

1 2 9 0



UNIVERSIDADE D  
COIMBRA

Sophia de Queiroz Soares

**ESCALONAMENTO DA PRODUÇÃO NUM  
SISTEMA JOB-SHOP FLEXÍVEL:  
APLICAÇÃO AO FABRICO DE PAVIMENTOS DE  
MADEIRA**

**Dissertação no âmbito do Mestrado em Engenharia e Gestão Industrial, orientada pelo Professor Doutor Cristóvão Silva e pelo Professor Pedro Miguel Fernandes Coelho e apresentada ao departamento de Engenharia Mecânica da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.**

Julho de 2021





FCTUC FACULDADE DE CIÊNCIAS  
E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA

DEPARTAMENTO DE  
ENGENHARIA MECÂNICA

# **Escalonamento da produção num sistema job-shop flexível: aplicação ao fabrico de pavimentos de madeira**

Dissertação apresentada para a obtenção do grau de Mestre em Engenharia e Gestão Industrial

## **Flexible job-shop scheduling: application to wood floor production**

**Autor**

**Sophia de Queiroz Soares**

**Orientadores**

**Professor Doutor Cristóvão Silva**

**Professor Pedro Coelho**

**Júri**

**Presidente** Professora Doutora Aldora Gabriela Gomes Fernandes  
Professora Auxiliar da Universidade de Coimbra

**Vogais** Professor Doutor Samuel de Oliveira Moniz  
Professor Auxiliar da Universidade de Coimbra

**Orientador** Professor Doutor Cristóvão Silva  
Professor Associado da Universidade de Coimbra

**Colaboração Institucional**

---



*SimpleAxis*

**Coimbra, Julho, 2021**







## Agradecimentos

Primeiramente, quero deixar um grande agradecimento à toda a minha família que é o meu alicerce. Aos meus pais e irmãos, por todo o incentivo, motivação, educação e exemplo que são para mim. Um obrigada especial à minha mãe por toda a paciência, dedicação e auxílio na realização deste trabalho.

Ao meu orientador, Professor Doutor Cristóvão Silva pelo acompanhamento e direcionamento em busca dos melhores caminhos e ao meu coorientador Professor Pedro Coelho por todos os ensinamentos, acompanhamento e atenção que disponibilizou. Também a todos os professores que fizeram parte do meu percurso académico e formaram a base fundamental da minha trajetória no curso.

À *SimpleAxis*, pela oportunidade de me acolher neste estágio curricular, em especial ao Doutor Luís Macedo, ao Doutor Miguel Castro e à Doutora Ana Catarina Leite, pelo tempo disponibilizado e por todo o interesse demonstrado desde o início para me auxiliar na procura das melhores soluções.

Um grande obrigada ao Leonardo Vizzotto, que foi responsável pelo auxílio na implementação do modelo, por toda atenção e esforço imensurável que dedicou ao meu projeto.

Por último, mas não menos importante, quero deixar um profundo agradecimento a todos os meus amigos que se fizeram presente ao longo desta trajetória. Aos meus amigos da UENF, que iniciaram o percurso académico comigo, aos meus colegas de curso, aos amigos do BEST Coimbra, que me proporcionaram experiências e oportunidades incríveis, em especial ao XXVIII Board. Aos amigos de Coimbra que se tornaram a minha família longe de casa, aos meus padrinhos, aos amigos de Campos que enfrentaram o desafio de vir morar em Portugal e devolveram o sentido de casa aqui. À Ana e à Georgia que estiveram comigo durante todo o semestre e me deram todo o suporte e energia que eu precisava para finalizar este trabalho.

A todos, todas as palavras não serão suficientes para enaltecer o quão grata estou por toda a ajuda que me prestaram nesta jornada, conduzindo, orientando e colocando desafios, a todos o meu muito obrigada!





## Resumo

O trabalho desenvolvido nesta dissertação resulta de um estágio curricular realizado na *startup SimpleAxis*, em Portugal, inserido no Mestrado em Engenharia e Gestão Industrial da Universidade de Coimbra. O seu objetivo principal consistiu em propor uma solução para escalonar as encomendas da empresa *Castro Wood Floors*, que se dedica à produção de pavimentos de madeira, minimizando o *makespan* e os atrasos nas entregas de suas encomendas, de forma a otimizar o processo de produção criando novas oportunidades de desenvolvimento para a empresa.

Deste modo, inicialmente foi feita a caracterização detalhada do problema e das suas restrições e, tendo-se concluído que este se poderia caracterizar-se como um *job shop* flexível, foi então realizada uma revisão da literatura de modo a identificar o método mais adequado para a resolução do problema.

A revisão da literatura permitiu concluir que um dos métodos mais utilizados na resolução de problemas do tipo *job shop* flexível é o Algoritmo Genético. Note-se que a *SimpleAxis* tinha já desenvolvido um sistema de escalonamento para a *Castro Wood Floors*, baseado num Algoritmo Genético e pretendia-se com este trabalho testar soluções alternativas que permitissem obter resultados com menor tempo de computação.

A implementação do Algoritmo Genético desenvolvido foi elaborada em linguagem de programação *Python* e testada para que se pudessem encontrar os parâmetros que trouxessem os melhores resultados para diferentes instâncias. Os melhores resultados foram analisados e comparados com os resultados obtidos pelo algoritmo criado pela *startup SimpleAxis*, para validar a capacidade do modelo desenvolvido.

A partir desta comparação, foi possível concluir que o modelo desenvolvido no âmbito desta dissertação é eficiente e não só gerou resultados melhores do que os alcançados pelo modelo desenvolvido pela *SimpleAxis*, como também o fez com tempos de computação menores tendo, portanto, alcançado os objetivos inicialmente propostos.

**Palavras-chave:** Escalonamento, Job-shop flexível, Algoritmo Genético, *Makespan*, *Tardiness*, Multiobjetivo.



## Abstract

This work is based on a curricular internship at the startup SimpleAxis, in Portugal, as part of the Masters in Industrial and Management Engineering at the University of Coimbra. Its main objective is to propose a solution related to the production scheduling of Castro Wood Floors enterprise, in order to minimize the makespan and the tardiness in deliveries to optimize the production process creating new opportunities of development to the company.

Deste modo, inicialmente foi feita a caracterização detalhada do problema e das suas restrições e, tendo-se concluído que este poderia caracterizar-se como um *job shop* flexível, foi então realizada uma revisão da literatura de modo a identificar o método mais adequado para a resolução do problema.

In this sense, initially, a detailed characterization of the problem and of its restrictions was carried out, showing that it could be characterized as a flexible job shop. Then a literature review was made to identify the most suitable method to solve it.

The literature review made it possible to conclude that one of the most used methods to solve flexible job shop problems is the Genetic Algorithm. It is important to highlight that SimpleAxis had already developed a scheduling system of its own to Castro Wood Floors, based on a Genetic Algorithm. In this sense, this work presents an alternative solution focusing on achieving better results with less computing time.

The Genetic Algorithm was implemented in *Python* programming language and tested to find the parameters that would achieve the best results for different instances. The best results were analyzed and compared with those obtained by the algorithm used by SimpleAxis, to validate the capacity of the developed model.

The comparison between the two algorithms, led to the conclusion that the model presented in this paper is efficient and not only produced better results than those achieved by SimpleAxis, but also ran in shorter computing time, meeting the initially proposed goals.

**Keywords** Scheduling, Flexible job-shop, Genetic Algorithm, Makespan, Tardiness, Multiobjective.



---

## Índice

Índice de Figuras .....	xi
Índice de Tabelas .....	xii
Siglas .....	xiii
1. Introdução .....	1
2. Caso de estudo .....	3
2.1. <i>SimpleAxis</i> .....	3
2.2. <i>Castro Wood Floors</i> .....	3
2.3. Descrição do problema .....	4
3. Revisão da literatura .....	7
3.1. Escalonamento .....	7
3.1.1. Caracterização do problema de escalonamento .....	8
3.1.2. Classes de escalonamento .....	10
3.2. <i>Job Shop</i> Flexível .....	10
3.3. Métodos de escalonamento .....	11
3.3.1. Formulações matemáticas .....	13
3.3.2. <i>Branch and bound</i> .....	14
3.3.3. Regras de despacho .....	14
3.3.4. <i>Shifing Bottleneck Procedure</i> .....	16
3.3.5. <i>Tabu seach</i> .....	16
3.3.6. <i>Simulated Annealing</i> .....	16
3.3.7. Algoritmos Genéticos .....	17
3.3.8. Discussão .....	22
4. Metodologia .....	25
4.1. Preparação dos dados .....	25
4.2. Desenvolvimento do modelo .....	27
4.2.1. Parâmetros de entrada .....	28
4.2.2. Inicialização e codificação da população .....	29
4.2.3. Avaliação da aptidão .....	30
4.2.4. Operadores Genéticos .....	31
4.2.5. Critério de parada .....	33
5. Resultados .....	35
5.1. Escolha dos parâmetros .....	35
5.2. Comparação dos resultados .....	38
6. Conclusões .....	43
REFERÊNCIAS BIBLIOGRÁFICAS .....	45
ANEXO A .....	51
ANEXO B .....	52

---

ANEXO C .....	53
ANEXO D .....	54
ANEXO E .....	55
ANEXO F .....	56
ANEXO H .....	60
ANEXO I .....	68
ANEXO J .....	71

---

## ÍNDICE DE FIGURAS

Figura 2.1. Pisos de madeira produzidos pela Castro <i>Wood Floors</i> .....	4
Figura 2.2. Possíveis seqüências de produção.....	5
Figura 3.1. Classificação para algoritmos de escalonamento (Amjad et al., 2018).....	12
Figura 3.2. Fluxograma de um Algoritmo Genético.....	18
Figura 4.1. Estrutura dos dados para entrada no algoritmo.....	27
Figura 4.2. Representação do vetor OS com a ordem das operações.....	29
Figura 4.3. Representação do vetor MS com escolha das máquinas para cada operação.....	30
Figura 5.1. Resultados para 20 <i>jobs</i> .....	39
Figura 5.2. Resultados da <i>SimpleAxis</i> para 20 <i>jobs</i> .....	39
Figura 5.3. Resultados para 50 <i>jobs</i> .....	40
Figura 5.4. Resultados da <i>SimpleAxis</i> para 50 <i>jobs</i> . ....	40
Figura 5.5. Resultados para 100 <i>jobs</i> . ....	41
Figura 5.6. Resultados da <i>SimpleAxis</i> para 100 <i>jobs</i> . ....	41

## ÍNDICE DE TABELAS

Tabela 2.1. Postos de trabalho para a produção dos pisos de madeira.....	4
Tabela 2.2. Código de encomenda.....	6
Tabela 4.1. Identificação das máquinas. ....	26
Tabela 5.1. Resultados obtidos para a instância de 20 jobs. ....	36
Tabela 5.2. Resultados obtidos para a instância de 50 jobs. ....	37
Tabela 5.3. Resultados obtidos para a instância de 100 jobs.....	37
Tabela 5.4. Melhores resultados obtidos pelo algoritmo desenvolvido para as instâncias de 20, 30 e 50 jobs.....	38
Tabela 5.5. Resultados obtidos pela SimpleAxis para as instâncias de 20, 30 e 50 jobs.....	38
Tabela 5.6. Comparação dos resultados obtidos pelo algoritmo desenvolvido neste trabalho e o da SimpleAxis.....	42



## **SIGLAS**

AG – Algoritmo Genético

FJS – *Flexible Job Shop*

FJSP – *Flexible Job Shop Scheduling Problem*

HFS – *Hybrid Flow Shop*

JSP – *Job Shop Problem*

MS - *Machine Selection*

OS – *Operation Sequencing*

PCP – Planeamento e Controlo da Produção



## 1. INTRODUÇÃO

Esta dissertação foi elaborada no âmbito da unidade curricular Estágio/Dissertação, do curso de Mestrado em Engenharia e Gestão Industrial da Faculdade de Ciências e Tecnologia da Universidade de Coimbra. Este trabalho é fruto de um projeto de estágio, em parceria com a *startup SimpleAxis*, para explorar possíveis soluções e desenvolver uma ferramenta de apoio à decisão para planear a produção de uma indústria de pisos de madeira, chamada *Castro Wood Floors*, para a qual a *SimpleAxis* presta serviços.

Um sistema de Planeamento e Controlo da Produção (PCP) é uma ferramenta crucial para que uma organização consiga satisfazer as exigências e expectativas cada vez mais elevadas dos clientes, no ambiente altamente competitivo que se vive atualmente. As funções típicas de um sistema de PCP incluem o planeamento dos requisitos de materiais, gerenciamento da procura, planeamento das capacidades e escalonamento e calendarização de *jobs* (Stevenson et al., 2005).

O escalonamento é um processo de tomada de decisão usado regularmente em muitas indústrias de manufatura e serviços. Este processo lida com a alocação de recursos a tarefas durante determinados períodos de tempo e sua finalidade é otimizar um ou mais objetivos (Pinedo, 1984). Normalmente, os recursos são apresentados como máquinas, e a tarefa ou conjunto de tarefas que serão processadas nas máquinas são conhecidas como um *job*.

Muitas empresas elaboram ainda o plano de escalonamento da produção manualmente, em papel, ou com ferramentas básicas como o Excel. Estas ferramentas podem até gerar uma solução exequível, porém a quantidade de restrições existentes dificulta o caminho para uma solução ideal.

A motivação para este trabalho surgiu da necessidade de criar um método para sequenciar o processamento das encomendas de uma indústria de pisos de madeira para otimizar a sua produção. A oportunidade de produzir um trabalho que trouxesse o conhecimento académico para a realidade de um ambiente industrial tornou-se fonte de interesse imediato. Aplicar a visão académica num contexto real de fábrica, demonstra não só a possibilidade, mas também a necessidade de interação entre estes dois mundos. A construção de pontes cada vez mais amplas e com via dupla que ligue o saber teórico às situações reais é fundamental.

É possível, de antemão, formular algumas questões que parecem pertinentes ao assunto: Quais são os métodos de escalonamento existentes? Quais são as vantagens e inconvenientes de cada abordagem? Como é possível averiguar se o método adotado é adequado ao ambiente de produção e às suas restrições? Quais são os critérios de desempenho? Existem estudos de caso que podem ser tomados como modelo para um estudo comparativo de forma a colaborar para a melhoria do processo adotado?

Para responder a essas questões, este trabalho explora as teorias e os conceitos desenvolvidos por alguns autores, que são essenciais para o desenvolvimento de uma solução.

Este trabalho tem como objetivo principal de, em parceria com a *SimpleAxis*, elaborar um método de escalonamento eficiente e eficaz para o processo produtivo da *Castro Wood Floors*. Para alcançar este objetivo foi necessário cumprir algumas etapas. Estas formaram a via que possibilitou a solução aqui desenvolvida e apresentada. Dessa forma, inicialmente, foi indispensável caracterizar e formular o problema com o máximo de detalhes possível. Esta etapa funcionou como uma anamnese para se chegar ao diagnóstico a ser tratado e foi a base que trouxe luz para o aporte teórico conceitual que fundamenta a proposição aqui descrita.

A partir desta anamnese, foi possível pesquisar os métodos mais eficientes e escolher o mais adequado para chegar a uma boa solução consoante os objetivos que este deveria cumprir: diminuir o *makespan* e o minimizar os atrasos nas entregas das encomendas.

Para além de formular um algoritmo que atendesse aos anseios da *Castro Wood Floors*, estabeleceu-se como meta neste estudo e tomou-se como desafio, mitigar o tempo computacional do algoritmo que, por ser complexo, pode levar muito tempo para ser processado.

A dissertação está organizada em 6 capítulos. No capítulo 1, é apresentada a introdução, contextualização do tema, o motivo da sua escolha e os objetivos deste estudo. No capítulo 2, é feita a descrição das empresas envolvidas no estudo e do caso a ser resolvido. O capítulo 3 baseia-se na revisão da literatura do escalonamento em *job shop* flexível e seus métodos de resolução, com foco nos Algoritmos Genéticos. A formulação do problema e o método utilizado para o resolver são apresentados no capítulo 4. No capítulo 5, são demonstrados os resultados encontrados e, por fim, o capítulo 6 apresenta as conclusões do trabalho desenvolvido e os trabalhos futuros.

## 2. CASO DE ESTUDO

O trabalho desenvolvido nesta dissertação por meio de um estágio curricular de cinco meses na *SimpleAxis*, consistiu em analisar o sistema produtivo da *Castro Wood Floors*, pesquisar os possíveis métodos de sequenciamento existentes para escolher o mais adequado e criar um algoritmo para o escalonamento da produção.

### 2.1. *SimpleAxis*

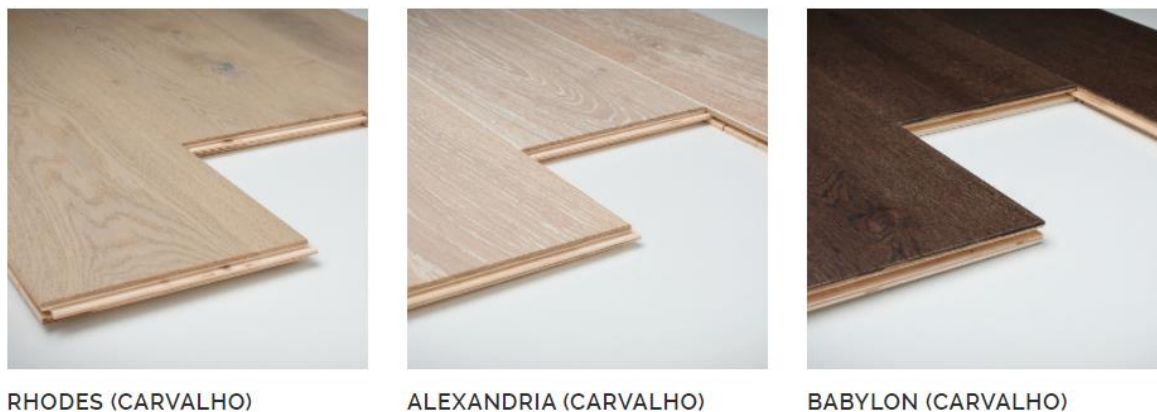
A *SimpleAxis* é uma *startup* tecnológica, fundada por uma equipa multidisciplinar de investigadores da Faculdade de Engenharia da Universidade do Porto, com competências avançadas em várias áreas da engenharia. Tem como foco, acelerar a incorporação das Tecnologias da Informação na indústria, combinando habilidades computacionais avançadas e analíticas. A *SimpleAxis* desenvolve soluções inovadoras, eficientes e funcionais baseadas na *cloud*, visando ganhos de produtividade, redução de custos e aumento da participação de mercado por meio de um melhor suporte à tomada de decisões.

Com um portfólio inicialmente desenvolvido na área da avaliação de riscos e otimização de soluções de engenharia civil estrutural, a *SimpleAxis* tem expandido progressivamente a sua atividade a outros sectores da indústria, contribuindo desta forma para a transformação digital das empresas e para a sua diferenciação em ambientes competitivos. Atualmente, seus principais serviços são: a criação de aplicativos com tecnologia de ponta, design criativo e computação em nuvem.

### 2.2. *Castro Wood Floors*

A *Castro Wood Floors* é uma empresa portuguesa pertencente à Castro & Filhos, que é especializada na produção de pavimentos em madeira desde 1970. Especializou-se durante décadas no tratamento de madeira dura, oriunda de todo o mundo. A experiência acumulada em todos estes anos de fabricação de produtos de madeira, permite que, atualmente, a Castro & Filhos seja líder nacional de produção de pavimentos de madeira e uma empresa de referência a nível mundial, com um volume de negócios de 20 M€. A gama

*Castro Wood Floors* inclui pisos maciços, pisos compostos e pisos de exterior, e está presente ativamente em mais de 40 países de 4 continentes. Na Figura 2.1 é possível ver exemplos dos produtos fabricados pela empresa.



**Figura 2.1.** Pisos de madeira produzidos pela *Castro Wood Floors*

### 2.3. Descrição do problema

O processo produtivo de pisos compostos da *Castro Wood Floors* conta com uma série de sete postos de trabalho. Cada um deles processa um tipo de operação, que atribui ao produto uma nova característica, tais como: colocação de massa, lixagem, escovagem, perfilagem, pintura, verniz e corte em chevron. A Tabela 2.1, apresenta os postos de trabalho existentes e suas respectivas máquinas. Os postos de trabalho A3 e B3 contam com máquinas paralelas idênticas.

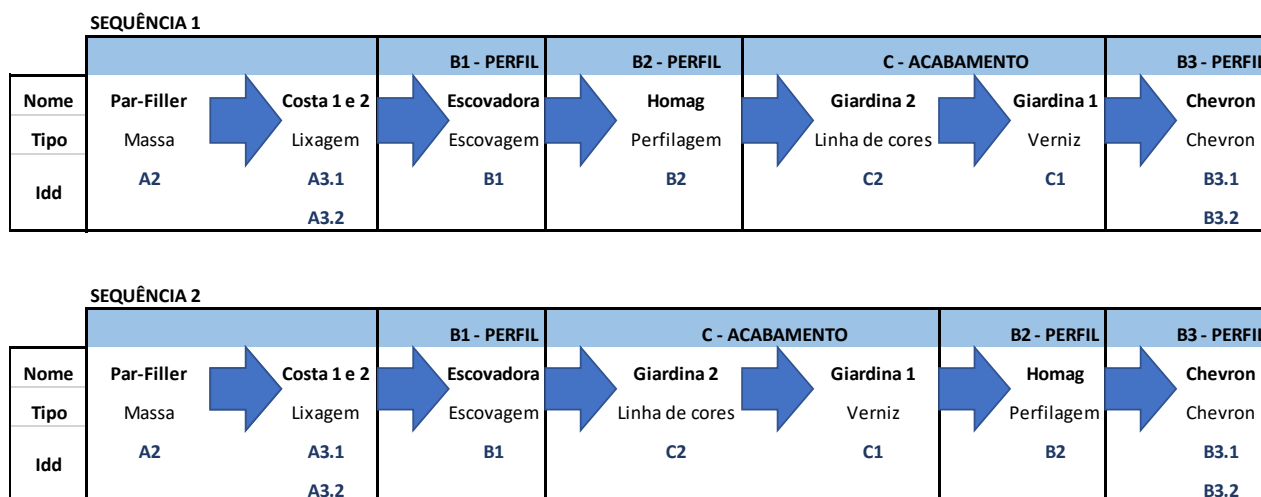
**Tabela 2.1.** Postos de trabalho para a produção dos pisos de madeira.

FASE	IDD	Máquina	Tipo
A - SANDWICH	A2	Par-Filler	Massa
A - SANDWICH	A3	Costa 1 e 2	Lixagem
B - PERFIL	B1	Escovadora	Escovagem
B - PERFIL	B2	Homag	Perfilagem
C - ACABAMENTO	C2	Giardina 2	Linha de cores/óleo
C - ACABAMENTO	C1	Giardina 1	Verniz
B - PERFIL	B3	Chevron 1 e 2	Chevron

As diferentes especificações estabelecidas pelos clientes para cada produto determinam como as máquinas terão de ser configuradas. A velocidade de uma mesma máquina é alterada em função das variáveis do produto a ser processado, e é calculada com base nas tabelas apresentadas pela empresa. (Anexos A a G)

As diferenças nos produtos também implicam um tempo de *setup* para trocas de ferramentas, limpeza e preparação das máquinas. Os tempos de *setup* dependem, também, da sequência em que as tarefas serão processadas. Considerando a margem de tempo necessária para a criação de um algoritmo que incluísse este último aspeto do processo, decidiu-se deixá-lo de fora por enquanto, podendo ser adicionado posteriormente em trabalhos futuros.

Embora o sistema produtivo completo contenha sete postos de trabalho, as encomendas não precisam de passar por todos eles e nem todas seguem a mesma rota. Existem duas possíveis sequências pelas quais as encomendas podem passar, ilustradas pela Figura 2.2. A sequência pela qual a encomenda passa é determinada pelo tipo de perfil do produto. Alguns postos dispõem de máquinas paralelas idênticas para divisão do trabalho.



**Figura 2.2.** Possíveis sequências de produção.

Cada encomenda possui um código de 22 dígitos, que determina as características que o produto precisa de ter no fim do processo produtivo. Este código refere-se às dimensões, ao tipo de produto, espécie da madeira, qualidade da madeira, formato, acabamento e cor, como pode ser visto no exemplo da Tabela 2.2.

Tabela 2.2. Código de encomenda.

<b>FA</b>	<b>14,0/0</b>	<b>CAF</b>	<b>0</b>	<b>00</b>	<b>0</b>	<b>00</b>	<b>0075</b>
Tipo de produto	Espessura	Espécie madeira	Qualidade	Formato	Acabamento verniz/óleo	Cor	Largura
A				B1+B2+B3	C1/C2	C2+C1	A

A partir do código de encomenda e das informações fornecidas sobre o rendimento de cada máquina de acordo com as especificações dos produtos, é possível calcular os tempos de processamento das operações a serem realizadas nas máquinas. Esses tempos serão os parâmetros que servirão como *input* para o algoritmo, tornando possível a elaboração de um cronograma para cada máquina, para que as encomendas sejam processadas no menor tempo possível e com o mínimo de atraso nas entregas.

Diante deste cenário, considerando a existência de duas possíveis sequências pelas quais os produtos podem seguir, é possível caracterizar o sistema de produção como um ambiente *job shop* que, por conter máquinas paralelas idênticas em dois postos de trabalho, configura-se como um caso especial, isto é, um FJS (*job shop* flexível), muito próximo de um HFS (*flow shop* híbrido).

Neste contexto, tornou-se um grande desafio encontrar uma solução eficiente para um problema que, além da complexidade de um ambiente FJS, envolve também minimizar mais do que um objetivo.

Quando a parceria com a *SimpleAxis* foi iniciada, um projeto de escalonamento, usando Algoritmo Genético, já havia sido desenvolvido para atender à procura da empresa *Castro Wood Floors*. No entanto, embora a solução criada já fosse eficiente, ainda apresentava alguns pontos que poderiam ser aprimorados e melhor desenvolvidos. Um dos pontos mais frágeis observado referia-se ao tempo de processamento computacional.

Nesse contexto, procurou-se, por meio deste trabalho, atacar os pontos frágeis da solução apresentada pela *SimpleAxis* e encontrar caminhos alternativos que trouxessem a eficiência e a eficácia necessárias às demandas da referida indústria.



### 3. REVISÃO DA LITERATURA

Para o bom desenvolvimento desta pesquisa, a leitura e análise de fundamentos teóricos já elaborados torna-se fundamental para embasar as questões aqui apresentadas. Deste modo, após caracterizar o caso de estudo encontrado, este enquadramento teórico apoia-se, principalmente, nos conceitos de escalonamento da produção, *flexible job shop* e seus métodos de resolução, com maior foco nos Algoritmos Genéticos.

#### 3.1. Escalonamento

O escalonamento refere-se à alocação de tarefas (por exemplo, *jobs* e operações) a recursos (por exemplo, máquinas) de tal forma que eles possam ser processados e / ou fabricados de uma maneira ideal (Emmons & Vairaktarakis, 2012), ou seja, consiste na determinação da sequência de operações que compõem as ordens de fabrico a processar nas diferentes máquinas de um determinado sistema industrial.

Segundo Jain e Meeran (1999), o escalonamento da produção é um problema de otimização combinatória de resolução complexa, mesmo para casos de pequena dimensão. O número de soluções possíveis, ou seja, de sequências de produção válidas, é demasiado grande para que de um modo simples e rápido se encontre uma solução de boa qualidade.

O problema de programação visa formular uma ordem de processamento que deve atingir um objetivo desejado da melhor maneira possível. Um bom escalonamento da produção permite melhorar o desempenho do sistema. Entre os indicadores de desempenho mais escolhidos, pode-se considerar: reduzir o tempo total de fabrico (*makespan*), aumentar as quantidades produzidas, aumentar a taxa de ocupação das máquinas, reduzir os stocks, diminuir tempos de entrega e atrasos das encomendas, diminuir os tempos de *setup* das máquinas e até reduzir o tempo de espera dos produtos entre operações.

Otimizar o processo produtivo possibilita saber a real capacidade de produção da empresa. Conhecer as capacidades do sistema, traz um aumento do potencial de negociação e torna mais fácil reagir a acontecimentos inesperados, trazendo ganhos financeiros à empresa.

Os sistemas de produção são muito diferentes uns dos outros, o que gera uma grande variedade de problemas. O escalonamento requer total conhecimento das operações, recursos e restrições do sistema. Esses problemas já foram estudados e descritos por muitos autores que propuseram diferentes métodos para resolvê-los. Por isso, a correta caracterização do problema é essencial para buscar casos parecidos na literatura e encontrar as melhores abordagens.

### **3.1.1. Caracterização do problema de escalonamento**

Graham et al. (1979) introduzem uma notação para descrever os problemas de escalonamento em que considerou o trio  $\alpha$ ,  $\beta$  e  $\gamma$ . O campo  $\alpha$  descreve o ambiente da máquina. O campo  $\beta$  fornece detalhes de características de processamento e restrições. O campo  $\gamma$  descreve o objetivo a ser minimizado, ou seja, suas medidas de desempenho.

Os possíveis ambientes de máquina especificados no campo  $\alpha$  são:

- Máquina única
- Máquinas idênticas em paralelo
- Máquinas em paralelo com velocidades diferentes
- Máquinas não relacionadas em paralelo
- *Flow Shop*
- *Flexible Flow Shop*
- *Job Shop*
- *Flexible Job Shop*
- *Open Shop*

As restrições de processamento especificadas no campo  $\beta$  podem incluir várias entradas. As entradas possíveis no campo  $\beta$  são:

- Datas de lançamento
- Preempções
- Restrições de precedência
- Tempos de *setup* dependentes da sequência
- Famílias de *jobs*
- Processamento em lote
- Avarias

- Restrições de elegibilidade da máquina
- Permutação
- Bloqueio
- Sem espera
- Recirculação

O terceiro campo ( $\gamma$ ), referente aos critérios de sequenciamento, permite qualificar uma solução por meio de uma função de avaliação que, para cada programa particular, terá um valor. O melhor programa é o que apresenta um melhor valor para a função de avaliação.

Cada objetivo traçado pela empresa está associado a uma medida de desempenho que, por sua vez, está vinculada a um ou mais indicadores de desempenho. Os indicadores mais utilizados dividem-se em três categorias:

1. Utilização de recursos:
  - a. Minimização do *makespan* (tempo total de processamento de todas as ordens de fabrico);
  - b. Maximização da percentagem de utilização de recursos.
2. Cumprimento dos prazos de entrega:
  - a. Minimização do número de ordens de fabrico atrasadas em relação à data de entrega;
  - b. Minimização do maior atraso na conclusão das ordens de fabrico;
  - c. Minimização do atraso médio na conclusão das ordens de fabrico;
  - d. Minimização do total de atrasos na conclusão das ordens de fabrico.
3. Critérios associados ao nível do estoque em processo:
  - a. Minimização do valor médio da quantidade de ordens abertas e ainda não finalizadas
  - b. Minimização do tempo médio de fluxo (média dos tempos decorridos entre o início e fim das ordens de fabrico)

O *makespan* é a medida de avaliação de performance mais usada na comparação de resultados obtidos por algoritmos de escalonamento. Um *makespan* mínimo geralmente implica uma boa utilização das máquinas, otimizando o sistema de produção.

### **3.1.2. Classes de escalonamento**

Segundo Pinedo (2002), o processo de escalonamento obriga a fazer algumas premissas. Essas premissas levam a que se possam considerar algumas classes de escalonamento:

1. Não-atrasados: Asseguram que nenhuma máquina é mantida ociosa enquanto pode ser utilizada para iniciar a execução de uma operação. Estes sequenciamentos fazem parte do conjunto dos ativos, logo, também são semi-ativos.
2. Ativos: Garantem que nenhuma operação pode ser antecipada sem provocar atraso em outras ou violar as restrições tecnológicas ou de recursos. Em outras palavras, nenhuma operação pode ser colocada em um buraco vazio no início do cronograma, preservando a viabilidade. Nesse tipo de sequenciamento, não existem tempos de inatividade desnecessários. Os sequenciamentos ativos formam um subconjunto dos semi-ativos, isto é, um sequenciamento ativo é necessariamente semi-ativo.
3. Semi-ativos: Asseguram que nenhuma operação pode ser concluída antes sem alterar a ordem de processamento em qualquer uma das máquinas.

### **3.2. Job Shop Flexível**

Num ambiente industrial, existem diversos tipos de sistemas de produção possíveis. A elaboração de um plano de escalonamento poderá ser mais ou menos eficaz considerando-se os diferentes cenários industriais. Neste contexto, o número de máquinas existentes, a variabilidade dos *jobs* a executar, os objetivos a atingir, a existência de imprevisibilidade na ocorrência de certos acontecimentos e outros fatores irão interferir na complexidade do plano de escalonamento. Apesar desta diversidade, a esmagadora maioria das situações reais existentes na indústria estão incluídas na categoria dos problemas *job shop* e suas variantes (Brucker & Knust, 2012). Ao longo dos anos, estes têm sido os tipos de problema de sequenciamento que têm suscitado maior interesse na comunidade científica.

Os sistemas *job shop* são capazes de produzir uma grande gama de produtos com diferentes rotas de processamento. Este fator traz consequências que, segundo Brucker

(2007), são problemas normalmente muito complexos, sendo poucos os que podem ser resolvidos em tempo polinomial.

Conforme descrito no capítulo 2 desta dissertação, foi possível concluir que o problema a ser resolvido no decorrer deste trabalho, se refere a um ambiente *job shop* flexível.

O *flexible job shop scheduling problem* (FJSP) é uma generalização do JSP clássico para sistemas de manufatura flexíveis (Pezzella et al., 2007), pois permite que cada operação seja processada em mais de uma máquina. Isso torna o FJSP mais difícil de resolver devido à consideração do roteamento de *jobs* e o sequenciamento de operações. Além disso, é um problema complexo de otimização combinatória. O JSP é conhecido por ser NP-hard (Garey et al., 1976), assim como o FJSP.

O FJSP envolve organizar a execução de  $N$  *jobs* em  $M$  máquinas. É dado um conjunto de *jobs* independentes:  $J = \{J_1, J_2, \dots, J_n\}$ . Cada *job*  $J_i$  contém um número de operações ordenadas imutáveis  $O_i, j \subseteq O_i$ , onde  $O_{ij}$  representa a  $j$  éxima operação da  $i$  éxima tarefa. É dado um conjunto  $U = \{M_1, M_2, \dots, M_m\}$  de máquinas. Cada operação  $O_{ij}$  requer pelo menos uma máquina para processamento de um conjunto de máquinas disponíveis  $U_{ij} \subseteq U$ . O tempo de processamento da operação  $O_{ij}$  atribuído à máquina  $M_k$  ( $M_k \in U_i, j$ ) é representado como  $P_{ijk}$ , no qual o *job*  $J_i = \{1 \leq i \leq N\}$ , operações  $O_{ij} = \{1 \leq i \leq N; 1 \leq j \leq O_i\}$ , e máquinas  $M_k = \{1 \leq k \leq M\}$ . Além disso, considera as seguintes premissas:

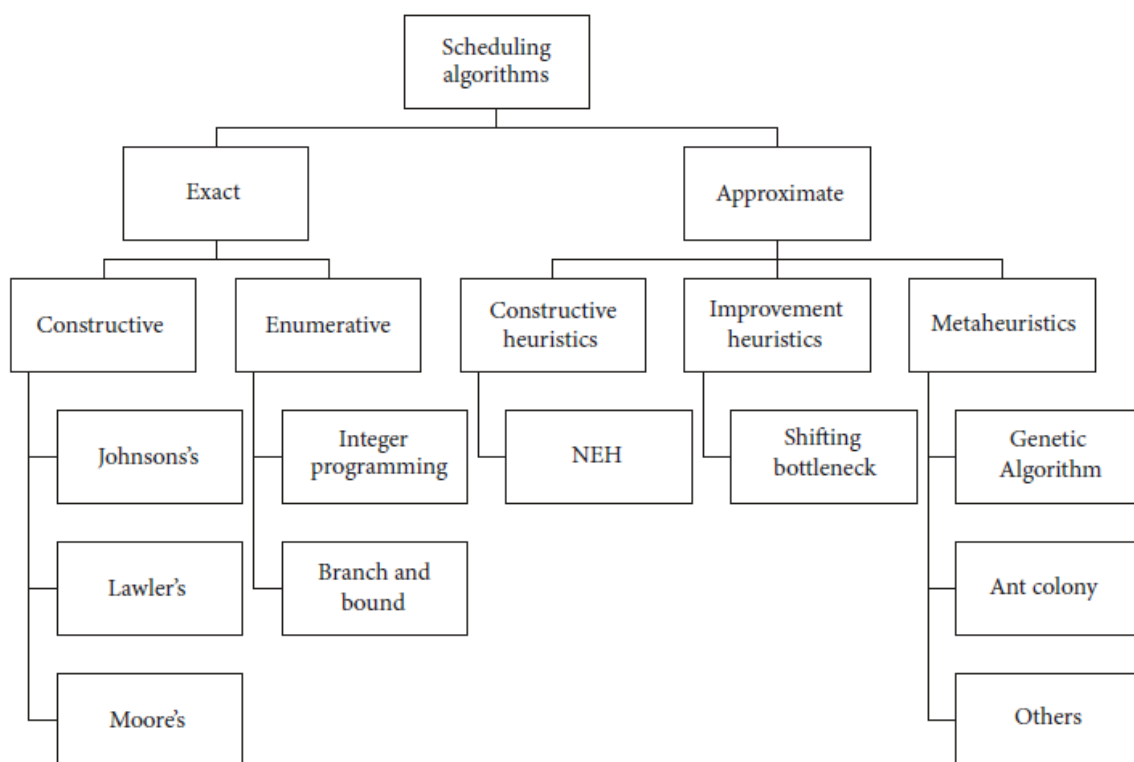
- Uma vez iniciada, cada operação deve ser concluída sem interrupção;
- Cada máquina pode processar uma operação por vez;
- As diferentes operações de uma ordem de fabrico não podem ser processadas simultaneamente, ou seja, cada *job* só pode estar a ser processado por uma máquina a cada momento;
- Todos os *jobs* e máquinas estão disponíveis no tempo 0;
- Os *jobs* são independentes. Não existem restrições de precedência entre operações de diferentes ordens de fabrico.

### 3.3. Métodos de escalonamento

A programação das operações em ambientes do tipo *job shop* consiste na determinação da sequência de operações que compõem as ordens de fabrico a processar nas

máquinas de um determinado sistema industrial. Isso é feito por meio da atribuição das operações dos *jobs* a uma posição específica na escala temporal da respetiva máquina.

Ao longo dos tempos, têm sido desenvolvidos e estudados variadíssimos métodos de resolução do problema de sequenciamento. A Figura 3.1 apresenta uma classificação para os algoritmos de escalonamento feita por (Amjad et al., 2018). Esta classificação não é exaustiva e contém apenas uma visão ampla das classes de algoritmo existentes.



**Figura 3.1.** Classificação para algoritmos de escalonamento (Amjad et al., 2018).

Alguns autores dividem os métodos entre exatos, heurísticos ou meta-heurísticos, entretanto Jain & Meeran (1999) classificaram-nos como métodos exatos (ou de otimização) e métodos de aproximação.

Os métodos exatos são aqueles que encontram sempre as melhores soluções ou as soluções mais eficientes, explorando o universo de soluções possíveis para o problema, encontrando a que cumpre melhor os objetivos (Jain e Meeran, 1999). Estão divididos entre métodos construtivos e métodos enumerativos.

Os métodos construtivos são resolvidos polinomialmente, enquanto nos métodos enumerativos temos as formulações matemáticas, que podem ser resolvidas por meio da programação linear, técnicas de decomposição, relaxação Lagrangeana e *Surrogate duality* ou da árvore de pesquisa *Branch and Bound*, com o grafo disjuntivo.

Devido ao obstáculo da complexidade de alguns problemas, passou-se a dar uma maior atenção e importância a métodos de aproximação, de tipo heurístico ou meta-heurístico.

As heurísticas são métodos que têm demonstrado grande aplicabilidade nos últimos anos tendo como vantagens o facto de serem intuitivos, de fácil implementação e não requerendo, geralmente, grande esforço computacional para a resolução dos problemas.

Algumas heurísticas são tão específicas e dependentes do tipo de problema que só podem ser usadas para um tipo de problema particular. Na década de 80, no entanto, a investigação orientou-se para métodos heurísticos que usam heurísticas tão gerais que podem ser usadas numa larga quantidade de problemas, podendo mesmo ser usadas em combinação com outros métodos, daí serem muitas vezes referidos na literatura por meta-heurísticas (Reeves, 1993).

As meta-heurísticas são um processo iterativo de pesquisa de melhores soluções na vizinhança, com o objetivo de encontrar um ótimo para o problema próximo do ótimo global. Os métodos de pesquisa local podem ser vistos como ferramentas para pesquisar um espaço de alternativas de modo a encontrar uma boa solução dentro dos limites.

A seguir serão apresentados alguns dos métodos mais utilizados na resolução destes problemas.

### **3.3.1. Formulações matemáticas**

As formulações matemáticas são um conjunto de técnicas para otimizar uma função cujas variáveis independentes estão sujeitas a restrições (French, 1982). Um desses métodos matemáticos é a programação linear, que estuda formas de resolver problemas de otimização cujas condições podem ser expressas por inequações lineares, isto é, inequações do primeiro grau.

Um problema de programação linear é composto por uma função objetivo, variáveis de decisão e algumas restrições a serem satisfeitas. A função objetivo e as

restrições são funções lineares das variáveis de decisão e a melhor solução depende das medidas de desempenho importantes para o sistema em questão.

### **3.3.2. Branch and bound**

O *branch and bound* é um método exato enumerativo que segue um princípio de enumeração implícita, isto é, considera certas soluções possíveis de um problema de otimização combinatória apenas indiretamente e sem avaliá-las explicitamente. Um atributo ou ramo da árvore de soluções define um subconjunto do conjunto, de todas as soluções possíveis do problema original.

Um outro princípio subjacente é o de subdividir um problema em subproblemas, em princípio independentes, e mais simples. O método consiste em considerar subconjuntos sucessivamente mais pequenos do conjunto de soluções possíveis (correspondendo a subproblemas do problema original cada vez mais pequenos) até obter conjuntos com uma única solução, ou conjuntos que, por certo, não contêm a solução ótima. Há três aspetos básicos neste método (Reis, 1996):

- O procedimento de *branch* (ramificar) - Particionar um problema grande em dois ou mais subproblemas.
- O procedimento de *bound* (limitar) - Calcular um limite inferior da solução ótima para cada subproblema.
- A estratégia de procura usada - Tem a ver com os subproblemas que são escolhidos para tomar em consideração em cada passo.

### **3.3.3. Regras de despacho**

A pesquisa de regras de prioridade tem sido alvo de variados estudos durante décadas, tendo sido estudadas na literatura muitas regras diferentes. Elas são provavelmente as heurísticas mais aplicadas à resolução de problemas de escalonamento devido à facilidade de implementação e reduzido tempo de processamento quando se pretende otimizar uma única medida de desempenho tal como o *makespan*, o tempo total de conclusão ou, por exemplo, o atraso máximo dos *jobs*.

Para cada processador, há uma fila de operações por programar. As operações são organizadas em filas e são processadas de acordo com uma das regras. De entre muitas



---

outras regras de prioridade, a seguir são destacadas as principais, adaptadas por Lustosa et al. (2008), Tubino (2007), Suresh e Sridharan (2007), Chan e Chan (2004):

- FIFO (*First In, First Out*) – Prioriza as tarefas que entram primeiro no sistema. Procura minimizar o tempo de permanência nas máquinas ou na fábrica.
- LIFO (*Last In, First Out*) – Prioriza a última peça a entrar no sistema. É considerada adversa e negativa no que tange à confiabilidade e rapidez de entrega. A sequência não é baseada em qualidade, flexibilidade ou custo.
- SPT (*Shortest Processing Time*) - Prioriza a tarefa com menor tempo de processamento total. Sua utilização visa a reduzir o tamanho das filas e o aumento do fluxo.
- LPT (*Longest Processing Time*) - Prioriza as tarefas com maior tempo de processamento total. Contrário à regra SPT. Sua utilização visa à redução de troca de máquinas.
- EDD (*Earliest Due Date*) - Prioriza a execução das ordens mais urgentes em termos de prazo de entrega. A finalidade é reduzir atrasos.
- LS (*Least Slack*) - Prioriza a menor folga entre a data de entrega e o tempo total de processamento entre as tarefas que estão à espera. É classificada por prazo de entrega e visa reduzir atrasos.
- SIPT (*Shortest Imminent Processing Time*) - Prioriza o menor tempo de processamento individual. Semelhante a SPT.
- LIPT (*Longest Imminent Processing Time*) - Prioriza o maior tempo de processamento individual. Semelhante a LPT.
- LWQ (*Least Work Next Queue*) - Prioriza as tarefas com destino à máquina ou estação de trabalho com menor fila no momento. Procura evitar o tempo morto de um processo subsequente.
- CR (*Critical Ratio*) - Prioriza a menor razão crítica (tempo até a data de vencimento dividido pelo tempo total de produção restante) entre as tarefas à espera. É uma regra dinâmica que procura combinar a EDD com a SPT, considerando apenas o tempo de processamento.
- DLS (*Dynamic Least Slack*) - Prioriza a menor folga (diferença entre a data prometida de entrega e o tempo total restante de processamento). Visa reduzir

atrasos, mas é um pouco mais complicada de aplicar que a LS por se tratar de uma regra dinâmica.

Estas são algumas das regras de despacho existentes, mas na prática, frequentemente, existe a necessidade de otimizar mais do que um critério de desempenho. Por isso, existem algumas regras de prioridade mais elaboradas, que contemplam vários parâmetros diferentes, para resolver funções objetivo mais complexas. Estas são designadas de regras de sequenciamento compostas. Uma regra de prioridade composta é uma expressão de classificação que combina uma série de regras de prioridade elementares.

#### **3.3.4. *Shifing Bottleneck Procedure***

Foi desenvolvido por Adams, Balas e Zawack (1988), é um método iterativo com o objetivo de melhorar a solução, que se baseia no grafo disjuntivo. Fundamenta-se na decomposição do problema principal em subproblemas de máquina única e cujo objetivo é melhorar a solução obtida. Em cada iteração, determinam-se as datas de disponibilidade e de entrega dos subproblemas de forma a sequenciar as máquinas que formam o problema inicial.

#### **3.3.5. *Tabu seach***

Faz uso da noção de vizinhança de uma solução, no espaço de soluções possíveis do problema e de uma analogia com um fenómeno natural que é, no caso, o uso de uma forma de memória. No método de *tabu search*, em cada iteração passa-se da solução atual para uma na vizinhança. O algoritmo verifica se essa solução já foi visitada recorrendo à lista tabu que guarda informação referente às soluções visitadas. A lista tabu consiste numa lista com o objetivo de guardar soluções dos últimos movimentos. Quando uma solução está na lista tabu esse movimento não é permitido, evitando ciclos e permitindo a exploração de outros espaços de soluções. A lista tabu tem uma dimensão fixa e quando fica cheia, o mais antigo é removido. O critério de paragem é baseado num número pré-definido de iterações, momento em que o algoritmo finaliza (Baker & Triestch, 2009).

#### **3.3.6. *Simulated Annealing***

À semelhança do *Tabu Search*, este método também faz uso da noção de vizinhança de uma solução, no espaço de soluções possíveis do problema. O método foi

criado na década de 1980 nos trabalhos de Kirkpatrick et al. (1983) e Cerny (1985). A designação deste método deve-se à analogia existente entre a otimização combinatória e a evolução do equilíbrio térmico dos sólidos.

Trata-se de um algoritmo estocástico que permite a degradação da solução atual através de movimentos para soluções de pior qualidade, de forma a escapar ao mínimo local e atrasar a convergência do algoritmo.

O processo é desencadeado com uma solução inicial, que pode ser obtida aleatoriamente ou por uma heurística. São geradas sequências do estado energético de um sólido. Para cada iteração, é selecionada aleatoriamente uma solução pertencente à vizinhança e é aplicada uma pequena perturbação gerada aleatoriamente ao estado energético atual. Se esta resultar num estado energético inferior, será aceite como nova solução. Se o nível energético aumentar, o novo estado será aceite dependendo de uma probabilidade de aceitação.

Numa fase inicial do processo iterativo, serão aceites quase todas as substituições da solução atual por uma nova da sua vizinhança, que é escolhida aleatoriamente permitindo explorar o espaço das soluções. Com o andamento do algoritmo, a temperatura vai diminuindo gradualmente e vai se tornando cada vez mais seletivo na aceitação de novas soluções.

### **3.3.7. Algoritmos Genéticos**

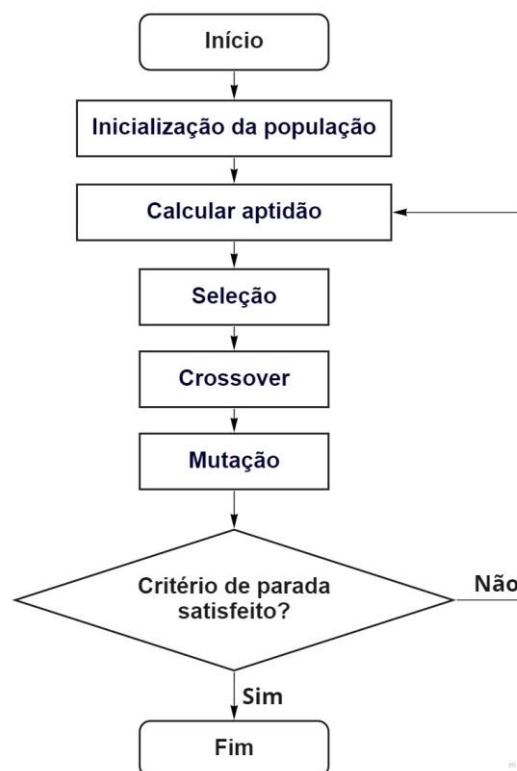
Os Algoritmos Genéticos são uma analogia entre o processo de encontrar boas soluções e a teoria da evolução das espécies, de Charles Darwin (1859), que diz que as populações evoluem de acordo com o processo de seleção natural com base na regra da sobrevivência dos indivíduos mais aptos. O trabalho inicial em AG foi conduzido por John Holland (1975), que foi então estendido principalmente por Goldberg (1989).

A busca pela melhor solução para os problemas de otimização é feita por meio de um processo iterativo de busca local. A partir de uma população inicial, o algoritmo aplica operadores genéticos para produzir descendentes, que herdaram características genéticas de ambos os pais. Como resultado, os descendentes dos melhores indivíduos são presumivelmente mais aptos do que seus ancestrais.

Os genes da nova geração que pontuarem mais na função de aptidão, serão também selecionados para a reprodução e criação da nova geração. A cada iteração, cada

novo indivíduo (cromossoma) corresponde a uma solução, ou seja, a um cronograma da instância FJSP fornecida. O ciclo evolucionário continua de geração para geração. Assim, os cromossomas com piores soluções desaparecem e os melhores reproduzem bons descendentes. Finalmente, a sobrevivência da população mais apta ter-se-á adaptado otimamente ou quase otimamente ao ambiente.

O fluxograma na Figura 3.2, mostra o funcionamento geral de um Algoritmo Genético tradicional.



**Figura 3.2.** Fluxograma de um Algoritmo Genético

1. Inicialização: Representa a criação da população inicial utilizando um esquema de codificação adequado. Existem diferentes formas de codificar a geração inicial: binária e não-binária, fixa ou de comprimento variável. O método tradicional descrito por Holland (1975), usa uma codificação binária de comprimento fixo. Na fase inicial o sistema gera os cromossomas de forma aleatória e codifica-os como uma população. Os genes dos cromossomas descrevem a atribuição de operações às

- 
- máquinas, e a ordem em que aparecem no cromossoma descreve a sequência de operações. Cada cromossoma representa uma solução para o problema.
2. Avaliação de aptidão: o objetivo de solucionar o FJSP é satisfazer um critério de desempenho predefinido para obter um cronograma ideal (Amjad et al., 2018). A aptidão de cada cromossoma é calculada por uma função, que pontua todas soluções da geração de acordo com o objetivo. A aptidão é calculada como um problema de otimização, com uma função que deve ser minimizada ou maximizada. Os critérios de desempenho mais utilizados são o *makespan*, a percentagem de utilização de recursos, o número de ordens de fabrico atrasadas em relação à data de entrega e o tempo médio de fluxo. No entanto, também existem casos em que mais de um objetivo deve ser considerado. A otimização multiobjetivo é mais complexa do que a otimização de objetivo único devido ao fato de que a minimização simultânea de duas ou mais funções pode levar a uma situação em que diminuir ainda mais uma função pode fazer com que a outra função aumente. Muitas abordagens foram desenvolvidas para resolver a otimização multiobjetivo. Essas abordagens podem ser classificadas em três categorias (Hsu et al., 2002).
    - a. Transformar o problema multiobjetivo em um problema mono objetivo atribuindo coeficientes de peso para cada objetivo;
    - b. A abordagem que trata cada objetivo de forma separada;
    - c. A abordagem de Frentes de Pareto, que é baseada no conceito de otimização de Pareto.
  3. Operadores genéticos: A população de cada geração é submetida aos operadores genéticos para obter uma nova geração. A nova geração é teoricamente melhor do que a anterior, pois segue o princípio da “sobrevivência do mais apto” e, portanto, substitui a geração anterior. Durante esse processo, toda a população pode ser alterada ou apenas o pior cromossomo pode ser substituído (El-Ghazali, 2009). A cada iteração as soluções são pontuadas de acordo com o critério de aptidão e os operadores genéticos são usados para guiar e trazer aleatoriedade às soluções. A seguir, são apresentados os operadores genéticos:
    - a. Seleção: Em cada iteração, os melhores cromossomas são escolhidos para reprodução. O operador de seleção escolhe os indivíduos arbitrariamente ou

conforme a sua aptidão. Existem vários tipos de operadores de seleção, alguns deles, descritos por Amjad (2018) são:

- i. Seleção da roda da roleta: a probabilidade de seleção de um cromossoma é diretamente proporcional à sua aptidão, conforme avaliado pela função objetivo. Assim, um cromossoma com maior aptidão terá uma maior probabilidade de ser selecionado; no entanto, cromossomas de menor aptidão também podem ser selecionados.
  - ii. Seleção por atribuição de aptidão baseada em classificação: associa a aptidão relativa entre cromossomas individualmente, evitando assim que uma geração contenha uma estrutura cromossômica totalmente adequada. O método é usado principalmente para manter a diversidade da população.
  - iii. Seleção por torneio: um conjunto de cromossomas é selecionado aleatoriamente e, em seguida, os cromossomas mais aptos são selecionados para operação posterior.
  - iv. Seleção por elitismo: mantém um número fixo de cromossomas mais aptos e o resto da população é gerado usando algum outro método. Assim, este método não só garante que as melhores soluções permaneçam na população, mas também garante a diversificação da população.
- b. *Crossover*: Este operador trabalha nos indivíduos selecionados pelo operador de seleção. Ele é aplicado nos genes de dois cromossomas pais para produzir dois descendentes que contêm a distinção dos cromossomas pais. São gerados dois novos indivíduos que herdaram as boas associações presentes em cada um dos pais, ou seja, esses descendentes têm maior probabilidade de sobrevivência. Alguns tipos de crossover existentes foram apresentados por Kora & Yadlapalli (2017) e Umbarkar1 & Sheth (2015):
- i. *Crossover* de um ponto: um ponto de *crossover* é selecionado aleatoriamente ao longo do comprimento dos vetores dos cromossomas pais e os genes que estão próximos aos locais de cruzamento são trocados, criando dois descendentes. Se o local

- 
- apropriado for escolhido, bons descendentes podem ser obtidos, caso contrário, a qualidade dos indivíduos será prejudicada.
- ii. *Crossover* de N pontos: usa o ponto de cruzamento aleatório para combinar os pais da mesma forma que para o cruzamento de um ponto. Para fornecer a grande combinação de pais, ele seleciona mais de um ponto de cruzamento para criar os dois descendentes (Gwiazda, 2006).
  - iii. *Crossover* de três pais: três pais são escolhidos arbitrariamente. Cada gene do primeiro progenitor é comparado com o gene equivalente do segundo progenitor. Se os dois genes são semelhantes, o gene é ocupado pelos descendentes ou então o gene equivalente do terceiro pai é usado pela prole. É usado principalmente no caso de cromossomas codificados em binários.
  - iv. *Crossover* parcialmente mapeado (PMX): é o operador de *crossover* usado com mais frequência. Dois cromossomas são associados e dois locais de *crossover* são escolhidos arbitrariamente. A fração de cromossomas entre os dois pontos de cruzamento dá uma seleção correspondente que passa pelo processo de cruzamento por meio de operações de troca posição por posição. O PMX tende a respeitar as posições absolutas.
  - v. *Crossover* baseado em ordens de fabrico (OBX): seleciona aleatoriamente várias posições em um dos tours pai, e a ordem das cidades nas posições selecionadas desse pai é imposta ao outro pai para produzir um filho. O outro filho é gerado de maneira análoga para o outro pai.
  - vi. *Crossover* de posição (PX): opera selecionando vários locais aleatórios ao longo dos cromossomas dos pais. Os elementos são então herdados pelos descendentes na ordem em que ocorrem no primeiro progenitor. Os elementos restantes necessários para completar os descendentes são doados pelo segundo progenitor (com os elementos doados pelo primeiro progenitor omitidos) na ordem em que aparecem no segundo progenitor.

- c. **Mutação:** O operador de mutação é aplicado nos cromossomas com o objetivo de alterar um ou mais de seus genes. A função deste operador é mudar algumas informações em cromossomas selecionados para diversificar o espaço de solução para exploração posterior, além de contornar o problema de mínimos locais. Muitas técnicas de mutação foram propostas na literatura, algumas delas foram descritas por Abdoun et al., (2012) e são apresentadas a seguir:
- i. **Mutação *Twors*:** permite a troca de posição de dois genes escolhidos aleatoriamente.
  - ii. **Mutação inversa do centro (CIM):** o cromossoma é dividido em duas secções. Todos os genes em cada secção são copiados e, em seguida, colocados inversamente na mesma seção de um descendente.
  - iii. **Mutação de sequência reversa (RSM):** é tomada uma sequência S limitada por duas posições  $i$  e  $j$  escolhidas aleatoriamente, de modo que  $i < j$ . A ordem dos genes nesta sequência será invertida da mesma maneira que foi abordada na operação anterior.
4. **Critério de parada:** É criada uma nova geração de cromossomas até que o critério de parada seja satisfeito. O critério pode ser o de atingir um número fixo de gerações, atingir um determinado valor como resultado da função objetivo, ou de não haver uma melhora significativa de uma geração para outra, após certa quantidade de iterações. Se o critério de parada for satisfeito, o algoritmo termina e o melhor cromossoma, e a programação correspondente é fornecida como saída. Caso contrário, o algoritmo volta novamente para a etapa 3.

### **3.3.8. Discussão**

Os métodos de otimização já foram estudados de forma exaustiva por diversos autores ao longo do tempo. Todo esse conhecimento trouxe diversos avanços para os problemas de sequenciamento, porém o estudo académico dos métodos exatos ainda apresenta diferenças significativas dos problemas encontrados na realidade industrial. A complexidade de alguns problemas faz com que esses métodos não sejam bem sucedidos em aplicações realistas, que muitas vezes exigem simplificações tão drásticas, que se afastam do problema inicial.



Os métodos de aproximação conseguem dar resposta a problemas de elevada dimensão, mas não garantem que a solução encontrada seja a ótima. Encontram soluções satisfatórias, num intervalo de tempo considerado razoável. São, por isso, os métodos que têm efetivamente aplicação prática na maioria dos casos reais. (Jain e Meeran, 1999).

As heurísticas são uma ótima solução para problemas de otimização combinatória, isto é, problemas de escalonamento em que a complexidade se eleva exponencialmente à medida que a dimensão dos problemas aumenta.

Diferentemente dos métodos exatos, alguns métodos de aproximação apresentam a grande vantagem de serem tão gerais que podem ser aplicáveis a uma larga quantidade de problemas. Eles podem ser usados em combinação com outros métodos, o que torna possível modelos muito mais aproximados dos problemas reais.

Além disso, os algoritmos dos métodos iterativos podem ser terminados com uma solução ao fim de uma iteração qualquer, mesmo que seja uma solução sub ótima, que seria melhorada na próxima iteração.

Vários procedimentos heurísticos foram desenvolvidos nos últimos anos para o FJSP. Eles podem ser classificados em duas categorias principais: abordagem hierárquica e abordagem integrada (Pezzela, 2007).

A abordagem hierárquica pode reduzir a dificuldade ao decompor o FJSP numa sequência de subproblemas. Brandimarte (1993) foi o primeiro a aplicar a abordagem hierárquica ao FJSP. Paulli (1995) resolveu o subproblema de roteamento utilizando algumas regras de despacho existentes e, em seguida, resolveu o subproblema de escalonamento por diferentes métodos de *tabu search*. A abordagem integrada poderia alcançar melhores resultados, mas é bastante difícil de ser implementada em operações reais. Hurink, Jurisch e Thole (1994) e Dautère-Pérès (1997) propuseram diferentes abordagens heurísticas de *tabu search* para resolver o FJSP usando uma abordagem integrada (Zhang et al., 2018).

Zhang et al. (2011) apresentam no seu artigo uma robusta revisão literária sobre os métodos de resolução do FJSP. Para além da conclusão de que as meta-heurísticas são mais eficientes para a resolução do FJSP, o artigo mostra que os Algoritmos Genéticos são a solução mais utilizada para resolver este tipo de problema. Foram considerados os artigos de Chen et al. (1999), Yang (2001), Zhang e Gen (2005), Jia, Nee, Fuh e Zhang (2003), Kacem, Hammadi e Borne (2002), Kacem (2003), Tay e Wibowo (2004) e Pezzella et al.

(2007) que adotam o AG em conjunto com diferentes abordagens. Na sua revisão, é relatado que os artigos analisados diferem uns dos outros em esquemas de codificação e decodificação, método de inicialização da população e estratégias de geração de descendentes.

A força do AG em relação a outros algoritmos de busca local deve-se ao facto de que mais estratégias podem ser adotadas em conjunto para manejar as gerações, tanto na fase de inicialização da população, quanto na fase de geração dinâmica. Desta forma, um espaço de busca maior pode ser explorado em cada etapa do algoritmo.

## 4. METODOLOGIA

O estudo da literatura mostrou que a utilização de Algoritmos Genéticos promove bons resultados para o FJSP, por este motivo esta foi a abordagem escolhida para o problema que se pretende resolver. Neste capítulo apresenta-se o desenvolvimento da solução para o problema em estudo e descreve-se o processo de preparação e simplificação dos dados e o funcionamento do algoritmo implementado.

### 4.1. Preparação dos dados

Ao escolher o método mais apropriado para resolver o problema, foi necessário fazer um tratamento dos dados para transformá-los em informação útil e exportá-la para o formato compatível com a entrada no algoritmo.

Conforme apresentado anteriormente, o sistema de produção da Castro *Wood Floors* contém sete postos de trabalho responsáveis pela aplicação de massa para corrigir a madeira, lixagem, escovagem, perfilagem, linha de cores e óleo, verniz e corte em chevron. Os processos de lixagem e corte em chevron possuem máquinas paralelas idênticas, sendo, portanto, um sistema com nove máquinas.

Após a passagem nas máquinas de massa (Par-Filler) e de cores (Giardina 2), os artigos têm um tempo de secagem, que será considerado como dez minutos e dois minutos, respetivamente, até se iniciar o processo seguinte. Para modelar o problema, os tempos de secagem foram considerados como parte do processo, pois limitam a entrada dos artigos na máquina seguinte. Por isso, foram criadas duas máquinas fictícias chamadas de salas de espera, por onde os artigos passam após serem processados pela Par-Filler e pela Giardina 2.

Desta forma, o sistema completo pode ser definido com onze máquinas, e cada uma delas recebe um número de identificação, pelo qual elas serão reconhecidas pelo algoritmo, conforme mostra a Tabela 4.1.

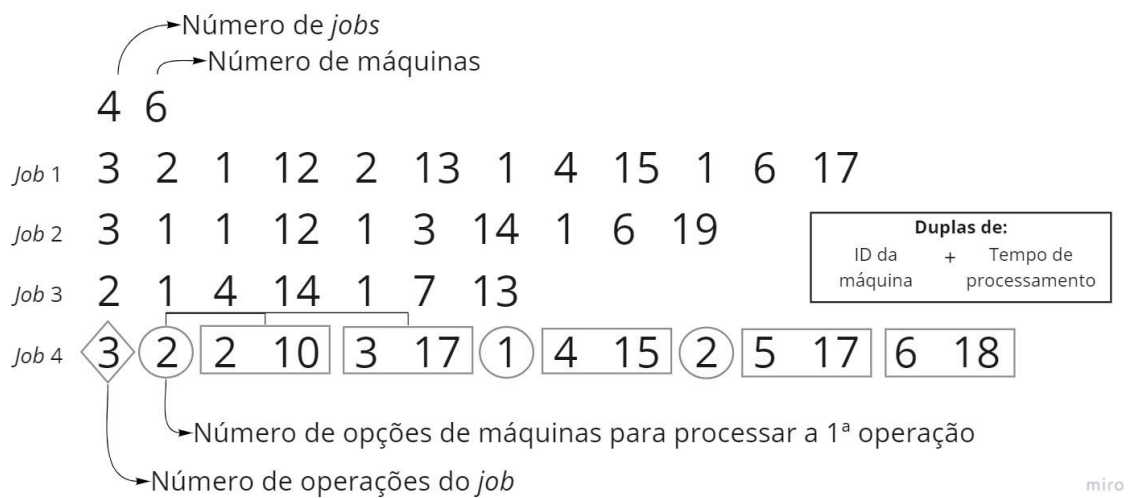
**Tabela 4.1.** Identificação das máquinas.

ID da máquina	Máquina	Tipo
1	Par-filler	Massa
2	Sala de espera 1	Secagem
3	Lixadora 1	Lixagem
4	Lixadora 2	Lixagem
5	Escovadora	Escovagem
6	Homag	Perfilagem
7	Giardina 2	Linha de cores /óleo
8	Sala de espera 2	Secagem
9	Giardina 1	Verniz
10	Chevron 1	Chevron
11	Chevron 2	Chevron

Também foram disponibilizadas, pela *SimpleAxis*, informações sobre as velocidades de produção das máquinas e processo produtivo, além de uma grande quantidade de dados contendo as diversas características que um produto pode ter ao ser encomendado. A informação recolhida foi organizada em tabelas, cujas informações mais relevantes estão disponíveis nos Anexos de A a G.

A utilização de uma folha de cálculo possibilita a automação dos cálculos necessários para gerar os tempos de processamento das operações das encomendas em cada máquina. A folha de cálculo utilizada foi o MS EXCEL e com o recurso ao *Visual Basic for Applications* - a linguagem de programação da aplicação - é possível exportar um ficheiro com a informação reorganizada em uma matriz que será utilizada como entrada para o algoritmo implementado em *Python*.

Na Figura 4.1 se encontra um exemplo da estrutura utilizada no ficheiro de entrada dos dados, que é um tipo de estrutura muito usada na literatura para a entrada de dados em algoritmos de resolução do FJSP.



**Figura 4.1.** Estrutura dos dados para entrada no algoritmo.

Conforme mostra a Figura 4.1, a primeira linha de dados refere-se ao número de *jobs* (ou ordens de fabrico) e ao número de máquinas do sistema. Da segunda linha em diante, cada linha representa um *job*. O primeiro valor da linha corresponde à quantidade de operações do *job*. Em seguida, para cada operação é apresentada a quantidade de opções de máquinas para processar aquela operação, e um par de valores que equivalem ao número identificador da máquina e ao tempo de processamento da operação na máquina em questão. Além dessas informações, também foi necessário criar um *input* utilizando as datas de entrega de cada ordem de fabrico, que foram consideradas como sendo o último segundo do dia de entrega, para possibilitar o cálculo dos atrasos em segundos.

## 4.2. Desenvolvimento do modelo

A solução desenvolvida se baseia num AG, que mostrou ser uma alternativa eficaz para resolver esse tipo de problema.

O problema é, portanto, determinar uma atribuição e uma sequência de operações nas máquinas para satisfazer alguns critérios. Ele contém dois subproblemas: o problema de seleção de máquina e o problema de sequenciamento de operação. Portanto, o FJSP é mais complicado e desafiante do que o JSP clássico porque requer uma seleção

adequada de uma máquina a partir de um conjunto de máquinas disponíveis para processar cada operação de cada *job* (Ho et al., 2007).

O procedimento geral da abordagem proposta é descrito pelas seguintes etapas:

1. Definição dos parâmetros de entrada;
2. Inicialização e codificação da população da primeira geração;
3. Avaliação da aptidão de cada indivíduo da população pela função objetivo;
4. Geração da nova população pelos operadores genéticos:
  1. Seleção dos indivíduos de acordo com a sua aptidão;
  2. *Crossover* dos indivíduos selecionados;
  3. Mutação dos indivíduos que sofreram *crossover*;
5. Verificação do critério de parada. Enquanto o critério não for satisfeito, voltar para a etapa 3. Se o critério for satisfeito, imprimir solução

#### **4.2.1. Parâmetros de entrada**

Para além dos dados com os tempos de processamento das encomendas e datas de entrega, também é necessário estabelecer outros parâmetros associados ao AG.

- Tamanho da população: quantidade de soluções (ou cromossomas) criadas a cada geração;
- Máximo de gerações: número máximo de iterações que serão feitas até chegar à solução final.;
- Data de início: dia em que dar-se-á início à produção das ordens de fabrico;
- Parâmetros genéticos:
  - Percentagem de reprodução;
  - Percentagem de *crossover*;
  - Percentagem de mutação;
- Peso dos indicadores de desempenho: por se tratar de um problema multiobjetivo, foi necessário definir o peso que cada indicador tem na avaliação de aptidão dos cromossomas.

#### 4.2.2. Inicialização e codificação da população

A inicialização da população é feita por meio da geração, de forma aleatória, dos cromossomos (ou indivíduos) que correspondem às soluções do FJSP. A codificação dos indivíduos é muito importante no AG.

Neste estudo, o método de codificação adotado é o de Gao et al. (2006). A codificação é feita por meio de dois vetores diferentes, uma vez que o FJSP contém dois subproblemas. O primeiro é o vetor OS, da ordem das operações e o segundo é o vetor MS, da seleção das máquinas e cada um deles tem um método de codificação diferente.

O método de codificação do vetor OS é uma representação baseada nas operações, que é composta pelos números dos *jobs*. Conforme a Figura 4.2.

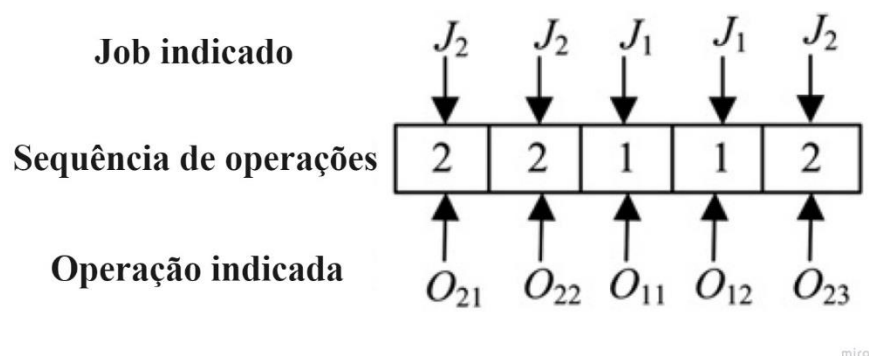


Figura 4.2. Representação do vetor OS com a ordem das operações.

Esta representação deve ser entendida da seguinte forma: o vetor apresenta o número dos *jobs* e cada vez que um *job* se repete, representa a sua operação seguinte. Ou seja, cada número de *job* aparece multiplicado pelo número de operações que ele possui. Ao digitalizar a sequência do OS da esquerda para a direita, a quarta aparição de um número de *job* refere-se à quarta operação deste *job*. Esta forma de representar a solução facilita a decodificação para uma solução viável.

Já o método de codificação do vetor MS é uma representação composta pela escolha das máquinas, que é baseada no número de opções de máquinas possíveis para processar cada uma das operações, conforme mostra a Figura 4.3.

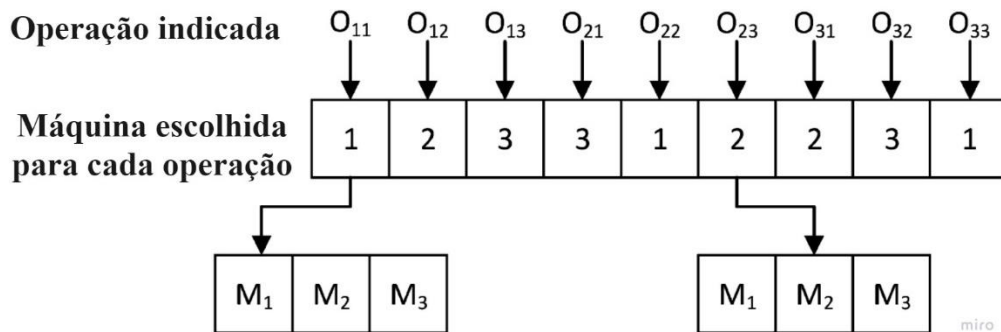


Figura 4.3. Representação do vetor MS com escolha das máquinas para cada operação.

Cada gene do vetor denota a máquina selecionada para as operações. O vetor MS inicial é gerado ao selecionar a máquina alternativa aleatoriamente para cada operação de cada *job*.

Um cromossoma é formado por um vetor OS e um vetor MS. As soluções são decodificadas em esquemas semi-ativos, que apesar de demorarem mais para alcançar boas soluções do que os ativos, requerem um tempo computacional menor.

### 4.2.3. Avaliação da aptidão

Após decodificar a solução, encaixando as operações nas máquinas num esquema semi-ativo, são calculados os tempos em que cada máquina finaliza as operações alocadas a ela. Ao obter estes resultados, sabe-se o valor do *makespan* da solução, que é o valor máximo dos tempos finais de cada máquina.

Para cada solução também são calculados os atrasos. Primeiramente, é calculado o tempo em que cada encomenda estará pronta, selecionando a sua última operação e somando o momento que ela chega à máquina ao seu tempo de processamento. Ao obter este valor, basta subtrair a suposta data de entrega da encomenda. Se o resultado for positivo, há atrasos. Isso é feito para todas as encomendas e todos os atrasos são somados (total *tardiness*).

Para calcular a pontuação de cada solução os valores são colocados numa função com pesos, que é apresentada abaixo:

$$Score = TotalTardiness * TardinessWeight + Makespan * MakespanWeight \quad (4.1)$$



---

A partir da pontuação, é feita a classificação das soluções das mais aptas para as menos aptas e seguem para as próximas etapas em que passam pelos operadores genéticos.

#### 4.2.4. Operadores Genéticos

O desempenho dos Algoritmos Genéticos e o alcance de boas soluções dependem principalmente dos tipos de operadores genéticos escolhidos. Eles podem ser divididos em três tipos: seleção, *crossover* e mutação. No método de codificação escolhido, a representação do cromossoma contém dois vetores, OS e MS, e cada um deles utiliza operadores genéticos diferentes para cada etapa.

O operador de seleção é utilizado para selecionar os indivíduos de acordo com a aptidão. Para resolver o problema em questão, foram escolhidos dois tipos de operadores de seleção, o primeiro é o esquema de seleção elitista, o outro é o esquema de seleção por torneio.

1. Seleção elitista: seleciona uma percentagem dos melhores indivíduos da população para serem reproduzidos na geração seguinte. O valor da percentagem é um dos parâmetros genéticos do algoritmo;
2. Seleção por torneio: alguns indivíduos da população são escolhidos aleatoriamente, a quantidade depende do parâmetro  $b$ , que normalmente varia entre 2 e 7, neste trabalho  $b$  foi considerado como 2, e o indivíduo com melhor aptidão de entre os escolhidos é selecionado.

Os operadores de *crossover* usam dois indivíduos para gerar descendentes. No AG criado, os dois pais são indicados como P1 e P2 e os dois descendentes são indicados como O1 e O2. Três operadores de *crossover* foram adotados: para aplicar ao vetor OS, é escolhida aleatoriamente uma opção entre o *crossover* de operação de precedência (POX) e o *crossover* baseado em *jobs* (JBX), já para o MS, é aplicado o *crossover* de dois pontos.

1. *Crossover* de operação de precedência (POX)

Etapa 1: O conjunto de *jobs* é dividido aleatoriamente em dois grupos *Jobset1* e *Jobset2*;

Etapa 2: Qualquer elemento em P1 que pertença a *Jobset1* é anexado à mesma posição em O1 e excluído em P1; qualquer elemento em P2 que pertença a *Jobset1* é anexado à mesma posição em O2 e excluído em P2;

Etapa 3: Os elementos restantes em P2 são acrescentados às posições vazias restantes em O1; e os elementos restantes em P1 são acrescentados às posições vazias restantes em O2.

## 2. Crossover baseado em *jobs* (JBX)

Etapa 1: O conjunto de *jobs* é dividido aleatoriamente em dois grupos *Jobset1* e *Jobset2*;

Etapa 2: Qualquer elemento em P1 que pertença a *Jobset1* é anexado à mesma posição em O1 e excluído em P1; qualquer elemento em P2 que pertença a *Jobset2* é anexado à mesma posição em O2 e excluído em P2;

Etapa 3: Os elementos restantes em P2 são acrescentados às posições vazias restantes em O1; e os elementos restantes em P1 são acrescentados às posições vazias restantes em O2.

## 3. *Crossover* de dois pontos

Etapa 1: Duas posições são selecionadas aleatoriamente;

Etapa 2: Os dois descendentes são gerados trocando todos os elementos entre as posições das sequências dos dois pais.

O operador de mutação é usado para a introdução e manutenção da diversidade genética da população, alterando arbitrariamente um ou mais genes dos vetores. O operador é aplicado aos indivíduos com uma probabilidade dada pela taxa de mutação. No modelo desenvolvido, dois operadores de mutação foram escolhidos para o vetor OS, mutação de troca e mutação de vizinhança, que são selecionados aleatoriamente, e um operador para o vetor MS, *half mutation*. Nos procedimentos de mutação, um dos pais é denotado como P e um descendente é denotado como O.

### 1. Mutação de troca

Etapa 1: Duas posições são selecionadas no P;

Etapa 2: O descendente é gerado trocando os elementos nas posições selecionadas.

## 2. Mutaç o de vizinhança

Etapa 1: Tr s elementos s o selecionados no P (os valores desses elementos s o diferentes);

Etapa 2: Os genes vizinhos a estes valores s o alterados aleatoriamente para gerar outros vetores;

Etapa 3: Um dos vetores gerados   escolhido para ser o O.

## 3. *Half Mutation*

Etapa 1: r posi es s o selecionadas no P (r   a metade do comprimento do vetor MS);

Etapa 2: Para cada posi o (de acordo com uma opera o), mudar o valor da m quina para outra dentre as dispon veis para processar aquela opera o.

### **4.2.5. Crit rio de parada**

O algoritmo desenvolvido termina quando o n mero de gera es atinge o valor m ximo, que   definido como um dos par metros de entrada. Quando este crit rio   satisfeito,   feita uma impress o das solu es e do gr fico de *Gantt*, que mostra como as opera es devem ser organizadas nas m quinas.

  dif cil definir um valor fixo do n mero de gera es necess rias para obter uma solu o satisfat ria. Geralmente este valor muda consoante o n mero de ordens de fabrico que precisam ser processadas.



## 5. RESULTADOS

Este capítulo apresenta os experimentos computacionais usados para avaliar o desempenho do algoritmo proposto. O algoritmo foi implementado em *Python* num PC Intel Core i5 1.7 GHz com 8 GB de memória RAM.

Para explorar a eficácia do modelo, foram realizados experimentos alterando os parâmetros do algoritmo para selecionar os que apresentam melhores resultados.

Para obter uma avaliação mais fidedigna das capacidades do modelo proposto, os resultados foram comparados com os encontrados pelo algoritmo desenvolvido pela *SimpleAxis*.

### 5.1. Escolha dos parâmetros

A configuração dos parâmetros do algoritmo pode afetar a sua performance, por isso, experimentos piloto foram conduzidos para avaliar o desempenho das possíveis combinações de parâmetros.

Geralmente, os parâmetros genéticos são testados e escolhidos como fixos para o algoritmo. No entanto, outros parâmetros são sensíveis à escala do problema, como o tamanho da população e o número de gerações. Por exemplo, uma população com tamanho 100 pode ser grande para uma instância de 10 *jobs*, mas pode ser pequena para uma instância de 100 *jobs*. Portanto, o tamanho da população deve mudar dinamicamente de acordo com a escala do problema, assim como o número de iterações, que é correspondente ao número de gerações.

A fim de escolher os melhores parâmetros genéticos e fixá-los, foram elaborados testes experimentais. Para comparar os resultados, serão avaliados os valores obtidos para o *makespan*, os atrasos e o tempo computacional necessário. Os pesos da função de avaliação foram mantidos como 0.5 para cada um dos objetivos.

Foram usadas três instâncias com dimensões de 20, 50 e 100 *jobs* e onze máquinas, com os dados obtidos por meio do tratamento das informações fornecidas pela *SimpleAxis*. A tabela com os dados encontra-se no Anexo I e J. Cada uma das instâncias foi executada 30 vezes, com algumas alterações nos parâmetros. A probabilidade de reprodução

variou entre 0.01 e 0.1, a probabilidade de *crossover* variou entre 0.7 e 0.9 e a probabilidade de mutação, entre 0.05 e 0.1. Os 10 melhores resultados dos 30 experimentos de cada instância foram coletados para comparação do desempenho dos parâmetros.

Os resultados encontrados para a instância de 20 *jobs* encontram-se na Tabela 5.1. Todos os valores obtidos para o tempo foram medidos em segundos. Neste caso, o total de atrasos para todas as corridas é zero, por este motivo, a coluna “*tardiness*” foi ocultada da tabela.

**Tabela 5.1.** Resultados obtidos para a instância de 20 *jobs*.

Número de <i>jobs</i>	Tamanho da população	Número de gerações	Probabilidade de reprodução	Probabilidade de <i>crossover</i>	Probabilidade de mutação	<i>Makespan</i>	Tempo computacional
20	100	500	0.1	0.8	0.05	103031	45.20
20	100	500	0.01	0.8	0.05	103137	75.04
20	100	500	0.01	0.8	0.05	103137	73.64
20	100	500	0.1	0.8	0.05	103137	48.55
20	100	500	0.1	0.8	0.05	103374	49.00
20	100	1000	0.01	0.8	0.05	103380	85.46
20	100	500	0.1	0.9	0.05	103380	84.66
20	100	500	0.1	0.8	0.1	103500	54.48s
20	100	500	0.01	0.8	0.05	103374	65.91
20	100	500	0.01	0.7	0.05	103613	49.29

Os resultados encontrados para a instância de 50 *jobs* encontram-se na Tabela 5.2. O total de atrasos também foi ocultado da tabela pois obteve o valor zero para todos os casos.

Tabela 5.2. Resultados obtidos para a instância de 50 *jobs*.

Número de <i>jobs</i>	Tamanho da população	Número de gerações	Probabilidade de reprodução	Probabilidade de <i>crossover</i>	Probabilidade de mutação	<i>Makespan</i>	Tempo computacional
50	300	2000	0.01	0.8	0.05	244774	1996.34
50	300	2000	0.01	0.8	0.05	246307	1562.81
50	200	2000	0.1	0.8	0.1	248919	1156.63
50	100	1500	0.01	0.8	0.05	252088	1004.64
50	300	1000	0.1	0.8	0.1	252409	921.53
50	200	1500	0.01	0.8	0.1	253278	705.03
50	200	2000	0.1	0.8	0.1	253387	1147.32
50	300	1500	0.01	0.8	0.05	254518	1597.96
50	100	1000	0.01	0.8	0.05	254950	600.03
50	200	2000	0.1	0.8	0.1	255194	1125.48

Os resultados do experimento com a instância de 100 *jobs* são mostrados na Tabela 5.3.

Tabela 5.3. Resultados obtidos para a instância de 100 *jobs*.

Número de <i>jobs</i>	Tamanho da população	Número de gerações	Probabilidade de reprodução	Probabilidade de <i>crossover</i>	Probabilidade de mutação	<i>Makespan</i>	<i>Tardiness</i>	Tempo computacional
100	200	10000	0.01	0.8	0.05	522882	49617	17233.63
100	200	4000	0.01	0.8	0.05	522935	38703	11180.46
100	200	3000	0.01	0.8	0.05	523695	225300	7728.11
100	100	16000	0.01	0.7	0.05	532354	122168	18903.89
100	200	2000	0.01	0.8	0.05	540520	87303	5073.54
100	100	2000	0.01	0.9	0.05	556538	1305747	1590.66
100	100	2000	0.01	0.8	0.01	568977	430958	1684.59
100	100	2000	0.1	0.8	0.01	572943	2320670	1240.27
100	100	2000	0.01	0.7	0.05	574272	1853416	1485.01
100	100	2000	0.01	0.9	0.05	592587	1022444	1538.45

Os resultados foram ordenados com base nos menores valores obtidos para o *makespan* que, na maioria dos casos, também representa um menor valor para o *tardiness*.

Ao ordenar os valores, é possível compreender que alguns parâmetros genéticos apresentam boa performance para todos os casos. Diante disto, foram fixados como:

- Probabilidade de reprodução: 0.01
- Probabilidade de *crossover*: 0.8
- Probabilidade de mutação: 0.05

## 5.2. Comparação dos resultados

Esta seção apresenta a comparação entre os resultados obtidos com o algoritmo desenvolvido neste trabalho e os obtidos com o algoritmo criado pela *SimpleAxis*.

Para o modelo desenvolvido, serão utilizados nesta comparação os melhores resultados alcançados para o valor do *makespan* para cada uma das instâncias testadas, ou seja, os resultados que estão na primeira linha de cada uma das tabelas anteriores, e são apresentados na tabela 5.4. Os resultados obtidos pela *SimpleAxis* são mostrados na Tabela 5.5.

**Tabela 5.4.** Melhores resultados obtidos pelo algoritmo desenvolvido para as instâncias de 20, 30 e 50 *jobs*.

Número de <i>jobs</i>	Tamanho da população	Número de gerações	<i>Makespan</i>	<i>Tardiness</i>	Tempo computacional
20	100	500	103031	0	45.20
50	300	2000	244774	0	1996.34
100	200	10000	522882	49617	17233.63

**Tabela 5.5.** Resultados obtidos pela *SimpleAxis* para as instâncias de 20, 30 e 50 *jobs*.

Número de <i>jobs</i>	Tamanho da população	Número de gerações	<i>Makespan</i>	<i>Tardiness</i>	Tempo computacional
20	50	500	103985	0	116
50	50	1500	251418	0	943
100	50	16000	536179	83626	18346



A primeira instância a ser testada, foi a de 20 *jobs*, que obteve como resultado um *makespan* de 103031 segundos, com um tempo de processamento computacional de 45 segundos. O gráfico de *Gantt* que correspondente a esta solução é apresentado na Figura 5.1

Na Figura 5.2, é mostrado o gráfico de *Gantt* com os resultados encontrados pela *SimpleAxis*, que obteve um *makespan* de 103985 segundos, com tempo de processamento de 116 segundos.

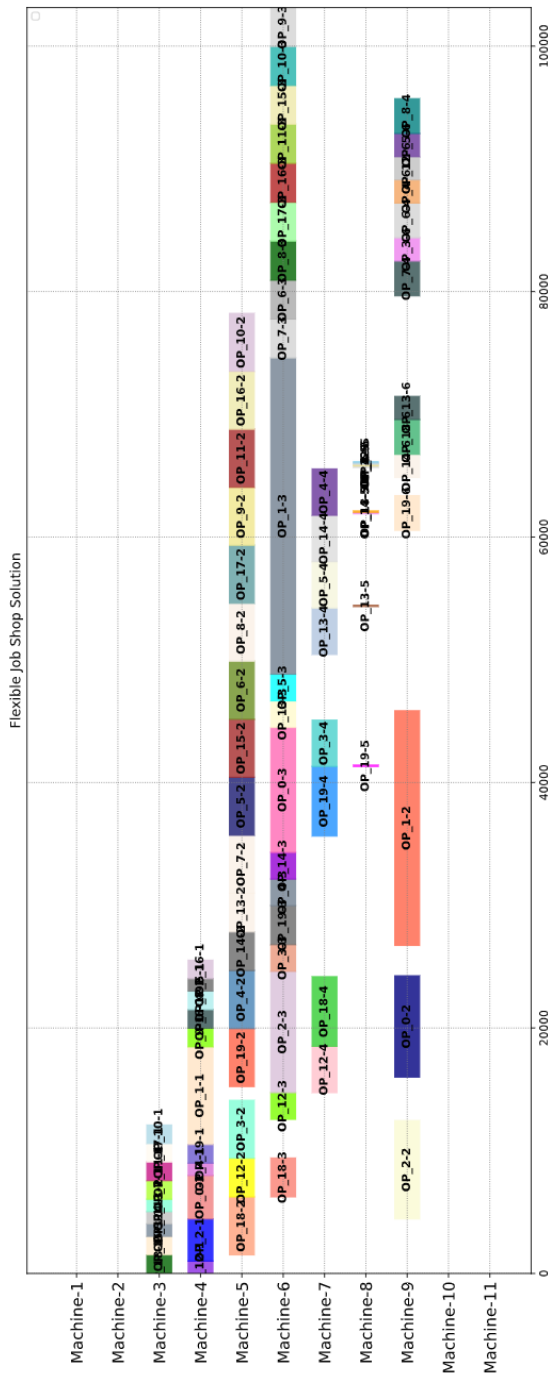


Figura 5.1. Resultados para 20 *jobs*.

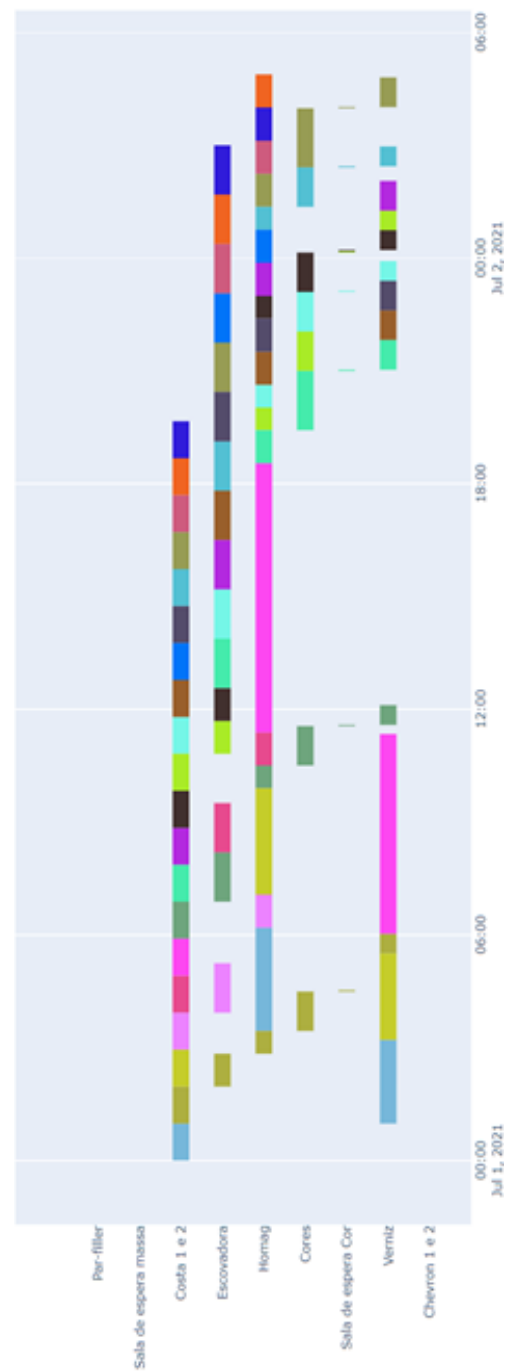


Figura 5.2. Resultados da SimpleAxis para 20 *jobs*.

Ao processar a instância de 50 *jobs*, os valores encontrados foram de 244774 segundos para o *makespan* e 1996 segundos para o tempo computacional.

A *SimpleAxis* obteve um *makespan* de 251418 segundos e 943 segundos para o tempo computacional. As Figuras 5.3 e 5.4 apresentam os Gráficos obtidos para o algoritmo desenvolvido e para o algoritmo da *SimpleAxis*, respetivamente.

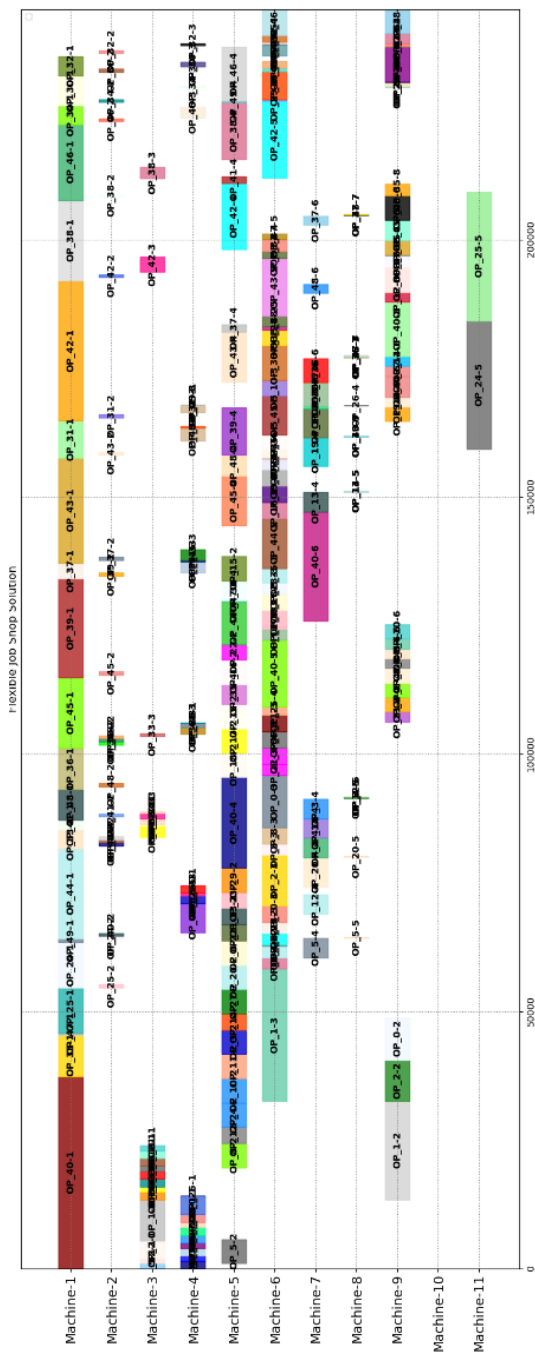


Figura 5.3. Resultados para 50 *jobs*.

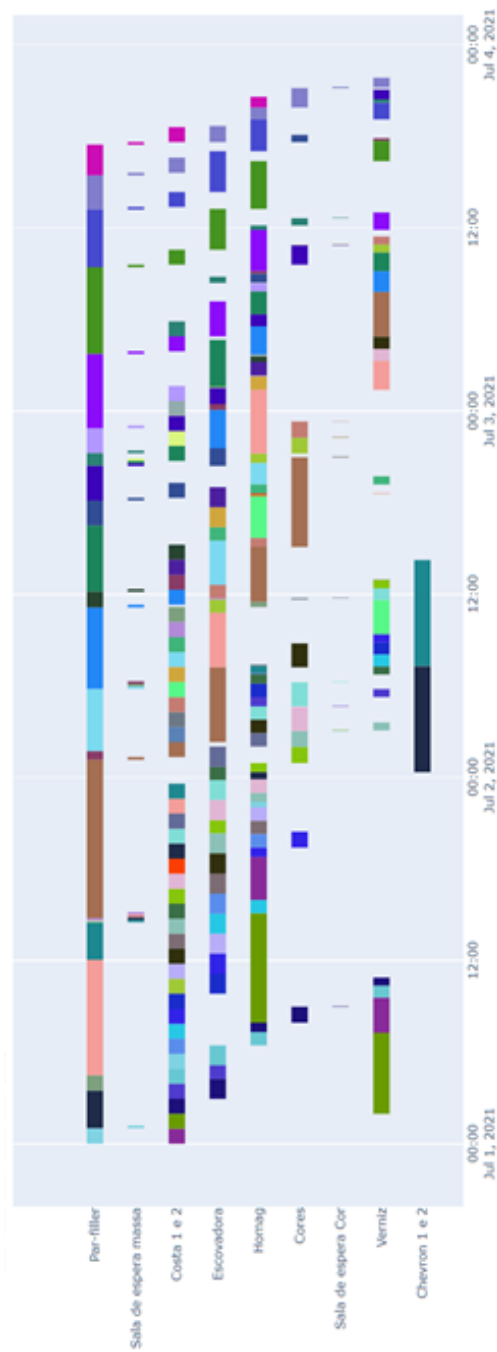


Figura 5.1. Resultados da SimpleAxis para 50 *jobs*.

Para a última instância testada, com 100 *jobs*, o algoritmo desenvolvido obteve um *makespan* de 522882 segundos e 49617 segundos de atraso, num período computacional de 17234 segundos.

Já a *SimpleAxis* obteve um *makespan* de 536179 segundos, 83626 segundos de atraso em 18346 segundos de tempo computacional. Os gráficos de *Gantt* correspondentes a estas soluções são mostrados nas Figuras 5.5 e 5.6.

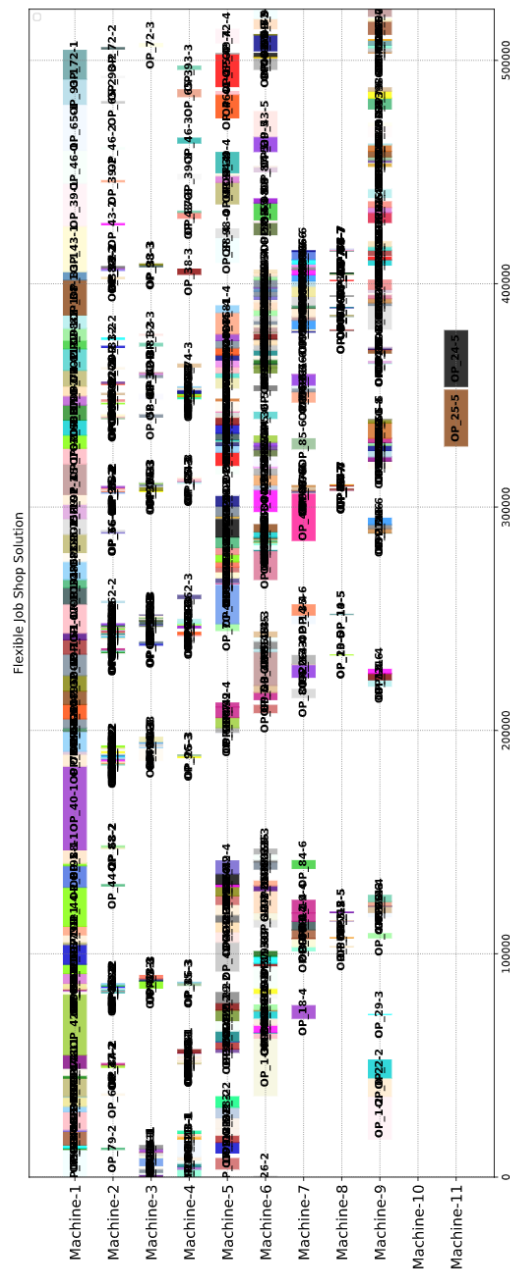


Figura 5.5. Resultados para 100 *jobs*.

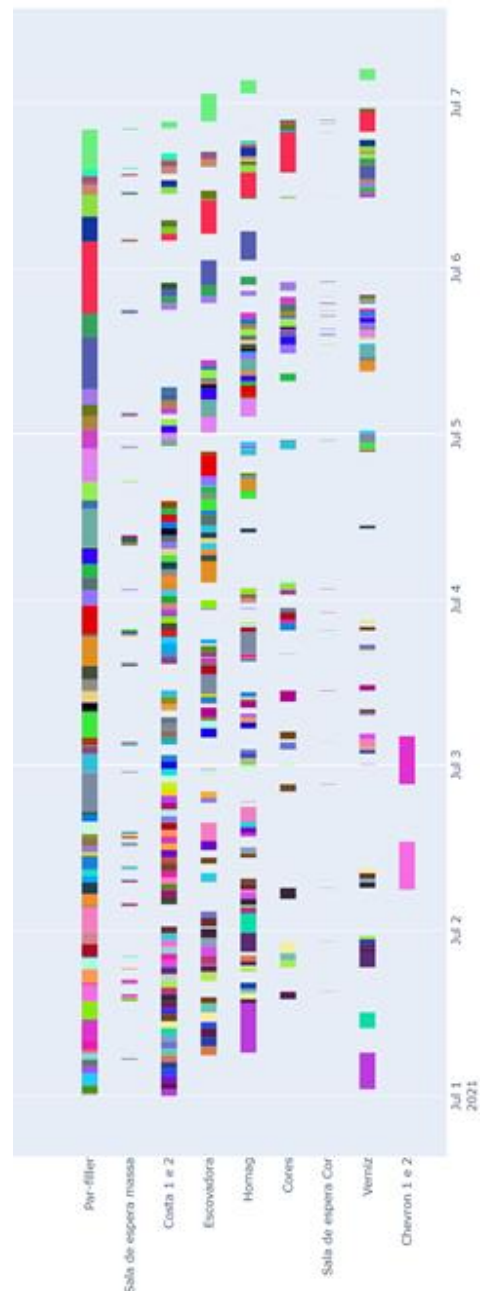


Figura 5.6. Resultados da SimpleAxis para 100 *jobs*.

Como mostra a Tabela 5.6, os resultados obtidos pelos dois algoritmos são compatíveis e não apresentam grandes discrepâncias, o que valida o funcionamento de ambos.

**Tabela 5.6.** Comparação dos resultados obtidos pelo algoritmo desenvolvido neste trabalho e o da SimpleAxis.

Número de <i>jobs</i>	20	50	100
<b>Makespan</b>	103031	244774	522882
<b>Makespan SimpleAxis</b>	103985	251418	536179
<b>Diferença (%)</b>	-0,93	-2,71	-2,54
<b>Tardiness</b>	0	0	49617
<b>Tardiness SimpleAxis</b>	0	0	83626
<b>Diferença (%)</b>	0	0	-68,54
<b>Tempo computacional</b>	45	1996	17233
<b>Tempo computacional SimpleAxis</b>	116	943	18346
<b>Diferença (%)</b>	-157,78	52,76	-6,46

O algoritmo desenvolvido neste trabalho apresentou resultados um pouco melhores para todas as instâncias. Em relação ao *makespan*, houve uma redução de 0,93% para a primeira instância, 2,71% para a segunda e 2,54% para a terceira instância.

O atraso nas encomendas diminuiu em 68,54%.

Os tempos computacionais apresentaram melhores resultados para as instâncias de 20 e 100 *jobs*, com uma redução de 157,78% e 6,46%. Para a instância de 50 *jobs*, houve um aumento no tempo, no entanto o algoritmo também conseguiu encontrar em tempos menores, resultados mais próximos aos da *SimpleAxis*.

Uma diferença a ser destacada entre os resultados obtidos se refere ao facto de que foram feitos vários experimentos com o AG desenvolvido para encontrar os menores valores, enquanto que a *SimpleAxis* rodou o algoritmo apenas uma vez para cada uma das instâncias. Esta diferença não garante a superioridade de um algoritmo sobre o outro, mas demonstra o potencial do método desenvolvido, uma vez que propõe um novo caminho para solucionar o problema exposto. Pode-se considerar também que o algoritmo desenvolvido poderá, futuramente, ser aprimorado produzindo resultados ainda mais promissores.

## 6. CONCLUSÕES

Neste trabalho é proposto um Algoritmo Genético para a resolução do problema de escalonamento das ordens de fabrico num ambiente *job shop* flexível da empresa *Castro Wood Floors*.

Neste ponto, considera-se que o trajeto percorrido até aqui trouxe desafios e uma grande aprendizagem. Antes de dar início ao trabalho, foi necessário perceber o funcionamento do sistema produtivo e os detalhes do problema em questão, que era de grande complexidade e possuía muitas restrições.

Ao dar início a esta pesquisa, deparou-se com a necessidade de fazer um estudo aprofundado sobre os métodos existentes, uma vez que o conhecimento sobre a sua maioria era superficial.

Ao optar por resolver o problema recorrendo a um Algoritmo Genético, ainda havia muitas possibilidades de aplicação. Foi preciso escolher a classe de escalonamento mais adequada e os operadores genéticos mais eficientes que diminuíssem o tempo de computação do algoritmo e otimizassem a solução.

Além disso, o conhecimento de linguagens de programação era limitado, o que se tornou um grande desafio.

O pouco tempo disponível para desenvolver uma solução, fez com que fosse necessário selecionar algumas restrições mais complexas do problema para serem excluídas temporariamente.

Apesar de todas as limitações apontadas, reconhece-se que o trabalho desenvolvido deu origem a resultados interessantes. O algoritmo proposto no âmbito deste trabalho foi testado e os seus resultados comparados com o anteriormente elaborado pela *SimpleAxis*. Verificou-se que os resultados obtidos com o AG desenvolvido são equivalentes aos obtidos pelo algoritmo da *SimpleAxis*, sendo inclusivamente superiores nalgumas instâncias, entre as quais sobressaem-se uma redução de até 68,54% nos atrasos e 157,78% no tempo computacional.

Embora fosse possível encontrar alguns resultados melhores, não é possível afirmar que o algoritmo desenvolvido é superior ao da *SimpleAxis*, visto que foram feitas 30

corridas para cada instância do algoritmo desenvolvido e apenas uma para o algoritmo da *SimpleAxis*.

Não obstante ao fator tempo, este trabalho pela procura de soluções e otimização de resultados poderia ser aprimorado. Deste modo, considera-se que, futuramente, será interessante investigar outros aspetos relacionados a esta pesquisa, tais como: a adição dos tempos de *setup* e turnos das máquinas; a hibridização com outro método para melhorar resultados e diminuir ainda mais o tempo computacional; teste de diferentes operadores genéticos; consideração de outros critérios de desempenho importantes para a empresa, como o peso dos clientes.

Que este seja, portanto, não um estudo que se encerra em si mesmo, mas sim um trabalho que abre portas a outras abordagens num momento propício.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Allahverdi, A., Gupta J.N.D. e Aldowaisan T. (1999) “A Review of Scheduling Research involving Setup Considerations”, *Omega The International Journal of Management Science*, vol. 27, pp. 219-239.
- Adams, J., Balas, E., Zawack, D. (1998), “The Shifting Bottleneck Procedure for Job Shop Scheduling”, *Management Science*, Volume 34, N° 3, 391-401.
- Amjad M. K., Butt S. I., Kousar R., Ahmad R., Agha M. H., Faping Z., Anjum N. e Asgher U. (2018), “Recent Research Trends in Genetic Algorithm Based Flexible Job Shop Scheduling Problems”, *Mathematical Problems in Engineering*, Volume 2018, Article ID 9270802.
- Baker, K.; Triestch, D. (2009), “Principles of Sequencing and Scheduling”, New Jersey: Jonh Wiley and Sons, Inc.
- Baker, K. (1974), “Introduction to Sequencing and Scheduling”. New York: Wiley.
- Barnes JW, Chambers JB. (1996), “Flexible Job Shop Scheduling by tabu search.”, Graduate program in operations research and industrial engineering. Technical Report ORP 9609, University of Texas, Austin.
- Brandimarte P. (1993), “Routing and scheduling in a flexible job shop by tabu search.”, *Annals of Operations Research*; 41: pp157–83.
- Brucker, P. e Knust, S. (2012), “Complex scheduling”, GOR Publications, Springer.
- Brucker, P. (2007), “Scheduling algorithms”, 5th Ed., Springer.
- Brucker P., Sotskov Y. N., e Werner F. (2007), “Complexity of shop scheduling problems with fixed number of jobs: a survey,” *Mathematical Methods of Operations Research*, vol. 65, no. 3, pp. 461–481.

- Chan, F. T. S.; Chan, H. K. (2004), “Analysis of dynamic control strategies of an FMS under different scenarios.”, *Robotics and Computer-Integrated Manufacturing*, n. 20, p. 423-437.
- Chen H, Ihlow J. e Lehmann C. (1999), “A genetic algorithm for flexible Job-shop scheduling.”, *IEEE international conference on robotics and automation*, Detroit; p. 1120–5.
- Conway, R.W.; Maxwell, W.L.; Miller, L.W. (1967), “Theory of Scheduling”, Mass: Addison Wesley.
- Dauzère-Pérès, S., e Paulli, J. (1997). “An integrated approach for modeling and solving the general multi-processor job-shop scheduling problem using tabu search.”, *Annals of Operations Research*, 70, 281–306.
- Darwin, C., (2003), “On the Origin of Species by Means of Natural Selection”, J. Carroll, Ed., Broadview, Toronto, Canada.
- El-Ghazali T., (2009), “Metaheuristics: From Design to Implementation”, vol. 74, JohnWiley & Sons.
- Emmons H. e Vairaktarakis G. (2012) “Flow Shop Scheduling: Theoretical Results, Algorithms, And Applications”, vol. 182, Springer Science & Business Media.
- French, S. (1982), “Sequencing and Scheduling: An introduction to the Mathematics of the Job Shop”, Chichester: Ellis Horwood.
- Gaither, N.; Frazier, G. (2002), “Administração da Produção e Operações”. 8. ed. São Paulo: Pioneira.
- Garey M. R. e Johnson D. S. (1979), “Computers and Intractability: A Guide to the Theory of NP-Completeness”, W. H. Freeman, San Francisco, Calif, USA.
- Garey MR, Johnson DS, Sethi R. (1976). “The complexity of flowshop and jobshop scheduling.”, *Mathematics of Operations Research*, volume 1, pp.117–29.
- Goldberg, D. E. (1989), “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley Longman Publishing Co., Inc.



- Graham, R. L.; Lawler, E. L.; Lenstra, J.K.; Rinnooy Kan, A.H.G. (1979), "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey", Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium. Elsevier. pp. (5) 287–326.
- Graves, S. C. (1981) "A Review of Production Scheduling, Operations Research".
- Grobler J., Engelbrecht A. P., Kok S., e Yadavalli S. (2010), "Metaheuristics for the multi-objective FJSP with sequence dependent set-up times, auxiliary resources and machine down time," *Annals of Operations Research*, vol. 180, no. 1, pp. 165–196.
- Gwiazda T. D. (2006), "Genetic Algorithms Reference", Volume –I, Poland: Tomasz Gwiazda.
- Ho N.B. e Tay J.C. (2004), "Genace: An Efficient Cultural Algorithm for Solving the Flexible Job-Shop Problem.", *Congress on Evolutionary Computation.*, vol. 2, pp. 1759-1766.
- Holland, J.H. (1975), "Adaptation in Natural And Artificial Systems", University of Michigan Press, Ann Arbor, Mich, USA.
- Hsu, T., Dupas, R., Jolly, D., & Goncalves, G. (2002), "Evaluation of mutation heuristics for the solving of multi-objective flexible job shop by an evolutionary algorithm.", In *Proceedings of the international conference on systems, man and cybernetics* (pp. 6–9). IEEE Press.
- Hurink, E., Jurisch, B., & Thole, M. (1994), "Tabu search for the job shop scheduling problem with multi-purpose machines.", *Operations Research Spectrum*, 15, 205–215.
- Jia, H. Z., Nee, A. Y. C., Fuh, J. Y. H., & Zhang, Y. F. (2003), "A modified genetic algorithm for distributed scheduling problems. *International Journal of Intelligent Manufacturing*", 14, 351–362.
- Kacem, I. (2003). "Genetic algorithm for the flexible job-shop scheduling problem. *IEEE International conference on Systems, Man and Cybernetics*, 4, 3464–3469.

- Kacem, I., Hammadi, S., e Borne, P. (2002). “Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic.”, *Mathematics and Computers in Simulation*, 60, 245–276.
- Kacem, I., Hammadi, S., e Borne, P. (2002), “Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems”. *IEEE Transactions on Systems, Man and Cybernetics*, 32(1), 1–13.
- Li, X. e Gao, L. (2016), “An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem”, *International Journal of Production Economics*, vol. 174, pp. 93–110.
- Li Y. e Zhou G., (2014), “The flexible job shop scheduling based on ATC and GATS hybrid algorithm,” in *Proceedings of the 2014 IEEE International Conference on Information and Automation, ICIA 2014*, pp. 860–864, China.
- Lustosa, L., Mesquita, M. A., Quelhas, O. e Oliveira, R.J. (2008), “Planejamento e Controle da Produção.”, Rio de Janeiro: Elsevier: Campus.
- Mastrolilli M, Gambardella LM. (1996), “Effective neighbourhood functions for the flexible job shop problem. *Journal of Scheduling*”, volume 3, pp. 3–20.
- Meeran, S.; Jain, A. (1999), “Deterministic Job Shop Scheduling: Past, present and future”. UK: University of Dundee.
- Padmavathi K. e Priyanka Y. (2017), “Crossover Operators in Genetic Algorithms: A Review”, *International Journal of Computer Applications (0975 – 8887)*, Volume 162
- Paulli, J. (1995), “A hierarchical approach for the FMS scheduling problem. *European Journal of Operational Research*”, 86(1), 32–42.
- Pezzella, F., Morganti, G., & Ciaschetti, G. (2007), “A genetic algorithm for the flexible job-shop scheduling problem.”, *Computers and Operations Research*, 35(10), 3202–3212.
- Pinedo, M. (1984), “Scheduling: Theory, Algorithms, and Systems”. New York: Springer Science + Business Media.

- Pinedo M. (2002), “Scheduling: Theory, algorithms and systems.”, Englewood cliffs, NJ: Prentice Hall.
- Reis, J. (1996) “Uma introdução ao Scheduling”, ISCTE-DCTI.
- Reeves, C.R.; Beasley, J.E. (1993), “Modern Heuristic Techniques for Combinatorial Problems”, Blackwell Scientific Publications.
- Rinnooy Kan, A.H.G. (1976), “Machine Scheduling Problems, Classification, Complexity and Computations”, Martinus Nijhoff, The Hague.
- Stevenson M., Hendry, L. C. e Kingsman B. G. (2005), “A review of production planning and control: the applicability of key concepts to the make-to-order industry”, International Journal of Production.
- Suresh, K.N., Sridhan, R. (2007), “Simulation modeling and analysis of tool sharing and part Scheduling decisions in single-stage multimachine flexible manufacturing systems”, Robotics and Computer Integrated Manufacturing.
- Talbi E. (2009), “Metaheuristics: from design to implementation.”, John Wiley & Sons, Hoboken
- Tay, J. C., & Wibowo, D. (2004). “An effective chromosome representation for evolving flexible job shop schedules”, GECCO 2004, Lecture notes in computer science (Vol. 3103, pp. 210–221). Berlin: Springer.
- Tubino, D. F. (2007), “Planejamento e controle da produção: teoria e prática”, São Paulo: Atlas.
- Umbarkar A.J. e Sheth P.D. (2015), “Crossover operators in genetic algorithms: a review”, Ictact journal on soft computing, volume: 06.
- Yang, J. B. (2001), “GA-based discrete dynamic programming approach for scheduling in FMS environments.”, IEEE Transaction on Systems, Man, and Cybernetics, Part B, 31(5), 824–835.
- Zhang G., Gao L., e Shi Y. (2010), “A novel variable neighborhood genetic algorithm for multi-objective flexible job-shop scheduling problems,” Advanced Materials Research, vol. 118-120, pp. 369–373, 2010.

Zhang G., Gao L., e Shi Y., (2010) “A genetic algorithm and tabu search for multi objective flexible job shop scheduling problems,” in Proceedings of the 1st International Conference on Computing Control and Industrial Engineering, CCIE 2010, pp. 251–254, China.

## ANEXO A

A2-Par-Filler					
Largura	Vel ml/min	Car.	m2/hora	m2/min	m2/seg
70	12	1	50,4	0,84	0,014
90	12	1	64,8	1,08	0,018
120	12	1	86,4	1,44	0,024
140	12	1	100,8	1,68	0,028
180	12	1	129,6	2,16	0,036
220	12	1	158,4	2,64	0,044
250	12	1	180	3	0,05
A3-Lixadora					
Largura	Vel ml/min	Car.	m2/hora	m2/min	m2/seg
70	15	8	504	8,4	0,14
90	15	7	567	9,45	0,1575
120	15	6	648	10,8	0,18
140	15	6	756	12,6	0,21
180	15	5	810	13,5	0,225
220	15	4	792	13,2	0,22
250	15	4	900	15	0,25
B1-Escovadora					
Largura	ml/min	Car.	m2/hora	m2/min	m2/seg
70	8	4	134,4	2,24	0,037333333
90	8	3	129,6	2,16	0,036
120	8	3	172,8	2,88	0,048
140	8	3	201,6	3,36	0,056
180	8	2	172,8	2,88	0,048
220	8	2	211,2	3,52	0,058666667
250	8	2	240	4	0,066666667

## ANEXO B

B2-Homag			
Macho	Esp.	Vel	
2	10-11.5mm	45	
3	12/13mm	40	
3	15/17mm	35	
6	17.5/21mm	25	
B2-Homag			
Macho/Largura	2,5	4	6
70	0,053	0,047	0,029
90	0,068	0,060	0,038
120	0,090	0,080	0,050
140	0,105	0,093	0,058
180	0,135	0,120	0,075
220	0,165	0,147	0,092
250	0,188	0,167	0,104
B2-Homag			
Qualidade			
rustic	(quantidade/100) x 15 min		
light	(quantidade/100) x 10 min		
select	(quantidade/100) x 10 min		
natur	(quantidade/100) x 10 min		
outra	20 min		

## ANEXO C

C1-Linha de Verniz					
	ml/min	Car.	m2/hora	m2/min	m2/seg
<b>70</b>	5	10	210	3,5	0,058333333
<b>90</b>	5	9	243	4,05	0,0675
<b>120</b>	8	6	346	5,76	0,096
<b>140</b>	8	6	403	6,72	0,112
<b>180</b>	8	5	432	7,2	0,12
<b>220</b>	8	4	422	7,04	0,117333333
<b>250</b>	8	3	360	6	0,1
C2-Linha de cores					
	ml/min	Car.	m2/hora	m2/min	m2/seg
<b>70</b>	10	5	210	3,5	0,058333333
<b>90</b>	10	4	216	3,6	0,06
<b>120</b>	10	3	216	3,6	0,06
<b>140</b>	10	3	252	4,2	0,07
<b>180</b>	10	2	216	3,6	0,06
<b>220</b>	10	2	264	4,4	0,073333333
<b>250</b>	10	2	300	5	0,083333333
B3 - Chevron					
				m2/min	m2/seg
<b>70</b>				0	0
<b>90</b>				0,18	0,003
<b>120</b>				0,23	0,003833333
<b>140</b>				0,27	0,0045
<b>180</b>				0	0
<b>220</b>				0	0
<b>250</b>				0	0

## ANEXO D

CODIGO	PRODUTO	ABREVPRODUTO
0A	LAMELAS (Produção)	Mercadorias
0B	RIPADO	Maciço
0C	CUTELO	Maciço
0D	DECK	Maciço
0F	FOLHINHA	Folhinha
0H	S4S	Maciço
0J	S2S	Maciço
0L	LAMPARQUET	Maciço
0P	RODAPÉ	Maciço
0R	RÉGUA	Maciço
0S	SOALHO	Maciço
1A	BÉTULA FSC MIX CREDIT, BV-COC-XXXXX	Mercadorias
1B	CP BETULA	Mercadorias
1C	CP CARB	Mercadorias
1D	CP CORTIÇA	Mercadorias
1F	CP FENOLICO	Mercadorias
1G	CP GLOBALPLEX	Mercadorias
1H	HDF	Flutuante
1P	CP POPLER	Mercadorias
2B	FL BET RASG	Flutuante
2C	FL CARB RASG	Flutuante
2D	FL CORT RASG	Flutuante
2F	FL FEN RASG	Flutuante
2G	FL GLO RASG	Flutuante
2P	FL POP RASG	Flutuante
3B	FL BET S RASG	Flutuante
3C	FL CARB S RASG	Flutuante
3D	FL CORT S RASG	Flutuante
3F	FL FEN S RASG	Flutuante
3G	FL GLO S RASG	Flutuante
3P	FL POP S RASG	Flutuante
BA	BARROTOS (MP)	Mercadorias
FG	FOLHINHA GAR	Folhinha
FL	FOLHA	Folhinha
GR	Grampos	Mercadorias
LA	LAMELAS (MP)	Mercadorias
PA	Parafusos	Mercadorias
PE	Pellets	Mercadorias
PR	PRANCHAS (MP)	Mercadorias
RE	RÉGUAS BRUTO (MP)	Mercadorias



## ANEXO E

CODIGO	QUALIDADE	ABREVQUALIDADE	A2	Carga
?			NÃO	
0			NÃO	
1	EXTRA	EXTRA	NÃO	
2	STD	STD	NÃO	
3	VAP	VAP	NÃO	
4	CLARO	CLARO	NÃO	
5	MÉDIO	MÉDIO	NÃO	
6	1º BIS	1º BIS	SIM	50%
7	AMARELO	AMARELO	NÃO	
8	A/B	A/B	NÃO	
9	WBP	WBP	NÃO	
A	RADIAL	RADIAL	NÃO	
B	TANGENCIAL	TANGEN	NÃO	
C	RUSTIC	RUSTIC	SIM	100%
D	D-C/D	D-C/D	SIM	100%
E	EXTRA LIGHT	EX LIGHT	NÃO	
F	SUPER SELECT	S.SELECT	NÃO	
G	BORNE (A/B)	BORNE-AB	NÃO	
H	A	A	NÃO	
I	CC/CP	CC/CP	SIM	100%
K	DARK	DARK	NÃO	
L	LIGHT	LIGHT	NÃO	
M	MÉDIUM	MÉDIUM	NÃO	
N	NATUR	NATUR	SIM	50%
O	R.A.	R.A.	SIM	100%
P	R.B.	R.B.	SIM	100%
Q	C/D	NÓS/MANC	SIM	100%
R	R	R.	SIM	100%
S	SELECT	SELECT	NÃO	
T	BB/CP	BB/CP	NÃO	
U	D	MANCHA	NÃO	
V	D/R	MAN/GALH	SIM	100%
W	HIDROFUGO	HIDRO	NÃO	
X	NATURAL LIGHT	N.LIGHT	NÃO	
Y	NATURAL MEDIUM	N.MEDIUM	NÃO	
Z	A/B/C	A/B/C	SIM	100%

## ANEXO F

CODIGO	FORMATO		B1	B2	Seq.	B3	Carga
??			NÃO			NÃO	
00			NÃO			NÃO	
01	GO0 LI	F	NÃO	GO0	S2	NÃO	
02	GO2 LI	F	NÃO	GO2	S1	NÃO	
03	GO4 LI	F	NÃO	GO4	S1	NÃO	
04	GO0 E.FR	F	SIM	GO0	S2	NÃO	1
05	GO2 E.FR	F	SIM	GO2	S1	NÃO	1
06	GO4 E.FR	F	SIM	GO4	S1	NÃO	1
07	GO0 E.FO	F	SIM	GO0	S2	NÃO	1,5
08	GO2 E.FO	F	SIM	GO2	S1	NÃO	1,5
09	GO4 E.FO	F	SIM	GO4	S1	NÃO	1,5
0A	AD/FO-BOLEADO	D	NÃO	soalho	ND	NÃO	
0B	AD/FO-Q.V.	D	NÃO	soalho	ND	NÃO	
0D	AD/LI	D	NÃO	soalho	ND	NÃO	
0E	AD/LI C/RASGO	D	NÃO	soalho	ND	NÃO	
0F	AD/LI-BOLEADO	D	NÃO	soalho	ND	NÃO	
0G	LI/LI	D	NÃO	soalho	ND	NÃO	
0H	LI-FO/CORTINA	D	NÃO	soalho	ND	NÃO	
0I	21VISTA(10LAM)	M	NÃO	soalho	ND	NÃO	
0J	17VISTA(12LAM)	M	NÃO	soalho	ND	NÃO	
0K	10VISTA(21LAM)	M	NÃO	soalho	ND	NÃO	
0L	2 FACES	D	NÃO	soalho	ND	NÃO	
0M	1 FACE	D	NÃO	soalho	ND	NÃO	
0N	BOLEADO	D	NÃO	soalho	ND	NÃO	
0O	1RASGO-C/F	D	NÃO	soalho	ND	NÃO	
0P	2RASGOS-C/F	D	NÃO	soalho	ND	NÃO	
0S	AD/LI-PERFIL B	D	NÃO	soalho	ND	NÃO	
0T	AD/FO-BOLEADO	D	NÃO	soalho	ND	NÃO	
0V	AD/FO-PERFIL B	D	NÃO	soalho	ND	NÃO	
0W	AD/FO	D	NÃO	soalho	ND	NÃO	
0X	LI/FO-PERFIL A	D	NÃO	soalho	ND	NÃO	
0Y	LI/FO	D	NÃO	soalho	ND	NÃO	
0Z	LI/LI-C/RASGO	D	NÃO	soalho	ND	NÃO	
10	GO0 LI CX BR	F	NÃO	GO0	S2	NÃO	
11	GO2 LI CX BR	F	NÃO	GO2	S1	NÃO	

12	GO4 LI CX BR	F	NÃO	GO4	S1	NÃO	
13	GO0 E.FR CX BR	F	SIM	GO0	S2	NÃO	1
14	GO2 E.FR CX BR	F	SIM	GO2	S1	NÃO	1
15	GO4 E.FR CX BR	F	SIM	GO4	S1	NÃO	1
16	GO0 E.FO CX BR	F	SIM	GO0	S2	NÃO	1,5
17	GO2 E.FO CX BR	F	SIM	GO2	S1	NÃO	1,5
18	GO4 E.FO CX BR	F	SIM	GO4	S1	NÃO	1,5
19	GO0 LI CX CWF	F	NÃO	GO0	S2	NÃO	
1A	AD/LI-BOLEADO C/RASGO	D	NÃO	soalho	ND	NÃO	
1F	AD/FO-PERFIL C	D	NÃO	soalho	ND	NÃO	
1G	AD/LI-PERFIL B	D	NÃO	soalho	ND	NÃO	
1H	2 FACE GR PLAS	D	NÃO	soalho	ND	NÃO	
1I	GO0 LI F/F CX BR	F	NÃO	GO0	S2	NÃO	
1J	GO2 LI F/F CX BR	F	NÃO	GO2	S1	NÃO	
1K	GO4 LI F/F CX BR	F	NÃO	GO4	S1	NÃO	
1L	GO0 E.FR F/F CX BR	F	SIM	GO0	S2	NÃO	1
1M	GO2 E.FR F/F CX BR	F	SIM	GO2	S1	NÃO	1
1N	GO4 E.FR F/F CX BR	F	SIM	GO4	S1	NÃO	1
1O	GO0 E.FO F/F CX BR	F	SIM	GO0	S2	NÃO	1,5
1P	GO2 E.FO F/F CX BR	F	SIM	GO2	S1	NÃO	1,5
1Q	GO4 E.FO F/F CX BR	F	SIM	GO4	S1	NÃO	1,5
1R	GO0 LI F/F CX CWF	F	NÃO	GO0	S2	NÃO	
1S	GO2 LI F/F CX CWF	F	NÃO	GO2	S1	NÃO	
1T	GO4 LI F/F CX CWF	F	NÃO	GO4	S1	NÃO	
1U	GO0 E.FR F/F CX CWF	F	SIM	GO0	S2	NÃO	1
1V	GO2 E.FR F/F CX CWF	F	SIM	GO2	S1	NÃO	1
1W	GO4 E.FR F/F CX CWF	F	SIM	GO4	S1	NÃO	1
1X	GO0 E.FO F/F CX CWF	F	SIM	GO0	S2	NÃO	1,5
1Y	GO2 E.FO F/F CX CWF	F	SIM	GO2	S1	NÃO	1,5
1Z	GO4 E.FO F/F CX CWF	F	SIM	GO4	S1	NÃO	1,5
20	GO2 LI CX CWF	F	NÃO	GO2	S1	NÃO	
21	GO4 LI CX CWF	F	NÃO	GO4	S1	NÃO	
22	GO0 E.FR CX CWF	F	SIM	GO0	S2	NÃO	1
23	GO2 E.FR CX CWF	F	SIM	GO2	S1	NÃO	1
24	GO4 E.FR CX CWF	F	SIM	GO4	S1	NÃO	1
25	GO0 E.FO CX CWF	F	SIM	GO0	S2	NÃO	1,5
26	GO2 E.FO CX CWF	F	SIM	GO2	S1	NÃO	1,5
27	GO4 E.FO CX CWF	F	SIM	GO4	S1	NÃO	1,5
28	GO0 LI CX RIB	F	NÃO	GO0	S2	NÃO	
29	GO2 LI CX RIB	F	NÃO	GO2	S1	NÃO	
2P	LI/LI- FO-2FACES-BOLEADO	D	NÃO	soalho	ND	NÃO	

30	GO4 LI CX RIB	F	NÃO	GO4	S1	NÃO	
31	GO0 E.FR CX RIB	F	SIM	GO0	S2	NÃO	1
32	GO2 E.FR CX RIB	F	SIM	GO2	S1	NÃO	1
33	GO4 E.FR CX RIB	F	SIM	GO4	S1	NÃO	1
34	GO0 E.FO CX RIB	F	SIM	GO0	S2	NÃO	1,5
35	GO2 E.FO CX RIB	F	SIM	GO2	S1	NÃO	1,5
36	GO4 E.FO CX RIB	F	SIM	GO4	S1	NÃO	1,5
39	SÓ TRAÇADO	NA	NÃO	soalho	ND	NÃO	
40	TACO LISO	F	NÃO	soalho	ND	NÃO	
41	SÓ COLADO	NA	NÃO	soalho	ND	NÃO	
42	GO0 C/ DEF	NA	NÃO	GO0	S2	NÃO	
43	GO2 C/ DEF	NA	NÃO	GO2	S1	NÃO	
44	GO4 C/ DEF	NA	NÃO	GO4	S1	NÃO	
45	C/ DEFEITO	NA	NÃO	soalho	ND	NÃO	
46	QUINA VIVA	F	NÃO	soalho	ND	NÃO	
51	GO0 LI CX DEC	F	NÃO	GO0	S2	NÃO	
52	GO2 LI CX DEC	F	NÃO	GO2	S1	NÃO	
53	GO4 LI CX DEC	F	NÃO	GO4	S1	NÃO	
54	GO0 E.FR CX DEC	F	SIM	GO0	S2	NÃO	1
55	GO2 E.FR CX DEC	F	SIM	GO2	S1	NÃO	1
56	GO4 E.FR CX DEC	F	SIM	GO4	S1	NÃO	1
57	GO0 E.FO CX DEC	F	SIM	GO0	S2	NÃO	1,5
58	GO2 E.FO CX DEC	F	SIM	GO2	S1	NÃO	1,5
59	GO4 E.FO CX DEC	F	SIM	GO4	S1	NÃO	1,5
60	GO0 LI HER.MLD	F	NÃO	GO0	S2	NÃO	
61	GO0 LI HER.MLE	F	NÃO	GO0	S2	NÃO	
62	GO2 LI HER.MLD	F	NÃO	GO2	S1	NÃO	
63	GO2 LI HER.MLE	F	NÃO	GO2	S1	NÃO	
64	GO4 LI HER.MLD	F	NÃO	GO4	S1	NÃO	
65	GO4 LI HER.MLE	F	NÃO	GO4	S1	NÃO	
66	GO0 E.FR HER.MLD	F	SIM	GO0	S2	NÃO	1
67	GO0 E.FR HER.MLE	F	SIM	GO0	S2	NÃO	1
68	GO2 E.FR HER.MLD	F	SIM	GO2	S1	NÃO	1
69	GO2 E.FR HER.MLE	F	SIM	GO2	S1	NÃO	1
70	GO4 E.FR HER.MLD	F	SIM	GO4	S1	NÃO	1
71	GO4 E.FR HER.MLE	F	SIM	GO4	S1	NÃO	1
72	GO0 E.FO HER.MLD	F	SIM	GO0	S2	NÃO	1,5
73	GO0 E.FO HER.MLE	F	SIM	GO0	S2	NÃO	1,5
74	GO2 E.FO HER.MLD	F	SIM	GO2	S1	NÃO	1,5
75	GO2 E.FO HER.MLE	F	SIM	GO2	S1	NÃO	1,5

76	GO4 E.FO HER.MLD	F	SIM	GO4	S1	NÃO	1,5
77	GO4 E.FO HER.MLE	F	SIM	GO4	S1	NÃO	1,5
78	GO0 LI C45º-MLD	F	NÃO	GO0	S2	SIM	
79	GO0 LI C45º-MLE	F	NÃO	GO0	S2	SIM	
80	GO2 LI C45º-MLD	F	NÃO	GO2	S1	SIM	
81	GO2 LI C45º-MLE	F	NÃO	GO2	S1	SIM	
82	GO0 E.FR C45º-MLD	F	SIM	GO0	S2	SIM	1
83	GO0 E.FR C45º-MLE	F	SIM	GO0	S2	SIM	1
84	GO2 E.FR C45º-MLD	F	SIM	GO2	S1	SIM	1
85	GO2 E.FR C45º-MLE	F	SIM	GO2	S1	SIM	1
86	GO0 E.FO C45º-MLD	F	SIM	GO0	S2	SIM	1,5
87	GO0 E.FO C45º-MLE	F	SIM	GO0	S2	SIM	1,5
88	GO2 E.FO C45º-MLD	F	SIM	GO2	S1	SIM	1,5
89	GO2 E.FO C45º-MLE	F	SIM	GO2	S1	SIM	1,5
90	SÓ/COLADO C/DEF	NA	NÃO	soalho	ND	NÃO	
91	F/F GO0 LI	F	NÃO	GO0	S2	NÃO	
92	F/F GO2 LI	F	NÃO	GO2	S1	NÃO	
93	F/F GO4 LI	F	NÃO	GO4	S1	NÃO	
94	F/F GO0 E.FR	F	SIM	GO0	S2	NÃO	1
95	F/F GO2 E.FR	F	SIM	GO2	S1	NÃO	1
96	F/F GO4 E.FR	F	SIM	GO4	S1	NÃO	1
97	LI/LI GO0 M/F TOPOS	D	NÃO	GO0	S2	NÃO	
98	LI/LI BOLEADO s/RASGO MF TOPO	D	NÃO	soalho	ND	NÃO	
99	Colado e alinhado	NA	NÃO	soalho	ND	NÃO	

## ANEXO H

CODIGO	COR	C2	Carga	C1	Carga	C2	C1
00	NATURAL			1		NÃO	SIM
01	NATURAL			1		NÃO	SIM
02	ALEXANDRIA	1		2		SIM	SIM
03	OLÍMPIA			1		NÃO	SIM
04	RHODES	1		2		SIM	SIM
05	GIZA	1		2		SIM	SIM
06	EPHESUS	1		2		SIM	SIM
07	BABYLON	1		2		SIM	SIM
08	HALICARNASSUS	1		2		SIM	SIM
09	PISA	1		2		SIM	SIM
0A	GRAND CANYON	1	2			SIM	NÃO
0B	EVEREST	1				SIM	NÃO
0C	AURORA	1	2			SIM	NÃO
0D	Rio de Janeiro (02 Pass.)	1	2			SIM	NÃO
0E	Great Barrier	1				SIM	NÃO
0F	Particutin (Cast.349)	1				SIM	NÃO
0G	Vitoria Falls (Preto 120)	1				SIM	NÃO
0H	Oil 349(Porto)	1				SIM	NÃO
0I	Oil branco 1582	1				SIM	NÃO
0J	Oil Extra grey 314	1				SIM	NÃO
0K	Oil Exterior 617911	1	2			SIM	NÃO
0L	Oil Metalic Bronze 358	1				SIM	NÃO
0M	OIL Umbra	1				SIM	NÃO
0N	OIL Extra White 384	1				SIM	NÃO
10	Hagia Sophia	1		2		SIM	SIM
11	Coliseum	1		2		SIM	SIM
12	Stonehenge	1		2		SIM	SIM
13	Great Wall			1	2	NÃO	SIM
14	Shoqafa	1		2		SIM	SIM
15	Nanjing	1		2		SIM	SIM
16	Cristo Rei	1		2		SIM	SIM
17	Petra	1		2		SIM	SIM
18	Taj Mahal			1		NÃO	SIM
19	Chichen Itza	1		2		SIM	SIM
1A	Autoban Color	1		2		SIM	SIM

1B	Elite Smoked ASTRA118	1		2		SIM	SIM
1C	Rovera OAK	1		2		SIM	SIM
1D	Cor Gobe	1		2		SIM	SIM
1E	White Grey	1		2		SIM	SIM
1F	Limed	1		2		SIM	SIM
1G	Toscana	1		2		SIM	SIM
1H	Smoked/ASTRA110	1		2		SIM	SIM
1I	VEL MAS DL 11187					NÃO	NÃO
1J	11/2016 Wen			1		NÃO	SIM
1K	Wen Preto			1		NÃO	SIM
1L	Teka			1		NÃO	SIM
1M	Polar White					NÃO	NÃO
1N	RK26			1		NÃO	SIM
1P	Napoli					NÃO	NÃO
1Q	Astra 122	1		2		SIM	SIM
1R	Firenze					NÃO	NÃO
1S	Davos	1		2		SIM	SIM
1T	Negro RWUVR0030			1		NÃO	SIM
1U	DL 11194 Iro médio			1		NÃO	SIM
1W	Tek Médio RWUVR0035			1		NÃO	SIM
1X	28/2016	1		2		SIM	SIM
1Y	Tiffany 2			1		NÃO	SIM
1Z	40/2016 - A.S.	1		2		SIM	SIM
20	Machu Pichu	1	2	2		SIM	SIM
21	China Wall	1		2		SIM	SIM
22	Rome	1		2		SIM	SIM
23	Pena	1		2		SIM	SIM
24	Guimarães	1		2		SIM	SIM
25	Vimaranes	1		2		SIM	SIM
26	Óbidos	1		2		SIM	SIM
27	Batalha	1		2		SIM	SIM
28	Alcobaça	1		2		SIM	SIM
29	Jerónimo	1		2		SIM	SIM
2A	32/2016	1		2		SIM	SIM
2B	Toledo 21/2016			1		NÃO	SIM
2C	Andaluzia XL (49/2016)	1		2		SIM	SIM
2D	66/2016	1		2		SIM	SIM
2E	Dark Brown			1		NÃO	SIM

2F	56/2016	1		2		SIM	SIM
2G	79/2016 XYZ	1		2		SIM	SIM
2H	77/2016 Mystic	1		2		SIM	SIM
2I	81/2016	1		2		SIM	SIM
2J	84/2016 AD9	1		2		SIM	SIM
2k	69/2016	1		2		SIM	SIM
2L	Sicilian Grigio - 86/2016	1		2		SIM	SIM
2M	AD9 - 97/2016-A	1		2		SIM	SIM
2N	80/2016 - JOHAN	1		2		SIM	SIM
2O	38/2016	1		2		SIM	SIM
2P	64/2016	1		2		SIM	SIM
2Q	65/2016	1		2		SIM	SIM
2R	93/2016 B	1		2		SIM	SIM
2S	96/2016 E	1		2		SIM	SIM
2T	88/2016 F	1		2		SIM	SIM
2U	94/2016 G - PROVENCE	1		2		SIM	SIM
2V	99/2016 H - NEW	1		2		SIM	SIM
2W	91/2016 I - LA FORET	1		2		SIM	SIM
2X	90/2016 J - CHOCOLATE NOIRE	1		2		SIM	SIM
2Y	92/2016 K	1		2		SIM	SIM
2Z	Astra 149	1		2		SIM	SIM
30	Belém	1		2		SIM	SIM
31	Bay of Fundy			1		NÃO	SIM
32	Yosemite			1		NÃO	SIM
33	Everglades	1		2		SIM	SIM
34	Mount Mckinley	1		2		SIM	SIM
35	Niagara Falls	1		2		SIM	SIM
36	Redwoods	1				SIM	NÃO
37	Yellowstone	1				SIM	NÃO
38	Kakadu (Cor Natural)			1		NÃO	SIM
39	Sydney (Ice White)			1		NÃO	SIM
3A	83/2016			1		NÃO	SIM
3B	41/2016	1		2		SIM	SIM
3C	100/2016 (Astra 153)	1		2		SIM	SIM
3D	101/2016	1		2		SIM	SIM
3E	Venécia	1		2		SIM	SIM
3F	RK RUVR0027			1		NÃO	SIM
3G	Astra 108	1		2		SIM	SIM



3H	Stockolm	1		2		SIM	SIM
3I	Gdansk	1		2		SIM	SIM
3J	Riga	1		2		SIM	SIM
3K	Helsinki	1		2		SIM	SIM
3L	St. Petersburg	1		2		SIM	SIM
3M	Vilnius	1		2		SIM	SIM
3N	Talin	1		2		SIM	SIM
3O	Granada	1		2		SIM	SIM
3P	Barcelona	1		2		SIM	SIM
3Q	Madrid	1		2		SIM	SIM
3R	Segovia	1		2		SIM	SIM
3S	Cordoba	1		2		SIM	SIM
3T	Santiago	1		2		SIM	SIM
3U	02/2017			1		NÃO	SIM
3V	15/2017					NÃO	NÃO
3W	78/2016					NÃO	NÃO
3X	05/2017					NÃO	NÃO
3Y	57/2016	1		2		SIM	SIM
3Z	19/2017 (Astra 155)	1		2		SIM	SIM
40	Uluru			1		NÃO	SIM
41	Bungle Bungles (AFZ)			1		NÃO	SIM
42	Great Reef (IRO)			1		NÃO	SIM
43	Tasmania (TB)			1		NÃO	SIM
44	Ocean Road			1		NÃO	SIM
45	Silver White	1		2		SIM	SIM
46	Silver Black	1		2		SIM	SIM
47	Zebra White	1		2		SIM	SIM
48	Zebra Brown	1		2		SIM	SIM
49	Coal Black	1		2		SIM	SIM
4A	27/2017	1		2		SIM	SIM
4B	39/2017					NÃO	NÃO
4C	30/2017 - AERO					NÃO	NÃO
4D	31/2017 - Shore					NÃO	NÃO
4E	Artico (29/2017)					NÃO	NÃO
4F	35/2017 - Lunar White	1		2		SIM	SIM
4G	47/2017			1		NÃO	SIM
4H	28/2017			1		NÃO	SIM
4I	Parati (46/2017)					NÃO	NÃO

4J	53/2017					NÃO	NÃO
4K	10/2016					NÃO	NÃO
4L	52/2017- Murano	1	2	2		SIM	SIM
4M	50/2017 - Reiner	1	2	2		SIM	SIM
4N	08/2016	1		2		SIM	SIM
4O	43/2017 (Astra 157)	1		2		SIM	SIM
4P	04/2017	1		2		SIM	SIM
4Q	49/2016 (Astra 134)	1		2		SIM	SIM
4R	60/2017	1		2		SIM	SIM
4S	61/2017					NÃO	NÃO
4T	62/2017	1		2		SIM	SIM
4U	102/2016	1		2		SIM	SIM
4V	71/2017 - CAFÉ AU LAIT	1		2		SIM	SIM
4W	67/2017 - MAISON	1		2		SIM	SIM
4X	68/2017 - BASTILLE	1		2		SIM	SIM
4Y	64/2017					NÃO	NÃO
4Z	14/2017					NÃO	NÃO
50	Hazel Brown			1		NÃO	SIM
51	Almond Brown			1		NÃO	SIM
52	AFZ Rec. RWUVR0026			1		NÃO	SIM
53	AFZ Rec. Jat. DL-10736			1		NÃO	SIM
54	Artic White			1		NÃO	SIM
55	WF11			1		NÃO	SIM
56	WF12			1		NÃO	SIM
57	WF13			1		NÃO	SIM
58	WF14			1		NÃO	SIM
59	WF16 C			1		NÃO	SIM
5A	56/2017	1		2		SIM	SIM
5B	76/2017	1	2	2		SIM	SIM
5C	78/2017 - BROADWAY	1		2		SIM	SIM
5D	37/2017					NÃO	NÃO
5E	85/2017 - CLINTON	1		2		SIM	SIM
5F	90/2017					NÃO	NÃO
5G	91/2017	1		2		SIM	SIM
5H	63/2017 -1500 HARBOUR SIDE	1		2		SIM	SIM
5I	TAUARI B			1		NÃO	SIM
5J	Teka Light			1		NÃO	SIM
5K	TEKA DARK			1		NÃO	SIM

5L	02/2018					NÃO	NÃO
5M	RK 0090			1		NÃO	SIM
5N	Mel 131	1		2		SIM	SIM
5O	88/2017					NÃO	NÃO
5P	03/2018 LINK	1		2		SIM	SIM
5Q	32/2017 PID	1		2		SIM	SIM
5R	Dubai	1		2		SIM	SIM
5S	New York	1		2		SIM	SIM
5T	London	1		2		SIM	SIM
5U	Miami	1		2		SIM	SIM
5V	Singapore	1		2		SIM	SIM
5W	Las Vegas			1		NÃO	SIM
5X	Shangai			1		NÃO	SIM
5Y	11/2018					NÃO	NÃO
5Z	31/2016					NÃO	NÃO
60	WF17			1		NÃO	SIM
61	WF18			1		NÃO	SIM
62	Creme WD FLR-03	1		2		SIM	SIM
63	Montblanc	1		2		SIM	SIM
64	Tiffany			1		NÃO	SIM
65	WD FLR-01			1		NÃO	SIM
66	Alexa. Cypres			1		NÃO	SIM
67	Flor de sal	1		2		SIM	SIM
68	Terre de Sienne	1		2		SIM	SIM
69	RESC 003	1		2		SIM	SIM
6A	14/2018					NÃO	NÃO
6B	15/2018					NÃO	NÃO
6C	19/2018 - Parker house	1		2		SIM	SIM
6D	08/2017	1		2		SIM	SIM
6E	42/2017	1		2		SIM	SIM
6F	30/2018 (AUTOBAN ALTERADO)	1		2		SIM	SIM
6G	80/2017	1		2		SIM	SIM
6H	24/2018					NÃO	NÃO
6I	17/2017					NÃO	NÃO
6J	28/2018 - Choupo St Petersburg			1		NÃO	SIM
70	Porto	1		2		SIM	SIM
71	Mel (Nadav)	1		2		SIM	SIM
72	Preto (Tagus)	1		2		SIM	SIM

73	Branco claro (Alva)			1		NÃO	SIM
74	Castanho (Douro)	1		2		SIM	SIM
75	Castanho (Mondego)	1		2		SIM	SIM
76	(Mondego) Chocolate	1		2		SIM	SIM
77	Cinza (Liz, cor antiga)	1		2		SIM	SIM
78	Cinza (Liz, cor nova )	1		2		SIM	SIM
79	White Washed	1		2		SIM	SIM
80	Sable (WF11)	1		2		SIM	SIM
81	Gris Pearl (Vodf)	1		2		SIM	SIM
82	Crema (mel cortado)	1		2		SIM	SIM
83	Cognac (WF14)			1		NÃO	SIM
84	Desert cêruse (Alexandria)	1		2		SIM	SIM
85	Athena pourpre (Rhodes)	1		2		SIM	SIM
86	Montblanc II	1		2		SIM	SIM
87	Ice White zebra	1		2		SIM	SIM
88	Fune	1		2		SIM	SIM
89	Amsterdan	1		2		SIM	SIM
90	Cinza Macedo	1		2		SIM	SIM
91	Cinza com veio negro	1	2	2		SIM	SIM
92	China Wall Dubai	1		2		SIM	SIM
93	Chocolate	1		2		SIM	SIM
94	Niebla	1		2		SIM	SIM
95	Granito	1	2	2		SIM	SIM
96	Cor mel Ma.	1		2		SIM	SIM
97	Cor Cinza Ma.	1		2		SIM	SIM
98	Cor branco claro Ma.	1		2		SIM	SIM
99	Oak Milano	1		2		SIM	SIM
6W	65/2018 - Lagrange	1		2		SIM	SIM
6X	61/2016 - Nizza	1		2		SIM	SIM
7C	13/2018	1		2		SIM	SIM
0P	05/19	1		2		SIM	SIM
0Q	44/2018 - Santana	1		2		SIM	SIM
0R	38/2018 - Santa Maria			1		NÃO	SIM
0S	43/2018 - Sao Vicente					NÃO	NÃO
0T	39/2019 - Camacha	1		2		SIM	SIM
0U	22/2019 - Serra					NÃO	NÃO
6L	61/2018 - castle	1		2		SIM	SIM

Erro! Fonte de referência não encontrada.

---

00	Carlos					NÃO	NÃO
6S	39/2018	1		2		SIM	SIM
7G	37/2019	1		2		SIM	SIM
7K	21/2019 - Machico	1		2		SIM	SIM

## ANEXO I

<b>Job</b>	<b>Vetores de entrada no algoritmo</b>
1	3 2 3 3538 4 3538 1 9 8256 1 6 10179
2	3 2 3 7968 4 7968 1 9 19123 1 6 25767
3	3 2 3 3435 4 3435 1 9 8016 1 6 9892
4	6 2 3 1008 4 1008 1 5 4725 1 6 2170 1 7 3780 1 8 120 1 9 1890
5	6 2 3 1008 4 1008 1 5 4725 1 6 2170 1 7 3780 1 8 120 1 9 1890
6	6 2 3 1008 4 1008 1 5 4725 1 6 2170 1 7 3780 1 8 120 1 9 1890
7	4 2 3 1512 4 1512 1 5 4725 1 6 3172 1 9 2835
8	4 2 3 1512 4 1512 1 5 4725 1 6 3172 1 9 2835
9	4 2 3 1512 4 1512 1 5 4725 1 6 3172 1 9 2835
10	3 2 3 1512 4 1512 1 5 4725 1 6 3172
11	3 2 3 1512 4 1512 1 5 4725 1 6 3172
12	3 2 3 1512 4 1512 1 5 4725 1 6 3172
13	6 2 3 1008 4 1008 1 5 3150 1 6 2170 1 7 3780 1 8 120 1 9 1890
14	6 2 3 1008 4 1008 1 5 3150 1 6 2170 1 7 3780 1 8 120 1 9 1890
15	6 2 3 1008 4 1008 1 5 3150 1 6 2170 1 7 3780 1 8 120 1 9 1890
16	3 2 3 1512 4 1512 1 5 4725 1 6 3172
17	3 2 3 1512 4 1512 1 5 4725 1 6 3172
18	3 2 3 1512 4 1512 1 5 4725 1 6 3172
19	6 2 3 1512 4 1512 1 5 4725 1 6 3172 1 7 5670 1 8 120 1 9 2835
20	6 2 3 1512 4 1512 1 5 4725 1 6 3172 1 7 5670 1 8 120 1 9 2835
21	6 2 3 1512 4 1512 1 5 4725 1 6 3172 1 7 5670 1 8 120 1 9 2835
22	4 2 3 1008 4 1008 1 5 3150 1 6 2170 1 9 1890
23	4 2 3 1008 4 1008 1 5 3150 1 6 2170 1 9 1890
24	4 2 3 1008 4 1008 1 5 3150 1 6 2170 1 9 1890
25	5 1 1 8833 1 2 600 2 3 476 4 476 1 6 1658 2 10 25000 11 25000
26	5 1 1 8833 1 2 600 2 3 476 4 476 1 6 1658 2 10 25000 11 25000
27	5 2 3 84 4 84 1 6 334 1 7 315 1 8 120 1 9 158
28	3 2 3 29 4 29 1 6 519 1 9 69
29	3 2 3 36 4 36 1 6 542 1 9 86
30	3 2 3 28 4 28 1 6 316 1 9 52
31	4 1 1 5889 1 2 600 2 3 889 4 889 1 6 2032
32	4 1 1 7227 1 2 600 2 3 1364 4 1364 1 6 2493
33	4 1 1 3666 1 2 600 2 3 461 4 461 1 6 1265
34	4 1 1 3666 1 2 600 2 3 461 4 461 1 6 1265
35	4 1 1 3666 1 2 600 2 3 461 4 461 1 6 1265
36	8 1 1 8093 1 2 600 2 3 1222 4 1222 1 5 3818 1 6 2792 1 7 4581 1 8 120 1 9 2291
37	8 1 1 8093 1 2 600 2 3 1222 4 1222 1 5 3818 1 6 2792 1 7 4581 1 8 120 1 9 2291
38	8 1 1 3053 1 2 600 2 3 461 4 461 1 5 1440 1 6 1053 1 7 1728 1 8 120 1 9 864
39	6 1 1 15671 1 2 600 2 3 2365 4 2365 1 5 11088 1 6 5406 1 9 4435
40	6 1 1 19233 1 2 600 2 3 2074 4 2074 1 5 9072 1 6 6635 1 9 4838
41	8 1 1 37397 1 2 600 2 3 5645 4 5645 1 5 17640 1 6 12902 1 7 21168 1 8 120 1 9 10584
42	6 1 1 1973 1 2 600 2 3 298 4 298 1 5 1396 1 6 681 1 9 558
43	6 1 1 27236 1 2 600 2 3 2937 4 2937 1 5 12847 1 6 15034 1 9 6852

44	6 1 1 20427 1 2 600 2 3 2569 4 2569 1 5 9635 1 6 11276 1 9 4818
45	6 1 1 17509 1 2 600 2 3 2202 4 2202 1 5 8259 1 6 9665 1 9 4129
46	6 1 1 13618 1 2 600 2 3 2056 4 2056 1 5 9635 1 6 7517 1 9 3854
47	5 1 1 14722 1 2 600 2 3 2222 4 2222 1 5 10417 1 6 5079
48	8 1 1 294 1 2 600 2 3 44 4 44 1 5 208 1 6 102 1 7 167 1 8 120 1 9 83
49	7 1 1 5889 1 2 600 2 3 889 4 889 1 5 4167 1 6 2032 1 7 1667 1 8 120
50	5 1 1 663 1 2 600 2 3 83 4 83 1 5 313 1 6 229
51	7 1 1 6928 1 2 600 2 3 871 4 871 1 5 3268 1 6 2390 1 7 1307 1 8 120
52	7 1 1 2944 1 2 600 2 3 444 4 444 1 5 2083 1 6 1016 1 7 833 1 8 120
53	7 1 1 4417 1 2 600 2 3 667 4 667 1 5 3125 1 6 1524 1 7 1250 1 8 120
54	6 1 1 624 1 2 600 2 3 67 4 67 1 5 295 1 6 215 1 9 157
55	6 1 1 624 1 2 600 2 3 67 4 67 1 5 295 1 6 215 1 9 157
56	6 1 1 624 1 2 600 2 3 67 4 67 1 5 295 1 6 215 1 9 157
57	8 1 1 1251 1 2 600 2 3 189 4 189 1 5 590 1 6 432 1 7 708 1 8 120 1 9 354
58	7 1 1 6704 1 2 600 2 3 1012 4 1012 1 5 4743 1 6 2313 1 7 3795 1 8 120
59	7 1 1 6704 1 2 600 2 3 1012 4 1012 1 5 4743 1 6 2313 1 7 3795 1 8 120
60	8 1 1 294 1 2 600 2 3 32 4 32 1 5 139 1 6 102 1 7 83 1 8 120 1 9 74
61	8 1 1 294 1 2 600 2 3 32 4 32 1 5 139 1 6 102 1 7 83 1 8 120 1 9 74
62	8 1 1 294 1 2 600 2 3 32 4 32 1 5 139 1 6 102 1 7 83 1 8 120 1 9 74
63	5 1 1 13108 1 2 600 2 3 1413 4 1413 1 6 4522 1 9 3298
64	8 1 1 9617 1 2 600 2 3 1037 4 1037 1 5 4536 1 6 3318 1 7 2722 1 8 120 1 9 2419
65	6 1 1 6411 1 2 600 2 3 691 4 691 1 5 3024 1 6 2212 1 9 1613
66	6 1 1 20606 1 2 600 2 3 3110 4 3110 1 5 14580 1 6 7109 1 9 5832
67	8 1 1 1311 1 2 600 2 3 141 4 141 1 5 412 1 6 452 1 7 371 1 8 120 1 9 330
68	8 1 1 1311 1 2 600 2 3 141 4 141 1 5 412 1 6 452 1 7 371 1 8 120 1 9 330
69	8 1 1 1311 1 2 600 2 3 141 4 141 1 5 412 1 6 452 1 7 371 1 8 120 1 9 330
70	6 1 1 6411 1 2 600 2 3 691 4 691 1 5 3024 1 6 2212 1 9 1613
71	8 1 1 6411 1 2 600 2 3 691 4 691 1 5 3024 1 6 2212 1 7 1814 1 8 120 1 9 1613
72	8 1 1 8243 1 2 600 2 3 1244 4 1244 1 5 5832 1 6 2844 1 7 4666 1 8 120 1 9 2333
73	6 1 1 12822 1 2 600 2 3 1382 4 1382 1 5 6048 1 6 4424 1 9 3226
74	6 1 1 3282 1 2 600 2 3 619 4 619 1 5 2322 1 6 1132 1 9 1161
75	8 1 1 7064 1 2 600 2 3 1333 4 1333 1 5 3332 1 6 2437 1 7 3998 1 8 120 1 9 2499
76	8 1 1 7064 1 2 600 2 3 1333 4 1333 1 5 3332 1 6 2437 1 7 3998 1 8 120 1 9 2499
77	8 1 1 7067 1 2 600 2 3 762 4 762 1 5 2222 1 6 2438 1 7 2000 1 8 120 1 9 1778
78	7 1 1 1089 1 2 600 2 3 164 4 164 1 5 771 1 6 376 1 7 617 1 8 120
79	8 1 1 5438 1 2 600 2 3 586 4 586 1 5 1710 1 6 1876 1 7 1539 1 8 120 1 9 1368
80	6 1 1 12364 1 2 600 2 3 1333 4 1333 1 5 5832 1 6 4265 1 9 3110
81	8 1 1 9616 1 2 600 2 3 1210 4 1210 1 5 4536 1 6 3318 1 7 3629 1 8 120 1 9 2268
82	8 1 1 9616 1 2 600 2 3 1210 4 1210 1 5 3024 1 6 3318 1 7 3629 1 8 120 1 9 2268
83	6 1 1 4121 1 2 600 2 3 622 4 622 1 5 2916 1 6 1422 1 9 1166
84	8 1 1 4121 1 2 600 2 3 622 4 622 1 5 2916 1 6 1422 1 7 2333 1 8 120 1 9 1166
85	8 1 1 6182 1 2 600 2 3 933 4 933 1 5 4374 1 6 2133 1 7 3499 1 8 120 1 9 1750
86	8 1 1 8014 1 2 600 2 3 1512 4 1512 1 5 3780 1 6 2765 1 7 4536 1 8 120 1 9 2835
87	8 1 1 2404 1 2 600 2 3 259 4 259 1 5 756 1 6 829 1 7 680 1 8 120 1 9 605
88	6 1 1 6034 1 2 600 2 3 651 4 651 1 5 2846 1 6 2082 1 9 1518
89	6 1 1 6034 1 2 600 2 3 651 4 651 1 5 2846 1 6 2082 1 9 1518
90	6 1 1 6034 1 2 600 2 3 651 4 651 1 5 2846 1 6 2082 1 9 1518
91	8 1 1 4121 1 2 600 2 3 622 4 622 1 5 1944 1 6 1422 1 7 2333 1 8 120 1 9 1166

92	6 1 1 4122 1 2 600 2 3 444 4 444 1 5 1944 1 9 1037 1 6 1422
93	6 1 1 4122 1 2 600 2 3 444 4 444 1 5 1944 1 9 1037 1 6 1422
94	4 1 1 11357 1 2 600 2 3 1429 4 1429 1 6 3918
95	6 1 1 1119 1 2 600 2 3 169 4 169 1 5 792 1 6 386 1 9 317
96	6 1 1 1119 1 2 600 2 3 169 4 169 1 5 792 1 6 386 1 9 317
97	7 1 1 4417 1 2 600 2 3 476 4 476 1 5 2083 1 6 1524 1 7 1250 1 8 120
98	7 1 1 4417 1 2 600 2 3 476 4 476 1 5 2083 1 6 1524 1 7 1250 1 8 120
99	6 1 1 5447 1 2 600 2 3 822 4 822 1 5 3854 1 6 1879 1 9 1542
100	6 1 1 3786 1 2 600 2 3 476 4 476 1 5 1786 1 6 1306 1 9 893



## ANEXO J

<i>Job</i>	<i>Due Dates</i>	
1	2021-07-02	23:59:59
2	2021-07-02	23:59:59
3	2021-07-02	23:59:59
4	2021-07-02	23:59:59
5	2021-07-02	23:59:59
6	2021-07-02	23:59:59
7	2021-07-03	23:59:59
8	2021-07-03	23:59:59
9	2021-07-03	23:59:59
10	2021-07-03	23:59:59
11	2021-07-03	23:59:59
12	2021-07-03	23:59:59
13	2021-07-04	23:59:59
14	2021-07-04	23:59:59
15	2021-07-04	23:59:59
16	2021-07-04	23:59:59
17	2021-07-04	23:59:59
18	2021-07-04	23:59:59
19	2021-07-04	23:59:59
20	2021-07-04	23:59:59
21	2021-07-04	23:59:59
22	2021-07-05	23:59:59
23	2021-07-05	23:59:59
24	2021-07-05	23:59:59
25	2021-07-05	23:59:59
26	2021-07-05	23:59:59
27	2021-07-05	23:59:59
28	2021-07-05	23:59:59
29	2021-07-05	23:59:59
30	2021-07-05	23:59:59
31	2021-07-05	23:59:59
32	2021-07-05	23:59:59
33	2021-07-05	23:59:59
34	2021-07-05	23:59:59
35	2021-07-05	23:59:59
36	2021-07-05	23:59:59
37	2021-07-05	23:59:59
38	2021-07-05	23:59:59
39	2021-07-05	23:59:59
40	2021-07-05	23:59:59
41	2021-07-06	23:59:59
42	2021-07-06	23:59:59
43	2021-07-06	23:59:59
44	2021-07-06	23:59:59

45	2021-07-06	23:59:59
46	2021-07-06	23:59:59
47	2021-07-06	23:59:59
48	2021-07-06	23:59:59
49	2021-07-06	23:59:59
50	2021-07-06	23:59:59
51	2021-07-06	23:59:59
52	2021-07-06	23:59:59
53	2021-07-06	23:59:59
54	2021-07-06	23:59:59
55	2021-07-06	23:59:59
56	2021-07-06	23:59:59
57	2021-07-06	23:59:59
58	2021-07-06	23:59:59
59	2021-07-06	23:59:59
60	2021-07-06	23:59:59
61	2021-07-06	23:59:59
62	2021-07-06	23:59:59
63	2021-07-07	23:59:59
64	2021-07-07	23:59:59
65	2021-07-07	23:59:59
66	2021-07-07	23:59:59
67	2021-07-07	23:59:59
68	2021-07-07	23:59:59
69	2021-07-07	23:59:59
70	2021-07-07	23:59:59
71	2021-07-07	23:59:59
72	2021-07-15	23:59:59
73	2021-07-15	23:59:59
74	2021-07-15	23:59:59
75	2021-07-15	23:59:59
76	2021-07-15	23:59:59
77	2021-07-15	23:59:59
78	2021-07-15	23:59:59
79	2021-07-15	23:59:59
80	2021-07-15	23:59:59
81	2021-07-20	23:59:59
82	2021-07-20	23:59:59
83	2021-07-20	23:59:59
84	2021-07-20	23:59:59
85	2021-07-20	23:59:59
86	2021-07-20	23:59:59
87	2021-07-20	23:59:59
88	2021-07-20	23:59:59
89	2021-07-20	23:59:59
90	2021-07-20	23:59:59
91	2021-07-20	23:59:59

92	2021-07-28	23:59:59
93	2021-07-28	23:59:59
94	2021-07-28	23:59:59
95	2021-07-28	23:59:59
96	2021-07-28	23:59:59
97	2021-07-28	23:59:59
98	2021-07-28	23:59:59
99	2021-07-28	23:59:59
100	2021-07-28	23:59:59