Universidade do Minho
Escola de Engenharia

João Manuel Silva Fonseca

Unrelated Parallel Machine Scheduling Problem: A Cement Industry Case Study

Outubro de 2018

Universidade do Minho
Escola de Engenharia
Departamento de Produção e Sistemas

João Manuel Silva Fonseca

Unrelated Parallel Machine Scheduling Problem: A Cement Industry Case Study

Dissertação de Mestrado
Mestrado em Engenharia de Sistemas

Trabalho realizado sob orientação de

Professor Doutor José António Vasconcelos Oliveira
Professor Doutor Luís Miguel Silva Dias

Outubro de 2018

DECLARAÇÃO

Nome: João Manuel Silva Fonseca

Endereço eletrónico: jmsf.14@gmail.com

Título da Dissertação: Unrelated Parallel Machine Scheduling: A Cement Industry Case Study

Orientadores:

Professor Doutor José António Vasconcelos Oliveira

Professor Doutor Luís Miguel Silva Dias

Ano de conclusão: 2018

Mestrado em Engenharia de Sistemas

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, ___ - ___ - ___

O Autor: ____

To my parents...

Acknowledgements

This document ends an important stage of my life. I use this space to express my gratitude to those who have helped me in so many ways.

To my supervisors, José António Oliveira, Ph.D. and Luís Dias, Ph.D., for the total availability, opinions, criticisms and for opening up the scientific research world to me.

To my family. Without them, this journey would not be possible. To my parents, to whom I dedicate this dissertation, for my education and for giving me great values, which I take to life. To my brother, for the constant support and to Filomena Alves, who always treated me like a son.

To Francisca Fonseca, who makes me a better man, every day. For being my confidant, for her love and affection, for her good mood, patience and for always being by my side.

To Ricardo Alves and Regina Macedo, who walked on this journey with me. For being the best group of work, for all the advice and words of encouragement, but specially for their friendship.

To André Souto, Afonso Rodrigues, Joaquim Santos, Miguel Nogueira, Miguel Sanches, Rui Costa and Zé Bruno, who accompanied me in these five years of learning. For the laughs, for their friendship and for being always prompt to help.

UNIVERSITY OF MINHO

Abstract

School of Engineering
Department of Production and Systems

by João Manuel Silva Fonseca

This dissertation considers the problem of scheduling unrelated parallel machines, with unequal release dates and machine eligibility constraints, to minimize the total flow time of the system. It establishes an analogy between this problem and an existing process in the cement industry – the loading of trucks by the customers. Hence, it intends to find opportunities for improvement in the reduction of the customers' interaction times and in their experience inside the cement plants. To achieve this goal, three optimization models are proposed, one exact and two heuristics. Also, an extensive series of computational tests are carried out to compare the performance of the methods. The exact method, based on a mathematical formulation of the problem, requires a high computational time and it is incapable of dealing with large instances. Consequently, it is not a viable solution for an industrial sized problem. However, it contributes to a better understanding of the structure of the problem and to develop efficient heuristics. The heuristics, one based on dispatching rules and the other on a simulated annealing algorithm, show potential for the implementation in a real life scenario. Although simulated annealing gives considerably better solutions than the other heuristic, it takes more time to give results and it is more complex to implement. The dispatching rules based heuristic gives solutions almost instantly and more easily includes certain characteristics of the problem. In general, these methods improve the quality of service provided, reducing the overall time the customers are spending inside the cement plants. Thus, cement industry can and should use optimization models to improve their operations and the customers' experience.

Keywords: Cement Industry, Machine Scheduling, Optimization Models, Mathematical Programming, Dispatching Rules, Simulated Annealing, Total Flow Time.

Supervised by:
Ph.D. Professor José António Oliveira
Ph.D. Professor Luís Dias

UNIVERSIDADE DO MINHO

Resumo

Escola de Engenharia
Departamento de Produção e Sistemas

feito por João Manuel Silva Fonseca

Esta dissertação considera o problema de agendamento de máquinas paralelas não relacionadas, com datas de disponibilidades diferentes e restrições de elegibilidade, para minimizar o tempo total de fluxo do sistema. Esta estabelece também uma analogia entre este problema e um processo existente na indústria cimenteira – o carregamento de camiões pelos clientes. Assim, pretende encontrar oportunidades de melhoria na redução dos tempos de interação dos clientes e na sua experiência dentro das cimenteiras. Para atingir este objetivo, três modelos de otimização são propostos, um exato e duas heurísticas. Além disso, uma extensa série de testes computacionais é realizada para comparar o desempenho dos métodos. O método exato, baseado numa formulação matemática do problema, requer bastante tempo computacional e é incapaz de lidar com instâncias grandes. Consequentemente, não é uma solução viável para um problema de tamanho industrial. No entanto, contribui para uma melhor compreensão da estrutura do problema e para desenvolver heurísticas eficientes. As heurísticas, uma baseada em regras de despacho e a outra num algoritmo de *simulated annealing*, mostram potencial para uma implementação num cenário da vida real. Embora o *simulated annealing* ofereça soluções consideravelmente melhores do que a outra heurística, este necessita de mais tempo para fornecer resultados e é mais complexo de implementar. A heurística baseada em regras de despacho fornece soluções quase instantaneamente e pode incluir mais facilmente certas características do problema. Em geral, estes métodos melhoram a qualidade do serviço prestado, reduzindo o tempo total que os clientes gastam dentro das cimenteiras. Assim, a indústria cimenteira pode e deve usar modelos de otimização, para melhorar as suas operações e a experiência dos clientes.

Palavras-Chave: Indústria Cimenteira, Agendamento de Máquinas, Modelos de Otimização, Programação Matemática, Regras de Despacho, *Simulated Annealing*, Tempo Total de Fluxo.

Orientado por:
Professor Dr. José António Oliveira
Professor Dr. Luís Dias

Contents

1 Introduction **1**

Part I State of the Art

2 Supply Chain Management **9**
2.1 Definition and Overview 9
2.2 Logistics Management 11
2.3 Customer Service and Value 13

3 Machine Scheduling **15**
3.1 Background 15
3.2 Problem Representation 16
3.3 Complexity 19

4 Solving Techniques **23**
4.1 Mathematical Programming 23
4.2 Constructive and Improvement Heuristics 26
4.2.1 Dispatching Rules 27
4.2.2 Simulated Annealing 30

Part II Case Study

5 Cement Industry **37**
5.1 Contextualization 37
5.2 Problem Description and Assumptions 41
5.3 Approaches 43
5.3.1 Mathematical Programming 45
5.3.2 Dispatching Rule 47
5.3.3 Simulated Annealing 47

6 Tests and Results **53**
6.1 Cement Plant I 55
6.2 Cement Plant II 57
6.3 Cement Plant III 59

7 Conclusions **67**

A Cement Plant I **71**

B Cement Plant II **81**

C Cement Plant III **91**

Bibliography **99**

List of Figures

2.1 Basic SC and its main flows. 9
2.2 Logistics management process, as part of a SC. 11
2.3 Product and its added value through customer service. 13

3.1 Complexity hierarchy on the machine environment of scheduling problems. 20
3.2 Complexity hierarchy on jobs and machines characteristics. 21
3.3 Complexity hierarchy on the performance criteria considered. 21

4.1 Acceptance function behavior with T and Δ. 31

5.1 Important locations of a typical CP (*SLV Cement by Cachapuz Bilanciai Group*, n.d.). 38
5.2 Cement manufacturing process, from raw material to end customer (Thomas and Lea, 2018). 40
5.3 Example of connections between silos and its respective LPs. 41
5.4 Example of a neighborhood solution by a switch. 50
5.5 Example of a neighborhood solution by a shift. 50
5.6 Example of a neighborhood solution by a swap. 51
5.7 Example of a neighborhood solution by a task move. 51

6.1 Combination between silos and LPs of CPI. 55
6.2 Combination between silos and LPs of CPII. 57
6.3 Combination between silos and LPs of CPIII. 59
6.4 SA improvements in CPIII. 61
6.5 Flow time behavior with R variation, using SA results. 63
6.6 Running time exponential behavior of the MP method. 65
6.7 Example of Gantt chart, using the SA on the n10R1.00 instance. 66

A.1 SA improvements in CPI. 77
A.2 Flow time behavior with R variation, using the SA results. 78
A.3 Flow time behavior with R variation, using the DR results. 78
A.4 Running time exponential behavior of the MP method. 79

B.1 SA improvements in CPII. 87
B.2 Flow time behavior with R variation, using the SA results. 88
B.3 Flow time behavior with R variation, using the DR results. 88
B.4 Running time exponential behavior of the MP method. 89

C.1 Flow time behavior with R variation, using the DR results. 97

List of Tables

4.1 Analogy between the annealing process and a combinatorial optimization problem. 32

6.1 Demand of each material available in CPI. 56
6.2 Flow time results in CPI. 56
6.3 Demand of each material available in CPII. 58
6.4 Flow time results in CPII. 58
6.5 Demand of each material available in CPIII. 59
6.6 Flow time results in CPIII. 60
6.7 Processing and Waiting Times Results in CPIII. 62
6.8 Running times of the computational tests in CPIII. 64
6.9 Example of schedule, using the SA on the n10R1.00 instance. 66

A.1 Generated data, for the customers of CPI - 10 jobs instance. 71
A.2 Generated data, for the customers of CPI - 12 jobs instance. 71
A.3 Generated data, for the customers of CPI - 50 jobs instance. 72
A.4 Generated data, for the customers of CPI - 100 jobs instance. 72
A.5 Generated data, for the customers of CPI - 200 jobs instance. 74
A.6 Processing and Waiting Times Results in CPI. 77
A.7 Running times of the computational tests in CPI. 79

B.1 Generated data, for the customers of CPII - 10 jobs instance. 81
B.2 Generated data, for the customers of CPII - 15 jobs instance. 81
B.3 Generated data, for the customers of CPII - 50 jobs instance. 82
B.4 Generated data, for the customers of CPII - 100 jobs instance. 82
B.5 Generated data, for the customers of CPII - 200 jobs instance. 84
B.6 Processing and Waiting Times Results in CPII. 87
B.7 Running times of the computational tests in CPII. 89

C.1 Generated data, for the customers of CPIII - 10 jobs instance. 91
C.2 Generated data, for the customers of CPIII - 15 jobs instance. 91
C.3 Generated data, for the customers of CPIII - 50 jobs instance. 92
C.4 Generated data, for the customers of CPIII - 100 jobs instance. 92
C.5 Generated data, for the customers of CPIII - 200 jobs instance. 94

Acronyms

ACO	**A**nt **C**olony **O**ptimization
CI	**C**ement **I**ndustry
CP	**C**ement **P**lant
DR	**D**ispatching **R**ule
ECT	**E**arliest **C**ompletion **T**ime
EDD	**E**arliest **D**ue **D**ate
ERD	**E**arliest **R**elease **D**ate
FAM	**F**irst **A**vailable **M**achine
FIFO	**F**irst **I**n **F**irst **O**ut
FCFS	**F**irst **C**ome **F**irst **S**erved
GA	**G**enetic **A**lgorithm
HC	**H**ill **C**limbing
ILS	**I**terated **L**ocal **S**earch
LFJ	**L**east **F**lexible **J**ob
LP	**L**oading **P**oint
LPT	**L**ongest **P**rocessing **T**ime
LST	**L**east **S**lack **T**ime
MS	**M**achine **S**cheduling
MP	**M**athematical **P**rogramming
SA	**S**imulated **A**nnealing
SC	**S**upply **C**hain
SCM	**S**upply **C**hain **M**anagement
SPT	**S**hortest **P**rocessing **T**ime
SQ	**S**hortest **Q**ueue
TS	**T**abu **S**earch
UH4SP	**U**nified **H**ub for **S**mart **P**lants

Chapter 1

Introduction

Contextualization

This dissertation emerges under the scope of the Unified Hub for Smart Plants (UH4SP) research project. In this project, the ALGORITMI Research Center, at University of Minho, joins several teams to help the company Cachapuz to take the current industry to another level. This company has its focus on the Cement Industry (CI), dealing with Cement Plants (CPs) geographically spread all over the world and with varied dimensions. Its main area of intervention is targeted to the control of logistics flows, seeking to improve all processes, from the entrance of a truck in the CP, to the cement delivery.

This research project appears at a time of renewal of this industry. The global market is changing from product oriented to customer oriented, and the CI also wants to make that change. Wanting to steer their businesses towards the customers, differentiated solutions that add value and promote a better level of service are sought. In order to develop the CI Supply Chain (SC), the UH4SP intends to introduce the concept of efficiency, effectiveness and intelligence to the CI. Here, the Smart Plant concept seeks to automate processes, reduce interaction times and improve the customers' experience for the partners involved in the logistics processes.

The delivery of orders to customers is one of the areas that has great opportunities for improvement in this industry. The daily arrival of hundreds of customers to a CP, looking for their trucks to be loaded, is responsible for most of the entropy within the facilities. Long waiting and processing times, the disruption of operations, and others, are just examples of the consequences of incorrect handling of a high flow of customers. These lead to customer's discontent and damages the service level, at the same time leading to high operating costs and inefficient use of resources and installed capacity.

In order to overcome this situation, the idea of scheduling, for the loading processes, aims to increase the organization within the CPs and to promote an improvement of the service levels. The scheduling of deliveries is one of the most important tasks in the Supply Chain Management (SCM), since it is directly linked to the customer. This is a process that determines the flow of resources and can be an indicator of the SC performance, through customer's satisfaction. Ensuring the right product, at the right time, in the exact quantity, to the authorized person and in perfect conditions are challenges that require a high level of organization. In this context, optimization models, applied to logistics flows, assume a special importance in order to create a favorable agenda for both the company and its customers.

In this work, a scientific research will be carried out and it aims to address the problem described above. The themes of SCM and logistics processes, which are aimed at customer's satisfaction, will be highlighted. An innovative approach to the loading schedule will be presented, establishing an analogy between the delivery of orders and the Machine Scheduling (MS) problems. Three optimization models will be developed from scratch and presented as solutions to the problem in question. Their performances will be analyzed and conclusions will be drawn, aiming a future implementation in real life. The ambition is to create a new paradigm in the scheduling of loading trucks by the customers in the CI.

Objectives

Purposing to tackle the problems stated before, this dissertation intends to:

- Recognize the lack of SCM in the CI and acknowledge the existence of improvement opportunities in the processes of delivering orders to the customers.
- Establish similarities between this problem and others already existing in the literature, in order to comprehend the best strategies to follow and to corroborate the chosen approaches.
- Recognize this challenge as a logistics optimization problem, which is highly combinatorial and difficult to solve.
- Identify the main variables and processes, which best describe and constraint the considered problem, to characterize it in an appropriate way as close as possible to the reality.

– Create scheduling solutions for loading trucks which reduce the interaction times and promote a better level of customer service and a greater organization of the logistics flows inside the premises.

– Perform computational tests, in multiple instances, allowing to draw conclusions about the performance of the developed solutions.

– Analyze the characteristics of the developed solutions in order to understand which is the best strategy for a future implementation in real life.

Dissertation Outline

To achieve these goals, this document is organized as follows. In the first chapter, a brief contextualization to the theme of this dissertation will be given. Also, the main objectives of this work and the structure of this document will be presented. Afterwards, two major parts divide the main contributions of this document, one more theoretical and the other more practical. The first part, which contains the following three chapters, presents a literature review that addresses this work main areas of intervention. The review intends to make a thorough study, based on scientific works, which will play as groundwork to the contribution developed in this dissertation. Thus, **Chapter 2** presents the main tasks of the SCM and logistics management. These address a major challenge, which is to keep all partners connected in a SC and ensure a correct management of all operations. Here, the concepts of level of service and customer's value are also addressed and customer's satisfaction emerges as the bigger goal of a SC. **Chapter 3** presents several types of problems that exist in the MS field. These can differ in the machine environment, the characteristics of the jobs to be processed, as well as the objective to minimize. Due to the high diversity of problems in this area, a systematic notation, commonly used in the literature, is also presented. In addition, emphasis is also given to the complexity of these problems and the impact it may have on building an efficient scheduling model. In **Chapter 4**, some solving techniques for the MS problems are presented – an exact method and two heuristics. Here, their main characteristics and advantages will be discussed and an extensive list of applications will be enumerated for each one of the three methods. The second part of the document begins with **Chapter 5**. Here, the CI is introduced, starting with its contextualization. Thus, a small characterization of this industry is made, highlighting the current challenges it faces and the need to turn its orientation towards the customers. A detailed description of this industry SC is also made, where all processes, from the extraction of raw materials to the delivery of the final product to the consumers, are explained. It is intended to exhibit all players in the SC and highlight its complexity. Afterwards, the problem that this work aims to solve is presented, which focuses on a specific part of the SC. Here, a detailed description of the process in question is presented, enumerating the main variables inherent to the problem as well as a list of the assumptions taken, before solving the problem. This chapter ends with the approach to the problem stated before and with the explanation of the three optimization models that were implemented to tackle it. These have several different characteristics and are based on the previously introduced solving techniques. This part ends with **Chapter 6**, which is dedicated to the testing phase of the developed optimization models, in several instances. This chapter begins with a description of the conditions under which the instances were constructed. Then, three CPs of different characteristics and dimensions are introduced, where the computational tests were performed. For each of them, the main results are shown and a discussion of the results is made. Finally, **Chapter 7** presents the main conclusions drawn from this work and gives indications to what should be the future work.

List of Publications

The study of the CI aroused great interest and there were found many opportunities for improvement. In the sense of sharing this knowledge with the rest of the world, 6 publications were made. Two of these were presented at conferences of scientific nature and the remaining four were published in specialized magazines of cement. These publications address not only the theme of this dissertation but also other challenges in the CI. Among these are the problems of routes, warehouse and quay management, simulation and the environmental impact of this industry. Hence, the full list of publications is as follows.

1. **Fonseca, J.**, Alves, R., Macedo, A. R., Oliveira, J. A., Pereira, G. and Carvalho, M. S. (2019), Integer programming model for ship loading management, in J. Machado, F. Soares and G. Veiga, eds, Innovation, Engineering and Entrepreneurship, Springer International Publishing, Cham, pp. 743-749.

2. Macedo, A. R., **Fonseca, J.**, Alves, R., Oliveira, J. A. , Carvalho, M. S., Pereira, G. (2018). The impact of Industry 4.0 to the environment in the cement industry supply chain. Proceedings of ECOS 2018 - The 31st International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems (ECOS). Presented at the ECOS 2018 Conference.

3. Alves, R., **Fonseca, J.**, Macedo, R., Veloso, H., Dias, L., Pereira, G., Carvalho, M. S., Figueiredo, M., Oliveira, J. A., Martins, C. and Abreu, R. (2018), Cement Industry - A Routing Problem, Cement Update by Daily Cement (5), 10-15.

4. **Fonseca, J.**, Macedo, R., Alves, R., Veloso, H., Dias, L., Carvalho, M. S., Pereira, G., Figueiredo, M., Oliveira, J. A., Abreu, R. and Martins, C. (2018), Rules for Dispatch, BMHR 2018 supplement in World Cement (September).

5. Macedo, A. R., Alves, R., **Fonseca, J.**, Veloso, H., Dias, L., Figueiredo, M., Pereira, G., Carvalho, M. S., Abreu, R. and Martins, C. (n.d.), What can we learn from Industry 4.0: Opportunities in the logistics field on Cement Industry.

6. Veloso, H., Vieira, A., Alves, R., **Fonseca, J.**, Macedo, A., Pereira, G., Dias, L., Carvalho, S., Figueiredo, M. (2018), Simulation in cement industry, CemWeek (July).

Part I

State of the Art

Chapter 2

Supply Chain Management

2.1 Definition and Overview

From suppliers to final customers, from acquisition of raw material to delivery of finished goods, a product development includes numerous stages and many entities. A Supply Chain (SC) encompasses all entities that influence, directly or indirectly, the making of a product and the fulfilling of a customer's request, as well as the links and interchanges between them. A basic SC typically involves suppliers, manufacturers, distributors, retailers and customers. In all these stages, each entity is responsible for a process that adds value to the product the customer wants. To achieve such goal, the elements of the SC are connected, mainly through the flow of materials, information and cash (Mentzer et al., 2001). To easily understand the dynamics of a basic SC and how its participants interact, Figure 2.1 is presented.

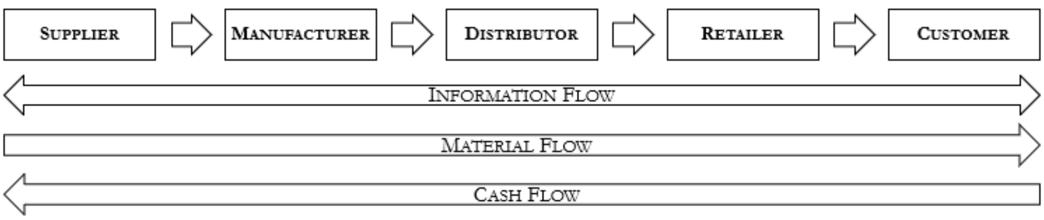

Figure 2.1: Basic SC and its main flows.

However, in the real world, SCs are not that simple. There are the suppliers' suppliers and the customers' customers. There are several players in each stage and, for example, a manufacturer may receive material from several suppliers and then supply several distributors. There are elements such as third party logistics, or others, which provide services to the ones inserted in the SC and that may also have influence in its efficiency.

When the number of participants increases, so does the SC and its complexity, often leading to the emergence of conflicting goals. In fact, every entity has its own goals and tries to improve the efficiency of its own operations. While manufacturers want high efficiency in production, to reduce costs, suppliers want stable volumes, and flexibility of delivery times. While distributors want to reduce transportation costs and inventory levels, retailers want to satisfy the customers by reducing lead times and increasing accurate deliveries. When each level of the SC optimizes its own operations, this is referred to as local optimization. But a small change at only one stage may damage the whole SC and affect the way customers are served. Therefore, there have to be some trade offs and a global optimization throughout the entire SC must be taken into account. Global optimization occurs when all entities work towards the same goal, seeking to balance efficiency with responsiveness to the final customer of the SC (Simchi-Levi et al., 1999).

Finding such balance is not an easy task and several issues may arise. Actually, today's marketplace is characterized by turbulence and uncertainty. Demand in almost every industrial sector seems to be more volatile than ever. Product and technology life cycles have shortened significantly and competitive product introductions make the demand difficult to predict (Christopher, 2016). This uncertainty in customer's demand can translate into increasingly large fluctuations in demand, for upstream manufacturers, occurring the so known bullwhip effect (Lee et al., 1997). Also, maintaining high levels of customer service calls for maintaining high levels of inventory, but operating efficiently calls for reducing inventory levels (Hugos, 2011).

The field that studies the best strategy and the set of approaches, to deal with these conflicting variables, is called Supply Chain Management (SCM). This concept is relatively new and according to Christopher (2016) it was firstly introduced in a white paper, by a consultancy firm, back in 1982. There, the authors alerted for a need of a new perspective and approach to fight the opposing objectives in a SC. This field has gained tremendous attention over the past decades, but despite its popularity, there is still disagreement about its definition. The Council of Supply Chain Management Professionals defines it as follows (*CSCMP Glossary*, 2013):

"Supply Chain Management encompasses the planning and management of all activities involved in sourcing and procurement, conversion, and all logistics management activities. Importantly, it also includes coordination and collaboration with channel partners, which can be suppliers, intermediaries, third party service providers, and customers. In essence, Supply Chain Management integrates supply and demand management within and across companies."

2.2 Logistics Management

The terms logistics and SCM are sometimes used interchangeably. Logistics is a term that has been around for a long time, emerging from its military roots, while SCM is a relatively new term (Rushton et al., 2014). Some say there is no distinction between the two terms and that SCM is the new logistics.

While these two fields do have some similarities, they are, in fact, different concepts with different meanings (Christopher, 2016). SCM is a wide concept that links together multiple processes to achieve competitive advantage. Logistics, on the other hand, is an activity within the SC and is just one small part of this larger concept, as suggested in Figure 2.2.

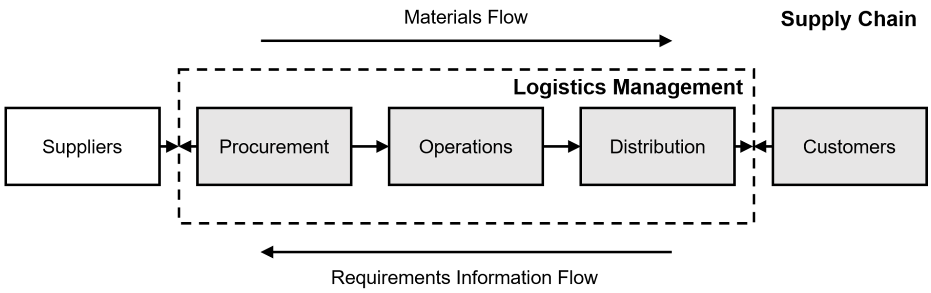

FIGURE 2.2: Logistics management process, as part of a SC.

Logistics refers to the movement, storage and flow of goods, services and information within the overall SC. It can be seen as the link between the delivery of the product to the marketplace and the management of raw materials given by the suppliers (Christopher, 2016). The main objective behind logistics is to make sure the customer receives the desired product, at the right time and place, with the right quality and price.

Every industry has its own characteristics, and for each company in that industry there can be major variations in strategy, size, range of product or market coverage. To resist these variations, logistics must be a diverse and dynamic function that has to be flexible and has to change according to the various constraints and demands imposed upon it and with respect to the environment in which it works (Rushton et al., 2014). Also, to achieve the desired levels of service and quality, at the lowest possible cost, a lot of planning and coordination, in all activities, are necessary. Logistics is essentially an integrative concept that seeks to develop a single plan to the SC, where no one acts independently. The Council of Supply Chain Management Professionals defines it as follows (*CSCMP Glossary*, 2013):

"Logistics Management is that part of Supply Chain Management that plans, implements, and controls the efficient, effective forward and reverse flow and storage of goods, services and related information between the point of origin and the point of consumption in order to meet customers' requirements."

According to Ballou (2004), logistics activities can be divided, by their importance to logistics management, into primary and support activities. Primary activities are key elements of any logistics system, since it is where companies invest the most and they are essential for an effective coordination. There are four primary activities, including:

Customer service refers to the quality with which the flow of goods and services is managed (Ballou, 2004). It is about getting the right product to the right customer at the right place, in the right condition and at the right time, at the lowest possible cost. This activity translates the experience and satisfaction of customers and plays an important role, since it represents the output of the logistics system (Kee-hung and Cheng, 2009).

Order processing is the information about demands, taken by customers. This is a core element of logistics activities, since it triggers product movement and service delivery. Its main goal is to shorten the order cycle time, delivering the product as fast as possible, which can give service differentiation (Niemelä, 2016).

Inventory management is concerned with the stock levels of a company. On the one hand, high inventory leads to high logistics costs. On the other hand, low inventory can harm the ability of a company to meet customers' demands and can lead to potential loss of customers. It is important to forecast fluctuations in demand and know how much inventory they should be keeping and when to replenish stock (Niemelä, 2016).

Transportation refers to the various methods for moving products between different entities in the SC. An effective management of this activity, concerns with selecting the best mode of transportation for a product, its routing and lead times, so everything is in the right place, at the right time and in the lower possible cost (Ballou, 2004).

Support activities differ from primary, since they are not necessarily a part of every logistics system. Despite the term *support*, they are also important and can help to reduce costs and improve service. These can include activities such as warehousing, purchasing, materials handling, packaging, production scheduling, information maintenance, or others (Kee-hung and Cheng, 2009).

2.3 Customer Service and Value

Most traditional SCs were designed to optimize internal operations and boost their efficiency. Typically, this would be achieved through mass production, the manufacturing in large batches and shipping in large quantities. Although this approach could benefit from the reduction of costs, it failed to understand the changing needs of customers, in an increasingly competitive marketplace.

The continuous increase in customers' expectations and the decrease in the difference of competing products made it harder to maintain a competitive edge, through only the product itself (Christopher, 2016). The power of the brand, to achieve customer retention, has declined and customers are more willing to accept substitutes. It is only when customer service is considered that a company can compete, in today's market. To be distinguished from the others and provide its customers a reason to remain loyal, a company must work to provide satisfaction to its customers. These reasons have contributed to a swing towards customer driven SCs.

Now, the majority of companies is centered on its customers. The designing of a SC starts with the identification of the customers' needs and the concept of customer value is the way of gaining competitive advantage. Customer value is the amount of benefits which customers get from purchasing products and services.

Levitt (1969) first introduced the idea that people do not buy products, they buy benefits and, as Figure 2.3 suggests, a product can no longer offer only its quality and features. Nowadays, it is not enough to have the right product, in the right quantities, delivered in the right place and time and in the right conditions. Customers want more. A product has to include also its service surround, where the basic product is augmented with value added services (Christopher, 2016).

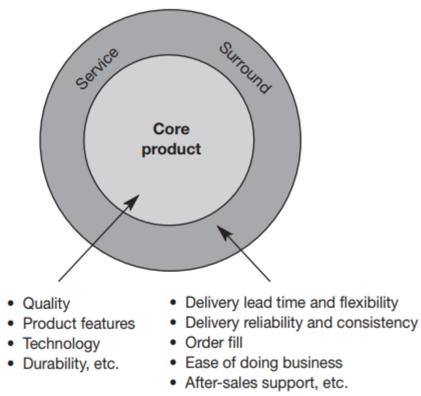

FIGURE 2.3: Product and its added value through customer service.

Value added services are additional benefits that consumers can receive, when they purchase a product or service. They are supplements that can bring competitive advantage to a company. Reduced lead times and flexibility in delivery can lead to the acquisition of new customers and retention of old ones. After sale support and maintenance, warranties and information access to the customers' personal data are also common services that can get the company closer to the customer (Simchi-Levi et al., 1999). These aspects often represent a small percentage of a product's cost, but have an enormous positive impact in the experience between buyer and seller (Rushton et al., 2014).

To assess the customer value a product can offer and clearly understand the impact that different elements have in this concept, Johansson et al. (1993) suggested the customer value ratio. It is defined as follows:

$$Customer\,Value = \frac{Quality \cdot Service}{Cost \cdot Time}$$

Where:

$Quality$: The functionality, performance and technical specification of the offer;

$Service$: The availability, support and commitment provided to the customer;

$Cost$: The customer's transaction costs including price and life cycle costs;

$Time$: The time taken to respond to customer's requirements.

Companies are struggling to satisfy their customers and fulfill their requirements. It is extremely difficult to keep a high level of service and, from the customer's perspective, there are only two possible levels. Either they get the perfect order, or they do not. To meet these almost impossible demands, Logistics and SCM play a crucial role. It is only through these fields that excellence can be achieved, in a consistent and cost effective way.

Chapter 3

Machine Scheduling

3.1 Background

In a Supply Chain (SC), time is essential and plays a vital role in a wide variety of situations. From production to distribution and delivery, scheduling is everywhere and has a huge impact in companies efficiency and in their relationship with customers.

Scheduling requires both sequencing and resource allocation decisions. Sequencing usually corresponds to a permutation of the jobs or the order in which they are processed on a machine. On the other hand, resource allocation refers to choosing which machine will process each job (Baker and Trietsch, 2009). Both jobs and machines may have different constraints or characteristics that will limit the productivity of the operations.

Machine Scheduling (MS) is responsible for covering the most important aspects of a certain environment and improve its operations. More specifically, it is the study of assigning jobs to machines or resources, in a way that one or more performance criteria are satisfied. The most common representation used in MS problems is as follows:

q_j - size or quantity of job j;

s_{ij} - speed of processing job j on machine i;

p_{ij} - processing time of job j on machine i;

r_j - release date of job j;

d_j - due date of job j;

w_j - weight or importance of job j;

3.2 Problem Representation

A great variety of MS problems in the literature demands the adoption of a formal and systematic manner of problem classification and representation (Varela et al., 2003). Graham et al. (1979) introduced a three-field classification ($\alpha|\beta|\gamma$), to categorize each type of problem, regarding its job, machine and scheduling characteristics. This notation allows identifying, unequivocally and precisely, the underlying characteristics of the problem that is intended to solve. Since its introduction, this notation has been extended and reformulated by several authors and many classifications have been added, as new problems appear. In this section, only some examples of types of MS problems will be analyzed. For further reading around this subject Varela (2007) is highly recommended.

Machine environment (α)

The first field $\alpha = \alpha 1 \alpha 2$ specifies the machine environment of the system in study. In this aspect, two important distinctions must be made, regarding the number of available machines and the number of stages a job must go by till it is finished.

The simplest case of MS regards the use of only one resource and was first studied by (Jackson, 1955) and (Smith, 1956). In single machine ($\alpha = 1$) models, there is no resource allocation decisions. One must only choose the order by which the jobs are processed in the only available machine. This is a special case of all other more complex machine environments.

When there is more than one machine available, but only one stage to go by, the problem facing is parallel MS. Firstly introduced by McNaughton (1959) this problem has received a lot of attention in the last decades due to its great importance and because the occurrence of resources in parallel is common in the real world. For this type of machine environment, $\alpha 1$ takes values P, Q and R, for identical, uniform or unrelated parallel machines, respectively. These values differ in the relationship established between the speed of each machine and the processed job. When speed is always the same, being independent of the type of job and the machine that processes it, $p_{ij} = p_j = q_j/s$ and the machines are called identical. When the speed only depends on the machine where the job is processed, $p_{ij} = q_j/s_i$ and the machines are called uniform. If speed is arbitrary and depends on both job and machine, $p_{ij} = q_j/s_{ij}$ then the machines are called unrelated. The unrelated parallel machine problem is a generalization of both identical and uniform machine problems. $\alpha 2$ indicates the number of machines considered.

If the problem requires that each job is executed on more than one machine, multistage scheduling is considered. In these types of problems, $\alpha 1$ takes values F, J and O, for flow shop, job shop or open shop environments. These values differ mostly in the required order of stages that a job must cross. In the case of flow shop, there are several machines, each one representing one stage of the process. Here, all jobs follow the same path of stages, that is, the order of machines by which the processing goes by is equal for all jobs. Job shop follows the same rules as flow shop, with the exception that, although predetermined, different jobs may have different paths. In these two types of environment, there are the special cases of flexible flow shop and flexible job shop, where in each stage there is a set of machines, instead of only one. In these cases, it is commonly considered that a job must be processed on only one machine per stage and $\alpha 2$ becomes the number of stages. In the open shop environment, there is also one machine per stage and there are no restrictions in the path taken by each job. Different jobs may have different paths and it is not required that a job must cross all stages. $\alpha 2$ refers to the number of existing machines.

Job and Machine characteristics (β)

The second field β consists of some job and machine characteristics that must be separated by commas and which better define the conditions of the problem and its constraints. Here, only some characteristics will be addressed, although there are a lot of values allowed in this field.

In MS, preemption (*prmp*) is the act of interrupting the processing of a job at any point in time and put a different job on the machine instead. The job interrupted may return later for further processing in the same, or other machine. On the contrary, in non-preemptive problems, once a job starts its process, it may not leave the machines, unless it is finished.

When there are constraints related with time windows and jobs are only available for processing, during a limited period of time, release dates (r_j) and due dates (d_j) can be added to each job. A job cannot start its processing before its release date and should end it before its due date. In contrast to release dates, due dates are not usually specified in this field, since the type of objective function can give sufficient indication whether or not there are due dates.

If a machine needs a period of time to be prepared to process a job, it is said that it requires setup times. In problems where multiple machines and different types of jobs are considered, these times are often different and may depend on the order of jobs processed. That way, a sequence dependent setup time (s_{jk}) exist if, after processing job j, a setup time s_{jk} is required before processing job k. When these times are also dependent on the machine, a subscript i is added to the variable. In a problem where job families ($fmls$) are considered, the setup time may be zero if job j and job k belong to the same family.

If a machine uses batch processing, it is capable of processing a set of jobs simultaneously. When the time for processing the set of jobs is equal to the sum of their processing times, the machine uses serial batch processing and $s - batch$ must be added. On the other hand, when the time for processing the set of jobs is equal to the maximum processing time among all jobs, the machine uses parallel batch processing and $p - batch$ must be added.

When in presence of precedence constraints ($prec$), it is required that one or more jobs have to be completed before another job is allowed to start its processing. There are also several special forms of precedence constraints and *intree*, *outtree* or *chains* must be added to this field if the job has at most one successor, at most one predecessor, or if the job has both, respectively.

There are also constraints in the ability of a machine to process a certain job. These constraints may be temporary or permanent. When not all machines are capable of processing certain job, it is said that there are machine eligibility constrains. Such constrains are permanent and the symbol M_j must be added in this field, representing that only a subset of all machines are able to process job j. When the machine is not available for a certain job, but only in a period of time (due to maintenance, shifts or other motives), machine availability restrictions are considered. In these cases, the symbol $brkdwn$ must be included, representing the period of machine breakdown.

Objective function (γ)

The last field γ, refers to the objective function, which is desired to minimize and that translates the performance of the system. This indicator is used as a comparator to select the best schedule, when more than one feasible schedule exists.

One important variable in MS is the jobs completion time C_j. Completion time refers to the instant when a job finishes its process and exits the system. The completion time of the last job (C_{max}) represents the length of the schedule, often called makespan. Minimizing the makespan guarantees that the set of jobs is processed as fast as possible and usually implies a good utilization of the machines. Other objective functions related with this variable are the minimization of work in process inventory levels. To achieve that, total weighted completion time ($\sum w_j C_j$) is the criteria to be analyzed. When jobs do not have different levels of importance, $w_j = 1$ and the function to be minimized is summed up in ($\sum C_j$).

Another way to assess a schedule is by the jobs lateness. Lateness, defined as $L_j = C_j - d_j$, represents how far the completion time of a job is to its due date. A positive lateness means that the job finished later than it was supposed to and a negative one means that the job was completed early. Tardy jobs are often related to late deliveries which may translate in the form of loss of goodwill between a company and its customers. In some cases, early jobs are also harmful to the scheduling, since they can represent an increase in the inventory levels. Here, the most common criteria are to minimize the total lateness ($\sum L_j$) or the maximum lateness (L_{max}) of all jobs. When the objective is to minimize the number of tardy jobs, often a unit penalty is given to each job that has a positive lateness. The best schedule, in this case, is the one which has the lowest sum of tardy jobs ($\sum U_j$). These last objective functions also have its weighted version, when there are jobs more important than others.

3.3 Complexity

When considering scheduling problems, an important issue is its complexity. Since scheduling is often related to manufacturing and services industries, the development of an algorithm capable of solving the problem, in an efficient way, is crucial. The efficiency of an algorithm may be measured by the maximum running time, i.e. the maximum number of steps it needs to solve a certain input or instance (Brucker and Knust, 2006).

The size of an instance takes a relevant role in a scheduling problem, since it refers to the length of the data necessary to specify that instance. Although there is data like the processing times or availability of jobs, only the number of jobs n is often referred to the size of the instance. This may seem an oversimplification but it is sufficiently accurate to make distinctions between the complexities of different problems (Pinedo, 2012).

It is obvious that, the larger the instance, the longer it will take to compute and solve the problem. But comparing two similar instances can become ambiguous, thus a more precise way to distinct running times is needed. That way for any input size n of the problem, there is a function $T(n)$ defined as an upper bound on the running time needed to solve that instance and a growth rate (or asymptotic order) $O(\cdot)$ that indicates how this time scales with the increase of the input size. $O(\cdot)$ is given by the term that has the largest impact on the maximum number of steps required, therefore ignoring all low terms and coefficients. For example, it is said that the asymptotic orders of $T_1(n) = 5n^3 + 10n^2 + 350$ and $T_2(n) = 2^n + 10n^{100}$ are $O(n^3)$ and $O(2^n)$, respectively, since these terms grow faster than the others, which become negligible.

When $T(n)$ is a polynomial function and refers to an algorithm capable of solving a problem, it is said that the problem is tractable and belongs to the class P of problems. However, there are functions, like the exponential function, which grow much faster than polynomial and that become unpractical for large size problems. So, when $T(n)$ is an exponential function it is said that the problem is intractable (Garey and Johnson, 1979). The complexity theory suggests that there is a large class of problems, namely, the NP-Hard problems, which may be intractable (Leung, 2004). The question whether they are, effectively, intractable or not is a major problem in computer science that remains unsolved.

It can be difficult to assess the precise complexity of a problem. For MS problems, Pinedo (2012) suggests a set of graphs that helps to determine the relative complexity between different scheduling problems. In the Figure 3.1, it is visible a graph that compares this complexity, regarding its machine environment.

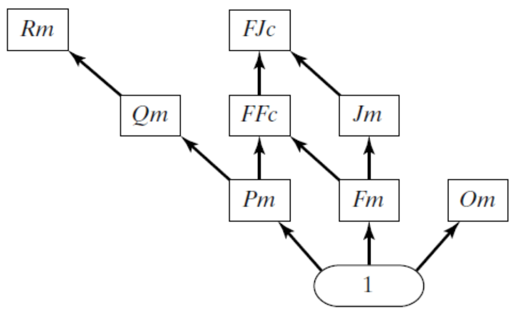

FIGURE 3.1: Complexity hierarchy on the machine environment of scheduling problems.

To better understand this graph, the term of problem reduction must be introduced. Often, an algorithm for one scheduling problem P can be applied to another scheduling problem Q as well. If this procedure can be applied correctly or if Q is a special case of P, than it is said that P reduces to Q ($P \propto Q$). The graph provides elementary reductions among problems. This concept is important because it allows to infer about the complexity of a problem, based on another. That way, if $P \rightarrow Q$ and Q is solvable in polynomial time, then P is also solved in polynomial time and if P is NP-Hard, then Q is also NP-Hard. Keeping this logic, a chain of reductions can be established and a complexity hierarchy of scheduling can be built.

Considering the single machine environment, where the goal is to minimize the makespan ($1| |C_{max}$), it is obvious that the makespan is equal to the sum of the processing times and is independent of the sequence. That way, the problem loses interest since it is easy to determine the optimal value. However, when machines are added to the problem, it gains an additional level of complexity due to the need of assigning machines. In fact, the overall complexity of multiple MS problems is inflated, often exponentially, as the number of machines m or jobs n increases (Cheng and Sin, 1990).

Garey and Johnson (1979) showed that scheduling jobs on two identical machines to minimize the makespan ($P2| |C_{max}$) is already NP-Hard. Following the complexity hierarchy previously introduced, since $Pm \rightarrow Qm \rightarrow Rm$, it is said that the problem of unrelated machines is a generalization of the identical ones, therefore belonging to the class of NP-Hard problems, too.

A similar hierarchy exists for the field of job and machine characteristics and can be seen in Figure 3.2. When release dates, sequence dependent setup times, precedence constraints and machine eligibility or availability constraints are included, in field β, the complexity of the problem is bigger than when these factors are ignored. Additionally, it appears that no relation in terms of problem reduction can be established in terms of including job preemption or not. However, French (1982) argued that scheduling with preemption gives the scheduler more flexibility and reduces the complexity of finding good schedules.

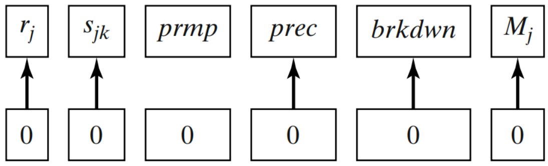

FIGURE 3.2: Complexity hierarchy on jobs and machines characteristics.

In the γ field, the relations among problems complexity can be established, as seen in Figure 3.3. Here, minimizing the total tardiness $\sum w_j T_j$ is said to be more complex to solve than when the objective function is to minimize the maximum lateness L_{max}, for example. Also, considering performance criteria like total completion time, total lateness or number of tardy jobs, it seems to be more complex to solve the problem when jobs have different weights.

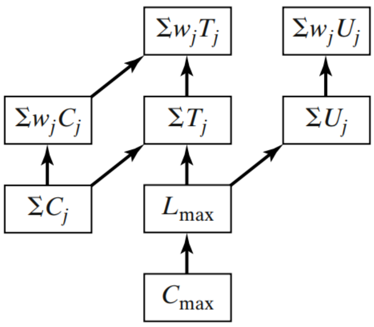

FIGURE 3.3: Complexity hierarchy on the performance criteria considered.

Chapter 4

Solving Techniques

4.1 Mathematical Programming

Mathematical Programming (MP) is a method for optimizing a function subject to constraints, upon the independent variables. In integer programming, the independent variables are constrained to be integral. The values 0 or 1 are often the only values allowed and are used to indicate the absence or presence of some property (French, 1982). The use of integer programming in solving scheduling problems can be traced back to 1959, when Wagner (1959) first formulated a flow shop problem as an all integer programming method.

MP is an exact method, meaning that it ensures an optimum solution. Although it can sound a very promising approach, due to the fact that most Machine Scheduling (MS) problems are NP-Hard, much computation is needed to solve each problem. This can lead to an exaggerated amount of time, before reaching a solution, being only applicable to small problems. As problems become larger, this method becomes inefficient (French, 1982). This statement remains valid in the current days despite the advances in the software and hardware industries.

But MP also has its advantages. It can handle different objective functions and incorporate other constraints in the model, which is often the case for real life scheduling problems. Also, a mathematical formulation is the first step to develop an effective heuristic and can be useful to understand the structure of the problem (Unlu and Mason, 2010). Rinnooy Kan (1976) even stated that a natural way to attack MS problems is to formulate them as MP models.

There are several approaches for MP, when dealing with MS problems. Since job completion time is a key metric in assessing the quality of a schedule, it is the kernel of all formulations. This metric can be used in several ways. Depending on the type of chosen decision variables, it can define the time or position of a job relatively to others. A summary of these variables is presented, based on the work of (Unlu and Mason, 2010).

Assignment and positional date variables, introduced by Wagner (1959). These variables define which job is scheduled next and at what time this job will start. Here, each machine has a fixed number of positions into which jobs can be assigned (Demir and Kürat Ileyen, 2013). Usually the number of positions is equal to the number of all jobs to be processed, allowing the extreme case, where all of them are processed in the same machine. For this approach, $x^i_{jl} = 1$ if job j is assigned to position l on machine i; $x^i_{jl} = 0$, otherwise. Additionally, the completion time of each job at position l, on machine i, will be determined by the y^i_l variable.

Another approach is to describe the precedence relationships among all jobs. To build a schedule, **linear ordering or sequencing variables** are used to denote the sequence of operations assigned to each machine. In this approach, proposed by Manne (1960), the processing order, in each machine, is based on three variables. x_{lj} will determine the precedence relationships, being equal to one, if job l precedes job j; otherwise $x_{lj} = 0$. To define which machine will process each job, y_{li} is used, being equal to one, if job l is positioned on machine i; $y_{li} = 0$, otherwise. Also, $z_{lj} = 1$ if job l and j are not scheduled on the same machine; $z_{lj} = 0$, otherwise.

When based on **time indexed variables**, the planning horizon is considered discrete and divided into time periods $1, 2, ..., T$ (Demir and Kürat Ileyen, 2013). Each job is assigned to these periods and, the job assigned to the last, T, will define the makespan of the schedule. The decision variable x^t_{ij} determines the time and machine where job j will be processed. If equal to one, it will start on machine i, at time t; $x^t_{ij} = 0$, otherwise. This type of formulation was firstly used by Bowman (1959).

Finally, **network variables** can also be used in mathematical formulations, for MS. The name of this approach is due to the similarity of MS with vehicle routing problems. Single MS relates with traveling salesman problem, where the nodes of the network are jobs that have to be visited and completed in the minimum amount of time (Picard and Queyranne, 1978). Also, parallel MS, resembles the capacitated vehicle routing problem, where jobs are again the nodes to be visited and the machines are the vehicles to be routed. To build the schedule, $x^i_{lj} = 1$ if job l is processed immediately before job j on machine i; $x^i_{lj} = 0$, otherwise.

Associated with these variables, a set of constraints is needed, for the mathematical model to work properly. These constraints ensure that all jobs are scheduled and that no jobs are processed simultaneously on the same machine. That is, at most, one job can be processed on each machine, at each period of time, or position. Also, when dealing with precedence relationships, for each machine, each job will have at most one predecessor and one successor. Beyond these constraints, others are added, specific of the problem, whether about the machine environment in question or the job characteristics. That way, release and due dates, weights of jobs or setup times can be added. The additional constraints, or objective functions to be minimized, can be added with few variations, depending on the chosen approach. Some examples of MP for MS problems are enumerated next.

Ángel-Bello et al. (2011) addressed the problem of availability constraints and sequence dependent setup costs, for the single machine environment. For minimizing the maximum completion time, the author presented a mathematical model. Also, two ways for reducing the execution time were presented. However, commercial solvers were only capable of solving the problem for small sized instances. In parallel MS, Lin and Hsieh (2014) focused on minimizing the total weighted tardiness of jobs, subject to release dates and setup times. The authors modified an existing mixed integer programming model and were capable of finding optimal solutions for 3 machines and 12 jobs. However, as the problem is NP-Hard, for larger instances they had to use alternative methods. In multiple machine problems, Zhu and Heady (2000) also tried to minimize the earliness and tardiness of jobs. In their work, the authors included sequence dependent setup times and different processing times depending on the chosen machine. Although their model only shows efficiency for small instances, the authors argue that it can be beneficial for developing and validating alternative methods for industrial scale problems. Fang et al. (2011) developed a multiple objective mixed integer programming formulation that consider both productivity and energy consumption. The authors described the problem in question as being a flow shop and allowed the operation speed of jobs to vary. This flexibility, although unusual in scheduling optimization, makes it possible to find a balance between energy spending and productivity measures. It is also referred that, with few modifications, the model can be adapted to other machine environments, such as job shop. Guo et al. (2006) addressed a job shop problem in the apparel industry. The main goal of this work was to minimize the total earliness and tardiness of jobs, taking advantage of the just in time philosophy. For more examples, Blazewicz et al. (1991) compiled a large number of mathematical formulations for MS problems. In this work, an extensive list of references, full with applications for these types of problems, is also presented.

4.2 Constructive and Improvement Heuristics

MS is, in general, a highly combinatorial problem. As said before, its complexity exponentially escalates, as instances grow, making it difficult to find an optimum solution, or even a good one. Also, real life problems often include a large number of variables and jobs, making this task even harder.

Exact methods can guarantee an optimum solution. However, towards a difficult or large problem, this solution can implicate the use of huge computational effort and take so long that, in many cases, it is inapplicable (Mart and Reinelt, 2011). Every solution obtained will be followed by a decision making and, in a competitive industry, managers need to take decisions as soon as possible, to achieve the desired results. It is not practicable to wait for long periods, sometimes hours, for a solution. Time is an essential aspect of every business and it cannot be wasted.

To overcome this problem, the use of heuristics has gained interest, in research and in applications for real life problems. Heuristics are practical approaches to problems that, although not perfect, are sufficiently good to achieve an immediate goal. These methods are not so dependent on the size of a problem as the exact methods are, since they often give a solution in reasonable amount of time. Heuristics are fast, easy to implement and although they do not guarantee an optimum solution, they can give good solutions, in less time (Pinedo, 2012). Heuristic algorithms can be divided into either constructive or improvement algorithms.

The constructive heuristics build solutions from scratch. They start without a schedule and gradually add one job at a time, being usually the fastest way to achieve feasible solutions. They often rely on a greedy approach of the problem, making always the choice that seems to be the best at the moment. Although this can ensure a fast and good solution, it will probably lead to a local optimum, ignoring all other better solutions. Constructive heuristics are mainly used if a reasonably good solution is acceptable, if the solution has to be found promptly or to provide initial solutions for improvement heuristics (Johannes Schneider, 2006).

Improvement heuristics differ from constructive heuristics, since they start with a complete schedule and try to obtain a better solution by manipulating the current one. Although they can find better solutions, the time required for computation is usually greater when compared to the constructive algorithms (Jungwattanakit et al., 2006). Metaheuristcs are often nature inspired improvement heuristics that are problem independent. Thus, unlike constructive heuristics, these procedures do not try to take advantage of any specificity of the problem.

An important class of improvement algorithms are the local search procedures. These procedures try to find a better solution, by searching in the neighborhood of the current schedule. Two schedules are neighbors if one can be obtained through a well defined modification of the other. At each iteration, a local search procedure evaluates a neighborhood solution, which is accepted or rejected based on a given criterion (Pinedo, 2012). The acceptance criterion is usually the aspect that distinguishes the local search procedures the most. The Hill Climbing (HC) method is the most basic local search procedure (Al-Betar, 2017). It starts with an initial solution and only accepts changes if the new solution is better than the previous. When no further improvements can be found, the algorithm stops. The Iterated Local Search (ILS) uses the HC approach, only accepting better solutions. It differs from the previous heuristic because when no improvement is found the algorithm starts again with a different initial solution. These methods only accept better solutions at each step, which leads easily to local optimums. Actually, the acceptance of worse solutions can be a mean of finding later a better solution. Simulated Annealing (SA) and Tabu Search (TS) are two well known local search metaheuristic procedures. They are able to accept worse solutions, but differ in their acceptance criterion. SA relies on a probabilistic process to accept or reject a solution and has its origin in the fields of material science and physics. TS uses a deterministic process for its acceptance criterion, based on a tabu list of movements, which the procedure is not allowed to make. These two local search methods only evaluate a schedule at a time. The Genetic Algorithm (GA), on the other hand, generates and evaluates a number of different schedules at each iteration. It is inspired by the process of natural selection and mutations, where only the best schedules will survive. There is also the method of Ant Colony Optimization (ACO), which combines local search procedures with constructive heuristics and other techniques (Pinedo, 2012). This method is inspired by the trail following behavior of ants and the use of their pheromones to attract the others to the best path or, in this case, the best schedule.

Apart from these, there are a lot of other methods, each one with its different characteristics. In this work, emphasis will be given to the Dispatching Rules (DRs) and to SA, as examples of constructive and improvement heuristics, respectively.

4.2.1 Dispatching Rules

In scheduling problems, DRs stands out as an example of constructive heuristic of great interest. A DR is a guideline that prioritizes all the jobs that are waiting for processing on a machine. Whenever a machine becomes idle, a DR inspects the waiting jobs and selects the job with the highest priority (Yildiz et al., 2011). These rules can be based on job attributes, considering its weight, processing time or release and due dates. Here, some of the most simple and common DRs are presented.

Shortest Processing Time (SPT) prioritizes jobs in the increasing order of their processing times. Dealing with shorter jobs first, allows to minimize their completion time and the number of jobs in the system. Also, this rule has strengths in maximizing machine utilization, avoiding the congestion of a machine, with a long duration job (Swamidass, 2000). Since SPT does not take into account information about due dates, it does not behave so well when lateness based performance criteria are used. If different jobs have different weights, the weighted SPT is proved to be optimal when minimizing total weighted completion time, in one machine environment (Pinedo, 2012).

Longest Processing Time (LPT) gives priority in the decreasing order of their processing times. This technique performs particularly well in reducing the makespan of parallel processors. Leaving the shorter jobs to the end of the schedule, allows to balance the loads in the several machines (Rajakumar et al., 2004). In the case of identical parallel machines, this rule has proved to achieve a makespan less than 4/3 of the optimal value (Williamson and Shmoys, 2011).

Earliest Release Date (ERD) sequences the jobs from their arrival time. This sequence attempts to minimize and equalize the waiting times of jobs (Pinedo, 2012). Although this rule does not consider due dates, failing to assess a job's urgency, it can be fair, when dealing with list of customers (Mukhopadhyay, 2015). This rule is the equivalent to **First In First Out (FIFO)** or the **First Come First Served (FCFS)** rules.

Earliest Due Date (EDD) organizes the jobs from their increasing due dates. This rule usually performs better than others, when considering tardiness based performance criteria. It is capable of minimizing maximum lateness (Baker and Trietsch, 2009) and reduce the number of tardy jobs (Moore, 1968). Although it is better at keeping promises to customers, this rule can be worse with respect to average flow time or total completion time (Ritzman and Krajewski, 2002).

Least Slack Time (LST) tries to measure the urgency of a job by its slack time. This time is calculated by the difference between its due date and its processing time. It indicates how much time there is left to process a job and finish it, without incurring in delays. The job with the lowest slack will be chosen, since it represents the highest priority and urgency. This rule is used to reduce mean tardiness of jobs (Barbosa et al., 2010).

DRs can also be constructed, based on machine information. That way, machine attributes are considered, such as its speed, the number of jobs waiting for processing or the total amount of processing waiting in the queue. Once the jobs are prioritized and ordered, they are assigned to a machine, according to the DR.

First Available Machine (FAM) rule schedules the first job of a sequence to the first machine that is ready to process it. This method can be used to improve the waiting times of the jobs, since they start processing as soon as possible.

When some machines are not capable of processing some jobs, it is said that there are machine eligibility constraints. When a job can only be processed in a subset of the available machines, it can be of interest to use the **Least Flexible Job (LFJ)** rule. A job is said to be less flexible than others, if it has less machines capable of processing it. Using this rule can be optimal to minimize makespan, in special cases of parallel identical machines (Pinedo and Reed, 2013).

If in the presence of identical machines and all jobs can be processed in any of the available machines, **Shortest Queue (SQ)** rule can also be of interest. This method attempts to minimize the idleness of machines, making sure that once finished a job, there is another to start processing. This rule usually reduces also the waiting times of jobs and balances the load of the several machines (Teixeira et al., 2014).

When using **Earliest Completion Time (ECT)** rule, both information of jobs and machines are used simultaneously. This is a rule for selecting the best machine, looking to reduce the total completion time of jobs. Here, each job will be allocated to the machine capable of finishing it earlier, considering all jobs in its queue (Framinan et al., 2014).

But the simplicity of these DRs, and others, can also be an obstacle to the construction of an efficient scheduling. These rules have limited use in practice, since most of them only focus on one job characteristic. In real problems, jobs characteristics and machine environments are usually more complex and using only a simple DR might be insufficient. That way, to achieve the desired results, often a combination of these procedures are utilized and can perform significantly better (Pinedo, 2012). Some examples of these procedures are here presented.

Baker and Trietsch (2009) shows how total completion time can be optimally minimized in a single machine environment, using SPT rule, when all jobs are available at the same time. But, when considering different release dates, the problem becomes NP-Hard and a simple DR becomes inefficient. To address the single machine problem, with unequal release dates, Potts (1980) used a combination of DRs. The ERD, FAM and LPT for tie breaks. This new sequence does not give a solution worse that 1.5 times the optimum solution. In the case of parallel machine problems, Weng et al. (2001) studied a problem that included setup times and different job weights. The authors tried to minimize total weighted completion time of jobs and proposed seven DRs based algorithms. All algorithms showed up to be extremely fast, even for large instances, and provided good solutions. Also in parallel machine environments, Na et al. (2006) dealt with a challenge in the wafer fabrication, at a semiconductor manufacturing facility. This problem was subject to job families and had the necessity of creating fixed sized batches, to reduce costs, over its processing and setup times. To minimize total weighted tardiness, the authors suggested heuristics based on existing DRs, such as weighted EDD and weighted SPT. On multiple stage machine problems, Chiang and Fu (2007) used DRs to solve the job shop scheduling problem. The authors showed that the existing rules usually focus on one objective and cannot provide good performance on multiple objectives at the same time. That way, to address due date based goals, the author suggested a procedure that combines several rules. Combining SPT, EDD and LPT rules, it was possible to outperform existing rules when the tardy rate and mean tardiness were simultaneously considered. In a flow shop environment, Johnson (1954) developed the very famous Johnson's rule. This method is capable of finding the optimum solution in minimizing the makespan of a two or three machines flow shop. Although this rule finds the best solution possible, it is limited on the number of machines. To address a flow shop scheduling with arbitrary number of machines, Campbell et al. (1970) expanded Johnson's rule and described a simple algorithm, which is capable of dealing with very large instances and achieve an optimum or near optimum solution. More examples of DRs can be found in (Panwalkar and Iskander, 1977). In the authors' work, more than one hundred DRs were presented. References with analyses for each rule and a classification scheme were also provided.

4.2.2 Simulated Annealing

SA is a local search metaheuristic procedure that has become a popular tool for tackling problems across a broad range of application areas (Dowsland and Thompson, 2012). SA was first introduced by Kirkpatrick et al. (1983) and Černý (1985) and has shown considerable success in optimization problems, both in academic research and in practical applications.

SA is inspired in the process of physical annealing with solids, which seeks their most regular possible crystal configuration (Henderson et al., 2003). The process of annealing begins with the heating of a material above its melting point, holding the temperature, and then cooling it again, very slowly. The final structural properties of the material depend on its residual energy, which in turn depends on the cooling rate. If cooled slowly, a low energy state can be found, which results in a perfect crystalline structure and a high quality material. On the contrary, if the rate of cooling is too fast, imperfections and defects on the material are going to appear. A defect free material corresponds then to the state of lowest energy, when the material reaches its ground state (Du and Swamy, 2016).

Metropolis et al. (1953) initially modeled the physical annealing, by simulating a material as a collection of atoms, at a given temperature. At each iteration, a random small displacement of the atoms takes place, resulting in a different rearrangement of the system and a respective variation of its energy, Δ. If $\Delta < 0$, the new organization of atoms is accepted. On the contrary, if $\Delta > 0$, it is only accepted with a probability of $e^{\frac{-\Delta}{k_B T}}$, where T is the temperature and k_B is the Boltzmann constant. The algorithm goes on for several iterations and, after a large number of them, the system would reach its thermal equilibrium, for temperature T (Eglese, 1990). This algorithm can be used to simulate the annealing process by repeatedly reducing the value of T, once the system has reached equilibrium at the current temperature, until the system freezes at its ground state.

The key feature of this algorithm is the possibility of accepting higher energy states through its acceptance function. These states have no interest as a final state, but can be a way of reaching a better one. The behavior of the acceptance function, due to variations of T and Δ is evidenced in Figure 4.1. For positive temperatures, this function has an asymptote for $P(T) = 1$, being closer to 0, for low temperatures, and closer to 1, for high ones. It is also possible to observe that, for the same value of temperature, changes in the system with higher Δ are harder to accept than lower ones. That way, at high T, the system easily accepts states of higher energy, therefore performing a gross search. But as these values decrease, so does the probability of accepting a worse state. For low temperatures, the function concentrates on the states with the lowest energy, performing a fine search in the neighborhood and finding a better minimum.

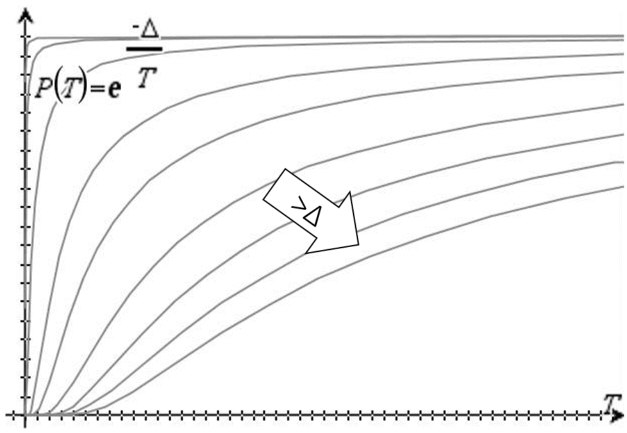

FIGURE 4.1: Acceptance function behavior with T and Δ.

It is possible to establish a bridge between combinatorial optimization and the physical annealing simulation of a material, like in Table 4.1. The system states and each one of its energy values become the set of feasible solutions and its cost. Instead of having a displacement of atoms, one can use a move and use the neighborhood to find other solutions. It is also the aim of combinatorial optimization to find the global minimum of a system, that is, the solution with the minimum cost. To achieve this goal, one can also use a control parameter, as temperature in the annealing process. Thus, for each value of this parameter, the cost of neighborhood solutions will be evaluated. When the algorithm reaches its stopping criteria, it is expected the returning of a good solution for the system (Van Laarhoven and Aarts, 1987).

TABLE 4.1: Analogy between the annealing process and a combinatorial optimization problem.

Thermodynamic Simulation	Combinatorial Optimization
System States	Feasible Solutions
Energy	Cost
Change of State	Neighborhood Move
Temperature	Control Parameter
Frozen State	Heuristic Solution

Four ingredients are needed to use SA for combinatorial optimization purposes (Kirkpatrick et al., 1983). First, a concise description of a configuration of the system, that is, an initial solution (S_0) for the problem. Then, a random generator of moves or rearrangements of the elements ($NEIGH(\cdot)$), so that other solutions and the neighborhood of the initial solution can be tested. To evaluate them, a quantitative objective function ($C(\cdot)$) is needed. This function must contain the trade offs that have to be made and must be capable of telling if one configuration is better, or not, than the previous configuration. Finally, it is necessary an annealing schedule that specifies how and when the system must be cooled ($UPDATE(T, I)$). A basic SA algorithm, addapted from Kim et al. (2002), is stated as follows:

Algorithm SA

```
initialize(S_0, T, I)
S ← S_0
repeat
   for i = 1 to I do
      S ← neigh(S_0)
      Δ ← C(S) − C(S_0)
      if ( Δ ≤ 0 or exp(−Δ/T) ≥ rand(0, 1) ) then
         S_0 ← S
      end if
   end for
   update (T, I)
until Stopping Criterion
```

SA has gained popularity in solving MS problems. Its flexibility and ease of implementation, allows one to adapt this algorithm to almost every kind of environment and use it for real life problems. Also, it is capable of escaping local optima and reaching good solutions, sometimes near optimal, in lower computing time. This feature makes it a viable and preferred alternative, when comparing to constructive heuristics. Some examples of its applications in MS are here presented.

On single stage MS, Potts and Van Wassenhove (1991) studied methods for solving the single machine total tardiness problem. Being too complex to solve optimally, for large instances, alternative approaches were considered. At the end, they conclude that SA is a viable heuristic alternative for this problem. Kim et al. (2002), on the other hand, applied SA to improve the production efficiency of compound semiconductors. The authors identified the dicing process of semiconductor wafers as being a major bottleneck operation that needed to be optimized. Referring to this issue as a typical unrelated parallel MS problem, they tried to minimize the total tardiness of jobs. On multiple stage MS, Raaymakers and Hoogeveen (2000) stands out by using SA on batch processes in the pharmaceutical industry. There, the authors formulated this scheduling issue as a job shop problem, with both overlapping operations and no-wait restrictions. The main goal was to minimize the makespan and near optimal solutions were obtained, with this metaheuristic. Flow shop scheduling problems are also addressed with this type of algorithms. Low (2005) purposed a heuristic based on SA, to minimize the total flow time of a multiple stage flow shop scheduling problem with unrelated parallel machines. In this work, the author observed that a good initial solution can be important and helpful for further improvement of the solution. Also, a good performance schedule was obtained in a reasonable running time, using the SA. More examples can be found in Koulamas et al. (1994) and in Suman and Kumar (2006), where the authors provided a survey of applications of SA, including to scheduling problems.

Part II

Case Study

Chapter 5

Cement Industry

5.1 Contextualization

Throughout history, cement has played a vital role in the development of civilization. The availability of basic raw materials and its endless applications make cement a very popular and widespread material. Now, it is the second most consumed substance in the world, after water, and it is used in several applications, such as houses, bridges and other infrastructures. Some would even say that cement is closely linked to economic development and cycle. In fact, cement sales are directly dependent on the growth of the construction sector, sector that itself follows the economic situation, prevailing at the time (Daugherty, 1973).

Cement belongs in a well established industry, dominated by few companies, which have huge geographical coverage, around the world. These companies are mature and have decades of experience. Being in business with cement since their foundation, they live to perfect its production. Also, Cement Industry (CI) is capital and energy intensive. Large manufacturing plants, with high level of production, are used to keep their companies efficient and to minimize costs based on economies of scale (Selim and Salem, 2011).

Due to the idea of established power and dominance, most cement companies stagnated their research and development and now their business are facing many threats. Although it is difficult for new players to enter in the CI and compete, this business remains attractive in emerging markets, where the quality requirements and purchase power is low. This happens mostly in underdeveloped countries, where small and medium firms try to compete with large companies, by offering lower prices (Agudelo, 2009). Also, because of the extreme heat required to produce it, cement manufacturing needs massive amounts of energy and is emissions intensive. In fact, it is estimated that 5-6%

of all carbon dioxide generated by human activities is derived from cement fabrication (Rodrigues and Joekes, 2011). Environmental aspects and its increasingly importance to society has led to the introduction of more and more restrict regulations. Although important, these regulations can limit their chances of growth. Finally, modern cement, as we know it, was developed in the 1800s and has not received many significant changes, since then. It is a commodity mainly selected by price, availability and quality. Thus, there is not much of a chance in gaining competitive advantage through product differentiation (Selim and Salem, 2011).

In a market ruled by rivalry and fierce competition and where customers' expectations and standards are rising, cement companies must now seek opportunities to compete and differentiate from others. The global market is changing from product oriented to customer oriented, and the CI also needs to make that change (Noche and Elhasia, 2013). To work around regulations and keep up with customers' demands, this industry needs to develop its operations to a new level, in terms of efficiency and in value offered to the customer. That being said, cement companies should focus their attention on progress of their Supply Chains (SCs) and there are a lot of processes in the making of cement (Agudelo, 2009). Every task and area have a chance of improvement and it must be sought.

From raw materials to the final customer, cement goes by several phases. Despite being considered a commodity, the manufacturing process of this material is very complex. A typical Cement Plant (CP) contains many distinct areas, each one with its specific function. In an industry so energy intensive, it is important to keep these areas strictly coordinated in order to achieve a high level of efficiency. Figure 5.1 illustrates the layout of a typical CP and its main areas of operation.

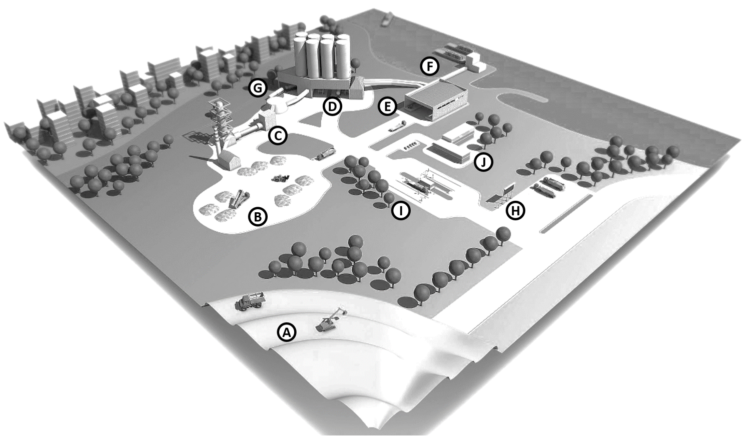

Figure 5.1: Important locations of a typical CP (*SLV Cement by Cachapuz Bilanciai Group*, n.d.).

As all tangible products, the process starts with the extraction of raw material. Cement most important raw material is limestone, which can be extracted from quarries (**A**), through drilling and heavy explosives. Near the surface, this material has high content of silica, iron and aluminum oxide. Deeper down, limestone is more pure, having less of those minerals and more of calcium carbonate, the most important substance. Varying in the proportions of both rocks, CPs can produce different types of cement (Thomas and Lea, 2018).

It is estimated that about 1.6 tons of raw materials are required to produce one ton of cement (Naik, 2005). That way, a huge amount of this material and others, in the form of big rocks, needs to be transported to the CP. This process is mainly done by dump trucks or wagons. To save in transportation costs and keep the production efficient, CPs are normally built close to the quarry (Afsar, 2012). After being transported to the CP, the materials are released in a storage area (**B**), usually open. To preserve their quality, different types of raw materials are kept on different piles, which are separated from each other.

The manufacturing process begins inside the CPs (**C**). Initially, the limestone rocks vary, in size, from few centimeters to meters, in diameter. To be more easily handled, they pass through two stages of crushing, primary and secondary, which will reduce their size up to 10 millimeters. Rocks with high concentration of calcium carbonate are crushed separately from those with lower concentration. Only then these two materials are blended together, in the correct proportions, to produce the type of cement required. Next, these materials go to a grinding machine, called roller mill, where it will mix and grind the minerals into a uniformly dry rock powder. Also, CPs use this stage to add silica or iron if the naturally minerals, in the crushed rocks, are not enough to produce high quality cement (*Cement Plant Layout*, 2018). The powder goes to a preheater, bonding the minerals together, so that they harden when hydrated with water. After that, the powder is sent to a rotary kiln. This machine is a huge cylindrical furnace, set at an angle, so that the powder moves from top to bottom. The kiln rotates very slowly and, close to the bottom, there is a flame that heats the powder at huge temperatures. This allows the powder to fuse together transforming it into small rocks called clinker. After produced, it is important to cool clinker very quickly, in order to achieve high quality cement. Being close to the final stage, clinker is then stored (*Understanding Cement*, n.d.). Some companies do not start the manufacturing process from raw materials. Instead, they buy clinker to other plants and start from there. Although this facilitates the production, it is more difficult to keep the business profitable. At the final stage, gypsum is added to the clinker and a final grinding is applied to the material. After grinding, all that remains is a fine, homogenized powder, called cement. The addition of gypsum is very important, to delay the setting time of cement. That way, it can be worked for about two hours, before hardening.

Cement is then stored in huge cylindrical structures, called cement storage silos (**D**). These silos are capable of storing thousands of tons of cement. Cement can be delivered to customers in bulk or in bags. To deliver in bulk, silos have hoses, which can connect to the customers trucks that go to the plant. To deliver in bags, cement flows to a warehouse (**E**), where it will be bagged and stored. After palletized, these bags are loaded in trucks, which also go to the facilities. If the plant is near to a quay or a railway, bulk and bagged cement can also be delivered to ships (**F**) or trains (**G**). These methods of transportation have gained interest in the CI due to the large quantities needed and its weight. Actually, according to (*Cembureau*, 2017), it is not profitable to move cement by truck, over distances longer than 300 kilometers.

Besides these areas, CPs also have at least one parking zone (**H**). Here, customers wait for their turn after being correctly identified. Before entering and leaving the premises, customers' trucks are normally weighed, using underground scales (**I**). This process is used to assure that the customer is loading or unloading the quantities previously agreed.

To control all these operations there is a central room (**J**). From there, the equipment can be turned on or off and its parameters are regulated. Also, information about quantities of products, energy spent, flaws in processes, and other, are available in this area, assuring the correct functioning of the plant.

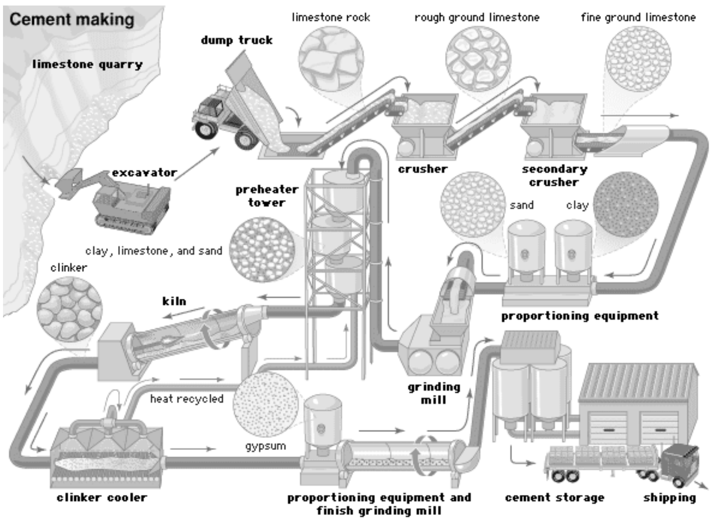

Figure 5.2: Cement manufacturing process, from raw material to end customer (Thomas and Lea, 2018).

5.2 Problem Description and Assumptions

The problem here presented deals with the supply of bulk cement to customers and the need of creating a schedule for that matter. Bulk cement is stored in huge structures, called silos, which can hold thousands of tons of material. Cement then flows to the Loading Points (LPs), through hoses, at a variable speed. Each silo may feed one or more LPs and each LP may receive material from different silos, thus establishing different connections. These connections form a combination between silos and LPs, which must be respected in order to get the correct materials. An example of such combination is visible in Figure 5.3.

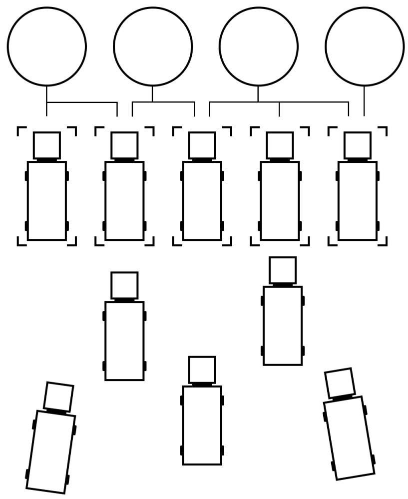

FIGURE 5.3: Example of connections between silos and its respective LPs.

To see their demand being fulfilled, customers must go to the CP with their bulk trucks, wait for their turn in the park, and enter the premises. Then, they will go to the silos area, where they must choose one LP that is available at the time and is capable of serving the material they demand. The customer himself connects and disconnects the hose to the truck, initializing and finishing the loading process, respectively. The type of material and quantities, taken by the customer, must be previously agreed upon and are controlled by scales at the entrance and exit of the CP.

In the present situation, there is a lack of control by the cement companies and they often see their park full with dozens of customers waiting to be served. Since there is no scheduling, customers do not have an estimated time of delivery and must stay in the park, to not lose their turn. The waiting periods are usually of hours, which is an unacceptable duration, considering that the load of a truck can take only few minutes. Also, the capacities of each plant are not very often respected. They allow more vehicles inside the premises than it can handle and a specific LP is not assigned to each customer.

This leads to a traffic jam at the silos area and a disruption of the process, which can also damage the rest of the operations inside the CP.

In order to improve these conditions, a different approach is needed. The waiting periods must be reduced and the customers' availability must be taken into account. Also, types of material and processing times of each customer as well as the resources and capacities of each CP must be included in the management of this operation. The main purpose is the creation of a schedule, favorable to the company and its respective customers. It is intended that the customers have an estimated time of delivery and that are forwarded to their respective LPs, according to the material they want. This schedule must reduce the waiting and operation times, respect the customers' availability and consider the CP capacities and characteristics. That way, the goal is not only to improve the operations inside the CP, but also to contribute to a higher service level. For this matter, some assumptions must be made before solving this issue, since real problems are volatile and full with aspects difficult to control. They are as follows:

- It will be considered an offline scheduling, where there are a number of customers to be processed. That way, all the problem data, such as quantities, materials ordered, release dates, or others, are known in advance and no customer can be added nor removed.

- Each LP is capable of dealing with only one customer at a time. There can be several LPs, capable of loading one or more materials and a material is available in one or more LPs.

- The combination between silos and LPs must be respected, when forwarding a customer. If a customer wants a product, he must go to a LP capable of supplying that product. Although it is possible to change these connections, this will not be considered, since it is not practical and the company would incur in high costs. Also, each silo has only one product and each LP may only serve one of the materials available, for each order. This relates with the reality of the process, since most silos and bulk trucks have only one compartment.

- Once inside the premises and at the silos area, only one customer per LP will be allowed. Therefore, at each moment, the number of customers will be limited to the number of LPs.

- Connecting and disconnecting the hose are necessary tasks, done by the customer. Since these times are of short duration in comparison with loading times and their fluctuations are minimal and hard to measure, they will be neglected.

- The processing time of each order depends on the speed at which the material flows through the hoses, to load the bulk truck. This speed varies over time, usually accelerating at the beginning and slowing down at the end of the load. Given the difficulty of measuring the speed at each instant, an average value will be considered. However, this value can be different, depending on the LP or material in question.

- Each loading process must be continuous. That way, it is assumed that, once a customer starts loading his truck, this operation must never be interrupted, nor canceled. Interrupting this process would damage the customers' satisfaction and would cost time to both customers and the company itself, since it is impractical and involves additional setup times.

- Machines breakdowns and rupture of inventory will not be considered.

5.3 Approaches

Assigning a set of customers, arriving in their trucks, to a set of LPs, is similar to a Machine Scheduling (MS) problem. Here, the arriving trucks are the jobs to be scheduled and the LPs are the machines, capable of processing them. Following this approach, it is possible to establish a bridge between the stated problem and the aspects of a typical machine environment.

In the described problem, only the process of loading trucks is considered. Comparing this with a machine environment, it is possible to assume a single stage with several machines, therefore a parallel MS problem. The average loading speed of cement into the trucks is dependent, not only on the type of required material by the customer, but also on the chosen LP. This means that the machines are unrelated and that the processing times of jobs depend on the machines and on the job itself. Following the combination between silos and LPs, it might happen that a material can be served in more than one LP. However, it can also happen that a LP is incapable of serving a certain material. Since a job can only be processed on a specific subset of the available machines it is said that machine eligibility constraints exist. Also, customers have availability that must be respected. These will be represented by release dates, determining the time by which customers are ready to start its processing and loading of their trucks.

All problems have to be solved according to a specific goal and this one is not different. When dealing with customers, a widely accepted measure of the quality of service provided is the total flow time of the system. It allows to determine the overall time the customers are spending in the system, both waiting for a service and being served. The flow time of a job can be calculated by the difference between the completion time of that job and its release date. Minimizing this values would mean a reduction in the waiting times and/or the processing times for the customers. This would bring advantages not only for the customers, but also to the company.

To summarize, the challenge in study is defined as an unrelated parallel MS problem, subject to unequal release dates and machine eligibility constraints and with the objective of minimizing the total flow time of the system. Following the three field notation, this problem can be represented as $(R_m|r_j, M_j|\sum F_j)$ and it is considered to be NP-Hard. To the best of our knowledge, this problem has not yet been addressed by the literature.

To address this problem, three different methods will be developed from scratch – one exact method and two heuristics. The first one is a Mathematical Programming (MP) formulation of the problem. Being a natural way to attack MS problems, this method is expected to give optimum solutions and to be useful to understand the structure of the problem. However, it is known that much computation is needed to solve problems with exact methods and that they are only applicable to small instances. That way, the utilization of this method will be restricted to small size problems and it will serve as comparison with the heuristics. The goal is to assess the quality of the other two methods, seeing how far their solutions are from the optimum ones. The second method is a heuristic, more specifically, a Dispatching Rule (DR). This method is presented as being highly flexible and easy to implement. It is mainly used in real life applications, in large size instances and when a good solution for a problem has to be found promptly. Although possibly giving a worse solution than the optimum, a good DR is usually the fastest way to achieve a good solution. The third method is an improvement heuristic, the Simulated Annealing (SA). Seeking to reach the optimum solution, this method will start with the solution given by the previous heuristic and will try to obtain a better one through its manipulation. Although SA will likely be slower than a simple DR, it is expected to be faster than an exact method, specially when dealing with large size instances. This method is expected to also give better solutions than the other heuristic, closer or equal to the optimum one. In the development and definition of the methods, the following notation will be used.

Sets:

M - set of machines, indexed $i = 1, ..., m$

J - set of jobs, indexed $j = 1, ..., n$

P - set of positions, indexed $k = 1, ..., n$

Parameters:

m_j - material ordered by job j

q_j - quantity ordered by job j

r_j - release date of job j

s_{ij} - speed of machine i for job j

Jobs Variables:

C_j - completion time of job j

F_j - flow time of job j

5.3.1 Mathematical Programming

To build a solution through a mathematical formulation of the problem, assignment and positional date variables were considered. This way, x_{ijk} is equal to 1, if job j is assigned to position k, on machine i, and equal to 0 otherwise. pos_{ik} is equal to 1, if position k, on machine i, is used, and equal to 0 otherwise. Regarding the dates, three more variables were created. pos_t_{ik} is the processing time of the job assigned to the position k, on machine i. pos_s_{ik} represents the start date of the job assigned to the position k, on machine i. pos_e_{ik} represents the ending date of the job assigned to the position k, on machine i. This method was developed and implemented in AMPL programming language and the complete MP model is shown below.

$$\min \sum_{j \in J} F_j \tag{5.1}$$

$$\sum_{i \in M} \sum_{k \in P} x_{ijk} = 1 \quad j \in J \tag{5.2}$$

$$\sum_{j \in J} x_{ijk} \leq pos_{ik} \quad i \in M, k \in P \tag{5.3}$$

$$\sum_{j \in J} \sum_{i \in M} x_{ijk} \geq \sum_{i \in M} pos_{ik} \quad k \in P \tag{5.4}$$

$$\sum_{i \in M} \sum_{k=2}^{n} pos_{ik} \leq pos_{ik-1} \tag{5.5}$$

$$\frac{q_j}{s_{ij}/60} \cdot x_{ijk} \leq pos\_t_{ik} \quad i \in M, j \in J, k \in P \tag{5.6}$$

$$pos\_s_{ik} \geq r_j \cdot x_{ijk} \quad i \in M, j \in J, k \in P \tag{5.7}$$

$$pos\_e_{ik} = pos\_s_{ik} + pos\_t_{ik} \quad i \in M, k \in P \tag{5.8}$$

$$pos\_s_{ik} \geq pos\_e_{ik-1} \quad i \in M, k \in \{2, \ldots, n\} \tag{5.9}$$

$$C_j \geq pos\_e_{ik} - B\,(1 - x_{ijk}) \quad i \in M, j \in J, k \in P \tag{5.10}$$

$$F_j \geq C_j - r_j \quad j \in J \tag{5.11}$$

Equation 5.1 denotes the objective to be minimized, the total flow time. Constraint 5.2 ensures that all jobs are assigned to exactly one position, on only one machine. Constraint 5.3 guarantees that each position, on every machine, contains at most one job and activates the utilization of position k on machine i. Constraint 5.4 assures that only the positions where the jobs are assigned are activated. Without this restriction, pos_{ik} would be 1, for every i and k. Constraint 5.5 is used to keep the the used positions in order. That way, if position k on machine i is used, position $k-1$ of the same machine, also has to be used. In Equation 5.6, it is calculated the processing time of the job assigned to the position k, on machine i. Here, the speed is divided by 60, to give the processing time in minutes. Constraint 5.7 assures that all jobs start only when they are ready. That way, starting a job before its release date is not allowed. In Equation 5.8, the completion time of the job assigned to position k, on machine i, is calculated. These values are given by the sum of the start date and the respective processing time, previously calculated. Constraint 5.9 forces each machine to process only one job at a time. That way, the job assigned to position k, on machine i, can only start after the end of the previous job, of that same machine. In Equation 5.10, the completion time of job j is calculated. Here B is a large arbitrary number, which guarantees that it will only be calculated for the positions and machines utilized. In

Equation 5.11, the flow time of each job is determined. Flow time is equal to the difference between the completion time of the job and its release date.

5.3.2 Dispatching Rule

This algorithm tries to assemble two DRs, in order to get a schedule that is both favorable to the customers and to the company. This method was developed and implemented in Java programming language and is here presented.

1. Let set U be the set of unscheduled jobs of an instance. Order the jobs in set U in non-decreasing order of release dates. Ties are broken arbitrarily.
2. Let job j be the first job in set U. Let C_{ij} be the completion time of job j, if scheduled on machine i. $C_{ij} = \max(r_j, l_i) + p_{ij}$, where r_j is the release date of job j, l_i is the completion time of the last job in the sequence of jobs, on machine i, and p_{ij} is the processing time of job j, on machine i. Let g be the machine that minimizes $C_{ij}, i \in M$. Ties are broken arbitrarily.
3. Add job j to the end of the sequence of jobs, on machine g. Let $C_j = C_{gj}$ and $l_g = C_j$. Remove job j from set U.
4. If U is an empty set, stop; otherwise go to step 2.

The algorithm will first serve the jobs in order of their Earliest Release Date (ERD) (step 1). This is a fair sequence, when dealing with a list of customers, since no one likes to be passed in a queue. By handling first the customers that first arrive to the system, allows also to minimize the waiting times of jobs. Then, after selecting the job to be scheduled, it is necessary to decide the machine where it will be allocated. Through the Earliest Completion Time (ECT) rule, the machine that gives the job the minimum completion time possible will be selected (step 2). Although this machine may not be the fastest or the first available one, for the job in question, it will allow for the job to spend less time in the system. This rules will be applied until all jobs are scheduled. This algorithm allows to reduce the waiting times and the time spent in the system, by all jobs, and, consequently, the total flow time.

5.3.3 Simulated Annealing

The SA method was designed to improve the solution given by the previous heuristic. Knowing that DRs, although fast can fall short in the expectations, the SA algorithm was created, seeking to obtain better solutions, in reasonable amount of time. This algorithm was also developed and implemented in Java programming language.

SA is a metaheuristic that has proved to be very effective for solving complicated combinatorial problems. However, to meet these expectations it is critical to adjust the initial values of parameters. That way, the algorithm begins with an initial solution (S_0), an initial temperature (T_0), an iteration number (I_{max}) and a time limit (t). S_0 will be the solution given by the previous heuristic. T_0 should be high enough so that the algorithm has the opportunity to pass through much of the neighborhood. However, high initial temperatures could consume too much time in the beginning of the algorithm. That way, an efficient value of temperature has to be considered. After an extensive number of tests, $n/10$ was the value considered. I_{max} represents the number of repetitions that must be made, before updating the temperature. This value is often proportional to the number of possible neighborhood solutions. Since this value is too large, $n \cdot m$ was the value considered. t will determine the end of the algorithm. After testing, $t = n$ seconds showed to be high enough to find good solutions, but small enough to keep this method fast, as a solution for real life problems should be. After determined the initial parameters, the algorithm can be initialized. The full algorithm is presented below.

Algorithm SA

initialize(S_0, T_0, I_{max}, t)
$S_{best} \leftarrow S_0$; $T \leftarrow T_0$; $end \leftarrow Current\ Time + t$

repeat

  for $I \leftarrow 0$ **to** I_{max} **do**
    $S_{new} \leftarrow move(S_0)$
    $\Delta \leftarrow F(S_{new}) - F(S_0)$
    if ($\Delta \leq 0$ **or** $\exp(-\Delta/T) \geq rand(0,1)$) **then**
      $S_0 \leftarrow S_{new}$
      if $(F(S_0) < F(S_{best}))$ **then**
        $S_{best} \leftarrow S_0$
      end if
    end if
  end for

  $T \leftarrow 0.99 \cdot T$

  if $(T \leq 1e-6)$ **then**
    $T \leftarrow T_0$
  end if

until $Current\ Time < end$

return S_{best}

Then, a new neighborhood solution $S_{new} = move(S_0)$ is generated, based on one simple move, which will change the position and/or machine of a job. The move is chosen randomly from four possible moves that will be explained later. This neighborhood solution becomes a new solution if an objective function $F(\cdot)$ is improved. In this case, the objective function is the total flow time of the system. To assess the variation in the objective function, the difference between the neighborhood solution and the previous one is computed, using $\Delta \leftarrow F(S_{new}) - F(S_0)$. If the solution improves the final schedule ($\Delta < 0$), the neighborhood solution becomes the new solution. The neighborhood solution can also be accepted, even if it is worse, with a probability based on $\exp(-\Delta/T)$. The possible acceptance of worse solutions allows to escape from a local optimum and keep the search for the global optimum solution.

After I_{max} iterations, the temperature is updated, using the cooling ratio. It was chosen a geometric ratio ($T_k = \alpha T_{k-1},\ k = 0, 1, ...$) which is widely accepted for practical applications. The value of α was chosen to be 0.99, allowing the temperature to cool very slowly, the algorithm to spend more time in low temperatures and, consequently, obtain better solutions. If the temperature reaches zero before the time limit, the system is re-heated to the the value of the initial temperature. This allows to take advantage of the remaining time and keep the search for a better solution, since it is possible that the algorithm converged in a local optimum.

The original SA algorithm gives the last solution found. However, in more recent formulations, often the best solution is returned. Thus, once reached the time limit, it is returned the best solution, found during all the execution of the algorithm.

Generation of Neighborhood Solutions

To find different solutions in the neighborhood of a schedule, it is necessary to develop a set of moves. These moves have to be well defined and will change the position and/or machine of one or more jobs. When the machine of a job is changed, it may happen that the new machine is not capable of processing that job due to the machine eligibility constraints. To prevent the formation of unfeasible schedules, the algorithm instantly rejects those moves. On the other hand, when the move forms a new and feasible schedule, the change of cost is evaluated. Four moves were developed and will be presented next.

1. **Switch** finds a new solution in the neighborhood by exchanging the order of two jobs in one machine. First, a random machine is chosen. Then, two random positions are chosen, within all scheduled positions of that machine. Finally, the

exchange is done. Figure 5.4 illustrates an example of a switch, if the selected machine was M1 and the positions were pos_{15} and pos_{17}.

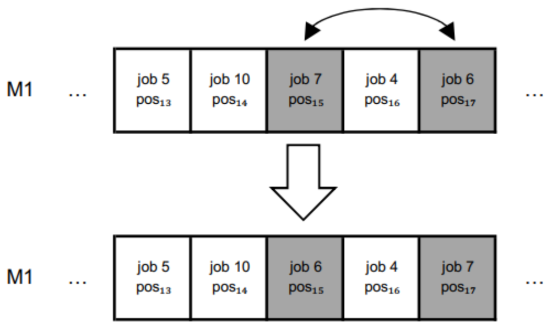

FIGURE 5.4: Example of a neighborhood solution by a switch.

2. **Shift** exchanges the position of only one job, inside a machine. First, a random machine is chosen. Then, two random positions are generated. The first one determines the job that will be moved. The second, determines the final position of that job. At last, the exchange is done. Figure 5.5 shows an example of a shift, if the selected machine was M1 and the positions were pos_{17} and pos_{14}.

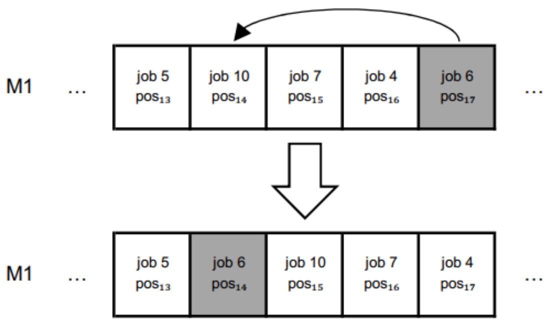

FIGURE 5.5: Example of a neighborhood solution by a shift.

3. The **Swap** move interchanges two jobs between two different machines. First, two random machines are chosen. Then, two random positions are generated, one per each machine, determining the jobs that will swap. Finally, the interchange is done. Figure 5.6 illustrates an example of a swap, if the random machines were M1 and M3 and if the chosen positions were pos_{15} and pos_{32}.

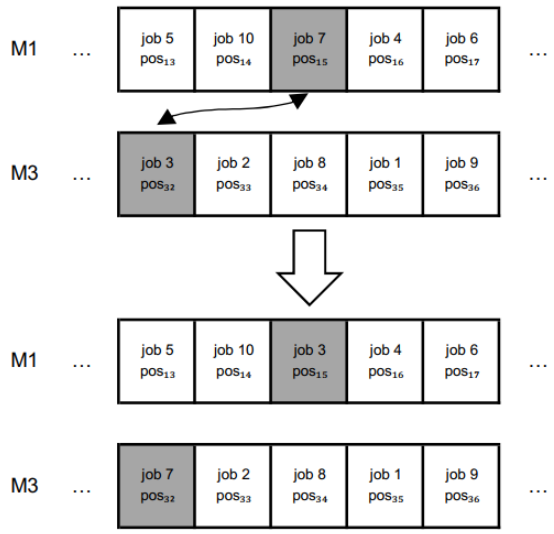

FIGURE 5.6: Example of a neighborhood solution by a swap.

4. **Task Move** finds a new solution in the neighborhood by moving a job from one machine, to another. First, two random machines are chosen. A first one, which contains the job that will be moved and a second one, its destination. Then, two random positions are also chosen. The first one, belonging to the first machine, determines the job that will participate in the move. The second one determines the position of that job, in the second machine. At last, the move is done. Figure 5.7 shows an example of a task move, if the random machines were M1 and M3 and if the chosen positions were pos_{13} and pos_{35}.

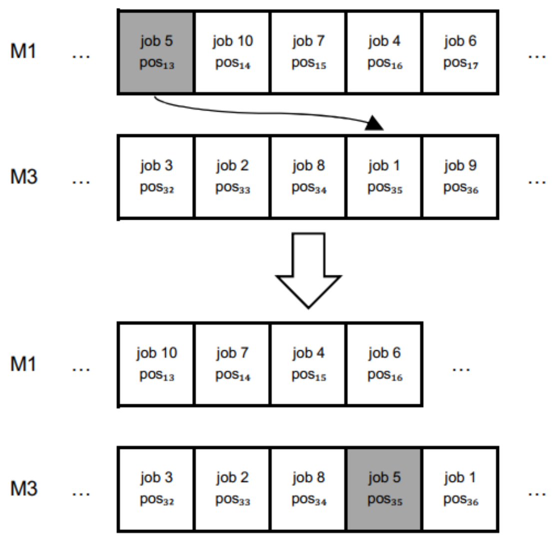

FIGURE 5.7: Example of a neighborhood solution by a task move.

Chapter 6

Tests and Results

After developed and implemented, the three methods were subjected to an extensive series of computational tests. The Mathematical Programming (MP) method was tested in the NEOS Server. It is a free internet based service for solving numerical optimization problems. This service allows to send the program to high performance machines, capable of dealing with problems that require high computational efforts. These machines contain several solvers, being the IBM ILOG CPLEX Optimizer the chosen one. The heuristics experiments were run on an Intel Core i7-4700HQ with 2.40GHz and 8Gb of RAM memory.

At this stage, there was the interest of testing the methods in public instances, to compare their results with the best known values. However, no public instances were found that met all the problem particularities. On the one hand, the Beasley (2018) and *Optsicom Project* (n.d.) repositories had no instance regarding parallel Machine Scheduling (MS) problems. On the other hand, the *SOA* (n.d.) repository had instances regarding the parallel MS problems, but with different characteristics and objective functions. To work around this issue, several instances were built from scratch.

First, three different Cement Plants (CPs) were chosen, CPI, CPII and CPIII, whose real names shall remain anonymous, for confidentiality reasons. Through several meetings it was possible to understand better the characteristics of each plant. These CPs, of various dimensions, differ in their number of silos and materials, in the number of Loading Points (LPs), in the combination between silos and LPs and in the hoses' speeds. From the analysis of raw data, it was possible to suit statistical distributions into the needed parameters to better describe the situation in question. The provided data lacked in information, but efforts were made to create instances that relate to a real life situation. Running tests on these instances is the first step to assess the developed solutions, before implementing them on the CPs.

To build each instance, it was necessary to generate random data about the customers that arrive to the CPs. This information had to consider the number of customers, the material each one seeks, the quantity of material previously agreed and their release dates, representing their availability. The following data generation scheme was considered.

For each one of the CPs, different dimensions of the instances were considered, with $n \in \{10, n_{mp}, 50, 100, 200\}$. Here, n_{mp} denotes the maximum number of jobs, where the MP method was capable of returning the optimum solution, in the eight hours available in the NEOS Server. The MP method was tested for 10 and n_{mp} jobs, whereas the heuristics were tested for all number of jobs mentioned before.

To generate the material each customer wants, a discrete distribution was used. This distribution is capable of defining the probabilities for distinct potential outcomes. Since different materials can have different demands, it was chosen a demand proportional to the number of hoses capable of serving each material. That way, if material A is only served in 1 of 10 possible hoses, it is said that its demand is 10%. Following this premise, in a 100 job instance, 10 customers, or at least approximately, should want material A.

The quantity of material most customers demand is about 30 tons. However, there are some fluctuations in these values, having customers demanding more or less quantity per order. That way, a normal distribution $q_j \sim N\left(30, 5^2\right)$ was used, to describe the quantities values.

To generate the release dates, a widely used distribution was chosen, the uniform distribution $r_j \sim U\left[a, b\right]$. When this is used, a should be set to zero. When not set to zero, a translation of the time axis occurs and this adds no new information to the experiment. Also, b should depend on the number of jobs or total processing time. If not, the comparability of results for different size problems becomes questionable, since as n grows, the release dates become closer to each other. This means that jobs with larger release dates do not have active release date restrictions. Following these assumptions, b was chosen to be an estimation of the maximum completion time of the jobs. The distribution used was $r_j \sim U\left[0, \bar{q}/\bar{s} \cdot n/m \cdot R\right]$. Here, $\bar{q}/\bar{s}$ is the quotient between the average quantity demanded and the average speed of the hoses, representing the expected average of the processing times. Also, n/m represents the expected number of jobs in each machine, if sorted equally for all machines. The multiplication of these two terms will give the expected makespan. To compare the effects of release dates tightness, a factor $R \in \{0.50, 0.75, 1.00, 1.25, 1.50\}$ was added. This factor will allow to test high and low periods of demands that may occur in a day of work at the CPs.

Thus, a total of 25 different instances were generated, for each one of the CPs, considering the 5 possible values for n and the 5 possible values for the factor R. In the case of Simulated Annealing (SA) method, it was tested 10 times, for the same instance, for all instances[1]. In the next sections, the main results, obtained from the computational tests, and their discussion are presented. A deeper analysis of the results will be made only to the CPIII, since the conclusions obtained in this CP are very similar to those obtained in the others. Nevertheless, the additional results of the other CPs are presented in the Appendix A and in the Appendix B.

6.1 Cement Plant I

CPI has four different materials and eight LPs. Material A is loaded by only one hose, located in LP 1 and has a speed of 150 tons per hour. Material B can be loaded by two hoses, in LPs 2 and 3, at speeds of 125 and 100 tons per hour, respectively. The silo holding material C, being the most wanted, can feed four different LPs, using four different hoses. These hoses connect with the LPs 4, 5, 6 and 7 at speeds of 90, 200, 140 and 150 tons of material per hour, respectively. Finally, the material D has three hoses capable of loading the trucks. They are located in LPs 6, 7 and 8, having speeds of 150, 160 and 250 tons per hour. The full combination between the silos and the LPs can be seen in Figure 6.1.

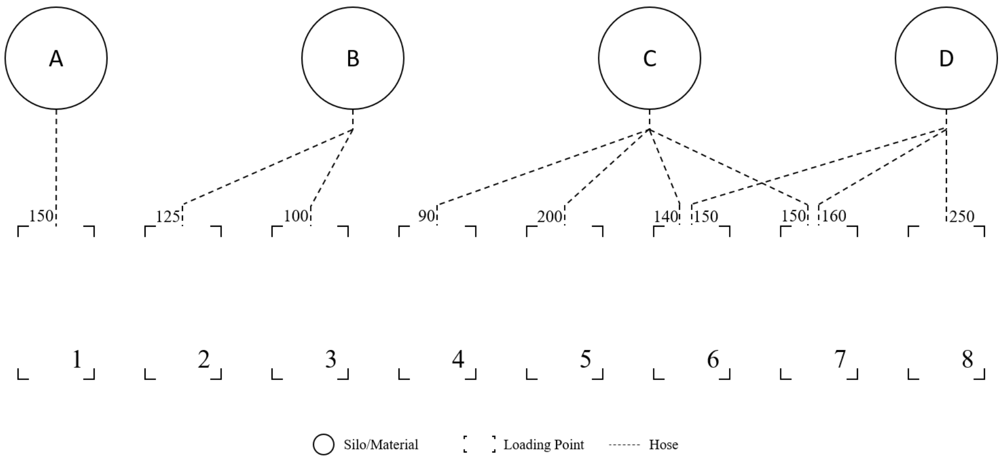

FIGURE 6.1: Combination between silos and LPs of CPI.

The generated demands, for the different materials, were based on the number of hoses capable of loading each material. Thus, material A will be the one with the less demand,

[1]Although the parameters of the SA method can be calibrated, it has always a probabilistic feature present in the acceptance function. That way, it might happen that the solution does not always converge to the same value. In this work, only the best result and the average of all 10 results, for each instance, will be presented.

followed by material B. Materials C and D represent the most wanted materials. In Table 6.1 it is possible to observe the considered demands of all materials.

Table 6.1: Demand of each material available in CPI.

Material	Hoses	Demand
A	1	10.0%
B	2	20.0%
C	4	40.0%
D	3	30.0%

This plant was tested for $n \in \{10, 12, 50, 100, 200\}$ jobs and the main results are presented in Table 6.2.

Table 6.2: Flow time results in CPI.

Instance	MP	DR	SA_{best}	SA_{avg}
n10R0.50	*115	116	*115	*115
n10R0.75	*105	111	*105	*105
n10R1.00	*105	108	*105	*105
n10R1.25	*103	108	*103	*103
n10R1.50	*99	100	*99	*99
n12R0.50	*172	175	*172	*172
n12R0.75	*148	155	152	152
n12R1.00	*151	153	*151	*151
n12R1.25	*160	163	161	161
n12R1.50	*147	148	*147	*147
n50R0.50	-	1485	1389	1389
n50R0.75	-	1188	1091	1092
n50R1.00	-	900	809	816
n50R1.25	-	871	799	803
n50R1.50	-	645	630	632
n100R0.50	-	5086	4536	4543
n100R0.75	-	3450	3058	3065
n100R1.00	-	2241	1972	1989
n100R1.25	-	1581	1520	1530
n100R1.50	-	1429	1383	1391
n200R0.50	-	19606	16796	16822
n200R0.75	-	10268	8986	9014
n200R1.00	-	4851	4489	4532
n200R1.25	-	4131	3892	3915
n200R1.50	-	3389	3303	3334

*optimum solution found.

The MP method was able to get optimum solutions for 10 and 12 jobs, but was unable to handle larger instances than these. The Dispatching Rule (DR) was never able to obtain the optimum solution, reaching an average deviation of 2.7% and a maximum of 5%, for these small instances. The SA method, in the instances that can be compared with the MP, achieved the optimum solutions in eight out of ten possible times. This method reached an average deviation of 0.3% of the optimum solution and a maximum of 3%. For larger instances, it was possible to compare the results of the heuristics. Here, the SA performed always better than the DR. For 50 jobs, the SA achieved a maximum improvement of 10.1%, in the n50R1.00 instance, and an average of 7.1%. For 100 jobs, the SA achieved a maximum improvement of 12%, in the n100R1.00 instance, and an average of 8.3%. For 200 jobs, the SA achieved a maximum improvement of 14.3%, in the n200R0.50 instance, and an average of 8.5%. It is also possible to notice a decrease in the flow time results, as the factor R increases, for each instance size, in both heuristics.

6.2 Cement Plant II

This CP has five different materials and five LPs. Materials A and B have one hose each, connected to the LP 1, with speeds of 150 and 160 tons per hour. The silo holding material C can feed three different LPs. With hoses connected to LPs 2, 3 and 4, this material can be loaded at speeds of 160, 160 and 120 tons per hour. Material D can be loaded by a single hose that connects the silo to LP 4. There, the customers can load this material at a speed of 130 tons per hour. Finally, material E can be loaded at LP 4, at a speed of 150 tons per hour and at LP 5 at 175 tons per hour. The full combination between the silos and the LPs can be seen in Figure 6.2.

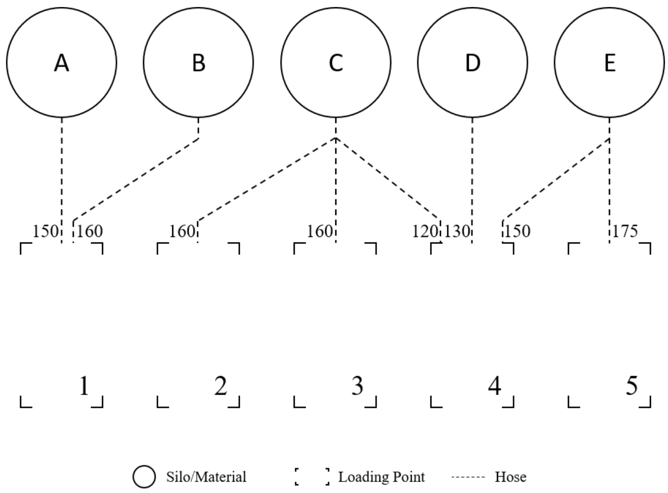

FIGURE 6.2: Combination between silos and LPs of CPII.

Like in the previous CP, the percentage of demand of each material, follows its number of hoses. Thus, material A, B and D will have the same demand. Material E will be the second most wanted material, only outnumbered by material C. To check all the materials' demands, Table 6.3 is presented.

Table 6.3: Demand of each material available in CPII.

Material	Hoses	Demand
A	1	12.5%
B	1	12.5%
C	3	37.5%
D	1	12.5%
E	2	25.0%

The CPII was tested for $n \in \{10, 15, 50, 100, 200\}$ jobs and the main results are presented in Table 6.4.

Table 6.4: Flow time results in CPII.

Instance	MP	DR	SA_{best}	SA_{avg}
n10R0.50	*159	*159	*159	*159
n10R0.75	*142	*142	*142	*142
n10R1.00	*136	139	*136	*136
n10R1.25	*123	*123	*123	*123
n10R1.50	*127	136	136	136
n15R0.50	*290	298	*290	*290
n15R0.75	*254	280	274	274
n15R1.00	*212	*212	*212	*212
n15R1.25	*192	193	*192	*192
n15R1.50	*203	*203	*203	*203
n50R0.50	-	1959	1779	1779
n50R0.75	-	1530	1418	1418
n50R1.00	-	1024	964	964
n50R1.25	-	1143	1050	1054
n50R1.50	-	818	784	790
n100R0.50	-	7218	6215	6222
n100R0.75	-	4626	4002	4008
n100R1.00	-	3621	3281	3284
n100R1.25	-	2044	1918	1933
n100R1.50	-	1906	1777	1783
n200R0.50	-	27086	23384	23408
n200R0.75	-	16142	13877	13913
n200R1.00	-	6774	5834	5879
n200R1.25	-	6977	6219	6248
n200R1.50	-	4123	3809	3836

*optimum solution found.

The MP method was able to get optimum solutions for 10 and 15 jobs, but was unable to handle larger instances than these. Comparing with the MP, the DR was able to obtain the optimum solution in five times, whereas the SA was able to find them in eight times. In these 10 instances, the DR reached a maximum deviation of the optimum solution of 9% and an average of 2%. The SA achieved a maximum deviation of 7% and an average of 1%, for these instances. For larger instances, it was not possible to know how far were the heuristics from the optimum solutions. However, a comparison between the DR and the SA can still be made. The SA method performed significantly better than the DR for all instances. For 50 jobs, the SA achieved a maximum improvement of 9.2%, in the n50R0.50 instance, and an average of 6.9%. For 100 jobs, the SA achieved a maximum improvement of 13.9%, in the n100R0.50 instance, and an average of 9.9%. For 200 jobs, the SA achieved a maximum improvement of 14.0%, in the n200R0.75 instance, and an average of 12%. As in the previous CP, it is also possible to notice a decrease in the flow time results, as the factor R increases, for each instance size, in both heuristics.

6.3 Cement Plant III

This last CP has three silos, each one with a different material, and three LPs. Material A can be loaded in LP 1 at a speed of 180 tons per hour. Material B can be loaded by two different hoses, located in LPs 1 and 2, at a speed of about 150 and 160 tons per hour, respectively. Material C has only one hose capable of loading the trucks. It is located in LP 3 and has a speed of 140 tons per hour. The full combination between the silos and the LPs can be seen in Figure 6.3.

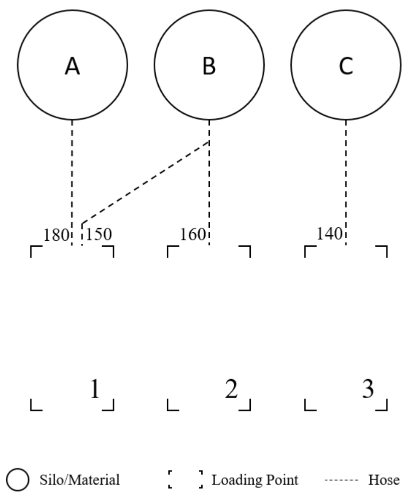

FIGURE 6.3: Combination between silos and LPs of CPIII.

As before, the generated demands are proportional to the number of hoses capable of loading each material. Thus, material B, which has twice the hoses of materials A or C, will also have twice the demand than these materials. In Table 6.5 it is possible to observe the considered demands of all materials.

TABLE 6.5: Demand of each material available in CPIII.

Material	Hoses	Demand
A	1	25.0%
B	2	50.0%
C	1	25.0%

This CP was tested for $n \in \{10, 15, 50, 100, 200\}$ jobs. The MP method was tested for only the size of 10 and 15 jobs, whereas the heuristics were tested for all values of n. In fact, the MP approach, showed to be incapable of solving the problem for more than 15 jobs, within the 8 hours available in the NEOS Server. A summary of the main results is presented in Table 6.6.

TABLE 6.6: Flow time results in CPIII.

Instance	MP	DR	SA_{best}	SA_{avg}
n10R0.50	*201	208	204	204
n10R0.75	*186	*186	*186	*186
n10R1.00	*147	151	*147	*147
n10R1.25	*163	170	*163	*163
n10R1.50	*131	137	137	137
n15R0.50	*328	345	330	330
n15R0.75	*268	273	*268	*268
n15R1.00	*232	255	*232	*232
n15R1.25	*215	241	218	218
n15R1.50	*192	*192	*192	*192
n50R0.50	-	3486	3065	3065
n50R0.75	-	1937	1725	1725
n50R1.00	-	1487	1303	1303
n50R1.25	-	901	857	857
n50R1.50	-	763	750	750
n100R0.50	-	11066	9291	9306
n100R0.75	-	7281	6038	6049
n100R1.00	-	2876	2591	2606
n100R1.25	-	2638	2406	2413
n100R1.50	-	1781	1671	1680
n200R0.50	-	47726	40039	40075
n200R0.75	-	25962	21721	21765
n200R1.00	-	14415	10621	10712
n200R1.25	-	5399	4794	4842
n200R1.50	-	3740	3550	3575

*optimum solution found.

As expected, the MP found the optimum solutions, for the tested instances. When compared with the other methods, it is possible to observe that the DR method found twice the optimum solutions, whereas the SA found them in six different instances. For 10 and 15 jobs, the DR approach reached an average deviation from the optimum solution of 4.1% and maximum deviation of about 10.8%, in the n15R1.25 instance. The SA, on the other hand, came closer to the MP approach, reaching an average deviation of only 0.5% and a maximum deviation of 4.4%, in the n10R1.50 instance. For larger instances, it was not possible to assess how far were the heuristics from the optimum solutions. However, a comparison between the DR and the SA and an analysis of the two methods can still be made.

Simulated Annealing Accuracy

Looking to the SA flow time results, it is possible to verify that, for 50 jobs and smaller instances, the best and the average solutions are always equal. Additionally, although for the instances with 100 and 200 jobs the best solution deviates a bit from the average solution, there is a maximum deviation of less than 1%. A deviation so small corroborates the validity of the initial parameters used in this method, since the SA always converged, in the 10 tests, to the same solution or came very close to it. However, there is no way to tell if the obtained solution is the optimum or not. Still, this allows one to use the best solution for comparison purposes.

Improvements with the Number of Jobs

In general, SA always improved the solutions of the DR. This was expected, since SA spends way more computational efforts than the other heuristic. Nevertheless, the improvements were, in some cases, very high, reaching a maximum of 26.3% for the n200R1.00 instance. This corroborates the quality of the search for neighborhood solutions and the cooling rate of the system.

Also, there is a relationship between the percentage of the average improvement and the size of the instances. In fact, a monotonically increasing behavior was found, giving greater improvements, as the size of the instances grows. This relationship is given by a logarithmic function and can be observed in the Figure 6.4. As the number of jobs increases, there are more chances of improvement, since there are more possible combinations, to form a feasible schedule.

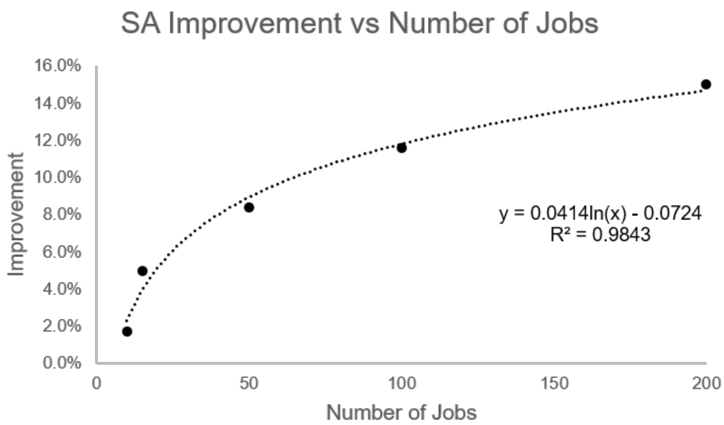

Figure 6.4: SA improvements in CPIII.

Processing Times vs Waiting Times

Total flow time is the sum of the the total processing times with the total waiting times that the customers spend in the system. Although the goal was to minimize the total flow time, it was possible to discriminate its components and evaluate their variation in the two heuristics.

Here, the SA shown to be highly effective in reducing the waiting times, but to have very little or no influence in the processing times. In fact, in average, the SA obtained about 14.7% of improvements, for the total waiting times, but only 0.3%, for the total processing times. This may be due to the similarity of all machines' speed and all the quantities ordered. Although these values may be different from each other, they may have not varied enough to show differences in the results. Even so, the improvements in the waiting times were very positive and important in the building of an efficient schedule, for the customers. To see all the obtained results for the total processing and waiting times, Table 6.7 is presented.

TABLE 6.7: Processing and Waiting Times Results in CPIII.

	Total Processing Time		Total Waiting Time	
Instance	DR	SA_{best}	DR	SA_{best}
n10R0.50	123	123	85	81
n10R0.75	123	123	63	63
n10R1.00	124	123	27	24
n10R1.25	123	123	47	40
n10R1.50	124	124	13	13
n15R0.50	176	175	169	155
n15R0.75	177	175	96	93
n15R1.00	176	175	79	57
n15R1.25	177	177	64	41
n15R1.50	175	175	17	17
n50R0.50	595	593	2891	2472
n50R0.75	595	594	1342	1131
n50R1.00	599	595	888	708
n50R1.25	600	598	301	259
n50R1.50	598	599	165	151
n100R0.50	1177	1170	9889	8121
n100R0.75	1178	1172	6103	4866
n100R1.00	1177	1171	1699	1420
n100R1.25	1178	1173	1460	1233
n100R1.50	1177	1175	604	496
n200R0.50	2392	2378	45334	37661
n200R0.75	2393	2383	23569	19338
n200R1.00	2391	2382	12024	8239
n200R1.25	2396	2392	3003	2402
n200R1.50	2396	2396	1344	1154

Flow Time Results with the factor R

The factor R tries to simulate the effects of release dates tightness. When R is small, there are more customers arriving, in the same period of time, than when R is bigger. The variation of this factor had high impact in the total waiting time of the customers. When customers arrive more frequently to the CP, it is likely that they will not have any available machine, at that moment, having to wait in the park for their turn. When R increases, the jobs and their release dates become more dispersed and the waiting times decrease. As suggested by Figure 6.5, it is expected that the total flow time approximates the value of the total processing time, for higher values of R. Eventually, this values will be equal, as the total waiting time reaches the value of zero. Although this chart was made with the results obtained by the SA method, this behavior was also verified for the DR method, as seen in the Appendix C.

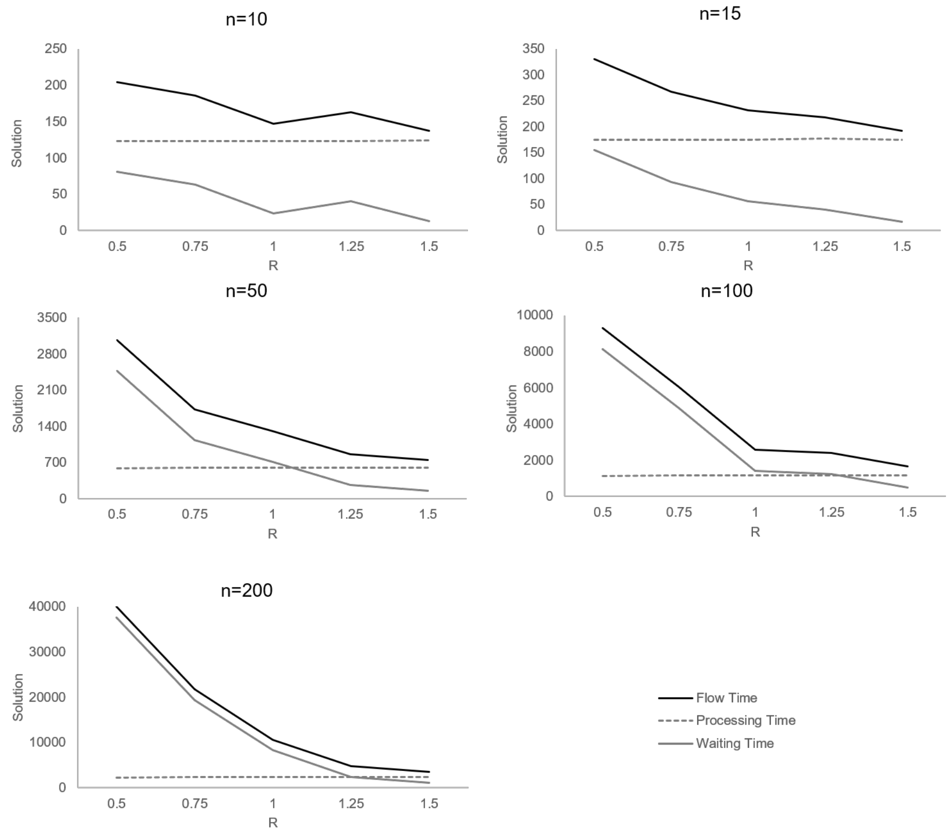

FIGURE 6.5: Flow time behavior with R variation, using SA results.

Running Time

In the Table 6.8 it is possible to observe the running times, obtained in the computational tests. The times are in seconds, and the value 0 is used when the method gives a solution in less than a second, almost instantly. In the case of the SA method, three different times are presented. The SA time is the time spent by the algorithm, during the computational test of that instance. As mentioned before, it was chosen to run the algorithm for a time equal to the number of jobs of the instance, n seconds. Also, 10 computational tests were made for each instance, using the SA method. That way, the SA_{best} time indicates the amount of time the algorithm needed to find the best solution, in each run. The SA_{avg} time is the average time the algorithm needed to find the best solution in the 10 computational tests.

TABLE 6.8: Running times of the computational tests in CPIII.

Instance	MP	DR	SA	SA_{best}	SA_{avg}
n10R0.50	10	0	10	0	0
n10R0.75	10	0	10	0	0
n10R1.00	10	0	10	0	0
n10R1.25	5	0	10	0	0
n10R1.50	5	0	10	0	0
n15R0.50	14000	0	15	0	0
n15R0.75	726	0	15	0	0
n15R1.00	475	0	15	0	0
n15R1.25	131	0	15	0	0
n15R1.50	100	0	15	0	0
n50R0.50	-	0	50	1	3
n50R0.75	-	0	50	1	1
n50R1.00	-	0	50	4	22
n50R1.25	-	0	50	2	14
n50R1.50	-	0	50	1	4
n100R0.50	-	0	100	28	50
n100R0.75	-	0	100	9	55
n100R1.00	-	0	100	79	76
n100R1.25	-	0	100	64	76
n100R1.50	-	0	100	88	78
n200R0.50	-	0	200	68	98
n200R0.75	-	0	200	171	146
n200R1.00	-	0	200	199	157
n200R1.25	-	0	200	163	172
n200R1.50	-	0	200	167	188

For 10 jobs the MP gave the optimum solutions in 10 seconds or less, remaining applicable to industrial problems. For the 15 jobs instances, a pattern was found, spending this method less time to find the optimum solution, as R increases. This can be due to the fact that, for larger values of R, there is a bigger dispersion of jobs, reducing the number of feasible solutions. However, the computational time, for this number of jobs, can reach almost 4 hours, which is too long, in an industry where decisions must be made promptly. When considering larger instances, this method was not capable of finding the optimum solutions, inside the available 8 hours of computational time, given by the NEOS Server. In fact, it was possible to notice an exponential growth of the computational time, as the number of jobs increased. This growth can be verified in the Figure 6.6. Following the trend line equation that best fits the collected data, it is estimated that, for 16 jobs, the MP would need around 16 hours. This is a rough estimation, but can help to understand why it was not possible to find the optimum solutions for instances larger than 15 jobs.

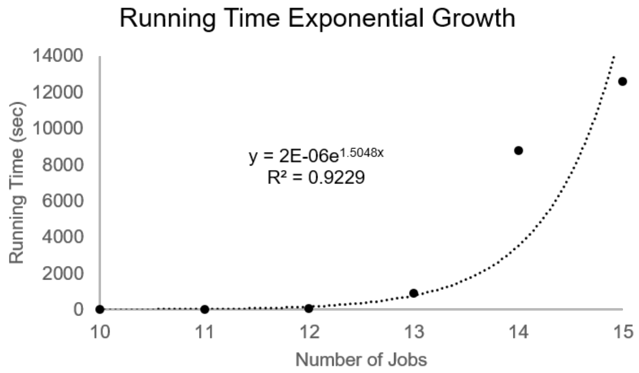

FIGURE 6.6: Running time exponential behavior of the MP method.

The heuristics obtained solutions in much less time than the MP. In particular, the DR method, which always gave solutions in less than a second. The SA algorithm was always run for n seconds. However, it is possible to verify that the best solutions were almost always found in much less time. This suggests that the running times may have not been chosen the best way and they may be subjected to improvements. Still, 200 seconds for a 200 jobs instance is not that long and may be applicable to most industrial problems.

Schedule and Allocation

So far, there has been a discussion about the performance of the developed methods and its applicability to the CPs. But there are more advantages than reducing waiting, processing and flow times. With these methods, it is also possible to take the service to another level and improve the relationship with the customers.

After running the models, information about the operations of each customer is gathered and a schedule is built from there. As an example, Table 6.9 suggests that each customer has estimated times for waiting, to start its operation, for how long the loading will take and at what time it is estimated that he leaves the plant. Also, he will know to which LP he must go to see his order fulfilled, no longer choosing the wrong LP. Through a simple interface, this information can be given to the customers, before or as they arrive to the CPs. Knowing the estimated times to be served can improve the service level, since the customers will no longer have to wait indefinitely, in the park. Assigning each customer to the correct LP brings benefits to the company that sees its entropy being reduced and to the customer that spends less time can have a better experience inside the CP. The example here presented refers to the SA results on the n10R.100 instance. However, it is possible to collect this data from all the methods and for all instances.

TABLE 6.9: Example of schedule, using the SA on the n10R1.00 instance.

id	LP	Arrived	Wait	Start	Process	End	Flow
1	1	0	0	0	8	8	8
2	2	9	0	9	12	21	12
3	2	26	7	33	15	48	22
4	1	9	0	9	12	21	12
5	3	24	0	24	16	40	16
6	3	3	0	3	14	17	14
7	1	38	5	43	12	55	17
8	1	13	8	21	11	32	19
9	1	30	2	32	11	43	13
10	2	19	2	21	12	33	14

To the company, information about the allocation of customers and the performance of each particular LP can be evaluated. For example, in Figure 6.7, it is possible to observe a Gantt chart, containing the sequence of customers allocated to each LP, the number of jobs each one processed and how long it took. In addition to this information, it would also be possible to know the average processing time of each LP, its occupation, its idle times, and others. Using this data, allows the company to plan changes that will improve the LPs' performance and the CP's performance as well. These changes might involve adding or reducing the number of LPs, implementing faster hoses or changing the combination between silos and LPs. Also, any modification would implicate the spending of a lot of money, and companies do not want to incur in high risks. That way, computational tests on a virtually modified CP would allow the companies to evaluate how better they would perform and at what cost. Again, it is possible to collect this data from all the methods and for all instances.

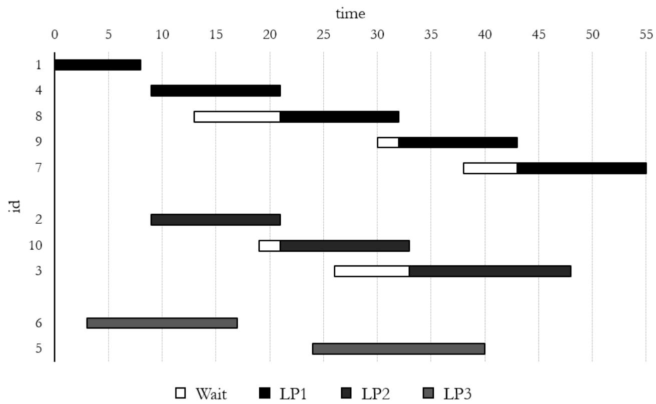

FIGURE 6.7: Example of Gantt chart, using the SA on the n10R1.00 instance.

Chapter 7

Conclusions

The major aim of this dissertation was to change the paradigm of customers' scheduling in the Cement Industry (CI). More specifically, it was intended to reduce their interaction times and improve the level of service. Thus, a study was carried out to establish an analogy between the truck loading scheduling problem and the Machine Scheduling (MS) problems. After analyzing the characteristics and constraints of the issue, its similarity with the $(Rm|r_j, M_j|\sum F_j)$ problem was clear. In order to tackle a problem which cannot be found in the literature, three different methods were developed from scratch. These were subjected to an extensive series of computational experiments to assess its quality and applicability to a real life problem. It was shown that a customer oriented CI can still be efficient and it was clear the need to include optimization models, which improve scheduling. These models have advantages not only for the customers, but also for the Cement Plants (CPs). Some major contributions were drawn from this work.

Major Contributions

It was presented a literature review that addressed this work main areas of intervention. It began by describing the tasks of Supply Chain Management (SCM) and logistics in an increasingly customer oriented market. Here, it was emphasized the influence of these two areas in maintaining a correct management of the operations and in achieving customers' satisfaction. Afterwards, several different types of MS problems were presented. Faced with a very long list, it was highlighted the need to adopt a systematic notation for its representation, which was also presented. In addition, an analysis was made on the complexity of these problems and on the impact this may have on building an efficient schedule. The literature review ended with the presentation of some solving techniques for MS problems – an exact method and two heuristics. The exact method not only stood out for being able to obtain optimum solutions for small instances, but also for the high computational time it required. The constructive heuristics stood out for their easy implementation, flexibility and speed in finding good solutions. The improvement heuristics had their main advantage in their ability to obtain solutions closer to the optimum solution. Also, both types of heuristics are capable of handling instances of any size, which is important in solving industrial problems.

The case study began with a characterization of the CI, highlighting the need to develop its SCM and to turn its orientation towards the customers. A detailed description of the existing processes in a CP was also made, thus understanding its size and complexity. Then, a specific process was approached – the scheduling of the loading of trucks by the customers. In this process, a lack of management by the cement companies and the long waiting times and low level of service offered to the clients were highlighted. In order to improve the customers' experience and reduce their interaction times, a list of assumptions and an analysis of the main variables of the problem was made before solving it. Also, the total flow time was chosen as the objective to be minimized, as it is widely accepted as a measure of a system's quality of service.

Afterwards, three optimization models were proposed. The Mathematical Programming (MP) method is an exact method and is based on a mathematical formulation of the problem. It was implemented in AMPL programming language and tested on the NEOS Server using the IBM ILOG CPLEX Optimizer solver. This method was developed with assignment and positional date variables. It allowed to better understand the structure of the problem and obtain optimum solutions. On the other hand, the heuristics were both developed in JAVA programming language and were run on an Intel Core i7-4700HQ with 2.40GHz and 8Gb of RAM memory. In the Dispatching Rule (DR) method, a combination of the Shortest Processing Time (SPT) and Earliest Completion Time (ECT) rules was used to obtain a good schedule. This method allowed to achieve a fair sequence for processing the customers and to reduce the time they spend in the system. The Simulated Annealing (SA) method used the solutions given by the previous heuristic and tried to find better schedules in its neighborhood. Here, a time limit was chosen as a stop criterion and the best solution found during the run was returned.

To test the developed methods, an extensive series of instances was built for three different CPs, whose characteristics were known through several meetings. To build each instance, statistical distributions were suited to the provided raw data. These distributions contained customers' information, such as the materials demand, ordered quantities and their release dates. Different release dates tightness were also considered, allowing to test high and low periods of demand in a day of work at the CPs. Each instance was tested for a different number of jobs, that varied from 10 to 200.

For the three CPs, the MP method allowed to find optimum solutions. However, it was not able to obtain solutions for instances with more than 12 or 15 jobs. This is due to the high computational efforts required to find a solution in this approach. In fact, it was shown in this document that the computing time of this method grows exponentially with the number of jobs. Knowing that the MP method is not applicable to larger instances, the computational tests proceeded with the heuristics.

The DR method, only found the optimum solutions in few small instances. For larger instances, this method behaved much worse than the SA, having obtained much larger total flow time values. The SA method was able to find the optimum solutions in most small instances. For larger instances there was no way to tell whether or not this method achieved the optimum solutions. For these instances, there was also the concern that this method would not always converge to the same solution due to its probabilistic properties. This concern was mitigated by having done 10 tests for each instance, which always almost got equal solutions.

When the results of the total flow times were analyzed, it was observed that the higher the number of jobs, the greater the improvement of the SA in relation to the DR. Although these improvements increase for larger instances, their growth rate decreases, being this behavior described by a logarithmic function. The results of the total processing and waiting times were also analyzed. Being these components part of the total flow time, it was intended to perceive the influence of the SA in these two measures. Here, it was noticed that the obtained improvements by the SA had a great impact on the waiting times and had little or no impact on the processing times. Furthermore, all instances were subject to different release dates tightness. Here, it was observed that this factor had a great influence on the waiting times in both heuristics. Thus, waiting times dropped abruptly as release dates became more dispersed.

From a real life implementation point of view, it was realized that the MP method should be discarded due to its high computational time. Actually, the CI deals with hundreds of trucks every day and decisions have to be made quickly, so the method must return solutions to any instance and in a short amount of time. As for the SA method, it obtained much better solutions than the DR and in reasonable amounts of time, being therefore plausible its implementation. However, the DR obtained much faster solutions and its implementation is much easier than the SA method. Moreover, this method is flexible and more easily includes certain characteristics of the problem. This way, the DR method implementation in real life is also plausible. Nevertheless, this method should be improved before implementation. Here, the results obtained by the SA method could tell how much the DR method could improve.

Through these methods it was also possible to know important times for each customer, such as their estimated waiting time and processing time. By making these times known to customers, something that at the moment does not happen, it is possible to improve the level of service. In addition, it was also possible to extract information about the loading points and its occupation. This can help the CPs find the bottlenecks of the system and in the decision making in order to make the processes more effective.

Future Work

Having in mind future work on this issue, it should be divided into two parts – the improvement of the developed models and their implementation in real life. Regarding the first part and the MP method, there is the need to make it more efficient through a better mathematical formulation. It would be expected that its computing time would not grow so quickly, making it possible to obtain solutions for slightly larger instances. As for the DR method, there are opportunities to get much better solutions. In fact, this is a method with a lot of potential and can be improved by including more rules and better tie breaks. As for the SA method, its initial parameters should be studied in order to find better values that increase the algorithm efficiency. Also, an evaluation on finding better solutions should be made, regarding the influence of the different neighborhood moves. From a more practical point of view, there is the need to implement optimization models in real life. In addition, two simple interfaces should be developed: one for the customers in order to provide them with estimated waiting and processing times, thus improving the level of service; another for the company, allowing it to be more informed and, with that, find improvement opportunities and make a better management of the CP.

Appendix A

Cement Plant I

Generated Instances

TABLE A.1: Generated data, for the customers of CPI - 10 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	C	29	3	9	8	9	8
2	C	22	2	1	9	1	14
3	C	23	3	0	7	14	0
4	D	25	5	2	12	7	18
5	B	28	0	4	10	10	12
6	C	29	3	3	5	6	2
7	B	25	1	8	9	0	8
8	A	24	3	5	3	5	9
9	D	22	5	3	0	12	19
10	D	26	2	7	9	7	5

TABLE A.2: Generated data, for the customers of CPI - 12 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	D	31	3	8	13	4	2
2	C	28	4	13	11	7	26
3	A	34	1	10	11	11	18
4	D	34	0	1	7	6	19
5	B	32	7	11	6	20	11
6	C	31	1	13	0	5	22
7	C	33	3	11	12	19	25
8	C	31	4	1	4	8	26
9	B	19	8	0	16	6	13
10	D	22	6	7	15	5	0
11	C	26	3	8	4	9	19
12	D	36	6	11	13	0	0

Table A.3: Generated data, for the customers of CPI - 50 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	A	30	6	36	37	81	27
2	D	27	4	16	43	40	79
3	D	34	0	7	72	86	62
4	D	30	15	37	18	57	106
5	C	35	32	27	24	33	48
6	D	32	35	0	3	42	17
7	C	27	20	2	54	70	70
8	C	39	12	24	41	75	50
9	B	28	15	38	51	87	20
10	C	32	25	4	3	46	110
11	B	26	10	38	47	91	41
12	C	33	24	21	46	70	92
13	D	20	13	24	45	23	5
14	D	33	13	18	41	24	2
15	D	35	36	43	1	44	68
16	C	28	10	31	24	48	63
17	B	28	9	31	28	50	14
18	D	25	15	47	54	80	81
19	A	34	1	49	2	44	108
20	D	36	37	3	53	3	105
21	D	33	24	51	42	47	78
22	C	29	9	15	54	31	59
23	C	28	23	54	71	32	102
24	B	22	24	3	23	48	1
25	C	33	1	12	21	6	91
26	C	35	17	5	22	59	23
27	C	31	27	22	69	40	2
28	B	31	23	23	71	46	101
29	B	23	15	35	42	45	0
30	D	29	35	11	37	0	82
31	B	29	10	26	37	47	110
32	C	29	36	37	27	11	93
33	A	31	26	37	5	46	3
34	C	38	2	9	7	19	30
35	B	23	8	43	10	57	66
36	C	30	5	35	37	14	64
37	B	22	9	37	68	40	99
38	A	14	6	13	49	77	22
39	D	29	21	18	19	81	89
40	C	28	34	30	63	35	4
41	C	29	22	16	71	13	98
42	D	45	19	31	31	9	28
43	D	34	26	50	52	85	44
44	C	26	29	50	5	69	73
45	A	22	3	24	20	78	97
46	C	28	19	47	41	87	56
47	C	33	21	21	18	56	42
48	C	35	10	44	43	80	19
49	B	32	4	8	0	44	95
50	D	29	0	38	63	52	12

Table A.4: Generated data, for the customers of CPI - 100 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	A	30	40	11	144	13	32
2	D	27	72	70	33	46	128
3	D	34	31	34	85	84	207
4	D	30	5	82	67	100	52
5	C	35	4	91	106	165	153
6	D	32	22	78	85	49	57
7	C	27	32	45	87	105	45
8	B	39	19	51	113	157	132
9	B	28	8	77	15	93	157
10	C	32	11	82	75	57	13
11	B	26	10	15	70	51	181
12	C	33	47	42	22	120	0
13	D	20	56	77	76	6	206
14	D	33	1	6	45	0	143
15	D	35	38	38	87	58	143

16	C	28	29	17	60	79	151
17	B	28	35	101	89	60	125
18	D	25	67	97	83	53	35
19	A	34	56	34	10	150	112
20	D	36	51	41	127	69	0
21	D	33	12	64	92	113	137
22	C	29	27	99	33	119	184
23	C	28	2	38	47	59	89
24	B	22	26	62	14	74	72
25	D	33	58	33	140	182	78
26	B	35	57	17	49	142	64
27	C	31	67	17	57	67	170
28	B	31	57	68	133	110	205
29	B	23	12	105	125	135	161
30	D	29	70	14	91	122	174
31	B	29	30	84	127	152	193
32	C	29	1	74	116	30	87
33	A	31	35	105	81	20	68
34	C	38	72	26	84	89	39
35	B	23	10	4	26	178	7
36	B	30	28	12	8	107	132
37	B	22	0	74	103	92	103
38	A	14	68	67	31	30	200
39	D	29	46	68	9	49	182
40	C	28	37	55	115	13	165
41	C	29	42	50	0	103	74
42	D	45	50	38	21	13	162
43	D	34	32	22	135	0	191
44	D	26	17	27	86	27	128
45	A	22	28	4	70	70	88
46	C	28	48	4	13	139	142
47	C	33	70	10	128	132	95
48	C	35	30	81	109	39	39
49	B	32	42	3	133	21	32
50	D	29	13	103	80	7	108
51	A	25	34	95	62	81	111
52	C	39	49	83	43	11	104
53	C	31	14	107	131	43	148
54	C	28	47	43	104	97	185
55	C	28	42	12	35	170	88
56	C	33	30	108	128	56	216
57	A	31	0	88	46	149	133
58	C	33	22	109	58	145	159
59	A	27	28	27	18	121	140
60	A	32	19	30	118	82	164
61	D	26	61	29	50	172	83
62	D	24	30	5	148	33	128
63	C	37	47	3	107	175	56
64	D	27	61	70	33	132	26
65	C	20	44	7	134	153	22
66	C	24	72	99	33	145	38
67	B	25	17	110	139	168	36
68	A	24	31	58	131	69	60
69	C	27	15	10	127	109	12
70	D	26	45	57	106	43	172
71	D	34	5	55	99	88	161
72	B	25	25	45	21	110	21
73	C	33	2	99	95	27	35
74	C	26	61	5	41	168	86
75	C	33	3	48	115	85	46
76	D	39	21	18	2	109	101
77	C	26	24	54	93	90	68
78	C	25	18	84	115	79	65
79	C	30	65	41	1	130	61
80	B	30	66	47	115	31	77
81	C	43	28	74	20	120	14
82	C	37	72	101	68	150	155
83	D	28	40	19	97	50	36
84	D	27	40	21	132	61	216
85	D	32	38	6	19	97	112
86	B	24	69	18	115	33	169
87	D	33	17	93	108	163	160
88	D	27	49	0	116	20	70
89	B	30	8	95	93	183	155
90	C	27	67	98	62	95	179
91	C	37	31	78	70	68	203
92	D	40	42	1	38	51	34
93	C	33	21	84	139	74	154
94	B	33	35	18	99	40	213
95	C	27	3	67	135	90	7
96	C	23	13	4	17	12	35

97	C	27	59	92	87	3	98
98	B	31	38	77	115	13	5
99	C	35	24	23	54	149	37
100	D	28	41	18	105	118	220

TABLE A.5: Generated data, for the customers of CPI - 200 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	C	29	0	99	35	304	135
2	B	24	119	157	177	186	226
3	C	37	28	170	143	208	153
4	A	32	78	51	42	164	416
5	A	34	114	199	262	369	2
6	B	33	78	57	19	44	150
7	D	34	76	2	35	294	6
8	C	28	21	217	68	360	278
9	C	41	11	159	114	121	168
10	C	24	95	21	293	229	146
11	D	34	117	26	267	41	193
12	C	34	56	57	202	264	0
13	D	26	38	166	44	245	401
14	C	31	1	93	206	28	65
15	D	22	23	182	193	349	361
16	A	34	39	74	69	335	152
17	B	28	86	61	282	65	232
18	C	23	53	138	136	308	32
19	D	38	58	151	65	49	211
20	D	29	143	190	217	328	287
21	A	30	70	193	125	277	380
22	B	37	15	204	101	2	437
23	D	30	101	174	48	211	36
24	D	34	28	190	256	296	269
25	C	32	70	11	16	19	141
26	C	39	20	104	205	315	283
27	D	29	44	4	89	71	379
28	B	34	119	21	284	78	234
29	C	24	72	181	261	271	231
30	C	31	105	199	262	302	35
31	B	26	10	27	213	359	229
32	A	24	146	215	222	285	76
33	C	21	73	129	97	236	395
34	A	35	136	136	78	192	186
35	C	29	20	85	285	29	31
36	B	27	92	132	91	133	22
37	D	30	60	124	154	163	396
38	B	34	112	182	266	177	349
39	C	30	120	44	282	314	27
40	C	30	49	53	287	303	94
41	B	29	28	17	192	225	341
42	C	28	55	145	40	311	320
43	A	24	1	69	137	264	232
44	C	26	119	168	154	277	35
45	C	27	139	200	159	39	96
46	C	27	69	75	138	329	359
47	D	35	20	214	175	289	99
48	C	20	117	37	122	164	136
49	C	32	7	23	32	271	388
50	C	20	65	163	237	241	356
51	C	20	120	187	33	61	173
52	D	33	75	18	93	174	114
53	C	22	17	100	254	196	26
54	D	24	35	61	182	121	24
55	B	24	58	213	43	296	243
56	B	35	108	13	223	340	33
57	D	28	117	202	19	44	283
58	C	33	79	128	10	330	230
59	D	33	96	84	3	294	291
60	C	34	56	139	46	324	413
61	B	34	18	59	230	151	203
62	D	27	22	166	115	209	43
63	C	34	76	220	210	36	80
64	B	22	126	12	12	54	189
65	D	26	108	82	110	213	100
66	B	26	54	151	76	196	278

67	B	35	56	32	271	193	111
68	A	27	134	132	199	275	325
69	C	34	36	23	215	269	109
70	C	24	11	86	158	216	126
71	C	31	97	92	263	320	261
72	B	35	54	63	274	68	199
73	C	29	50	117	210	39	410
74	D	27	53	134	165	31	303
75	C	39	45	64	66	84	144
76	D	31	59	9	203	45	340
77	D	28	140	66	51	124	204
78	C	33	68	162	243	340	200
79	C	33	134	9	248	86	174
80	D	27	111	27	150	132	401
81	B	37	123	183	8	189	268
82	B	22	110	89	92	294	256
83	C	35	143	135	26	97	284
84	C	30	83	153	260	275	186
85	C	37	17	61	190	227	401
86	C	38	100	149	292	238	122
87	D	37	25	153	135	139	140
88	D	35	28	128	240	8	159
89	C	27	0	196	42	48	155
90	A	30	36	36	131	147	312
91	C	30	10	201	196	37	40
92	B	29	109	203	22	44	362
93	C	29	49	146	252	101	362
94	C	33	5	24	87	353	109
95	D	34	22	69	199	259	119
96	A	30	88	185	258	348	416
97	C	21	9	213	129	265	87
98	B	26	53	157	87	203	325
99	D	32	80	209	132	40	153
100	D	27	112	179	45	75	385
101	B	24	58	9	294	103	299
102	C	31	65	147	216	148	117
103	D	28	68	180	260	68	248
104	C	28	37	195	141	107	313
105	D	36	102	5	204	151	437
106	C	38	54	81	39	311	15
107	C	28	100	9	236	160	275
108	D	36	57	84	170	270	433
109	D	28	124	83	34	338	53
110	C	27	89	122	241	271	379
111	C	26	130	98	142	311	100
112	A	26	21	62	146	102	296
113	D	28	26	95	11	315	158
114	A	32	122	205	83	177	311
115	B	26	72	72	156	9	255
116	D	32	69	201	162	53	154
117	B	38	103	208	193	347	6
118	D	25	138	126	15	137	15
119	C	35	100	185	167	105	216
120	B	28	119	16	255	152	179
121	D	24	99	52	17	162	384
122	A	30	123	144	260	209	96
123	D	31	14	121	189	290	143
124	D	29	86	84	242	62	414
125	A	34	44	0	18	333	265
126	D	30	9	45	292	185	339
127	D	36	21	149	211	168	146
128	C	20	136	174	50	137	139
129	D	38	90	37	289	247	96
130	D	35	113	110	72	367	328
131	C	24	41	127	60	308	261
132	A	26	68	20	23	0	440
133	B	32	34	13	178	207	362
134	C	36	141	193	269	298	26
135	B	25	24	213	272	162	95
136	B	23	65	128	266	189	224
137	C	27	29	35	293	234	278
138	C	31	116	86	264	1	3
139	D	36	25	104	101	331	428
140	B	32	136	73	292	47	128
141	C	26	64	159	63	160	320
142	B	33	98	65	45	299	314
143	C	37	1	211	189	156	403
144	D	29	55	115	75	368	128
145	A	31	13	121	217	235	140
146	C	30	129	141	47	121	390
147	C	30	7	52	98	311	141

148	C	26	68	50	251	280	402
149	D	33	58	43	165	178	18
150	C	27	104	214	15	310	273
151	C	36	19	148	289	293	196
152	B	32	31	219	190	200	140
153	C	37	20	200	254	344	53
154	C	16	79	113	133	365	220
155	B	27	38	34	253	112	134
156	B	25	73	158	263	334	304
157	C	32	72	102	43	334	374
158	D	34	11	173	148	236	350
159	D	31	31	15	287	120	84
160	C	25	141	36	229	297	145
161	D	31	41	102	218	212	102
162	C	27	15	202	40	254	25
163	D	32	40	164	105	173	0
164	B	31	32	129	51	22	133
165	D	30	63	216	179	20	150
166	C	26	129	171	43	216	406
167	D	31	35	185	8	194	33
168	A	35	113	155	114	143	235
169	D	26	87	42	90	158	22
170	C	30	101	33	9	323	106
171	D	32	125	84	157	357	310
172	D	35	5	160	38	193	415
173	C	28	124	15	29	190	361
174	B	33	82	214	59	196	267
175	B	29	113	171	140	29	404
176	C	28	39	108	170	170	271
177	C	28	56	107	0	99	86
178	D	29	108	200	161	135	101
179	C	31	51	111	279	324	348
180	D	38	102	96	211	125	272
181	A	32	1	212	240	91	298
182	B	27	120	83	265	105	97
183	D	42	38	191	68	164	376
184	C	26	66	119	235	7	156
185	B	36	76	103	256	5	260
186	B	30	36	53	126	247	168
187	A	30	24	106	224	330	336
188	D	22	146	197	58	319	60
189	D	29	96	81	183	338	273
190	C	30	10	27	156	260	163
191	B	29	88	201	181	204	425
192	C	29	30	3	127	283	398
193	D	40	114	213	218	38	319
194	B	14	81	181	43	320	223
195	C	31	36	13	180	214	169
196	C	24	19	156	172	352	284
197	A	27	135	173	251	105	419
198	C	30	91	144	51	155	434
199	D	21	100	180	75	234	420
200	D	29	69	82	8	223	275

Obtained Results

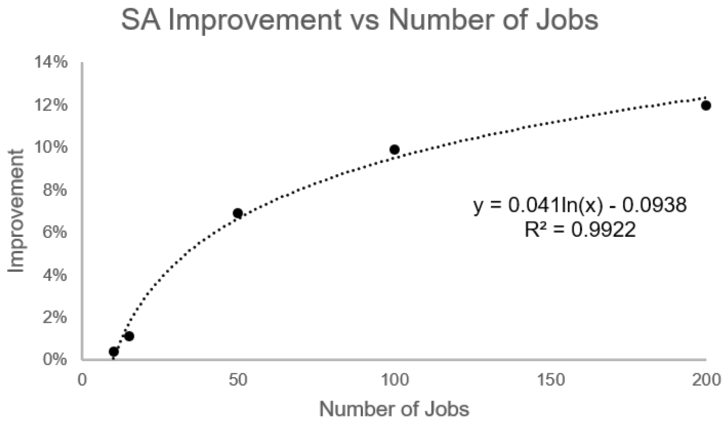

FIGURE A.1: SA improvements in CPI.

TABLE A.6: Processing and Waiting Times Results in CPI.

Instance	Total Processing Time DR	Total Processing Time SA_{best}	Total Waiting Time DR	Total Waiting Time SA_{best}
n10R0.50	96	104	20	11
n10R0.75	106	98	5	7
n10R1.00	104	96	4	9
n10R1.25	106	95	2	8
n10R1.50	100	98	0	1
n12R0.50	143	142	32	30
n12R0.75	143	140	12	12
n12R1.00	145	145	8	6
n12R1.25	148	146	15	15
n12R1.50	147	145	1	2
n50R0.50	596	587	889	802
n50R0.75	602	592	586	499
n50R1.00	598	595	302	214
n50R1.25	602	588	269	211
n50R1.50	606	601	39	29
n100R0.50	1221	1201	3865	3335
n100R0.75	1229	1210	2221	1848
n100R1.00	1211	1201	1030	771
n100R1.25	1206	1208	375	312
n100R1.50	1216	1201	213	182
n200R0.50	2453	2430	17153	14366
n200R0.75	2451	2439	7817	6547
n200R1.00	2454	2456	2397	2033
n200R1.25	2454	2449	1677	1443
n200R1.50	2444	2438	945	865

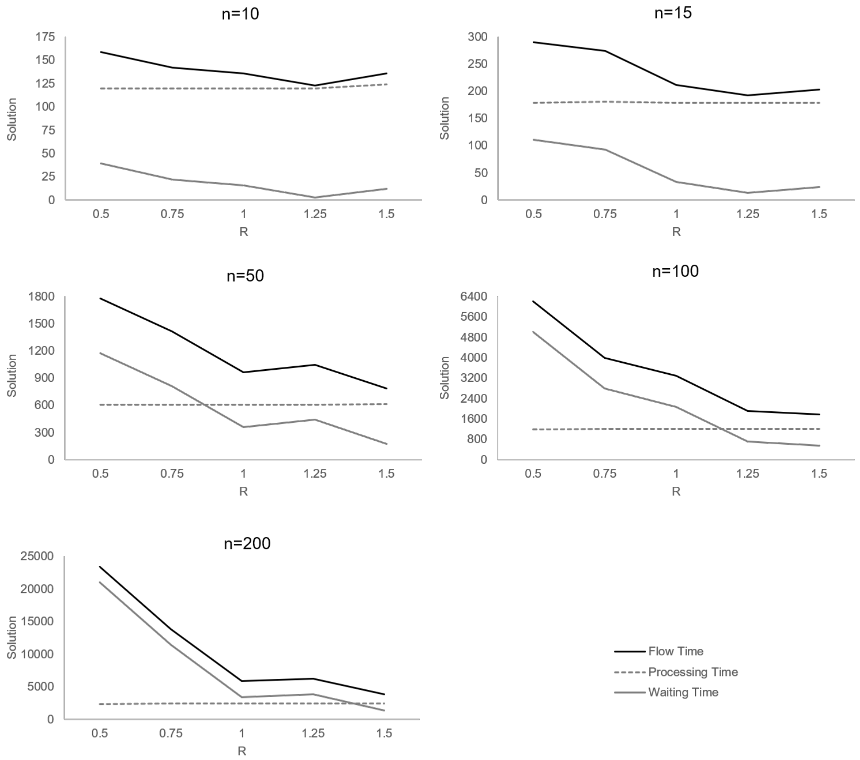

FIGURE A.2: Flow time behavior with R variation, using the SA results.

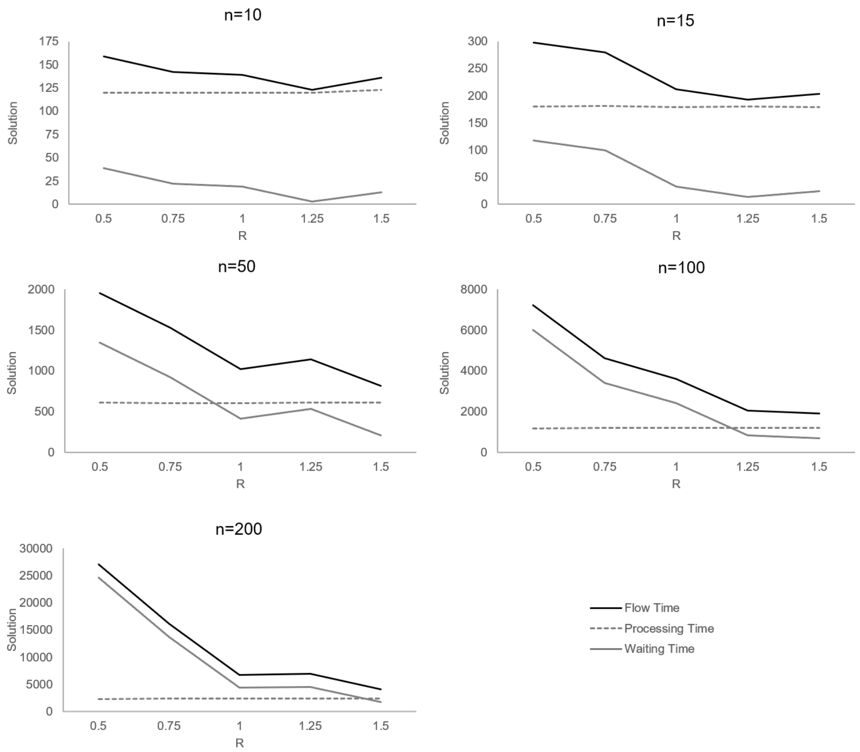

FIGURE A.3: Flow time behavior with R variation, using the DR results.

TABLE A.7: Running times of the computational tests in CPI.

Instance	MP	DR	SA	SA_{best}	SA_{avg}
n10R0.50	20	0	10	0	0
n10R0.75	15	0	10	0	0
n10R1.00	15	0	10	0	0
n10R1.25	21	0	10	0	0
n10R1.50	20	0	10	0	0
n12R0.50	255	0	12	0	0
n12R0.75	190	0	12	0	0
n12R1.00	175	0	12	0	0
n12R1.25	150	0	12	0	0
n12R1.50	101	0	12	0	0
n50R0.50	-	0	50	2	7
n50R0.75	-	0	50	2	16
n50R1.00	-	0	50	22	35
n50R1.25	-	0	50	21	35
n50R1.50	-	0	50	37	14
n100R0.50	-	0	100	20	46
n100R0.75	-	0	100	19	47
n100R1.00	-	0	100	89	76
n100R1.25	-	0	100	45	59
n100R1.50	-	0	100	99	57
n200R0.50	-	0	200	198	128
n200R0.75	-	0	200	123	133
n200R1.00	-	0	200	121	137
n200R1.25	-	0	200	100	153
n200R1.50	-	0	200	157	154

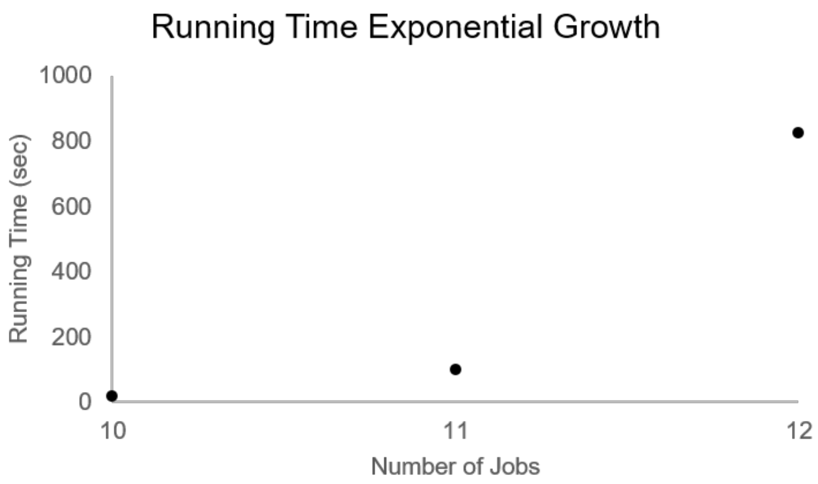

FIGURE A.4: Running time exponential behavior of the MP method.

Appendix B

Cement Plant II

Generated Instances

TABLE B.1: Generated data, for the customers of CPII - 10 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	E	31	9	13	12	12	9
2	C	32	0	3	0	10	2
3	B	31	3	17	23	18	0
4	A	26	5	0	19	27	30
5	E	31	10	0	2	1	10
6	C	36	4	1	22	28	20
7	E	29	5	4	16	26	6
8	C	27	3	10	7	28	29
9	C	33	2	0	23	4	15
10	D	26	1	0	14	0	36

TABLE B.2: Generated data, for the customers of CPII - 15 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	C	30	4	9	18	19	7
2	E	26	0	9	17	15	51
3	A	32	13	25	30	37	19
4	C	30	6	9	31	0	26
5	D	26	7	17	29	24	24
6	C	35	11	7	4	14	41
7	D	35	11	17	34	10	0
8	C	16	2	0	14	42	9
9	B	35	10	1	0	26	22
10	E	32	18	4	12	28	20
11	B	29	17	19	13	2	48
12	A	30	11	18	15	11	9
13	E	27	0	19	14	34	18
14	C	32	1	13	34	35	50
15	E	31	2	15	31	42	8

Table B.3: Generated data, for the customers of CPII - 50 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	C	30	41	4	51	84	136
2	E	25	26	14	32	108	12
3	C	29	4	3	119	3	161
4	D	30	10	29	113	109	107
5	E	31	35	41	64	77	16
6	A	37	46	29	77	111	4
7	E	25	40	67	61	135	155
8	D	27	55	45	21	82	35
9	E	22	57	33	43	123	152
10	B	24	52	27	99	10	157
11	C	35	21	56	22	86	100
12	C	30	58	71	2	74	120
13	B	30	0	80	94	41	0
14	C	35	24	26	118	43	5
15	E	20	50	81	62	117	25
16	C	30	51	38	42	62	136
17	A	37	46	86	62	97	16
18	E	31	54	1	53	96	15
19	C	37	8	36	71	109	169
20	A	33	59	34	35	41	152
21	B	31	54	74	68	96	80
22	C	25	57	68	60	97	13
23	C	35	22	63	22	66	18
24	E	22	3	53	116	131	0
25	B	33	53	37	89	143	115
26	E	32	13	0	109	95	70
27	A	28	37	82	3	28	125
28	E	36	53	87	104	0	3
29	A	35	13	42	105	103	14
30	C	18	33	54	41	8	10
31	B	29	53	77	102	119	55
32	E	24	29	74	107	49	85
33	C	32	44	33	89	26	60
34	E	31	35	88	115	111	171
35	C	26	19	10	68	129	35
36	C	35	37	9	91	23	109
37	D	22	47	50	9	67	33
38	A	26	53	32	16	150	123
39	C	22	58	64	54	101	8
40	E	32	30	14	96	77	15
41	C	33	19	40	0	64	172
42	C	32	13	75	78	115	145
43	C	34	4	7	48	127	44
44	C	31	14	24	19	106	60
45	E	33	41	71	11	15	28
46	D	34	52	63	51	95	168
47	C	34	12	84	33	65	162
48	B	28	23	78	118	108	103
49	D	36	10	27	37	108	177
50	D	38	42	38	105	145	84

Table B.4: Generated data, for the customers of CPII - 100 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	E	27	30	98	192	295	125
2	C	30	87	64	189	141	112
3	E	31	93	64	121	75	277
4	B	28	110	155	99	29	0
5	A	31	11	1	172	227	18
6	C	34	106	105	155	102	169
7	E	24	81	56	237	270	208
8	C	25	100	146	22	249	174
9	E	24	27	94	90	41	215
10	B	23	68	87	65	194	250
11	C	23	10	102	58	48	290
12	D	29	12	36	35	0	228
13	B	27	22	152	31	91	9
14	C	26	68	26	23	173	228
15	A	20	112	82	236	221	250

16	D	40	72	129	212	63	248
17	C	26	109	68	50	235	84
18	C	28	104	74	117	9	107
19	A	35	47	123	101	0	226
20	E	34	103	1	60	179	246
21	B	38	0	63	124	268	11
22	E	33	52	110	237	117	352
23	B	27	81	138	156	99	30
24	E	27	56	98	74	291	268
25	D	26	92	148	82	284	332
26	E	38	98	108	201	4	21
27	E	28	41	96	47	77	252
28	B	24	61	34	47	128	348
29	B	28	99	119	143	4	19
30	E	22	94	177	194	24	96
31	C	42	66	9	161	157	101
32	C	24	92	104	199	111	199
33	E	30	107	173	196	151	164
34	C	33	101	175	210	1	41
35	E	25	69	123	209	152	359
36	C	34	57	4	41	99	128
37	E	30	41	172	131	100	5
38	C	28	102	139	177	2	26
39	A	34	9	116	76	29	193
40	E	32	48	17	70	144	179
41	C	31	39	175	44	1	101
42	D	28	76	150	185	18	105
43	D	37	57	126	132	153	201
44	D	31	13	85	13	80	66
45	C	32	119	78	29	95	216
46	C	34	82	2	91	241	117
47	C	29	105	84	42	42	54
48	C	25	1	54	164	18	152
49	A	31	59	79	33	18	222
50	C	31	112	163	84	197	261
51	B	37	116	171	231	265	229
52	B	31	28	118	0	70	273
53	C	31	57	175	203	127	248
54	C	30	87	66	37	212	351
55	C	19	14	119	6	19	165
56	C	32	98	57	140	165	252
57	E	25	83	49	70	117	75
58	E	35	102	98	194	178	210
59	E	32	59	20	88	125	174
60	C	29	31	79	164	95	0
61	C	36	42	65	84	291	152
62	E	24	111	69	70	2	294
63	E	39	119	161	195	66	154
64	C	32	4	64	36	50	158
65	D	27	104	54	22	141	256
66	A	36	99	0	6	297	339
67	A	28	11	45	149	9	278
68	D	30	112	21	29	262	84
69	C	23	94	34	194	184	327
70	C	26	66	147	116	194	141
71	C	29	43	46	187	49	314
72	D	31	117	171	196	47	297
73	E	37	38	80	191	248	246
74	B	36	29	155	116	139	276
75	C	43	72	100	81	97	220
76	C	27	88	48	40	246	21
77	A	33	49	63	134	125	184
78	C	23	73	29	26	271	21
79	B	27	67	144	165	159	110
80	D	30	86	70	137	49	204
81	A	33	83	95	158	285	84
82	C	37	79	110	209	110	13
83	A	24	1	121	50	233	79
84	E	31	100	124	143	21	136
85	C	33	37	30	185	183	295
86	C	24	103	43	31	74	58
87	A	24	32	126	45	160	148
88	E	33	48	37	204	207	177
89	C	40	118	134	42	245	93
90	A	29	77	51	58	60	351
91	D	35	104	124	97	168	78
92	E	33	94	91	200	24	197
93	D	36	18	24	73	270	72
94	D	27	55	171	140	85	116
95	B	28	61	7	73	30	53
96	E	26	25	152	20	224	85

97	C	30	90	125	118	19	117
98	C	23	27	52	203	39	143
99	C	24	110	131	36	294	184
100	E	30	88	65	72	155	228

TABLE B.5: Generated data, for the customers of CPII - 200 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	E	23	62	268	248	160	245
2	A	35	11	246	206	481	360
3	E	29	123	72	443	327	273
4	B	32	223	135	114	467	471
5	B	28	207	14	463	390	437
6	E	38	38	331	62	540	257
7	E	37	122	292	12	184	691
8	C	25	144	193	181	527	63
9	D	31	201	219	57	326	673
10	E	28	56	305	119	400	53
11	B	31	71	246	13	396	714
12	E	26	36	74	464	569	426
13	C	25	154	343	227	395	120
14	A	28	0	32	0	75	360
15	C	28	218	46	347	300	209
16	E	25	70	160	421	544	102
17	C	26	19	152	94	595	459
18	A	29	158	177	36	341	258
19	D	29	86	137	444	367	0
20	B	24	171	220	237	538	634
21	E	29	119	63	89	68	490
22	C	31	115	142	58	479	617
23	B	41	99	180	90	347	262
24	E	32	185	309	75	418	679
25	C	24	64	102	424	336	570
26	B	31	101	150	432	32	231
27	C	27	110	207	216	377	353
28	E	31	24	214	417	203	209
29	A	25	166	16	134	378	537
30	A	27	142	257	445	464	690
31	A	31	159	213	118	552	459
32	E	27	45	3	158	198	695
33	E	33	183	340	466	193	317
34	E	23	11	33	419	540	336
35	E	38	60	254	292	209	44
36	C	32	171	129	362	238	501
37	E	32	214	163	360	441	539
38	D	33	182	322	243	473	423
39	A	26	41	83	190	215	675
40	B	23	163	72	307	92	150
41	C	32	17	2	252	419	120
42	A	33	60	24	216	282	217
43	E	32	217	27	48	492	314
44	E	31	16	44	407	353	146
45	C	39	131	127	48	540	378
46	E	34	162	97	270	220	577
47	E	29	179	346	426	300	285
48	C	25	36	89	96	490	268
49	E	38	123	261	296	191	408
50	D	29	214	191	86	86	653
51	B	34	74	70	462	80	526
52	C	24	29	33	460	561	270
53	C	34	211	311	62	125	466
54	E	33	74	87	16	499	53
55	B	27	123	75	448	399	12
56	A	37	42	345	179	277	120
57	E	21	138	169	92	283	256
58	C	38	2	354	56	547	469
59	D	38	221	277	166	399	237
60	B	33	179	192	332	122	435
61	E	30	130	19	414	376	241
62	C	33	240	16	138	278	405
63	C	35	138	15	14	272	625
64	C	28	239	287	372	5	522
65	E	25	197	279	27	135	56
66	C	36	1	15	412	55	18

67	C	34	61	106	193	583	273
68	D	29	67	354	370	158	430
69	E	27	204	103	291	560	709
70	E	37	172	212	2	451	368
71	C	27	75	218	202	562	653
72	C	19	162	234	437	353	120
73	C	31	113	16	477	80	335
74	C	30	224	129	100	85	422
75	E	30	231	93	145	172	671
76	E	26	66	213	468	353	712
77	C	24	160	91	56	144	248
78	E	32	98	171	292	482	548
79	A	29	64	55	327	285	291
80	E	28	91	258	200	62	191
81	D	30	142	304	1	355	290
82	C	24	226	243	211	376	42
83	C	34	137	6	185	383	499
84	E	24	70	198	340	512	58
85	A	30	216	213	433	92	200
86	E	31	118	27	194	514	288
87	D	30	171	247	205	93	18
88	C	41	182	190	453	528	181
89	D	23	219	316	311	486	383
90	C	24	195	2	116	185	377
91	B	34	194	88	208	377	284
92	C	36	211	254	92	411	503
93	E	30	156	50	317	341	137
94	D	35	113	285	132	554	381
95	E	35	188	98	64	516	370
96	B	28	111	36	322	296	531
97	A	19	66	189	274	243	338
98	B	32	155	331	105	200	550
99	C	43	143	18	26	472	582
100	B	24	155	90	131	313	140
101	E	35	227	290	388	42	42
102	C	34	160	152	16	340	116
103	C	35	66	332	51	577	9
104	C	38	193	72	202	407	556
105	A	34	162	17	42	374	469
106	B	36	154	357	37	351	514
107	A	31	156	245	98	353	266
108	D	31	116	198	419	358	393
109	C	24	30	34	86	583	65
110	E	30	22	228	193	470	169
111	E	33	11	56	268	381	1
112	D	33	122	262	202	232	9
113	C	24	235	142	442	120	589
114	E	33	12	119	189	75	31
115	C	35	162	319	402	49	312
116	D	27	209	44	30	35	431
117	E	20	203	74	385	585	10
118	D	28	104	210	292	24	549
119	D	35	16	46	391	478	622
120	B	24	148	95	325	575	18
121	C	34	58	238	42	487	72
122	D	32	51	96	17	471	621
123	A	25	60	319	325	80	109
124	B	22	84	249	347	122	314
125	C	23	172	294	408	274	705
126	C	34	39	239	290	237	75
127	A	23	15	161	312	407	61
128	C	35	123	244	472	257	513
129	A	27	149	49	412	119	533
130	C	32	174	54	91	521	404
131	E	27	64	346	48	191	701
132	C	20	27	151	434	506	617
133	C	39	129	162	323	225	78
134	C	34	185	127	438	474	253
135	C	35	196	205	356	8	41
136	C	29	12	334	266	565	623
137	A	38	232	0	431	516	328
138	C	36	183	98	12	593	543
139	D	26	12	246	78	253	567
140	A	28	165	78	351	477	718
141	B	29	47	169	472	484	455
142	C	26	37	76	54	262	615
143	C	38	14	298	120	509	379
144	C	32	201	340	126	195	603
145	C	31	2	234	231	257	282
146	E	30	117	335	470	11	272
147	B	35	16	272	357	335	473

148	B	36	191	79	474	292	48
149	D	19	37	257	74	66	506
150	D	31	16	356	439	197	625
151	C	39	6	41	253	0	479
152	C	22	151	232	256	440	248
153	E	24	41	227	415	372	528
154	D	20	144	211	320	103	634
155	B	25	70	180	394	565	156
156	D	29	70	126	147	419	149
157	C	25	142	276	307	423	563
158	E	26	184	316	433	535	226
159	D	26	143	47	459	62	542
160	A	26	200	29	405	286	281
161	C	29	3	74	377	509	665
162	A	33	7	63	199	133	410
163	C	36	36	336	410	149	475
164	A	28	59	19	467	261	349
165	D	24	35	290	258	292	495
166	C	31	176	309	421	458	490
167	C	29	23	120	430	191	360
168	E	28	43	257	408	559	231
169	E	26	70	195	27	263	365
170	C	28	172	192	203	492	44
171	C	26	37	185	150	376	276
172	C	21	38	33	188	537	4
173	C	29	75	48	122	515	181
174	C	40	130	70	394	401	31
175	B	35	168	162	369	380	707
176	E	28	111	290	71	360	717
177	B	22	158	193	150	285	437
178	B	23	133	339	262	579	618
179	C	30	20	169	149	202	164
180	C	26	70	191	61	366	107
181	E	37	101	104	149	238	432
182	D	28	203	16	242	271	147
183	C	29	110	232	132	491	156
184	C	25	108	336	304	163	70
185	C	29	235	92	421	168	18
186	E	28	101	293	342	344	50
187	C	30	209	280	189	589	265
188	B	24	28	295	291	392	672
189	A	31	68	292	128	77	54
190	C	36	120	73	338	132	695
191	C	21	234	107	338	589	249
192	E	27	219	231	255	207	389
193	E	27	64	357	360	302	688
194	C	33	91	174	455	116	165
195	C	36	12	276	200	248	580
196	D	28	219	352	211	18	365
197	A	34	107	222	370	268	543
198	A	25	185	347	26	363	379
199	C	32	164	151	202	461	329
200	C	29	179	74	5	140	29

Obtained Results

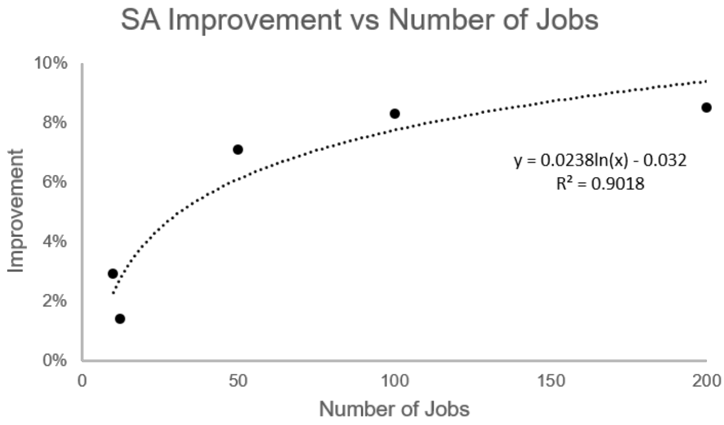

Figure B.1: SA improvements in CPII.

Table B.6: Processing and Waiting Times Results in CPII.

	Total Processing Time		Total Waiting Time	
Instance	DR	SA_{best}	DR	SA_{best}
n10R0.50	120	120	39	39
n10R0.75	120	120	22	22
n10R1.00	120	120	19	16
n10R1.25	120	120	3	3
n10R1.50	123	124	13	12
n15R0.50	180	179	118	111
n15R0.75	181	181	99	93
n15R1.00	179	179	33	33
n15R1.25	180	179	13	13
n15R1.50	179	179	24	24
n50R0.50	609	607	1350	1172
n50R0.75	606	607	924	811
n50R1.00	608	606	416	358
n50R1.25	612	609	531	441
n50R1.50	612	611	206	173
n100R0.50	1211	1207	6007	5008
n100R0.75	1213	1209	3413	2793
n100R1.00	1212	1211	2409	2070
n100R1.25	1214	1212	830	706
n100R1.50	1217	1218	689	559
n200R0.50	2395	2392	24691	20992
n200R0.75	2405	2398	13737	11479
n200R1.00	2398	2393	4376	3441
n200R1.25	2411	2408	4566	3811
n200R1.50	2421	2420	1702	1389

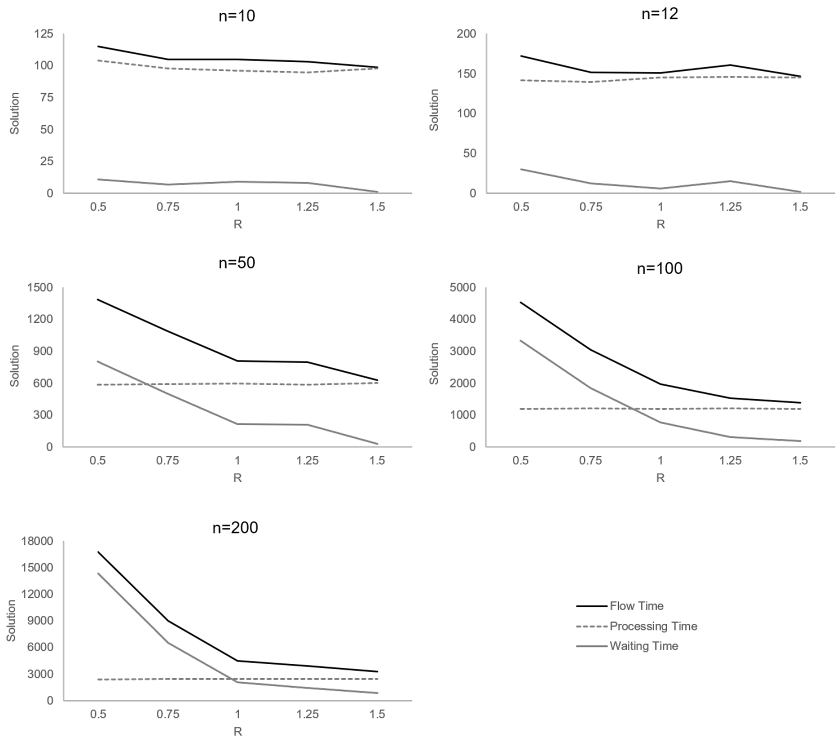

FIGURE B.2: Flow time behavior with R variation, using the SA results.

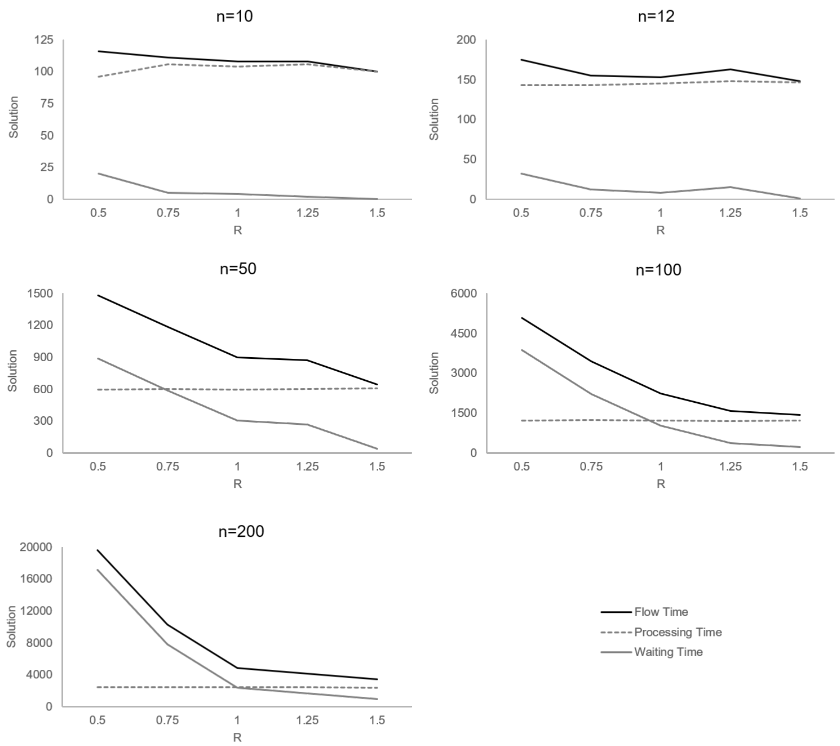

FIGURE B.3: Flow time behavior with R variation, using the DR results.

Table B.7: Running times of the computational tests in CPII.

Instance	MP	DR	SA	SA_{best}	SA_{avg}
n10R0.50	10	0	10	0	0
n10R0.75	20	0	10	0	0
n10R1.00	10	0	10	0	0
n10R1.25	10	0	10	0	0
n10R1.50	10	0	10	0	0
n15R0.50	2287	0	15	0	0
n15R0.75	2262	0	15	0	0
n15R1.00	2963	0	15	0	0
n15R1.25	200	0	15	0	0
n15R1.50	766	0	15	0	0
n50R0.50	-	0	50	6	21
n50R0.75	-	0	50	2	11
n50R1.00	-	0	50	3	12
n50R1.25	-	0	50	8	14
n50R1.50	-	0	50	4	7
n100R0.50	-	0	100	32	58
n100R0.75	-	0	100	76	73
n100R1.00	-	0	100	23	45
n100R1.25	-	0	100	88	74
n100R1.50	-	0	100	64	72
n200R0.50	-	0	200	153	102
n200R0.75	-	0	200	146	118
n200R1.00	-	0	200	182	148
n200R1.25	-	0	200	171	113
n200R1.50	-	0	200	91	120

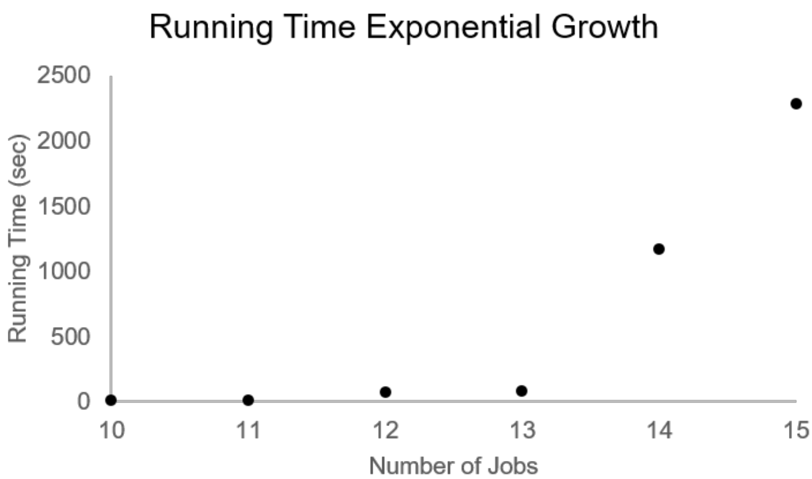

Figure B.4: Running time exponential behavior of the MP method.

Appendix C

Cement Plant III

Generated Instances

TABLE C.1: Generated data, for the customers of CPIII - 10 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	A	24	9	0	0	23	53
2	B	31	19	5	9	28	31
3	B	39	14	25	26	14	24
4	B	30	0	24	9	44	5
5	C	36	6	16	24	40	44
6	C	32	1	9	3	12	8
7	B	30	9	14	38	16	50
8	A	31	16	1	13	27	33
9	B	27	13	7	30	18	0
10	B	30	16	16	19	0	16

TABLE C.2: Generated data, for the customers of CPIII - 15 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	C	31	26	28	48	69	63
2	A	37	8	0	16	0	15
3	C	28	11	40	54	32	55
4	B	36	23	21	0	29	30
5	C	37	12	5	6	10	83
6	B	25	22	39	28	5	79
7	A	29	20	26	14	55	0
8	B	30	2	28	36	55	26
9	A	36	6	14	46	62	29
10	B	20	0	37	44	42	45
11	C	35	1	1	28	13	16
12	A	24	26	6	7	31	53
13	B	26	4	41	28	21	19
14	B	31	23	37	1	25	69
15	B	24	9	4	50	10	6

Table C.3: Generated data, for the customers of CPIII - 50 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	A	27	51	91	46	41	194
2	C	27	74	66	66	37	38
3	B	28	44	124	167	140	280
4	B	31	35	134	154	226	68
5	A	23	58	99	171	0	84
6	B	34	74	85	27	160	127
7	C	38	2	95	148	222	4
8	A	31	44	98	91	119	243
9	B	26	69	105	184	200	104
10	A	26	39	68	6	225	69
11	A	26	83	75	160	0	133
12	A	29	49	23	24	16	113
13	B	37	17	71	71	140	236
14	B	29	89	57	21	1	95
15	B	35	35	69	109	200	158
16	C	34	58	139	60	129	209
17	C	27	77	84	127	4	263
18	B	35	17	5	109	178	0
19	A	25	50	98	90	191	240
20	B	35	59	85	169	72	136
21	B	29	77	26	162	23	126
22	C	27	64	41	47	95	131
23	C	29	81	76	186	42	137
24	B	29	31	22	33	10	84
25	B	23	40	77	83	153	142
26	B	27	27	5	117	85	176
27	B	31	43	131	20	209	151
28	A	20	62	105	51	59	38
29	C	26	53	88	37	5	110
30	B	28	93	48	127	129	173
31	C	31	0	61	89	110	4
32	A	30	52	134	17	186	35
33	B	31	2	0	126	119	165
34	A	39	61	112	101	28	267
35	B	27	20	104	0	20	29
36	B	38	8	123	58	161	64
37	B	23	55	81	169	229	106
38	B	37	64	89	59	211	199
39	A	35	40	130	30	126	26
40	C	33	27	25	166	159	94
41	B	29	41	139	101	128	58
42	B	32	10	62	176	2	206
43	A	35	12	18	170	236	160
44	B	30	17	52	49	95	276
45	B	31	25	69	151	0	53
46	C	28	5	108	99	132	29
47	B	30	91	34	23	191	123
48	C	38	69	127	11	234	67
49	C	23	56	23	31	71	253
50	B	36	35	94	42	90	286

Table C.4: Generated data, for the customers of CPIII - 100 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	C	28	39	166	91	281	437
2	A	27	1	270	184	227	157
3	B	30	185	226	24	350	338
4	A	31	156	43	261	463	9
5	A	19	105	216	115	144	304
6	C	25	15	156	280	168	236
7	B	31	91	262	279	130	350
8	B	30	142	54	345	233	214
9	B	25	152	260	17	388	474
10	B	25	74	173	277	273	127
11	B	32	164	28	324	199	193
12	B	27	176	230	62	23	546
13	B	42	38	33	319	389	169
14	B	25	120	127	252	332	412
15	B	38	66	124	48	174	271

16	B	33	91	267	313	54	371
17	B	33	127	33	169	165	380
18	C	31	102	201	15	350	0
19	C	33	94	178	198	91	193
20	A	19	167	52	32	281	327
21	A	28	125	212	190	343	524
22	A	39	131	130	362	0	306
23	C	25	81	92	337	245	361
24	B	29	117	167	97	245	154
25	B	37	156	66	361	332	519
26	B	28	75	103	90	190	507
27	A	24	170	38	177	286	545
28	B	25	172	255	129	195	500
29	B	34	67	46	88	146	494
30	B	30	170	213	345	212	459
31	A	30	36	47	260	343	179
32	B	29	69	11	265	352	154
33	B	37	151	163	267	353	173
34	C	33	7	173	345	79	72
35	C	32	170	179	17	153	311
36	B	32	91	101	244	63	95
37	A	30	7	227	239	433	286
38	B	26	87	56	189	111	494
39	B	25	124	174	93	366	134
40	B	23	169	168	17	277	184
41	B	28	49	95	337	319	190
42	B	33	138	132	23	356	359
43	A	29	147	118	188	122	422
44	A	21	63	225	235	332	473
45	B	27	143	170	272	268	208
46	C	23	48	258	39	335	355
47	B	38	164	187	65	227	28
48	B	25	1	68	52	414	243
49	B	32	48	243	327	75	79
50	C	20	49	12	322	280	370
51	B	33	102	161	224	416	20
52	C	25	119	194	0	247	6
53	C	32	91	6	107	280	335
54	B	32	32	4	146	158	200
55	C	33	184	137	137	16	499
56	C	38	137	269	228	201	153
57	B	34	19	171	301	286	237
58	A	25	34	230	238	454	461
59	C	22	157	53	305	324	172
60	C	31	43	28	358	59	349
61	B	27	41	225	336	250	235
62	B	30	101	111	108	60	85
63	C	31	182	276	16	239	23
64	C	31	47	64	277	243	105
65	B	29	166	177	5	279	304
66	C	42	74	15	337	403	508
67	A	26	97	48	79	320	354
68	C	35	9	76	214	249	137
69	B	29	157	85	204	21	458
70	A	32	124	206	45	388	8
71	B	31	0	244	153	224	425
72	B	22	154	184	185	46	433
73	B	27	114	217	343	160	406
74	B	26	136	83	194	184	125
75	B	33	140	197	333	47	58
76	C	33	115	79	156	113	273
77	B	27	107	135	334	8	156
78	B	16	94	151	306	302	289
79	B	18	75	132	348	342	90
80	C	27	158	195	248	258	51
81	B	29	169	122	215	181	219
82	A	35	124	3	7	416	351
83	C	24	157	75	37	94	471
84	B	33	133	244	95	384	17
85	B	38	1	135	53	357	400
86	B	23	90	106	110	293	69
87	B	29	47	21	93	74	180
88	C	39	18	145	46	55	304
89	B	33	82	0	335	158	346
90	A	22	40	109	7	263	69
91	B	31	119	111	357	86	43
92	C	41	23	91	325	56	160
93	A	25	156	243	208	166	250
94	B	37	73	58	351	313	82
95	A	26	112	265	273	338	228
96	A	35	8	0	91	428	175

97	A	33	173	242	171	365	400
98	B	23	173	54	288	392	86
99	A	29	103	260	293	116	245
100	A	34	103	161	9	60	383

TABLE C.5: Generated data, for the customers of CPIII - 200 jobs instance.

id	m_j	q_j	r_j (R0.50)	r_j (R0.75)	r_j (R1.00)	r_j (R1.25)	r_j (R1.50)
1	B	33	372	346	650	631	445
2	B	26	172	395	84	294	815
3	B	39	20	198	384	2	923
4	B	30	85	322	496	257	595
5	A	24	355	23	128	53	450
6	C	28	92	354	375	484	208
7	C	29	148	78	58	517	755
8	C	29	47	246	416	342	1092
9	B	19	201	427	235	452	771
10	A	32	214	273	387	907	1100
11	A	30	7	27	3	6	593
12	B	31	234	69	6	71	922
13	B	34	167	406	476	461	1049
14	A	36	47	278	754	766	662
15	A	31	61	132	480	639	480
16	C	29	197	138	53	801	496
17	B	32	119	260	330	785	149
18	B	27	30	360	223	457	268
19	B	30	215	340	240	931	652
20	C	21	182	5	495	677	454
21	B	32	4	559	262	444	758
22	A	30	362	249	22	918	503
23	B	16	325	311	426	688	511
24	B	29	228	374	726	587	841
25	A	28	320	432	139	181	487
26	B	28	221	287	628	756	901
27	C	33	277	353	516	141	714
28	A	34	164	82	726	633	190
29	B	34	235	336	555	669	507
30	A	29	181	412	581	304	129
31	B	35	80	487	452	385	454
32	A	36	342	415	535	371	921
33	C	24	49	146	655	953	1052
34	C	33	59	463	432	146	971
35	A	35	222	194	422	237	1101
36	B	26	39	465	605	196	303
37	A	27	282	469	249	183	977
38	B	30	374	78	700	391	844
39	B	43	49	513	160	465	218
40	A	32	262	424	549	788	1063
41	B	20	216	324	595	357	688
42	C	27	17	236	20	714	1017
43	B	40	172	71	756	883	922
44	A	21	209	2	343	882	141
45	A	25	109	238	108	634	269
46	B	30	246	103	41	732	287
47	B	25	142	2	57	324	1050
48	C	32	274	200	750	848	35
49	A	32	292	453	112	171	178
50	C	21	317	457	175	725	510
51	A	40	82	84	61	349	903
52	C	27	80	242	96	810	106
53	B	36	214	401	0	42	963
54	B	24	227	440	293	348	114
55	B	34	190	486	444	590	946
56	A	33	191	210	301	790	741
57	B	28	253	105	761	39	145
58	C	29	107	56	640	153	704
59	A	27	303	110	477	68	599
60	B	27	225	17	92	261	797
61	B	34	137	431	63	145	9
62	B	35	218	359	122	677	559
63	C	38	329	141	395	46	579
64	B	29	99	95	597	738	499
65	B	18	258	540	182	556	242
66	B	38	327	481	702	83	139

67	A	21	270	336	620	662	365
68	B	23	180	181	465	154	560
69	C	30	89	435	428	416	680
70	B	35	76	176	610	471	329
71	A	38	293	127	683	366	151
72	B	27	193	423	347	911	486
73	A	26	3	23	718	704	800
74	B	26	295	106	246	794	80
75	C	31	306	137	87	6	7
76	B	31	313	61	29	155	682
77	C	36	320	270	280	838	451
78	B	21	350	464	401	907	685
79	A	30	283	41	644	400	1007
80	C	32	80	181	355	520	914
81	B	33	30	190	439	213	0
82	B	31	284	551	532	442	313
83	B	35	24	483	222	65	1040
84	C	26	100	41	155	273	930
85	A	35	159	447	78	7	926
86	B	32	183	472	558	721	1009
87	A	24	353	359	524	181	459
88	B	29	246	402	327	141	384
89	A	31	24	355	419	122	420
90	C	31	123	80	62	106	261
91	B	32	230	74	354	396	1017
92	C	28	96	459	660	173	418
93	B	39	167	238	33	71	626
94	B	27	61	357	677	605	868
95	B	37	220	339	79	258	610
96	A	34	305	79	8	849	578
97	A	35	221	551	565	573	732
98	C	32	333	60	161	791	831
99	B	32	266	203	287	421	685
100	C	24	299	299	31	52	350
101	B	23	115	463	224	266	589
102	B	30	194	325	87	499	480
103	B	27	287	35	338	181	11
104	A	30	177	508	21	109	134
105	B	26	179	32	165	648	230
106	B	41	62	239	266	68	54
107	B	23	11	457	582	852	909
108	B	27	304	271	733	190	566
109	A	23	88	0	27	175	903
110	A	32	293	126	304	379	762
111	B	33	13	69	93	955	460
112	C	41	351	187	116	755	780
113	C	32	289	453	63	256	44
114	B	30	232	155	623	847	983
115	A	32	5	259	623	442	893
116	A	26	65	492	399	143	270
117	A	29	91	452	737	230	831
118	B	27	344	67	281	684	626
119	C	34	178	332	160	878	106
120	A	30	90	467	211	395	347
121	B	28	257	471	319	206	1029
122	C	33	32	326	79	570	707
123	C	31	95	487	629	535	6
124	B	35	138	249	409	463	621
125	B	32	97	493	572	405	244
126	C	26	102	80	587	453	772
127	A	28	202	480	60	163	532
128	B	32	96	13	367	639	336
129	C	38	322	232	276	486	662
130	B	27	169	345	507	221	842
131	B	34	35	547	681	132	900
132	B	24	248	441	339	202	1143
133	C	33	137	101	563	0	39
134	A	34	347	292	385	877	120
135	A	31	123	502	170	861	152
136	B	38	37	22	477	575	383
137	C	28	220	154	250	582	445
138	B	34	214	215	173	489	291
139	B	26	145	127	619	682	95
140	C	23	1	23	589	796	301
141	B	23	54	246	525	421	392
142	B	28	162	246	268	792	1045
143	B	33	314	212	127	183	424
144	C	29	157	560	444	479	343
145	B	24	117	362	255	878	775
146	B	40	40	312	667	450	260
147	B	25	97	461	32	429	612

148	B	32	238	324	99	929	263
149	C	23	77	336	514	390	316
150	C	26	215	191	200	475	330
151	C	24	98	537	406	283	693
152	C	27	57	154	55	338	805
153	C	20	51	193	348	334	575
154	A	28	331	562	458	693	763
155	A	39	46	456	654	401	535
156	A	34	10	483	704	654	851
157	B	27	287	360	77	894	819
158	B	30	125	442	235	258	457
159	B	25	255	266	151	854	79
160	C	24	12	355	123	909	863
161	B	28	193	84	313	46	518
162	C	33	271	551	478	321	162
163	B	32	279	405	390	922	171
164	C	36	320	174	431	346	1132
165	C	29	133	215	668	635	129
166	B	57	306	512	186	718	850
167	C	36	192	373	476	502	675
168	A	32	231	274	245	569	60
169	C	33	0	66	200	203	638
170	B	30	78	569	179	318	35
171	B	33	144	571	524	152	125
172	C	35	48	386	350	250	573
173	A	30	297	252	470	257	604
174	B	31	203	345	542	575	223
175	B	34	165	328	185	155	506
176	B	26	356	142	136	452	1069
177	B	32	109	70	199	43	389
178	C	35	321	139	233	791	387
179	A	36	96	424	704	449	675
180	B	32	177	521	375	644	922
181	B	28	8	178	85	920	790
182	A	20	222	169	18	2	801
183	B	40	80	257	36	834	468
184	B	30	345	24	658	585	390
185	A	31	53	291	607	822	1152
186	C	31	325	361	318	731	195
187	C	30	337	257	189	425	136
188	B	21	193	470	150	661	733
189	B	33	150	322	624	115	307
190	C	37	68	143	508	685	576
191	A	24	275	284	187	572	1039
192	C	31	75	65	730	452	542
193	A	42	243	482	350	237	817
194	B	34	274	270	473	304	982
195	B	29	271	565	719	584	289
196	B	24	102	165	431	614	995
197	C	28	286	100	360	93	266
198	B	33	4	354	218	726	124
199	B	34	33	525	41	116	280
200	A	24	319	502	87	179	1056

Obtained Results

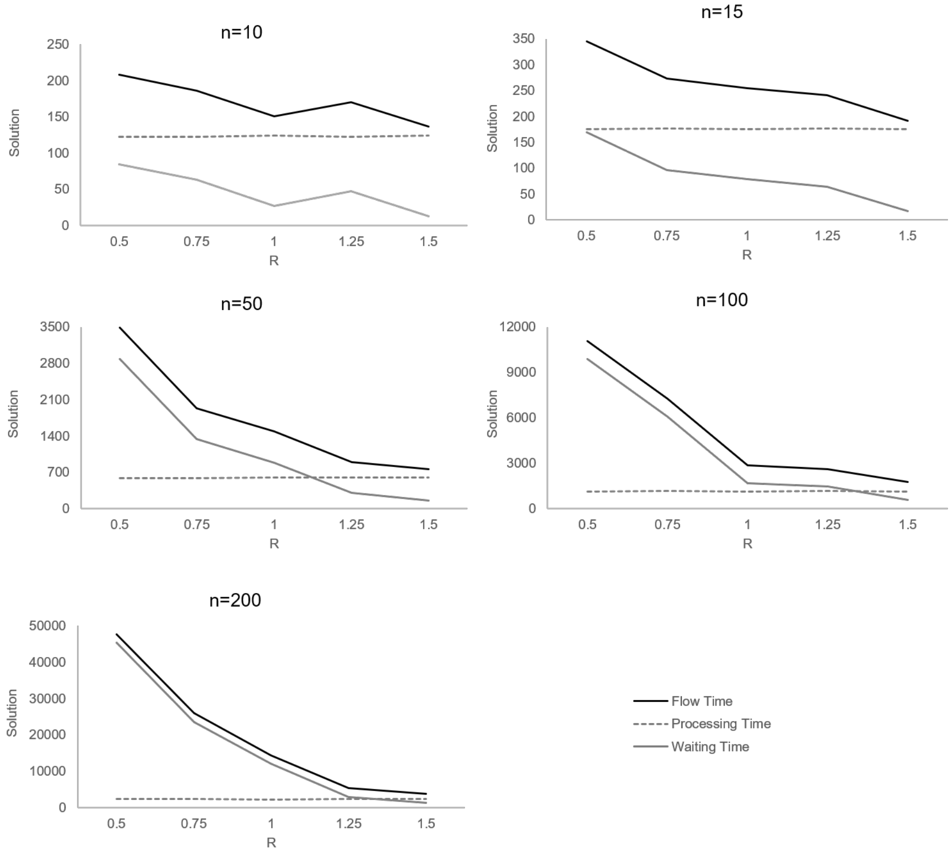

FIGURE C.1: Flow time behavior with R variation, using the DR results.

Bibliography

Afsar, J. (2012), 'Cement manufacturing process', `http://www.engineeringintro.com/uncategorized/cement-manufacturing-process/`. Accessed: 2018-07-07.

Agudelo, I. (2009), Supply Chain Management in the Cement Industry Master of Engineering in Logistics, Master's thesis, Massachusetts Institute of Technology.

Al-Betar, M. A. (2017), 'β-hill climbing: an exploratory local search', *Neural Computing and Applications* **28**(1), 153–168.

Ángel-Bello, F., Álvarez, A., Pacheco, J. and Martínez, I. (2011), 'A single machine scheduling problem with availability constraints and sequence-dependent setup costs', *Applied Mathematical Modelling* **35**(4), 2041–2050.

Baker, K. R. and Trietsch, D. (2009), *Principles of Sequencing and Scheduling*, John Wiley.

Ballou, R. H. (2004), *Business Logistics/supply Chain Management: Planning, Organizing, and Controlling the Supply Chain*, Pearson/Prentice Hall.

Barbosa, E., Costa, M. G., Fátima, M., Souza, D. and Pereira, F. H. (2010), 'Evaluation of Production Sequencing Rules in Job Shop and Flow Shop Environment through Computer Simulation', *Icieom* pp. 1–12.

Beasley, J. E. (2018), 'Or–library', `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`. Accessed: 2018-07-30.

Blazewicz, J., Informatyki, I., Poznahska, P., Dror, M., Weglarz, J., Informatyki, I. and Poznahska, P. (1991), 'Mathematical programming formulations for machine scheduling : A survey', *European Journal of Operational Research* **51**, 283–300.

Bowman, E. H. (1959), 'The Schedule-Sequencing Problem', *Operations Research* **7**(5), 621–624.

Brucker, P. and Knust, S. (2006), *Complex Scheduling*, Srpinger.

Campbell, H. G., Dudek, R. A. and Smith, M. L. (1970), 'A Heuristic Algorithm for The n Job, m Machine Sequencing Problem', *Management science* **16**(10), B630– B637.

Cembureau (2017), `https://cembureau.eu/`. Accessed: 2018-07-07.

Cement Plant Layout (2018), `http://cemtechsanghi.com/index.php/current-project/cement-plant-layout`. Accessed: 2018-07-07.

Černý, V. (1985), 'Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm', *Journal of Optimization Theory and Applications* **45**(1), 41–51.

Cheng, T. C. E. and Sin, C. C. S. (1990), 'A state-of-the-art review of parallel-machine scheduling research', *European Journal of Operational Research* **47**, 271–292.

Chiang, T. C. and Fu, L. C. (2007), 'Using dispatching rules for job shop scheduling with due-date based objectives', *International Journal of Production Research* **45**(May), 3245–3262.

Christopher, M. (2016), *Logistics and supply chain management*, Vol. 4, FT Press.

CSCMP Glossary (2013), `https://cscmp.org/CSCMP/Educate/SCM_Definitions_and_Glossary_of_Terms/`. Accessed: 2018-05-12.

Daugherty, K. E. (1973), 'Cement as an indicator of economic development', *Cement and Concrete Research* **3**(2), 163–183.

Demir, Y. and Kürat Ileyen, S. (2013), 'Evaluation of mathematical models for flexible job-shop scheduling problems', *Applied Mathematical Modelling* **37**(3), 977–988.

Dowsland, K. A. and Thompson, J. M. (2012), *Simulated Annealing*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1623–1655.

Du, K. L. and Swamy, M. N. (2016), *Search and optimization by metaheuristics: Techniques and algorithms inspired by nature*, Birkhäuser Basel.

Eglese, R. W. (1990), 'Simulated annealing: A tool for operational research', *European Journal of Operational Research* **46**(3), 271–281.

Fang, K., Uhan, N., Zhao, F. and Sutherland, J. W. (2011), 'A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction', *Journal of Manufacturing Systems* **30**(4), 234–240.

Framinan, J. M., Leisten, R. and García, R. R. (2014), *Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools*, Springer-Verlag London.

French, S. (1982), *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*, Ellis Horwood.

Garey, M. R. and Johnson, D. S. (1979), 'Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)', *Computers and Intractability* p. 340.

Graham, R. L., Lawler, E. L., K., L. J. and Rinnooy Kan, A. H. G. (1979), 'Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey', *Annals of Discrete Mathematics* **5**, 287–326.

Guo, Z. X., Wong, W. K., Leung, S. Y., Fan, J. T. and Chan, S. F. (2006), 'Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: A case study based on the apparel industry', *Computers and Industrial Engineering* **50**(3), 202–219.

Henderson, D., Jacobson, S. H. and Johnson, A. W. (2003), *The Theory and Practice of Simulated Annealing*, Vol. 57, Springer, Boston, MA, pp. 287–319.

Hugos, M. (2011), *Essentials of supply chain management*, 3rd edn, Wiley.

Jackson, J. R. (1955), Scheduling a production to minimize maximum tardiness, Technical report, University of California, Los Angeles.

Johannes Schneider, S. K. (2006), *Stochastic optimization*, Scientific computation, 1 edn, Springer.

Johansson, H. J., McHugh, P., Pendlebury, A. J. and Wheeler, W. A. (1993), *Business process reengineering : breakpoint strategies for market dominance*, Vol. 27, Wiley.

Johnson, S. M. (1954), 'Optimal two- and three-stage production schedules with setup times included', *Naval Research Logistics Quarterly* **1**(1), 61–68.

Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P. and Werner, F. (2006), 'Sequencing and Tabu Search Heuristics for Hybrid Flow Shops with Unrelated Parallel Machines and Setup Times', *Proceedings of the 7th Asia Pacific Industrial Engineering and Management Systems Conference* pp. 1330–1342.

Kee-hung, L. and Cheng, T. (2009), *Just-in-Time Logistics*, Gower Publishing.

Kim, D. W., Kim, K. H., Jang, W. and Frank Chen, F. (2002), 'Unrelated parallel machine scheduling with setup times using simulated annealing', *11th International Conference on Flexible Manufacturing* **18**(3-4), 223–231.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983), 'Optimization by Simulated Annealing', *Science, Number 4598, 13 May 1983* **220**(4598), 671–680.

Koulamas, C., Antony, S. and Jaen, R. (1994), 'A survey of simulated annealing applications to operations research problems', *Omega* **22**(1), 41–56.

Lee, H. L., Padmanabhan, V. and Whang, S. (1997), 'Information Distortion in a Supply Chain: The Bullwhip Effect', *Management Science* **43**(4), 546–558.

Leung, J. Y. (2004), *Handbook of Scheduling: Algorithms, Models and Perfomance Analysis*, Chapman and Hall CRC.

Levitt, T. (1969), *The marketing mode: pathways to corporate growth*, McGraw-Hill.

Lin, Y.-K. and Hsieh, F.-Y. (2014), 'Unrelated parallel machine scheduling with setup times and ready times', *International Journal of Production Research* **52**(4), 1200–1214.

Low, C. (2005), 'Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines', *Computers and Operations Research* **32**(8), 2013–2025.

Manne, A. S. (1960), 'On the job-shop scheduling problem', *Operations Research* **8**(2), 219–223.

Mart, R. and Reinelt, G. (2011), *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*, Applied Mathematical Sciences 175, 1 edn, Springer-Verlag, Berlin, Heidelberg.

McNaughton, R. (1959), 'Scheduling with Deadlines and Loss Functions', *Management Science* **6**, 1–12.

Mentzer, J. T., Keebler, J. S., Nix, N. W., Smith, C. D. and Zacharia, Z. G. (2001), 'Defining Supply Chain Management', *Journal of Business Logistics* **22**(2), 1–25.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953), 'Equation of state calculations by fast computing machines', *J. Chem. Phys.* **21**(6), 1087–1092.

Moore, J. M. (1968), 'An n Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs', *Management Science* **15**(1), 102–109.

Mukhopadhyay, S. (2015), *Production Planning and Control: Text and Cases*, 3rd edn, Prentice-Hall of India.

Na, D.-G., Kim, D. W., Jang, W. and Chen, F. F. (2006), Scheduling unrelated parallel machines to minimize total weighted tardiness, *in* '2006 IEEE International Conference on Service Operations and Logistics, and Informatics', pp. 758–763.

Naik, T. R. (2005), 'Sustainability of Cement and Concrete Industries', *Achieving Sustainability in Construction* pp. 141–150.
URL: *http://www.icevirtuallibrary.com/doi/10.1680/asic.34044.0017*

Niemelä, J. (2016), Outbound Transportation Management Analysis, Master's thesis, Tampere University of Applied Sciences.

Noche, B. and Elhasia, T. (2013), 'Approach to innovative supply chain strategies in cement industry ; Analysis and Model simulation', *Physics Procedia* **75**, 359–369.

Optsicom Project (n.d.), `http://www.optsicom.es`. Accessed: 2018-07-30.

Panwalkar, S. S. and Iskander, W. (1977), 'A Survey of Scheduling Rules', *Operations Research* **25**(1), 45–61.

Picard, J.-C. and Queyranne, M. (1978), 'The Time-Dependent Traveling Salesman Problem and Its Application to the Tardiness Problem in One-Machine Scheduling', *Operations Research* **26**(1), 86–110.

Pinedo, M. L. (2012), *Scheduling: Theory, Algorithms, and Systems*, 4 edn, Springer-Verlag, New York.

Pinedo, M. and Reed, J. (2013), 'The "least flexible job first" rule in scheduling and in queueing', *Operations Research Letters* **41**(6), 618–621.

Potts, C. N. (1980), 'Technical Note Analysis of a Heuristic for One Machine Sequencing with Release Dates and Delivery Times', *Operations Research* **28**(6), 1436–1441.

Potts, C. N. and Van Wassenhove, L. N. V. (1991), 'Single Machine Tardiness Sequencing Heuristics', *IIE Transactions (Institute of Industrial Engineers)* **23**(4), 346–354.

Raaymakers, W. H. and Hoogeveen, J. A. (2000), 'Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing', *European Journal of Operational Research* **126**(1), 131–151.

Rajakumar, S., Arunachalam, V. P. and Selladurai, V. (2004), 'Workflow balancing strategies in parallel machine scheduling', *International Journal of Advanced Manufacturing Technology* **23**(5-6), 366–374.

Rinnooy Kan, A. H. G. (1976), *Machine Scheduling Problems: Classification, Complexity and Computations*, 1 edn, Springer US.

Ritzman, L. and Krajewski, L. (2002), *Foundations of Operations Management*, Pearson.

Rodrigues, F. A. and Joekes, I. (2011), 'Cement industry: Sustainability, challenges and perspectives', *Environmental Chemistry Letters* **9**(2), 151–166.

Rushton, A., Croucher, P. and Baker, P. (2014), *The Handbook of Distribution Management*, 5th edn, Kogan Page.

Selim, T. H. and Salem, A. S. (2011), 'Global Cement Industry: Competitive and Institutional Frameworks', *Journal of Strategic Management Education* **7**, 181–200.

Simchi-Levi, D., Kaminsky, P. and Simchi-Levi, E. (1999), *Designing and managing the supply chain: concepts, strategies, and case studies*, McGraw-Hill/Irwin.

SLV Cement by Cachapuz Bilanciai Group (n.d.), `http://www.slvcement.com/EN/`. Accessed: 2018-07-07.

Smith, W. E. (1956), 'Various optimizers for single stage production', *Naval Research Logistics Quarterly* **3**(1-2), 59–66.

SOA (n.d.), http://soa.iti.es/problem-instances.

Suman, B. and Kumar, P. (2006), 'A survey of simulated annealing as a tool for single and multiobjective optimization', *Journal of the Operational Research Society* **57**(10), 1143–1160.

Swamidass, P. M. (2000), *Encyclopedia of Production and Manufacturing Management*, Kluwe Academic Publishers.

Teixeira, F. d. O., Oliveira, L. and Varela, L. R. (2014), 'Comparative analysis of scheduling rules through arena for parallel machines', *2014 6th World Congress on Nature and Biologically Inspired Computing, NaBIC 2014* pp. 170–175.

Thomas, O. M. and Lea, F. M. (2018), 'Cement: Building material', `https://www.britannica.com/technology/cement-building-material`. Accessed: 2018-07-07.

Understanding Cement (n.d.), `https://www.understanding-cement.com/`. Accessed: 2018-07-07.

Unlu, Y. and Mason, S. J. (2010), 'Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems', *Computers and Industrial Engineering* **58**(4), 785–800.

Van Laarhoven, P. J. M. and Aarts, E. H. L. (1987), *Simulated Annealing: Theory and Applications*, Springer Netherlands.

Varela, M. L. R. (2007), Uma Contribuição para o Escalonamento da Produção baseado em Métodos Globalmente Distribuídos Uma Contribuição para o Escalonamento da Produção, PhD thesis, University of Minho.

Varela, M. L. R., Aparício, J. N. and do Carmo Silva, S. (2003), A scheduling web service based on xml-rpc.

Wagner, H. M. (1959), 'An Integer Linear Programming Model for Machine Scheduling', *Naval Research Logistics Quarterly* **6**(2), 131–140.

Weng, M. X., Lu, J. and Ren, H. (2001), 'Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective', *International Journal of Production Economics* **70**(3), 215–226.

Williamson, D. P. and Shmoys, D. B. (2011), *The design of approximation algorithms*, Cambridge University Press.

Yildiz, O. T., zlem Aslan, Alpaydin, E. and Coello, C. A. C. (2011), *Learning and Intelligent Optimization*, Lecture Notes in Computer Science 6683, 1 edn, Springer-Verlag Berlin Heidelberg.

Zhu, Z. and Heady, R. B. (2000), 'Minimizing the sum of earliness / tardiness in multi-machine scheduling : a mixed integer programming approach', *Computers & Industrial Engineering* **38**, 297–305.