary indicators that specifies whether a collaborator is preferred (1) or not to open the store
Classification	Top Performer, Normal or Rookie. Relevant to guarantee balancing between top performers and rookies at every working block
Contract	Part-time 4h, part-time 5h, part-time 6h or full-time 8h
Maternity License	Binary indicator that specifies whether a collaborator is on maternity license or not. Subject to a specific schedule (as detailed in chapter 4)
Days-off	Specify specific days of the month that the worker rests
[Monday...Sunday] day off	Binary indicator that specifies whether the workers always has a day off on that specific day of the week
From...To	Specify the starting and ending periods of the worker on that specific day of the week. If, for example, it is defined [9-21], all schedules that include working blocks before 09:00h or after 21:00 are excluded

Appendix C

Figures

Recursos de loja

							Período Folgas #1		Período Folgas #2		Período Ausência #1			Período Ausência #2			Período Ausência #3		
Código Empregado	Recurso	Função	Sénior?	Classificação	Contrato	Licença Amamentação	Início folga	Fim folga	Início folga	Fim folga	Início	Fim	Tipo	Início	Fim	Tipo	Início	Fim	Tipo
7324689343	Diana	GL	1	Top performer	Full time	0													
3299459056	Edgar	CE	1	Top performer	Full time	0	17-06-23	18-06-23	12-06-23	12-06-23	26-06-23	30-06-23	Férias						
2346546423	Cristina	CE	1	Normal	Full time	0													
2395484329	Catarina	AL	1	Top performer	Full time	0					01-06-23	09-06-23	Férias						
3240459329	Alessandra	AL	0	Normal	Full time	1	01-06-23	02-06-23			12-06-23	23-06-23	Férias						
2348218432	Nuno	AL	0	Normal	Full time	0	03-06-23	04-06-23											
1670549832	Ademir	AL	0	Rookie	Half time 4h	0	01-06-23	02-06-23	12-06-23	30-06-23									

Figure C.1: Store Parameters

Recursos de loja

							2ª feira			3ª feira			4ª feira			5ª feira			6ª feira			Sábado			Domingo		
Código Empregado	**Recurso**	**Função**	**Sénior?**	**Classificação**	**Contrato**	**Licença Amamentação**	**Folga?**	**De**	**Até**	**Folga?**	**De**	**Até**	**Folga?**	**De**	**Até**	**Folga?**	**De**	**Até**	**Folga?**	**De**	**Até**	**Folga?**	**De**	**Até**	**Folga?**	**De**	**Até**
7324689343	Diana	GL	1	Top performer	Full time	0	0			0			0			0			0			1			1		
3299459056	Edgar	AL	1	Top performer	Full time	0	0			0			0			0			0			0			0		
2346546423	Cristina	AL	1	Normal	Full time	0	0			0			0			0			0			1			1		
2395484329	Catarina	AL	1	Top performer	Full time	0	0			0			0			0			0			0			0		
3240459329	Alessandra	AL	0	Normal	Full time	1	0			0			0			0			0			0			0		
2348218432	Nuno	AL	0	Normal	Full time	0	0			0			0			0			0			0			0		
1670549832	Ademir	AL	0	Rookie	Half time 4h	0	0			0			0			0			0			0			0		

Figure C.2: Store Parameters (Continuation)

Dia	Bloco	9:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00	14:30	15:00	15:30	16:00	16:30	17:00	17:30	18:00	18:30	19:00	19:30	20:00	20:30	21:00	21:30	22:00	22:30
02-06-23	Total	0.5	4.5	4.2	3.2	5.2	5.4	3.8	3.9	3.8	4.2	3.7	4.3	4.3	4.0	4.0	3.2	3.2	2.6	2.6	3.0	3.0	2.8	2.8	2.4	2.6	0.4	0.4
Sexta	Atendimento	0.5	3.7	3.7	3.2	3.2	2.8	2.8	3.3	3.3	3.7	3.7	4.3	4.3	4.0	4.0	3.2	3.2	2.6	2.6	3.0	3.0	1.8	1.8	1.9	1.9	0.4	0.4
	Tarefas indirectas	0.0	0.8	0.5	0.0	2.0	2.7	1.0	0.7	0.5	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.5	0.7	0.0	0.0
	Reunião IOS Daily Abertura	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Reunião IOS Daily 14 h	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Declaração de vendas/Ac. Vendas Pend	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	CAT envio/recepção	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0
	Tratamento Material Danificado	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Envio SPV/Logic	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0
	Tratamento de PLI	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	IOP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure C.3: Visualization of Daily Demand

Dia	Hora	Dia da Semana	Peso do dia no mês (%)	Peso da hora no dia	Forecast automático	Alteração	Forecast final
01-06-23	0:00	5ª feira	0.00%	0.00%	0		0
01-06-23	1:00	5ª feira	0.00%	0.00%	0		0
01-06-23	2:00	5ª feira	0.00%	0.00%	0		0
01-06-23	3:00	5ª feira	0.00%	0.00%	0		0
01-06-23	4:00	5ª feira	0.00%	0.00%	0		0
01-06-23	5:00	5ª feira	0.00%	0.00%	0		0
01-06-23	6:00	5ª feira	0.00%	0.00%	0		0
01-06-23	7:00	5ª feira	0.00%	0.00%	0		0
01-06-23	8:00	5ª feira	0.00%	0.00%	0		0
01-06-23	9:00	5ª feira	3.60%	1.37%	2		2
01-06-23	10:00	5ª feira	3.60%	10.52%	14		14
01-06-23	11:00	5ª feira	3.60%	9.59%	13	17	17
01-06-23	12:00	5ª feira	3.60%	7.55%	10		10
01-06-23	13:00	5ª feira	3.60%	7.32%	10		10
01-06-23	14:00	5ª feira	3.60%	9.73%	13		13

Figure C.4: Visualization of Hourly Demand

Escolher Semana: **05-06-23**

Recurso	Função	2ºa feira			3ºa feira			4ºa feira			5ºa feira			6ºa feira			Sábado			Domingo		
		Ent	Ref	Saí	Ent	Ref	Saí	Ent	Ref	Saí	Ent	Ref	Saí	Ent	Ref	Saí	Ent	Ref	Saí	Ent	Ref	Saí
Diana	GL	9:00	12:30	18:00	9:00	12:30	18:00	9:00	12:30	18:00	9:00	12:30	18:00	9:00	12:30	18:00	F	F	F	F	F	F
Edgar	CE	F	F	F	9:30	13:30	18:30	10:00	14:30	19:00	10:00	13:30	19:00	F	F	F	14:00	17:00	23:00	11:00	14:00	20:00
Cristina	CE	12:00	16:00	21:00	12:00	15:00	21:00	14:00	17:30	23:00	12:00	15:30	21:00	12:00	16:00	21:00	F	F	F	F	F	F
Catarina	AL	Férias	Férias	Férias	Férias	Férias	Férias	Férias	Férias	Férias	Férias	Férias	Férias	Férias	Férias	Férias	F	F	F	F	F	F
Alessandra	AL	F	F	F	10:00	14:30	17:00	10:00	15:00	17:00	9:30	14:30	16:30	10:00	15:00	17:00	9:00	13:30	16:00	F	F	F
Nuno	AL	14:00	17:00	23:00	F	F	F	9:30	14:00	18:30	F	F	F	9:30	14:00	18:30	11:30	14:30	20:30	9:00	13:00	18:00
Ademir	AL	9:30	14:00	18:30	14:00	17:00	23:00	F	F	F	14:00	17:00	23:00	14:00	17:30	23:00	F	F	F	13:00	17:00	22:00

Figure C.5: Visualization of Weekly Results

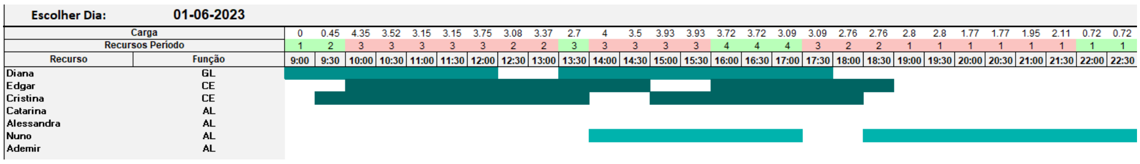

Figure C.6: Visualization of Daily Results

Appendix D

Equations

$$\begin{aligned}\hat{y} =& -1.22 + 0.28x_A + 0.21x_B + 0.16x_Ax_B + 0.11x_C + -0.02x_Ax_C + 0.28x_Bx_C - 0.15x_Ax_Bx_C \\ &+ 0.10x_D - 0.03x_Ax_D - 0.02x_Bx_D + 0.01x_Ax_Bx_D + 0.04x_Cx_D + 0.21x_Ax_Cx_D - 0.18x_Bx_Cx_D \\ &+ 0.02x_Ax_Bx_Cx_D + 0.12x_E - 0.05x_Ax_E - 0.04x_Bx_E + 0.15x_Ax_Bx_E + 0.03x_Cx_E + 0.12x_Ax_Cx_E \\ &+ 0.15x_Bx_Cx_E + 0.02x_Ax_Bx_Cx_E + 0.04x_Dx_E + 0.07x_Ax_Dx_E + 0.30x_Bx_Dx_E - 0.40x_Ax_Bx_Dx_E + \\ &+ 0.21x_Cx_Dx_E - 0.37x_Ax_Cx_Dx_E - 0.39x_Bx_Cx_Dx_E + 0.62x_Ax_Bx_Cx_Dx_E\end{aligned} \tag{D.1}$$

$$\hat{y} = -2.74 + 0.38x_A + 0.43x_B + 0.28x_C + 0.20x_D + 0.24x_E \tag{D.2}$$

$$\begin{aligned}\hat{y} =& -2.61 + 0.37x_A + 0.39x_B - 0.10x_Ax_B + 0.25x_C + -0.14x_Ax_C - 0.16x_Bx_C + 0.35x_Ax_Bx_C \\ &+ 0.21x_D - 0.12x_Ax_D - 0.12x_Bx_D + 0.08x_Ax_Bx_D - 0.12x_Cx_D + 0.14x_Ax_Cx_D + 0.16x_Bx_Cx_D \\ &- 0.13x_Ax_Bx_Cx_D + 0.29x_E - 0.18x_Ax_E - 0.14x_Bx_E + 0.16x_Ax_Bx_E - 0.16x_Cx_E + 0.26x_Ax_Cx_E \\ &+ 0.18x_Bx_Cx_E - 0.35x_Ax_Bx_Cx_E - 0.15x_Dx_E + 0.18x_Ax_Dx_E + 0.14x_Bx_Dx_E - 0.12x_Ax_Bx_Dx_E + \\ &+ 0.11x_Cx_Dx_E - 0.37x_Ax_Cx_Dx_E - 0.31x_Bx_Cx_Dx_E + 1.58x_Ax_Bx_Cx_Dx_E\end{aligned} \tag{D.3}$$