



UNIVERSIDADE D  
COIMBRA

Tiago Miguel Pestana da Silva

**APLICAÇÃO DE UM ALGORITMO DE  
ESCALONAMENTO EM MÁQUINAS PARALELAS  
COM TEMPOS DE SETUP DEPENDENTES DA  
SEQUÊNCIA**

**Dissertação no âmbito do Mestrado em Engenharia e Gestão Industrial, orientada pelo Professor Doutor Samuel de Oliveira Moniz e apresentada ao Departamento de Engenharia Mecânica, da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.**

julho de 2022





FCTUC FACULDADE DE CIÊNCIAS  
E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA

DEPARTAMENTO DE  
ENGENHARIA MECÂNICA

# **Aplicação de um algoritmo de escalonamento em máquinas paralelas com tempos de setup dependentes da sequência**

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

## **Application of a scheduling algorithm on parallel machines with sequence-dependent setup times**

**Autor**

**Tiago Miguel Pestana da Silva**

**Orientador**

**Professor Doutor Samuel de Oliveira Moniz**

**Júri**

**Presidente** Professor Doutor **Cristóvão Silva**  
Professor Auxiliar da Universidade de Coimbra

**Vogais** Professor Doutor **Miguel Vieira**  
Professor Auxiliar da Universidade Lusófona

**Orientador** Professor Doutor **Samuel Moniz**  
Professor Auxiliar da Universidade de Coimbra

**Colaboração Institucional**

---

**SRAM**® SRAMPOR, Transmissões Mecânicas, Lda

**Coimbra, julho, 2022**



“Todas as vitórias ocultam uma abdicação.”

Simone de Beauvoir

Aos meus pais.



## Agradecimentos

O presente trabalho representa o final da etapa mais importante da minha vida até então. Ao longo dos passados cinco anos estive rodeado de pessoas e entidades que foram incansáveis no apoio à concretização deste sonho, às quais não poderia deixar de agradecer.

O maior dos meus agradecimentos vai para os meus pais pois, através de sacrifícios, paciência e apoio incondicional, caminharam sempre comigo e contribuíram para que este percurso fosse o mais estável possível.

Agradeço a todos os professores com que me fui cruzando e em especial ao meu orientador, Professor Doutor Samuel Moniz, que foi incansável neste último esforço fazendo sempre questão de estar presente, apoiando e direcionando para a melhor abordagem adotar para o problema.

Aos colegas e amigos que Coimbra me deu, particularmente ao Afonso, Bruno e Francisco, que foram fundamentais durante todo este percurso pela entreaajuda e socialização que, sendo eu estudante deslocado, teve sempre um especial impacto.

À cidade e Universidade de Coimbra pelo excelente acolhimento e condições disponibilizadas, tornando a integração no ambiente académico agradável e bastante natural.

Por fim, mas não menos importante, à SRAMPORT pela oportunidade de estágio, e em especial ao Engenheiro Cláudio Silva, Engenheiro Hélio Palrilha e Engenheiro Tiago Seiça pela cooperação, acompanhamento e disponibilidade demonstrada ao longo do estágio.



## Resumo

A presente dissertação tem como objetivo expor, da melhor forma possível, o trabalho realizado ao longo do estágio curricular na empresa SRAMPORT – Transmissões mecânicas, Lda, para obtenção do grau de Mestre em Engenharia e Gestão Industrial pela Faculdade de Ciências e Tecnologias da Universidade de Coimbra.

Com o crescente aumento da competitividade de mercado e custos operacionais, em conjunto com o aumento do volume de negócios pressentido pela empresa, torna-se cada vez mais imperativo maximizar a rentabilidade de recursos e eficiência de sistemas. Com isto, surgiu a necessidade de rentabilizar a operacionalidade das linhas de tratamento térmico em contínuo da fábrica, sendo estas um dos *bottlenecks* do sistema e uma das secções de grande investimento financeiro.

Assim sendo, o presente trabalho desenvolve um algoritmo de escalonamento, para as linhas de tratamento térmico, com tempos de *setup* dependentes da sequência, e prazos de entrega associados, que otimize a utilização dos fornos por obtenção de sequências de tratamento próximas de uma solução ótima.

O modelo proposto foi desenvolvido em linguagem de programação em *Python*, seguindo a lógica do algoritmo heurístico *Iterated Greedy*, para um ambiente de  $m$  máquinas paralelas idênticas e  $n$  jobs, que como indicado por Garey (1979), caracteriza-se como um problema *NP-hard*.

Através de um desenho fatorial  $2^k$  e de experiências foram definidos os valores para os parâmetros de modelação do algoritmo.

Os resultados obtidos demonstraram um possível aumento da disponibilidade das linhas de tratamento térmico, capaz de fazer face à procura futura e ainda ter alguma margem para amortizar picos de procura.

**Palavras-chave:** Sequenciamento, Máquinas Paralelas, Setup Dependente, Algoritmo Heurístico, Iterated Greedy



## Abstract

The main goal of this dissertation is to present the work carried out at SRAMPOR – Transmissões mecânicas, Lda., to obtain the Master's degree in Industrial and Management Engineering of Faculty of Sciences and Technology of University of Coimbra.

Fierce market competition and increasing operational costs, combined with the continuous growth of business volume, makes it imperative to optimize the use of resources and system efficiency. With that said, the need to maximize the use of the continuous thermic treatment lines arises, due to the fact that these are one of the system bottlenecks and a section of high investment.

Along this report, it is explained the development of a scheduling algorithm on parallel thermic treatment lines with sequence-dependent setup times and associated delivery dates, to obtain a treatment sequence near an optimal solution.

The proposed model was developed in Python programming language and following Iterated Greedy heuristic algorithm logic with  $m$  identical parallel machines and  $n$  jobs. As said by Garey (1979) this is a NP-Hard problem.

With a  $2^k$  factorial design and design of experiments, values for the algorithm's modeling parameters were defined.

The results obtained showed a possible increase of availability in the thermic treatment lines, capable of facing demand and absorb demand peaks.

**Keywords** Sequencing, Parallel Machines, Dependent setup, Heuristic Algorithm, Iterated Greedy.



## Índice

Índice de Figuras .....	xi
Índice de Tabelas .....	xiii
Siglas e Acrónimos .....	xv
1. Introdução .....	1
1.1. Enquadramento .....	1
1.2. Objetivos .....	2
1.3. Metodologia .....	2
1.4. Estrutura .....	3
2. Enquadramento Teórico .....	5
2.1. Sequenciamento da produção .....	5
2.1.1. Características dos modelos .....	6
2.1.2. Parâmetros de performance .....	8
2.1.3. Regras base de sequenciamento .....	8
2.2. Balanceamento de linhas de produção .....	9
2.3. Tempo de setup .....	10
2.4. Overall Equipment Effectiveness .....	11
2.5. Abordagens ao problema .....	11
2.5.1. Programação Linear .....	12
2.5.2. Tabu Search .....	13
2.5.3. Iterated Greedy .....	14
3. Estudo de Caso .....	17
3.1. Apresentação da empresa .....	17
3.1.1. SRAM .....	17
3.1.2. SRAM Portugal .....	18
3.2. Análise da situação atual .....	18
3.2.1. Fluxo de processamento .....	19
3.2.2. Organização dos cartões kanban .....	23
3.2.3. Fornos de Tratamento Térmico em Contínuo .....	25
3.2.4. Setups considerados .....	30
3.3. Descrição do problema .....	31
3.3.1. Restrições .....	32
4. Metodologia .....	35
4.1. Dados de input .....	35
4.1.1. Recolha de dados .....	36
4.1.2. Tratamento de dados .....	36
4.2. Modelo heurístico .....	37
4.2.1. Inputs do modelo .....	37
4.2.2. Desenvolvimento do modelo .....	38
4.2.3. Medidas de desempenho .....	44
5. Resultados .....	47
5.1. Desenho fatorial <b>2k</b> .....	47
5.2. Desenho de experiências .....	50

5.3. Outputs de programação .....	51
6. Conclusões .....	55
6.1. Limitações e recomendações para trabalhos futuros .....	56
Referências Bibliográficas .....	59
ANEXO A .....	63
ANEXO B .....	65
ANEXO C .....	67
ANEXO D .....	69
ANEXO E .....	71
ANEXO F .....	73
ANEXO G .....	75

---

## ÍNDICE DE FIGURAS

Figura 1: Mapa de Gantt (Baker & Trietsch, 2019) .....	5
Figura 2: Pseudocódigo de um algoritmo Iterated Greedy (Ying & Cheng, 2010).....	14
Figura 3: Solução de um mínimo local e global (Alrabghi, 2015) .....	15
Figura 4: Marcas do grupo (SRAM,2022). .....	17
Figura 5: Dispersão geográfica de fábricas e escritórios do grupo (SRAM,2022).....	18
Figura 6: Armazém de peças soltas .....	19
Figura 7: Esquema de peças soltas constituintes de uma corrente (SRAM,2022). .....	20
Figura 8: Rolo de aço trefilado .....	20
Figura 9: Rolo de aço laminado.....	20
Figura 10: Secagem em contínuo da EBS .....	21
Figura 11: Tambores de bariagem .....	21
Figura 12: Centro de trabalho para montagem da corrente .....	23
Figura 13: Quadro kanban .....	24
Figura 14: Layout da linha 1 de TTC (SRAM,2022) .....	26
Figura 15: Excerto de parâmetros de tratamento de peças soltas nas linhas de TTC (SRAM,2022).....	27
Figura 16: Buffer das linhas de TTC .....	28
Figura 17: Descarga de peças soltas após RVC .....	29
Figura 18: Input de referências para tratamento .....	38
Figura 19: Output do cenário modelo com a lista de peças a tratamento .....	52
Figura 20: Sequência de tratamento obtida para o cenário modelo.....	52
Figura 21: Tabela output do modelo para as referências em atraso .....	54



## ÍNDICE DE TABELAS

Tabela 1: Famílias próximas de referências .....	28
Tabela 2: Taxa de variação de temperatura na linha de TTC .....	30
Tabela 3: Cenários considerados para os parâmetros do algoritmo IG .....	47
Tabela 4: Cenários de combinação fatorial .....	48
Tabela 5: Impacto de cada critério na solução final .....	49
Tabela 6: Evolução do fator de disponibilidade por variação do número de iterações .....	51
Tabela 7: Evolução da temperatura ao longo da sequência para a linha 2 .....	53



## **SIGLAS E ACRÓNIMOS**

CTC – Cementação e Têmpera em Contínuo

EBS – Bariagem por Centrifugação

FCH – Forno de Cementação e Têmpera

FIFO – *First In First Out*

IG – *Iterated Greedy*

JIT – *Just In Time*

MTO – *Make To Order*

MTS – *Make To Stock*

OEE – *Overall Equipment Effectiveness*

RVC – Revenido em Contínuo

SCC – Lavagem e Secagem em Contínuo

TCC – Têmpera e Cementação em Contínuo

TTC – Tratamento Térmico em Contínuo

TTD – Tratamento Térmico Delta

WIP – *Work In Process*



## 1. INTRODUÇÃO

A presente dissertação tem como finalidade a apresentação do trabalho desenvolvido ao longo de um semestre no âmbito do Mestrado em Engenharia e Gestão Industrial, pela Faculdade de Ciências e Tecnologia da Universidade de Coimbra - FCTUC. O projeto desenvolvido nesta dissertação deu-se na SRAMPOR – Transmissões Mecânicas, Lda., uma empresa com vasta experiência na produção de correntes de transmissão.

Ao longo deste capítulo será feito um breve enquadramento do tema desenvolvido em estágio curricular, assim como a apresentação dos objetivos e estrutura do documento.

### 1.1. Enquadramento

Com o crescente aumento da competitividade e diversificação de mercado, bem como a evolução tecnológica, é cada vez mais importante garantir a qualidade dos produtos e serviços fornecidos. Para tal, é indispensável um eficaz planeamento e controlo da produção.

O planeamento da produção e alocação de recursos é um dos problemas mais desafiantes na gestão do setor industrial, podendo até comprometer a eficiência e competitividade de uma organização (Drexl & Kimms, 1997). Uma abordagem adequada ao problema irá impactar de forma direta a rentabilidade e performance produtiva de uma organização, que por sua vez é fundamental para alcançar ou manter uma competitividade de mercado (Karimi et al., 2003).

O problema de loteamento e sequenciamento é um dos problemas mais importantes a nível operacional e surge pela necessidade de rentabilizar os recursos disponíveis. Este relaciona a capacidade disponível dos recursos de produção com as necessidades produtivas, procurando sempre otimizar um ou mais parâmetros de performance. A modelação da abordagem mais indicada considera parâmetros como eficiência ou rentabilidade do sistema e/ou dos recursos, satisfação do cliente, redução de custos produtivos e entrega atempada das encomendas (Fuchigami & Rangel, 2018).

De forma a otimizar o planeamento produtivo e garantir que todos os objetivos são cumpridos, podem ser estudadas técnicas de balanceamento, loteamento e/ou sequenciamento que otimizem o processamento das ordens de fabrico de acordo com as necessidades e características do sistema.

## 1.2. Objetivos

Atualmente é verificada uma evolução da indústria para uma filosofia MTO (*Make To Order*) com produtos cada vez mais diferenciados entre si. O crescente *mix* de produtos induz uma maior complexidade na gestão e planeamento das tarefas a processar que, aliado a um crescente volume de produção e restrições de capacidade, exige uma definição clara e objetiva de regras ou algoritmos de planeamento.

O trabalho em questão tem como objetivo o estudo e desenvolvimento de uma lógica de sequenciamento para fornos paralelos de tratamento térmico em contínuo, considerando a otimização dos tempos de *setup* dependentes da sequência e respeitando prazos de entrega associados. Tendo em conta a *performance* produtiva atual medida pela empresa, não será possível num futuro próximo fazer face à procura, surgindo assim a necessidade de aumentar a eficiência no chão de fábrica.

A otimização da sequência de tratamento irá traduzir-se numa maior disponibilidade produtiva dos fornos, como também numa maior facilidade em absorver picos de procura, diminuindo ou evitando atrasos na entrega dos produtos.

## 1.3. Metodologia

Por forma a alcançar uma maior objetividade e caracterização do sistema, e seguindo a abordagem usual para aplicação de modelos heurísticos, o presente trabalho teve início com o reconhecimento do *modus operandi* do chão de fábrica, por forma a caracterizar a lógica de funcionamento, particularidades e recursos utilizados.

Após caracterizado o sistema e definidos os objetivos, foram analisadas heurísticas possíveis de se enquadrar no modelo em análise e identificados os dados necessários para tratamento.

Passou-se ao desenvolvimento e aplicação do algoritmo heurístico *Iterated Greedy*, com recurso a folhas de cálculo em Microsoft Excel para os *inputs* do sistema e tratamento dos dados em *Python*.

Assim que modelada a lógica heurística, foi feita uma verificação dos dados junto dos responsáveis pela gestão da produção, comparando os resultados obtidos pelo algoritmo, para o cenário modelo, com aqueles que seriam esperados com base na experiência e *know-how*. Uma validação mais objetiva e fundamentada do algoritmo não foi possível pois, à data de conclusão da lógica desenvolvida e até à data de entrega da presente dissertação, a fábrica não operava com as suas condições normais de funcionamento como exposto no capítulo final.

## **1.4. Estrutura**

Esta dissertação encontra-se dividida por seis capítulos.

No primeiro capítulo é dada uma introdução ao trabalho realizado e uma breve exposição aos objetivos e metodologia de análise ao problema.

No capítulo 2 é feito o enquadramento teórico onde são abordados conceitos como sequenciamento da produção, *setup*, regras de sequenciamento e abordagens de otimização para problemas de sequenciamento.

No terceiro capítulo é apresentado o estudo de caso através de uma breve exposição da empresa, análise da situação atual e descrição do problema.

No quarto capítulo é descrito a metodologia utilizada desde a recolha de dados até ao desenvolvimento da lógica de escalonamento.

No quinto capítulo realiza-se uma análise dos parâmetros a considerar no algoritmo e por fim, no último capítulo, são apresentadas as conclusões, limitações e sugestões para trabalhos futuros.



## 2. ENQUADRAMENTO TEÓRICO

Neste capítulo é feita uma abordagem teórica aos conceitos e modelo abordado na presente dissertação.

### 2.1. Sequenciamento da produção

Sequenciamento é um processo de tomada de decisão utilizado regularmente e transversal a indústrias de manufatura ou serviços, entendendo-se como a alocação de *jobs* a recursos ao longo de um determinado intervalo de tempo, com o objetivo de otimizar um ou mais objetivos. Num contexto de ambiente industrial e de sequenciamento, o conceito *job* entende-se como a tradução de ordens de produção com prazos de entrega associados (Pinedo et al., 1997).

Os modelos existentes servem para compreender e resolver as dificuldades e exigências do sequenciamento de ordens de fabrico. Como *input* necessitam de um conjunto variado de informação acerca das tarefas a realizar, dos recursos disponíveis, restrições do sistema e uma função objetivo a otimizar que dependerá de modelo para modelo conforme os parâmetros em análise (Baker & Trietsch, 2019).

Um dos modelos mais simples para representar o sequenciamento e alocação de *jobs* é o mapa de *Gantt*. Neste mapa é representado graficamente a alocação de *jobs* aos recursos ao longo de um determinado intervalo de tempo, como demonstrado na Figura 1, sendo que as zonas a sombreado podem representar tempos inatividade ou de preparação de *jobs*. Os tempos de processamento são fixos, não pode existir sobreposição de *jobs* na mesma máquina e para definir a lógica de sequenciamento existem regras de priorização como algumas representadas mais à frente no subcapítulo 3.2.3.3..

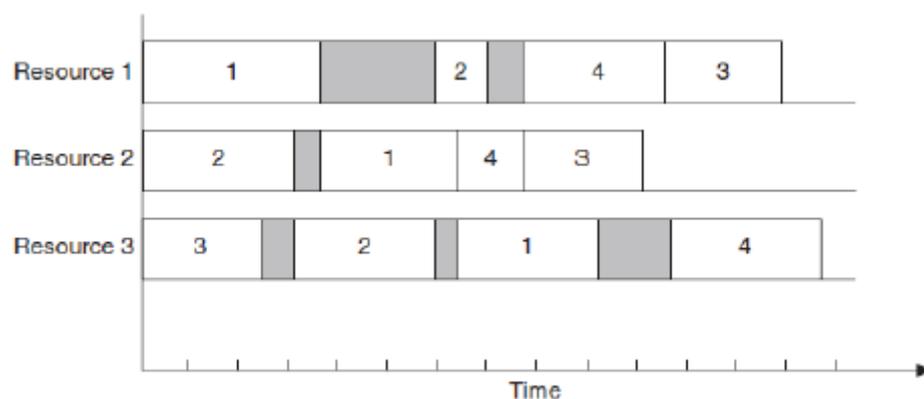


Figura 1: Mapa de Gantt (Baker & Trietsch, 2019)

### 2.1.1. Características dos modelos

O processamento do fluxo de trabalho pode ser feito através de uma ou mais máquinas sendo que, quando na presença de várias máquinas, estas podem ser organizadas em série ou em paralelo. Como tal, é necessário definir o sequenciamento mais adequado e tecnologicamente exequível de acordo com as características do sistema. Caso os dados utilizados na modelação sejam fixos e conhecidos *a priori*, o modelo diz-se determinístico, caso contrário define-se como estocástico (Baker & Trietsch, 2019).

As regras de priorização podem inicialmente ser classificadas segundo a informação considerada, podendo ser locais considerando apenas a informação das operações em espera numa máquina, ou globais considerando a informação de todas as máquinas necessárias. Posteriormente classificam-se pela sua evolução cronológica e podem ser estáticas quando as prioridades não variam com o tempo, ou dinâmicas quando as prioridades variam com o tempo (Baker & Trietsch, 2019).

Segundo Pinedo et al. (1997), um problema de sequenciamento é caracterizado através de três parâmetros sendo eles, a interação de máquinas, as características de processamento incluindo restrições e por fim, pela função objetivo que poderá ser otimizada segundo critérios apresentados no subcapítulo 2.1.2..

Pela interação entre máquinas e *jobs* a classificação é feita da seguinte forma:

- **Máquina única:** Os *jobs* para processamento precisam apenas de uma máquina, sendo este o ambiente de máquinas mais simples.
- **Máquinas paralelas:** Os *jobs* são processados uma vez e podem ser alocados a qualquer de duas ou mais máquinas disponíveis. Estas máquinas podem ser: idênticas, sendo que o tempo de processamento é invariável entre máquinas; uniformes. em que apesar de semelhantes podem ter velocidades de processamento diferentes justificadas por avanços tecnológicos; ou não relacionadas, quando o tempo de processamento depende da máquina a utilizar.
- **Flow Shop:** Existem várias máquinas posicionadas em série e todos *jobs* são processados uma vez em cada máquina, seguindo sempre a mesma sequência unidirecional entre eles.

- ***Flexible Flow Shop***: Este é um modelo generalizado do *Flow Shop*, sendo que em vez de um posicionamento de máquinas em série, existem vários centros de trabalho em série onde cada centro é composto por um conjunto de máquinas paralelas idênticas.
- ***Job Shop***: O fluxo de processamento não é unidirecional, sendo que cada *job* possui uma sequência de máquinas específica e pode visitar uma máquina mais que uma vez.
- ***Flexible Job Shop***: Este modelo é uma generalização do *Job Shop*, seguindo a mesma lógica de processamento, mas através de vários centros de trabalho.
- ***Open Shop***: Cada *job* terá de passar novamente em cada máquina, no entanto alguns tempos de processamento podem ser nulos. A sequência a seguir poderá ser variável, não existindo restrições.

Como características e restrições de processamento, podemos ter várias tais como:

- **Data de lançamento**: Data a partir da qual o *job* dá entrada no sistema e pode começar a ser processado.
- **Restrições de precedência**: Podem aparecer em ambientes de máquina única ou paralela e exigem que um ou mais *jobs* estejam finalizados para que se possa dar início ao processamento de um outro.
- **Tempos de *setup* dependentes da sequência**: Tempo necessário para preparar o processamento de uma tarefa, sendo que este dependerá não só do *job* precedente, como também do *job* a processar. O tempo incorrido poderá depender da sequência de *jobs*, como também das máquinas a utilizadas.
- **Avarias**: Períodos sob os quais a máquina poderá não estar continuamente disponível, justificados por anomalias ou incompatibilidades de processamento.
- **Restrições de elegibilidade de máquina**: Num ambiente de máquinas paralelas, poderão existir *jobs* incompatíveis com algumas máquinas.
- **Recirculação**: Acontece quando um *job* pode visitar, mais que uma vez, uma máquina ou centro de trabalho.

### 2.1.2. Parâmetros de performance

Ao longo do sequenciamento pode ser gerado um conjunto de *outputs* possíveis de utilizar para avaliar a eficácia de uma sequência. Segundo Allahverdi et al. (1999), de forma a adotar o modelo de sequenciamento mais eficiente é então necessário definir um parâmetro de *performance* adequado às características do sistema. Abaixo são apresentados alguns desses parâmetros que podem ser otimizados através de uma função objetivo a otimizar.

- **Tempo total de finalização:** Somatório do tempo total de todos os jobs, desde que deram entrada no sistema até à sua conclusão. Inclui tempos de *setup* e de espera.
- **Tempo total de finalização ponderado:** Somatório do tempo total de todos os jobs com a adição de um fator de importância atribuído a cada uma deles.
- **Makespan:** Equivale ao tempo de finalização da última tarefa processada pelo sistema.
- **Lateness:** Mede o desvio entre a data de entrega prevista e a data de conclusão.
- **Tardiness:** Mede apenas o atraso de entrega sofrido pelos jobs, não considerando aqueles que concluem antecipadamente ao prazo de entrega.
- **Total de jobs em atraso:** Contabiliza o número de jobs em que o prazo de conclusão ultrapassou a data de entrega prevista.

De acordo com Fuchigami & Rangel (2018), e no que se refere a medidas de *performance*, os critérios mais utilizados no setor industrial foram, até então, os relacionados com o *makespan* (54.35%) e com os atrasos de entrega (34.79%).

### 2.1.3. Regras base de sequenciamento

De acordo com Baker & Trietsch, (2019) existem várias regras de priorização de *jobs*, sendo que abaixo representam-se apenas algumas relacionadas com o contexto do presente documento.

- **Shortest processing time (SPT):** As tarefas são organizadas por ordem ascendente de tempo de processamento, sendo primeiramente alocadas às máquinas os *jobs* de menor tempo de processamento.
- **Earliest Due Date (EDD):** São consideradas as datas de entrega de cada *job* e é dada prioridade àquele cuja data de entrega está mais próxima.
- **First In First Out (FIFO):** Os *jobs* são processados respeitando a ordem de chegada ao recurso, ou seja, os primeiros a chegar à máquina ou estação de trabalho são também os primeiros a serem processados.
- **Critical Ratio (CR):** As sequências são organizadas segundo um nível de urgência obtido pelo rácio entre o intervalo de tempo restante até à entrega e o tempo de processamento restante até à conclusão.
- **Random:** São geradas sequências de processamento aleatórias, sem qualquer critério específico de organização.

## 2.2. Balanceamento de linhas de produção

O balanceamento de linhas de produção relaciona a alocação de jobs aos recursos disponíveis de modo a alcançar determinados parâmetros de produção e otimizar objetivos preestabelecidos (Eghtesadifard et al., 2020; Boysen et al., 2021).

O equilíbrio entre recursos ou máquinas é geralmente abordado em sistemas de custos controlados e cuja capacidade utilizada está próxima do limite máximo (Özcan & Toklu, 2008).

Na análise e equilíbrio de capacidades podem existir diferentes restrições, no entanto e segundo Nof et al. (1997), as de maior impacto são: as de ocorrência que asseguram que cada *job* é alocado especificamente a um recurso; de precedência de forma a garantir que são respeitadas as sequências de processamento; e as de capacidade que se referem à carga máxima de trabalho possível de alocar.

O principal objetivo deste problema é a alocação de *jobs* a recursos, procurando sempre manter um equilíbrio de capacidades considerando restrições de precedência (Abdullah Make et al., 2016).

### 2.3. Tempo de setup

Segundo Allahverdi et al. (1999) e Shingo (2019), *setup* define-se como o conjunto de operações necessárias à alteração de máquinas, processos ou outros recursos quando se pretende mudar o produto a processar. As operações em questão podem incluir calibragens, alteração e reposicionamento de ferramentas, posicionamento de WIP (*Work In Process*), limpeza, alteração de materiais ou ambientes produtivos, inspeção de material a utilizar e outras atividades que dependam do produto a produzir.

Referindo Shingo (2019), o *setup* pode ser dividido por *setup* interno ou externo. O *setup* interno refere-se ao tipo de operações possíveis de efetuar apenas quando a máquina está parada, já o *setup* externo diz respeito às operações que podem ser praticadas com a máquina em funcionamento.

O objetivo de uma organização é ser o mais rentável possível, como tal, as tarefas devem ser consideradas de modo que seja possível converter ao máximo as atividades de *setup* interno em externo, diminuindo assim o tempo de paragem das máquinas e aumentando a sua eficiência de utilização. Um aumento da rentabilidade operacional pode ser alcançado através da revisão e reconfiguração de atividades de *setup* internas em atividades externas (Shingo, 2019).

Numa ótica de produção, o *setup* pode ser classificado como dependente ou independente (Shingo, 2019). No *setup* dependente as operações de preparação para a produção podem depender da sequência de *jobs* a processar e/ou das máquinas utilizadas, já no *setup* independente as operações de *setup* dependem apenas do *job* que irá ser processado.

Num contexto real existem várias linhas de produção ou processamento onde os tempos de *setup* são dependentes da sequência de operação como em linhas de processamento de tintas ou de produtos químicos. Neste contexto, não é válido integrar os tempos de *setup* nos tempos de processamento e como tal é necessário adotar modelos que considerem este parâmetro dependente (Baker & Trietsch, 2019).

As atividades de *setup* não adicionam valor ao produto ou serviço fornecido, no entanto, é fundamental uma consideração das mesmas aquando do planeamento produtivo, pois influenciam diretamente a disponibilidade e eficiência produtiva dos recursos.

## 2.4. Overall Equipment Effectiveness

Por *Overall Equipment Effectiveness* (OEE) entende-se como uma medida de eficiência geral de um equipamento que torna possível identificar oportunidades de melhoria no desempenho operacional (Binti Aminuddin et al., 2016).

O cálculo deste indicador está predefinido e é utilizado a nível global para avaliar a produtividade dos recursos. São considerados três parâmetros relacionados com perdas em ambientes fabris, sendo eles disponibilidade, *performance* e qualidade.

$$OEE = \text{disponibilidade} \times \text{performance} \times \text{qualidade} \quad (1)$$

O fator de disponibilidade serve para medir as perdas por paragens não planeadas como falta de operador, mudanças de série, mudanças de ferramenta, manutenção, perdas no arranque, faltas de energia ou falta de abastecimento ocorridas ao longo do tempo planeado de produção. O seu cálculo é obtido como indicado na Equação 2.

$$\text{Disponibilidade} = \frac{\text{Tempo útil de produção}}{\text{Tempo útil de produção} + \text{Tempos de paragem não planeadas}} \quad (2)$$

As perdas relacionadas com o fator de velocidade consideram micro paragens, ciclos em vazio e perdas de ritmo de produtividade.

$$\text{Performance} = \frac{\text{Quantidade real produzida}}{\text{Quantidade teórica a produzir}} \quad (3)$$

O fator de qualidade aborda as perdas relacionadas com produtos não conformes produzidos, retrabalhos ou peças ensaios de teste realizados.

$$\text{Qualidade} = \frac{\text{Produção total} - (\text{Produção não conforme} + \text{Produção de ensaios})}{\text{Produção total}} \quad (4)$$

## 2.5. Abordagens ao problema

O problema de sequenciamento e balanceamento de máquinas paralelas é uma área de estudo considerada atrativa para muitos investigadores, como tal, existem diversas possíveis abordagens que podem conduzir a uma solução ótima ou próxima dela.

As abordagens a ter em conta podem ser classificadas como métodos de otimização exatos ou algoritmos heurísticos. Nos métodos de otimização exatos é alcançada

a solução ótima, geralmente através de modelos matemáticos, no entanto o tempo computacional exigido tende a tornar esta uma abordagem inviável para aplicação em contexto real. Em algoritmos heurísticos, abordagem seguida no presente trabalho, são alcançadas soluções quase ótimas com um tempo computacional aceitável, podendo estes ter por base formulações matemáticas (Low & Wu, 2001).

Os algoritmos heurísticos podem ser caracterizados como construtivos, de melhoria ou compostos. Uma heurística construtiva aplicada ao sequenciamento de tarefas, começa inicialmente com uma tarefa e, de forma iterativa, adiciona uma tarefa seguinte considerando apenas a otimização entre ambas o que induz uma elevada vulnerabilidade da solução obtida à existência de ótimos locais (Ruiz & Maroto, 2005). Numa abordagem algorítmica de melhoria, parte-se de uma solução inicial e de forma iterativa é procurada na vizinhança a otimização de um determinado critério de performance (Blum & Roli, 2003). Algoritmos compostos entendem-se como aqueles que inicialmente recorrem a um algoritmo inicial construtivo para obtenção de uma solução inicial e posteriormente passam para uma fase de melhoria utilizando meta-heurísticas.

Apesar de num contexto real de fábrica serem frequentemente verificados tempos de *setup* dependentes da sequência, e que a chegada de lotes para tratamento poderá ser dinâmica, grande parte dos algoritmos utilizados na literatura não consideram estes pressupostos (Ying & Cheng, 2010). Segundo Allahverdi (2015), a ponderação de tempos de *setup* em decisões de sequenciamento ajuda a aumentar a produtividade, reduzir desperdícios e atividades de valor não acrescentado, levando a uma melhor utilização dos recursos e cumprimento dos prazos de entrega.

O cenário em análise é caracterizado pelo sequenciamento de  $n$  jobs em  $m$  máquinas paralelas idênticas, que como indicado por Garey (1979), caracteriza-se como um problema *NP-hard*.

### **2.5.1. Programação Linear**

O método de Programação Linear (PL) é uma abordagem matemática de otimização que considera um conjunto independente de variáveis de relação linear. As abordagens de programação matemática garantem uma solução ótima para os problemas abordados, no entanto, para ambientes de sequenciamento de tarefas considerados *NP-Hard*,

o elevado tempo computacional exigido torna esta uma abordagem impraticável num contexto real de operação (French,1982).

Em Programação Linear Inteira o conjunto de variáveis consideradas apenas podem assumir valores inteiros. Se o problema analisado possuir apenas variáveis pertencentes ao conjunto de números inteiros o modelo diz-se de Programação Linear Inteira Pura, por outro lado, se for também verificada a existência de variáveis contínuas caracteriza-se como Programação Linear Inteira Mista. Além do referido, muitas das abordagens que recorrem a modelos de Programação linear para problemas de sequenciamento utilizam uma abordagem de Programação Linear Binária em que as variáveis consideradas assumem apenas valores binários Wagner (1959).

### **2.5.2. Tabu Search**

O algoritmo de *Tabu Search*, é uma meta-heurística desenvolvida para resolver problemas de otimização combinatória (Özcan & Toklu, 2008).

A técnica de pesquisa utilizada consiste numa pesquisa local, a partir de uma solução inicial, em busca da iteração que melhor otimiza a solução. Com recurso a uma lógica de memória adaptativa, as últimas soluções consideradas são mantidas numa lista de histórico, proibindo a evolução do algoritmo em direção a essas soluções, evitando desta forma um ciclo de pesquisa infinito e, forçando a lógica desenvolvida a considerar soluções diferentes daquelas obtidas na vizinhança. Por sua vez, esta lógica de pesquisa induz também um aumento da robustez de pesquisa em cenários com a possibilidade de existência de ótimos locais (Blum & Roli, 2003).

A meta-heurística possui diferentes parâmetros que irão influenciar o desenvolvimento e qualidade da solução obtida, sendo que os mesmos carecem de afinação para diferentes cenários (Xu et al., 1998). Entre os fatores a definir nesta abordagem está o critério de paragem das iterações, a estrutura de armazenamento de dados, que poderá guardar um histórico das melhores soluções encontradas e regiões não exploradas, ou registo de orientações que induzam a melhores soluções, o espaço da vizinhança a considerar e o critério a utilizar na escolha de uma nova solução.

### 2.5.3. Iterated Greedy

O algoritmo de escalonamento *Iterated Greedy* tem como parâmetros de *input* o nível de destruição, a probabilidade de aceitar uma solução pior à encontrada até então e um critério de paragem que irá definir o horizonte máximo de simulação.

O algoritmo IG começa a partir de uma solução inicial procurando melhorar um ou mais parâmetros de performance associados a essa solução, como por exemplo o *makespan* total de uma sequência de produção. A qualidade da solução inicial influencia a *performance* do algoritmo, no entanto não garantirá uma solução ótima (Ribas et al., 2021).

---

```
Procedure Iterated_Greedy
1 begin
2    $\xi_0 := \text{Generate\_Initial\_Solution}$ 
3    $\xi := \text{Local\_Search}(\xi_0)$  % optional
4   while termination conditions not meet do
5      $\xi_d := \text{Destruction}(\xi)$ 
6      $\xi_c := \text{Construction}(\xi_d)$ 
7      $\xi' := \text{Local\_Search}(\xi_c)$  % optional
8      $\xi := \text{Acceptance\_Criterion}(\xi, \xi')$ 
9   return  $\xi_{best}$ 
10 end
```

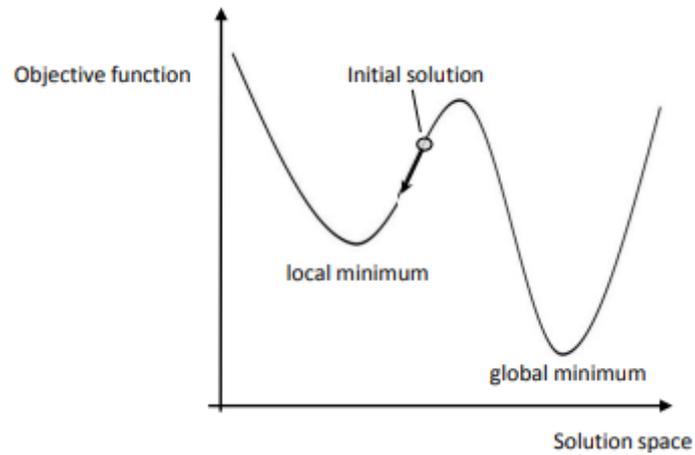
---

**Figura 2: Pseudocódigo de um algoritmo Iterated Greedy (Ying & Cheng, 2010)**

O processo de melhoria da solução é realizado através de duas principais fases até ser atingido um determinado critério de paragem. A primeira fase diz-se de destruição e consiste na remoção aleatória de lotes à solução inicial, obtendo assim uma solução candidata parcial. O número de lotes a retirar à solução inicial é dado pelo nível de destruição definido *a priori* como *input* à heurística. Na segunda fase são percorridas todas as posições da solução candidata parcial e os lotes retirados são realocados um a um na posição em que o *setup* seja mínimo na sequência (Ying & Cheng, 2010).

Após concluída a realocação de todos os lotes removidos, é aplicado um critério de aceitação para decidir se a solução é ou não aceite. O critério mais simples de aceitação é aquele que aceita uma nova solução apenas se esta melhorar a anterior, no entanto isso poderá levar a uma convergência prematura da solução obtida (Ruiz & Stützle, 2007), como demonstrado na Figura 3. De forma a desconsiderar ótimos locais é necessário aceitar com

uma determinada probabilidade de aceitação uma solução pior àquela já existente, procurando assim ótimos globais (Ying & Cheng, 2010).



**Figura 3: Solução de um mínimo local e global (Alrabghi, 2015)**

Adicionalmente pode ser realizada uma pesquisa local antes de se aplicar o critério de aceitação e/ou após obtida uma solução candidata procurando assim otimizar a nova solução. As fases supramencionadas são repetidas até que atingido um critério de paragem que poderá por exemplo ser um tempo limite computacional ou um número máximo de iterações (Arroyo & Leung, 2017).



## 3. ESTUDO DE CASO

### 3.1. Apresentação da empresa

#### 3.1.1. SRAM

A SRAM é uma empresa dedicada à produção de componentes para bicicletas, com início no ano de 1987 nos Estados Unidos da América.

O grupo é um dos maiores do setor de ciclismo a nível mundial e através das várias marcas do grupo (Figura 4) são vendidos grande parte dos componentes necessários para ciclismo como por exemplo: correntes, suspensões, travões, pedaleiras, rodas, medidores de desempenho, entre outros. Além das empresas com fins lucrativos o grupo possui ainda duas marcas de cariz social focadas na manutenção de trilhos e ciclovias e na distribuição em larga escala de bicicletas por áreas economicamente desfavorecidas.

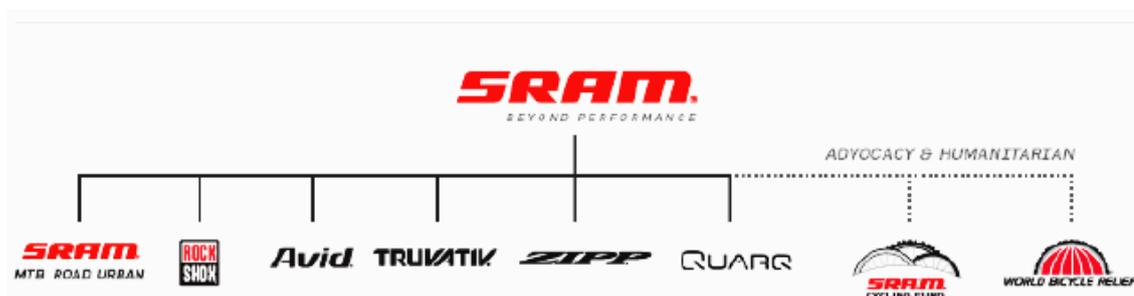


Figura 4: Marcas do grupo (SRAM,2022).

A visibilidade do grupo no mercado é alcançada pela qualidade de produtos e alavancada pela dispersão geográfica, tendo presença física em nove países (Figura 5).



Figura 5: Dispersão geográfica de fábricas e escritórios do grupo (SRAM,2022).

### 3.1.2. SRAM Portugal

Quanto à SRAM Portugal (SRAMPOR), é uma empresa pertencente ao grupo SRAM com elevada experiência na produção de transmissões mecânicas, produzindo atualmente correntes de transmissão para bicicletas, sendo esta a única empresa do grupo responsável pela produção de correntes, sendo que grande parte do seu volume de negócios, cerca de 96%, é em exportações.

Até à data, a empresa expandiu o seu negócio para a montagem de cubos, rodas e pedais e está em fase de estudo a integração de outros novos produtos.

## 3.2. Análise da situação atual

No chão de fábrica são produzidas 82 referências diferentes que podem diferir entre si pela sua geometria, acabamento e características estruturais. Cada referência de peça solta tem uma sequência, parâmetros de processamento e tamanho de lote predefinido, percorrendo o chão de fábrica através de uma abordagem *Flexible Job Shop*.

### 3.2.1. Fluxo de processamento

Como referido anteriormente a fábrica vende vários componentes de ciclismo, no entanto, e como as montagens são independentes, será apenas abordada a produção e montagem de correntes, área objeto de estudo da presente dissertação.

O primeiro setor do chão de fábrica é onde se dá a produção das peças soltas, peças essas que atravessam vários processos de transformação que alteram as suas características físicas e químicas. No segundo setor é feita a montagem da corrente através de oito linhas, sendo que cada linha possui um leque específico de correntes compatíveis. Entre o primeiro e segundo setor existe uma *pool* de peças soltas, designada por armazém (Figura 6).

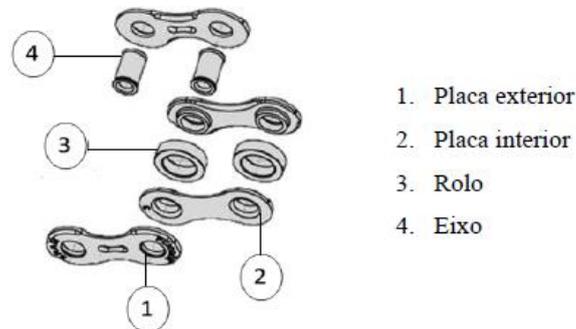


Figura 6: Armazém de peças soltas

#### 3.2.1.1. Setor de peças soltas

A constituição de uma corrente varia conforme a gama a considerar, possuindo em média 108 malhas. Cada malha possui quatro componentes como ilustrado na Figura 7, sendo que os rolos e alguns dos eixos utilizados não são produzidos internamente e são

adquiridos a fornecedores externos, mas os tratamentos térmicos e superficiais são realizados pela SRAMPORT.



**Figura 7: Esquema de peças soltas constituintes de uma corrente (SRAM,2022).**

O fluxo de produção das peças soltas, peças constituintes da corrente, pode seguir várias rotas de processamento dependendo do tipo de peça e características desejadas, no entanto, existem lógicas e parâmetros de processamento comuns entre algumas peças.

O processo inicia-se com o corte por estampagem de rolos de aço laminado através de prensas mecânicas com ferramentas progressivas dando origem a placas interiores e exteriores, ou com a cisalhagem de rolos de aço trefilado a eixos. Grande parte dos eixos e todos os rolos utilizados são obtidos por subcontratação, entrando no sistema a partir do processo descrito abaixo.



**Figura 8: Rolo de aço trefilado**



**Figura 9: Rolo de aço laminado**

Seguidamente, as peças passam por um processo de bariagem inicial, termo com influência francesa, que consiste numa lavagem por rotação de tambores (Fig. 11), removendo rebarbas, óleo e outros possíveis contaminantes provenientes de processos anteriores, deixando a peça apenas com o seu material puro. Este processo pode ser feito de forma automática e em contínuo numa EBS (Bariagem por Centrifugação) ou manualmente onde são carregados lotes de peças em tambores de bariagem, posteriormente descarregados e carregados em secadoras. Cada módulo de EBS é constituído por uma bariagem por centrifugação, onde é descarregado o lote, e por uma secadora em contínuo tornando o processo de lavagem e secagem menos dependente do operador.



**Figura 10: Secagem em contínuo da EBS**



**Figura 11: Tambores de bariagem**

Após secagem, algumas peças passam pelo TTD seguidas de um arrefecimento, peneiragem, lavagem e secagem. Excepcionalmente, existe um pequeno grupo de eixos que começa o seu processo de tratamento a partir do TTD sem passar pelos processos anteriores. O tratamento em questão serve para conferir às peças uma maior resistência superficial ao desgaste.

Posteriormente, existe um tratamento térmico que poderá ser de cementação ou tempera e revenido de forma a induzir uma maior dureza e resistência mecânica às peças, garantindo o desempenho adequado quando sujeitas ao desgaste por abrasão ou erosão. O tratamento de cementação é realizado em fornos FCH (Forno de Cementação e Têmpera) e a tempera e revenido é realizada nos fornos CTC (Cementação e Têmpera em Contínuo).

Após o primeiro tratamento térmico as peças passam por uma bariagem intermédia para que sejam removidos os óxidos de revenido e tons resultantes do tratamento anterior. Posteriormente ao revenido, o processo é de bariagem e secagem final, garantindo um melhor acabamento.

Como exceção existem referências que: passam por um processo de polimento seguido de secagem; peças que são apenas passivadas nos tambores de bariagem e seguidas de uma secagem; peças que mantêm o óxido de revenido; e peças que após bariagem final são sujeitas a processos de tratamento superficial como niquelagem, latunagem ou oxidação a negro. Além destas exceções, existe uma referência que passa pelos FCH seguido de TTD, finalizando com têmpera e revenido nas linhas de TTC e operações de lavagem e secagem pressupostas.

Os possíveis fluxos de peças existentes no chão de fábrica estão representados no Anexo A.

#### **3.2.1.2. Setor da montagem**

Este sector é composto por oito linhas de montagem e num futuro próximo é esperada a implementação de duas novas linhas. O planeamento da produção é feito por uma lógica *Just In Time* (JIT), produzindo assim de acordo com as necessidades atuais.

Cada unidade autónoma de montagem possui quatro cubas diferentes onde são carregadas as peças soltas necessárias à corrente a produzir. A circulação de peças soltas é feita através de alimentadores em calha semifechados e o movimento é promovido por gravidade e mecanismos de sopro.

A sequência de montagem inicia-se pela junção de dois eixos e uma placa exterior inferior, seguida de uma placa interior inferior. O terceiro passo é a adição dos rolos a cada eixo prosseguindo-se a placa interior superior e placa exterior superior.

Por fim temos a rebitagem, sendo esta a última etapa para obter a corrente como projetada, onde é deformada a extremidade superior dos eixos fixando assim as placas exteriores e assegurando a integridade do produto.



**Figura 12: Centro de trabalho para montagem da corrente**

Após montada, a corrente é sujeita a forças de tensão de modo a assegurar uma resistência mínima desejada à rotura. Em linhas de montagem onde estejam a ser produzidas as correntes topo de gama da marca, existe ainda após o tensionamento um sistema de inspeção por visão artificial para controlo de possíveis defeitos que possam existir.

Por fim, a corrente é lubrificada e aprovisionada segundo especificações do cliente que podem ser sob a forma de bobine, caixas de corrente unitárias, ou caixas que podem ser aprovisionadas com diferentes comprimentos de corrente ou unidades de corrente.

### **3.2.2. Organização dos cartões kanban**

As necessidades de produção são geradas sob a forma de cartões *kanban* no setor da montagem à medida que os lotes em armazém são consumidos, seguindo uma filosofia MTS (*Make To Stock*). Os cartões gerados anteriormente são então colocados num quadro *kanban*, respeitando uma ordem FIFO, no setor das peças soltas e reorganizados por uma lógica MTO como indicado adiante.

O quadro *kanban* organiza as ordens de produção pela mesma ordem que as peças são consumidas nas linhas de montagem, no entanto, a procura verificada não segue um padrão constante e as linhas de produção, nas circunstâncias atuais, estão a operar próximo da capacidade máxima.

Dado que a procura é variável entre peças poderemos ter um cenário em que apenas sejam consumidos anualmente, por exemplo, dois lotes de uma referência com um intervalo de tempo considerável entre consumos, fazendo com que a reposição prematura do lote consumido induza a custos de posse e carga da linha de produção desnecessária. Para evitar tais ineficiências o gestor de produção retira do quadro os cartões *kanban* desnecessários a curto prazo e volta a ejetar apenas quando necessário.

Um outro possível cenário que torna inviável a organização FIFO do quadro acontece quando temos referências de elevada procura. O espaço de armazém é limitado e podem existir várias linhas a consumir a mesma referência, fazendo com que o stock em inventário não seja suficiente e a taxa de consumo superior à de reposição. Quando verificadas estas condições são ejetados, manualmente, cartões *kanban* no quadro prevendo assim um consumo futuro. Os cartões ejetados, após produzidos e consumidos os lotes, são retirados do sistema.



Figura 13: Quadro kanban

### **3.2.3. Fornos de Tratamento Térmico em Contínuo**

Os fornos de tratamento térmico em contínuo são utilizados para alterar as características químicas e físicas de peças soltas através da variação de temperatura e atmosfera.

#### **3.2.3.1. Constituição dos fornos**

Até à data a empresa possui dois fornos de TTC semelhantes e aguarda a receção e montagem de um terceiro. Estes funcionam em paralelo de forma independente entre si, e são constituídos por três módulos.

O primeiro módulo, identificado como forno de Tempera e Cementação em Contínuo varia a sua temperatura entre os 880°C e os 910°C, através de 8 queimadores de gás natural distribuídos por três zonas. Além da temperatura pode também ser variada a concentração de polímero no banho de têmpera, dependendo do tipo de aço utilizado e características finais desejadas.

Devido a restrições de capacidade das linhas de TTC e disponibilidade dos fornos FCH, o tratamento de cementação era feito nos fornos FCH, no entanto devido à futura falta de capacidade dos mesmos, está a ser ponderada a inclusão de tratamentos de cementação nos fornos CTC.

O módulo intermédio responsável pela Lavagem e Secagem em Contínuo retira, através da passagem por água, polímero e outras impurezas acumuladas na superfície das peças durante o processo anterior.

Por fim, no terceiro módulo, decorre um processo de Revenido em Contínuo e os parâmetros a variar são a concentração da atmosfera e a temperatura de tratamento que varia entre os 180°C e os 350°C com o auxílio de 3 queimadores, dependendo do tipo de aço e características finais desejadas. A atmosfera é concentrada por gás endotérmico e, futuramente quando existirem cementações, por propano.

Além dos módulos referidos anteriormente, esta secção de tratamento possui ainda um *buffer* para lotes em espera para tratamento.

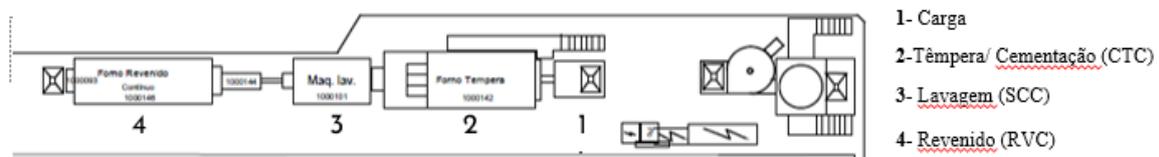


Figura 14: Layout da linha 1 de TTC (SRAM,2022)

### 3.2.3.2. Parâmetros de tratamento

As temperaturas de tratamento podem depender do tipo de peça, aço utilizado e fornecedor, procurando sempre utilizar o mesmo tipo de aço de forma a normalizar os parâmetros de tratamento para cada referência. Os parâmetros considerados são revistos ciclicamente ao longo do ano.

O tratamento por têmpera opera maioritariamente a 880°C e ocasionalmente sobe até aos 910°C. A concentração de gás endotérmico varia com um caudal volúmico mínimo e máximo de 10 L/h a 16 L/h. Quando necessária uma concentração de propano na atmosfera, é utilizado um caudal volúmico de 600 L/h de propano.

Finalizando a passagem pelo forno de TCC as peças passam por um banho de têmpera, sendo este o processo diferenciador entre linhas pela diferente concentração de polímero entre banhos pois a linha 2 utiliza uma maior concentração que a linha 1. Esta concentração poderia ser uniformizada entre linhas à concentração mais elevada, no entanto os custos de produção iriam sofrer um incremento, e existiriam referências sem qualquer valor acrescentado para essa diferença de concentração. Em situações de sobrecarga da linha de menor concentração, linha 1 de TTC, é feita uma equilibragem utilizando também a linha 2 de TTC.

Para o tratamento por revenido as temperaturas variam entre os 180°C e os 350°C.

MATERIAL	TÊMPERA					REVENIDO		DÉBITO DE GASES		ATMOSFERA
	Aço	Zonas 1	Retorta	Tempo	Carga	Tempera	Tempo	Endogas L/h	Propano L/h	% C
Ref.1	C35E_a	880°C	855°C	25min	3kg	225°C	45min	Min.=10 Max.=16	-	Min.=0,35 Máx.=0,60
Ref.2										
Ref.3										
Ref.4										
Ref.5										
Ref.6										
Ref.7										
Ref.8										
Ref.9										
Ref.10										
Ref.11										
Ref.12										
Ref.13										
Ref.14										
Ref.15										
Ref.16										
Ref.17	C55	880°C	855°C	25min	3kg	350°C	45min	Min.=10 Max.=16	-	Min.=0,35 Max.=0,70
Ref.18										

Figura 15: Excerto de parâmetros de tratamento de peças soltas nas linhas de TTC (SRAM,2022)

### 3.2.3.3. Lógica de sequenciamento atual

Neste subcapítulo serão apresentadas as regras e lógica de sequenciamento seguida atualmente pelo gestor de produção.

Os lotes são organizados respeitando quatro regras essenciais. Os agrupamentos são realizados com um máximo de quatro lotes, os prazos de entrega devem ser respeitados, o sequenciamento deve ser feito considerando uma lógica de subida ou descida de temperatura de tratamento e por fim, se uma linha estiver sobrecarregada pode ser feito um equilíbrio de lotes entre linhas, exceto com os lotes de referências exclusivas à linha 2.

Para organizar o sequenciamento, é analisada uma folha de cálculo que considera o stock em armazém e necessidades futuras, onde é obtida a urgência de produção e quantidade de lotes necessária para cada referência.

Se for verificado que uma referência possui maior urgência de reposição em relação a outra, mas que ainda assim não possui o número mínimo de lotes, existem duas possíveis abordagens. Como primeira abordagem temos a situação em que apesar de mais urgente, essa referência pode aguardar pela chegada ao *buffer* dos CTC outros lotes da mesma referência já em processamento, segundo o conhecimento empírico do gestor, possibilitando o tratamento de referências menos urgentes, mas que já possuam o lote mínimo estipulado, procurando sempre respeitar os prazos de entrega. Como segunda abordagem temos a situação em que é processada a referência de maior prioridade apesar de

não possuir o número de lotes mínimo estipulado, pois não seria possível aguardar pela chegada de outros lotes da mesma referência, respeitando os prazos de entrega predefinidos.

Exceccionalmente existem conjuntos de referências em que poderia ser considerado um ponto de desacoplamento após TTC, no entanto o sistema atual faz a distinção entre as mesmas ao iniciar o fluxo de processamento. Para estas peças, apesar de referências diferentes, é possível realizar um agrupamento como o indicado no parágrafo anterior respeitando sempre as famílias (Tabela 1) em que se inserem.

**Tabela 1: Famílias próximas de referências**

<b>3.1</b>	<b>3.2</b>	<b>3.3</b>	<b>3.4</b>	<b>3.5</b>
500 39	500 17	500 19	500 28	400 79
500 40	500 18	500 20	500 29	400 86
500 42	500 41		500 34	400 95

#### **3.2.3.4. Lógica de processamento**

O processamento dos lotes neste setor tem início a partir dos *buffers* situados antes das linhas de TT, em que um operador com o auxílio de uma empilhadora faz a recolha do lote e posiciona-o nas linhas de TTC.



**Figura 16: Buffer das linhas de TTC**

Em cada linha apenas pode ser carregado um lote de cada vez, sendo essa carga realizada em ciclos de 57 segundos com o equivalente a 2.5 quilogramas ou 3 quilogramas de peças dependendo da geometria. O carregamento do forno tem início após ordens de um operador e os ciclos de carga são automáticos.

Caso a referência a tratar posteriormente não exija qualquer *setup*, é dado um compasso de espera de 5 minutos para voltar a carregar o forno. Este compasso de espera serve para evitar a mistura de peças entre lotes por motivos de controlo de qualidade.

Se a referência de maior prioridade exigir uma temperatura de tratamento diferente da atual, a carga do novo lote só pode ser feita após o respetivo *setup* e estabilização de temperatura. Na situação em que o parâmetro a variar é a atmosfera, aplicam-se os mesmos critérios, mas referentes à concentração atmosférica.

Ao serem carregadas no forno de TCC as peças percorrem toda a extensão do mesmo pelo movimento rotativo de uma retorta helicoidal e posteriormente passam pelo banho de tempera, assegurando desta forma um tratamento uniforme entre peças e respeitando o tempo de tratamento predefinido.

De seguida as peças passam pelo processo de SCC onde são descontaminadas e posteriormente são revenidas, percorrendo o forno de revenido e por fim descarregadas para um novo contentor e dando continuidade ao fluxo de processamento.



**Figura 17: Descarga de peças soltas após RVC**

Há exceção de todas as outras peças, existem referências de rolos e eixos, que não passam pelo processo de revenido e as peças são retiradas após SCC.

### 3.2.4 Setups considerados

Dada a complexidade do fluxo de processamento, os equipamentos do chão de fábrica passam por diversos *setups*, no entanto serão abordados abaixo apenas os que ocorrem nos fornos de TTC e que sejam dependentes da sequência de processamento, pois apenas estes serão objeto de análise no presente estágio.

O principal *setup* incorrido nestes equipamentos são os de temperatura, variando em média pelas taxas indicadas na Tabela 2. Sempre que necessária uma variação de temperatura ou concentração atmosférica, esta só pode ocorrer quando não existam quaisquer peças soltas no interior da secção a variar.

**Tabela 2: Taxa de variação de temperatura na linha de TTC**

	<b>Taxa de subida</b> [°C/min]	<b>Taxa de descida</b> [°C/min]
<b>Forno TCC</b>	1.755	-1.944
<b>Forno RVC</b>	1.033	-1.816

Apesar do arrefecimento não ser um arrefecimento forçado, existe uma maior rapidez de arrefecimento do que aquecimento que é justificada pelas elevadas perdas de calor para o exterior sofridas pelo forno datado de 1992.

O *setup* térmico ocorre sempre que a referência a tratar exija uma diferente temperatura de tratamento da atual, sendo que as peças só podem dar entrada no respetivo forno após estabilizada a nova temperatura de tratamento. Caso a variação de temperatura aconteça só no forno de RVC, deve ser considerado o tempo necessário para variar e estabilizar até essa nova temperatura, considerando também o tempo desde a entrada na secção de TCC até à entrada da secção de RVC. A dedução do tempo de CTC ao tempo necessário para variar e estabilizar a temperatura, acrescido de uma margem de segurança, indica a antecedência ideal para dar início à carga do forno com a nova referência, no entanto esta é uma consideração que poderá não estar a ser praticada no planeamento realizado em

chão de fábrica sendo que por vezes a carga da linha só é feita após uma estabilização da temperatura independentemente da secção a efetuar *setup*.

Um outro parâmetro de tratamento a variar é a concentração atmosférica do forno que pode operar entre duas concentrações distintas. Caso seja necessária uma subida da concentração o *setup* necessário é de 1h30min, já para a descida é necessário um tempo de paragem consideravelmente superior pois é necessário seguir uma inertização da atmosfera através de azoto, uma limpeza da atmosfera através de injeção de ar seguida de uma nova inertização a azoto e, posteriormente a gasificação da atmosfera até atingir a concentração desejada, processo que demora cerca de 15 horas. Dado o elevado tempo de *setup*, este processo só é feito no final da semana, aproveitando o período de inatividade do fim de semana ou paragens mínimas de dois dias para restabelecer os parâmetros normais de atmosfera.

Outro *setup* realizado é a limpeza do forno através da passagem de porcas por toda a extensão das linhas de TT sendo que a carga das mesmas ao forno é realizada da mesma forma que as peças soltas. O processo de limpeza consiste na passagem das porcas ao longo da retorta que irá, por arrastão, retirar quaisquer peças soltas de lotes anteriores que ficaram retidas no forno. A passagem é feita sempre que for alterada a referência a tratar, exceto se pertencer à mesma família próxima que a peça anterior, ou a cada 3 a 4 lotes processados.

No Anexo B estão representadas as famílias de peças soltas agrupadas por temperatura de tratamento e na Tabela 1 estão representados os agrupamentos de peças, designados por família próxima, que não exigem qualquer *setup* entre peças à exceção da limpeza a cada 3 a 4 lotes processados.

Por fim, podem existir paragens para manutenção, no entanto algumas são realizadas durante o *setup* de inertização atmosférica ou durante o período de férias.

### **3.3. Descrição do problema**

Com o crescente aumento do volume de negócios, cerca de 30% superior no ano de 2021 em relação ao ano anterior, a fábrica viu o seu processamento estrangulado e surgiram ineficiências de processamento impercetíveis, camufladas até então, pelo excesso de capacidade disponível. Com isto, surgiu a necessidade de aproximar a capacidade real

dos recursos à capacidade teórica máxima e para tal, era necessário resolver os *bottlenecks* do sistema.

Um dos *bottlenecks* é as linhas de TTC, pelo que a empresa decidiu implementar um terceiro forno, ao mesmo tempo que implementa novas linhas de montagem. No entanto, pelas previsões do departamento de Engenharia, assim que implementado um novo forno, o sector de TTC continuaria a ser um dos estrangulamentos do sistema mantendo o OEE semelhante ao verificado atualmente, surgindo assim a necessidade de um modelo de sequenciamento que otimize a sequência produtiva.

As ordens de fabrico, na forma de *kanban*, são geradas considerando o stock de peças soltas e necessidades das linhas de produção que após seguirem diferentes fluxos de tratamento chegam ao *buffer* das linhas de TTC, podendo ou não chegar pela mesma ordem inicial de produção pois diferentes fluxos possuem diferentes *lead times*.

A sequência de tratamento atual é então definida de forma empírica como indicado no subcapítulo 3.2.3.3, podendo induzir num acréscimo desnecessário dos custos de *setup* como também em roturas de stock por falhas no planeamento, fazendo parar as linhas de montagem e adiando o prazo de entrega. Caso a linha de montagem afetada seja compatível com outro tipo de corrente, e procurando evitar uma paragem completa, é então feita a produção antecipada de outra corrente até ser repostos o stock em falta.

A abordagem a tomar, de modo a aumentar a rentabilidade das linhas e respetiva disponibilidade, passa pelo estudo de heurísticas e definição de regras que permitam o adequado balanceamento e sequenciamento nas linhas de TTC que ajudará a evitar paragens ou perturbações nas linhas de montagem por falta de stock de peças soltas e/ou a existência de *setups* acrescidos.

### **3.3.1. Restrições**

Além de um fluxo de processamento extenso e complexo, o sistema apresenta ainda várias restrições que dificultam as possibilidades de sequenciamento com vista à minimização dos tempos de *setups* incorridos, associadas também a uma lógica JIT.

Como restrição inicial temos o tempo máximo de operação disponível que se estende pelos dias úteis da semana, ao longo de três turnos, funcionando horas extra ao longo do fim de semana apenas quando a capacidade normal de funcionamento não é capaz de satisfazer a procura.

O tratamento das peças, como referido anteriormente, só pode ter início aquando da estabilização da temperatura no forno de tratamento para os parâmetros predefinidos, e a carga do forno realizada com um tempo de ciclo predefinido e cargas de 2.5kg ou 3kg dependendo do tipo de peça solta.

Em situações que não sejam necessários *setups*, é sempre dado um intervalo de segurança entre lotes de modo a evitar mistura de lotes para que em caso de controlos de qualidade desfavoráveis, não sejam retidos os dois lotes de peças.

A linha 2 de TTC possui uma concentração de polímero superior no banho de tempera pois algumas peças requerem menores velocidades de arrefecimento, evitando distorções ou empenos geométricos aliados também ao maior teor de carbono, como tal, apenas essas peças podem ir a esta linha. Já na linha 1 de TTC, existe uma menor concentração de polímero no banho de tempera fazendo com que algumas peças não possam ser tratadas nesta linha. À exceção das peças exclusivas à linha 2, pode ser feito um equilíbrio de *jobs* entre linhas.

O tempo de tratamento pode também ser visto como uma restrição do processo, no entanto, e segundo o departamento de Engenharia, este tempo já está otimizado de forma a alcançar as características físicas e químicas necessárias.



## 4. METODOLOGIA

Neste capítulo é descrito o modo como os dados foram recolhidos e utilizados, e de que forma foi desenvolvido, validado e aplicado o algoritmo heurístico no contexto real do problema.

O planeamento produtivo praticado atualmente pela empresa exige um grande envolvimento e dependência do gestor de produção que tem como tarefa diária percorrer o chão de fábrica, registar em que etapas de processamento encontra-se cada lote de peças soltas e através da sua experiência e *know-how*, definir a sequência de tratamento para as linhas de TTC que lhe parece mais adequada.

Na presente dissertação foi utilizado o algoritmo heurístico *Iterated Greedy* (IG) dada a boa performance demonstrada em problemas de sequenciamento em máquinas paralelas e pela sua simplicidade de implementação em relação a outras heurísticas com maior número de parâmetros a calibrar (Ribas et al., 2021).

O número de parâmetros a considerar pela heurística teve grande importância na escolha da mesma, pois, o ambiente fabril considerado possui uma grande variabilidade no que respeita à quantidade de lotes a serem processados, podendo fazer variar os mais diferentes parâmetros de input e a adequabilidade dos mesmos a cada cenário a considerar pela heurística.

Com o modelo de sequenciamento proposto pretende-se diminuir essa dependência e praticar um sequenciamento mais assertivo que, dadas as inúmeras considerações e cenários possíveis de sequenciamento, seria inviável realizar de forma manual num intervalo de tempo considerado útil em contexto real de chão de fábrica.

### 4.1. Dados de input

A abordagem considerada para o problema em questão exigia que fossem conhecidos os parâmetros de tratamento para cada referência, as restrições associadas, assim como os tempos necessários para determinados processos e entre determinadas fases de processamento.

#### 4.1.1. Recolha de dados

Os dados necessários à análise em questão foram recolhidos por formas distintas, uns através de documentação interna à empresa, outros pela recolha manual em chão de fábrica complementados também pelo conhecimento baseado na experiência do gestor de produção.

Para o cálculo da taxa de variação da temperatura nos fornos foram extraídos da base de dados do sistema de informação *Navigo*, utilizado pela empresa para monitorização de processos, dados gerados minuto a minuto ao longo de cinco meses da temperatura do forno em cada secção de tratamento. Com a evolução da temperatura ao longo do tempo em cada secção, foi calculada então a taxa de variação de temperatura em graus centígrados por minuto ( $^{\circ}\text{C}/\text{min}$ ) tanto para o processo de TCC como para o de RVC. Dado que os dois fornos existentes atualmente são iguais, essa variação foi assumida como sendo igual entre fornos. Para o terceiro forno, não existiam dados suficientes para o calcular ou conjeturar a taxa de variação de temperatura e como tal, foi assumido que este seria semelhante aos restantes.

Os parâmetros de tratamento como os indicados na Figura 15 foram obtidos através de documentação interna da empresa, sem a necessidade de tratamento de dados para obtenção dos mesmos.

Os *lead times* médios entre a saída das linhas de TTC e a entrada em armazém foram obtidos por recolha manual em contexto de chão de fábrica. Esse registo era feito pelo preenchimento de uma folha A4, que acompanhava cada lote, contendo o respetivo fluxo de tratamento de cada peça e espaços designados à introdução da data e hora a que cada peça iniciou e acabou cada processo de tratamento.

Dados como restrições de tratamento, tempo incorrido pelo *setup* de porcas e intervalo de carga entre lotes foram obtidos através do gestor de produção. Como restrições de tratamento no contexto das linhas de TTC entendem-se como referências incompatíveis de tratar em contínuo entre elas, sem a necessidade de limpeza do forno pela passagem de porcas.

#### 4.1.2. Tratamento de dados

O tratamento de dados começou inicialmente pela extração, a partir das folhas de registo A4, de tempos incorridos por cada referência desde a saída das linhas de tratamento térmico em contínuo até estas darem entrada em armazém. Os valores médios de

*lead time* para cada referência foram calculados através de folhas de cálculo em *Excel*, no entanto, devido à grande variabilidade nos *lead times* associada à instabilidade de algumas secções de tratamento e ausência de processamento de algumas referências no período de recolha de informação, foi realizado um ajuste a alguns tempos médios e uma inferência daqueles em falta em conjunto com o gestor de produção, por forma a aproximar os dados utilizados a um cenário próximo do habitual. Estes dados poderiam ser sustentados com uma maior fiabilidade, caso existissem pontos de picagem ao longo do fluxo de produção para registo e controlo do estado normal do sistema e tempos médios entre estações.

Para obtenção da taxa de variação da temperatura nos fornos, foram utilizados registos em *Excel* e manipulados através da linguagem de programação em *Python*.

## **4.2. Modelo heurístico**

No desenvolvimento do problema considerado, foi adotada uma lógica de abordagem seguindo o algoritmo *Iterated Greedy*.

### **4.2.1. Inputs do modelo**

Os inputs do modelo podem ser subentendidos ou introduzidos pelo utilizador.

Dados já subentendidos entendem-se como parâmetros e tempos de tratamento, tempos entre diferentes pontos de processamento e compatibilidade das referências a cada forno.

Os dados a introduzir pelo utilizador são as referências para tratamento com os respetivos prazos limite de entrada em armazém e, caso não se encontrem no *buffer* das linhas de TTC em espera para tratamento, o *lead time* desses lotes até ao respetivo *buffer* (Figura 18). Além da informação referida é também necessária a entrada das temperaturas a que cada secção de tratamento de cada forno se encontra e a partir de quando é que será possível alocar lotes a cada linha, pois, no momento em que é realizado o sequenciamento os fornos poderão estar ocupados com o tratamento de lotes anteriores.

Prazo entrada armazém			
Referência	Data	Hora	Lead Time até forno
500 20	17/05/2022	13:00:29	0
500 20	17/05/2022	13:30:00	0
500 20	17/05/2022	05:00:00	0
500 20	18/05/2022	13:00:00	0
500 39	22/05/2022	00:26:04	0
500 42	16/05/2022	00:00:17	0
500 42	16/05/2022	08:30:00	0
500 17	16/05/2022	00:15	0
500 17	16/05/2022	04:45:00	0
500 17	16/05/2022	08:15:00	0
500 17	17/05/2022	00:30:00	0
500 17	17/05/2022	00:30:00	0
500 18	17/05/2022	19:20:00	0

Figura 18: Input de referências para tratamento

#### 4.2.2. Desenvolvimento do modelo

O modelo desenvolvido foi conseguido através da linguagem de programação em *Python*, seguindo as lógicas descritas abaixo.

##### 4.2.2.1. Solução inicial

A heurística de IG começa a partir de uma solução inicial, solução essa que terá impacto no desempenho do modelo (Ribas et al., 2021). De forma a obter uma solução inicial não aleatória e próxima daquela praticada na realidade, foi desenvolvida uma lógica de programação aproximada às considerações tidas em conta pelo gestor de produção.

##### 4.2.2.2. Cálculo do makespan

Dado que o fluxo a analisar é processado em linhas contínuas, os intervalos de utilização ou em vazio verificados à entrada da linha das linhas de TTC, serão também verificados em qualquer outro ponto da linha. Como tal, os cálculos para a utilização do forno e outros associados, foram realizados considerando à entrada de cada forno.

Após obtida a solução inicial, solução essa com uma sequência de tratamento para cada forno, é calculado o somatório do *makespan* total que servirá posteriormente como critério de otimização e comparação entre soluções.

O cálculo do *makespan* total para problemas de sequenciamento é normalmente obtido pelo somatório dos tempos de processamento, no entanto, no problema em análise um forno poderá processar mais que um lote ao mesmo tempo, tornando essa abordagem de *makespan* inviável pela possibilidade de estar contabilizando um mesmo intervalo de forno mais que uma vez. Como processamento de mais que um lote em simultâneo entenda-se por exemplo um cenário em que exista um lote na fase final de tratamento do forno e um outro lote na fase inicial de tratamento ou fase de carga.

Para contornar o cenário anteriormente indicado, o *makespan* para efeitos de comparação entre soluções foi calculado segundo a Equação 5, considerando a utilização do forno apenas à entrada do módulo de têmpera. Dado que os fornos em análise são fornos contínuos, os intervalos de carga e em vazio ao longo do forno e até à saída serão os mesmos que os verificados à entrada, podendo desta forma simplificar o cálculo do *makespan*. Os parâmetros considerados para efeitos de cálculo são os referentes à entrada dos fornos sendo que o parâmetro  $t_{carga\ da\ referência}$  representa o tempo total utilizado para carregar um lote dessa referência no forno de tratamento, o  $t_{vazio}$  refere-se ao tempo que o forno esteve em espera para que essa referência chegasse até ao *buffer* para tratamento e o  $t_{setup}$  ao tempo utilizado pelo forno para que este atingisse os parâmetros de tratamento necessários, ou para efetuar o *setup* de limpeza por passagem de porcas.

$$makespan_{referência} [min] = makespan_{ref\ anterior} [min] + \max(t_{vazio}, t_{setup}) [min] + t_{carga\ da\ referência} [min] \quad (5)$$

$$t_{carga\ da\ referência} [min] = \frac{Peso\ do\ lote\ [kg]}{Peso\ da\ carga\ [kg]} \times t_{ciclo\ de\ carga} [min] \quad (6)$$

O cálculo do *makespan* final é obtido percorrendo cada posição de cada sequência para cada forno, onde em cada iteração é calculado o *makespan* da referência na posição em análise. O valor obtido para a referência na última posição de cada sequência, após entrada no módulo de CTC é então considerado como sendo o *makespan* total para essa linha.

#### 4.2.2.3. Setup de temperatura

O cálculo do *setup* por variação da temperatura foi realizado através de duas lógicas diferentes, mas utilizando os mesmos princípios sendo que uma era referente ao *setup* entre a referência da posição  $i$  em análise e a referência anterior ( $i-1$ ), e a outra para com a referência posterior ( $i+1$ ), sendo que esta segunda lógica foi utilizada apenas no decorrer da fase de construção da heurística, para a escolha da posição que minimizava o tempo de *setup* dependente da sequência.

Sabendo qual a referência anterior na sequência, são também sabidas as temperaturas de tratamento a que se encontra cada secção do forno e desta forma calcular o tempo necessário para atingir e estabilizar a temperatura até aos parâmetros de tratamento desejados. Se para ambas as secções a variação de temperatura for igual a zero, significa que não existe qualquer *setup* de temperatura, caso contrário é seguida a lógica indicada abaixo.

Quando é verificada uma necessidade de variação de temperatura na secção inicial, secção de têmpera e cementação em contínuo, o tempo de preparação do forno é o tempo necessário para variar e estabilizar a temperatura nessa secção. Visto que a variação de temperatura só pode ter início após a última peça do último lote sair desse tratamento, ao tempo de variação de temperatura é também acrescentado o tempo de tratamento de TCC da referência anterior, como representado na Equação 7, visto ser esse o tempo em que o forno não pôde receber carga e esteve a ser inutilizado.

$$setup_{TCC}[min] = \Delta T[^\circ C] * taxa\ de\ variação\ de\ T\ [min/^\circ C] + t_{TCC\ da\ referência\ anterior}[min] \quad (7)$$

Caso a variação de temperatura seja verificada na secção de RVC, esta só poderá ocorrer após a última peça do lote anterior dar saída da secção de RVC, como tal, ao tempo necessário para variar a temperatura é acrescentado o tempo de tratamento da referência anterior na secção de TCC e RVC. Visto que o *setup* incorrido não é à entrada da linha de TTC, mas sim na segunda secção do forno, é possível carregar a linha pouco antes do *setup* estar concluído sendo esse tempo de antecedência igual ao tempo de tratamento de TCC da referência em análise (Equação 8).

$$\begin{aligned} \text{setup}_{RVC}[\text{min}] = & \Delta T[^\circ\text{C}] * \text{taxa de variação de } T [\text{min}/^\circ\text{C}] + t_{TCC \text{ da referência anterior}}[\text{min}] \\ & + t_{RVC \text{ da referência anterior}}[\text{min}] - t_{TCC \text{ da referência}}[\text{min}] \end{aligned} \quad (8)$$

#### 4.2.2.4. Setup de limpeza por porcas

O *setup* de limpeza com recurso a porcas sempre que necessário provoca uma inutilização do forno de 25 minutos sendo que 15 minutos são de carga ao forno e 5 minutos entre o lote anterior e o seguinte.

Sempre que a referência a tratar seja diferente da anterior, e estas não se incluam na mesma família próxima de peças soltas (Tabela 1), é imperativa a limpeza do forno. Caso as referências a tratar sejam iguais ou da mesma família próxima, não existe necessidade de limpeza à exceção de quando é atingido o número máximo de lotes (quatro) sem limpeza de forno por porcas.

#### 4.2.2.5. Fase de destruição

O *setup* de limpeza com recurso a porcas sempre que necessário provoca uma inutilização do forno de 25 minutos sendo que 15 minutos são de carga ao forno e 5 minutos entre o lote anterior e o seguinte.

A fase de destruição tem início com a definição do número de lotes a remover da solução inicial, também conhecido como critério de destruição.

A lógica de remoção é repetida em *loop* até que sejam removidos o número de lotes definido. A remoção é feita de forma aleatória para qualquer posição de qualquer forno. Os lotes removidos, vão depois sendo armazenados numa lista de lotes removidos para serem realocados na fase de destruição.

No Anexo D, encontra-se o pseudocódigo para lógica seguida.

#### 4.2.2.6. Fase de construção

A fase de construção tem início a partir das listas parciais obtidas após a fase anterior, listas essas referentes à sequência parcial de cada forno e uma lista com os lotes removidos na fase de destruição. Os lotes são realocados pela mesma ordem que foram removidos.

Num contexto real de chão de fábrica poderão existir ineficiências de processo ou paragens não planeadas que afetam a continuidade do fluxo de processamento planeado,

como tal, podem ser verificados atrasos de processamento e respetivamente atrasos na entrega do produto. Como o algoritmo desenvolvido e cenários analisados são de um curto intervalo temporal, no máximo um dia, sempre que um lote chega em atraso até ao *buffer* para tratamento este não é considerado na fase de construção e é alocado através de uma lógica ligeiramente diferente e descrita no subcapítulo seguinte. Os lotes em atraso referidos são registados numa lista que é considerada numa outra lógica de alocação.

Na lógica de processamento atual existem referências que não são compatíveis entre linhas, tendo isso em conta o algoritmo de construção começa por verificar a compatibilidade entre o lote e os fornos de tratamento. Sempre que verificada uma compatibilidade positiva é aplicada a lógica de construção na sequência parcial obtida até então para esse forno.

Em cada iteração realizada para cada lote removido, é atribuído um valor de *setup* suficientemente grande para que este seja depois otimizado.

Para cada posição  $i$  da sequência parcial, é verificado o tempo incorrido desde o início do sequenciamento. Caso se verifique que o lote a realocar já se encontra no *buffer* entre o fim do processamento do lote  $i-1$  e início de  $i+1$ , são identificados e calculados os *setups* necessários para essa posição na sequência.

Se o tempo de *setup* obtido em  $i$  for inferior ao encontrado até então, passa-se para a verificação de atrasos na sequência. Através do *lead time* em sistema calculado pelas Equações 9, 10 e 11, é verificado se a data de entrega prevista não é ultrapassada.

$$\begin{aligned} \text{lead time}_i[\text{min}] = & \text{makespan}_{\text{referência em } i}[\text{min}] + t_{\text{TCC da referência em } i}[\text{min}] + t_{\text{lavagem}}[\text{min}] \\ & + t_{\text{RVC da referência em } i}[\text{min}] + \text{lead time}_{\text{forno-armazém da referência}}[\text{min}] \end{aligned} \quad (9)$$

$$\begin{aligned} \text{tempo inicial disponível}_{\text{referência}}[\text{min}] \\ = \text{data de entrega}_{\text{referência}} - \text{data de sequenciamento} \end{aligned} \quad (10)$$

$$\text{lead time}_i[\text{min}] < \text{tempo inicial disponível}_{\text{referência}}[\text{min}] \quad (11)$$

Sendo *lead time* em sistema de cada referência o tempo desde que o algoritmo teve início, até que essa referência dá entrada em armazém, e sendo *lead time*<sub>forno-armazém</sub> o tempo incorrido por cada referência desde o fim do tratamento nas linhas de TTC até estas darem

---

entrada em armazém, enquanto que *tempo inicial disponível*<sub>referência</sub> refere-se ao tempo, em minutos, entre a data de início do sequenciamento e o prazo de entrega da referência.

Após consideradas as três sequências parciais obtidas na fase de destruição, o lote é realocado na sequência que minimiza o *setup*. Quando o valor de *setup* mínimo é verificado em mais que uma sequência, é utilizado o *lead time* em sistema como critério de desempate, sendo escolhido o menor valor.

No Anexo E, encontra-se o pseudocódigo para lógica seguida.

#### **4.2.2.7. Alocação de lotes em atraso**

Semelhante à lógica de construção, foi desenvolvida uma lógica de alocação de lotes atrasados a aplicar após a fase de construção pois, é preferível pelos gestores de produção, que não sejam gerados mais atrasos por alocação antecipada de lotes previamente em atraso.

Caso se verifique um atraso no processamento que provoque a paragem das linhas de montagem, é feita uma troca da corrente a produzir nas linhas, produzindo por antecipação uma outra corrente que possua o *stock* necessário de peças soltas em armazém. Com a alteração referida, o nível de urgência na entrega do lote em atraso diminui e não obriga a uma alocação forçada, excetuando-se o cenário em que uma linha produz apenas uma corrente ou que não existe em armazém *stock* suficiente de peças alternativas a produção de outra corrente.

A lógica utilizada para os lotes em atraso difere apenas da utilizada na fase de construção na restrição de lotes em atraso. Nesta abordagem seria redundante verificar se o lote a alocar se atrasa ou não na posição *i*. Além disso, ao percorrer a sequência para verificar atrasos em posições subsequentes é considerada uma exceção para os lotes registados como atrasados nas fases pré linhas de tratamento térmico.

Após analisado o *setup* incorrido para cada posição em cada sequência parcial, o lote é alocado na posição em que possui o menor *setup* e o menor *lead time* em sistema como critério de desempate, respeitando sempre a exigência de que não podem ser verificados atrasos nas referências que chegam atempadamente ao *buffer*.

#### 4.2.2.8. Critério de aceitação

Ao fim de cada iteração, e após aplicada a fase de destruição e construção, é aplicada uma verificação à solução obtida ponderando assim a aceitação, ou não, da mesma. Sempre que atingida uma solução, a solução atual é atualizada e a melhor solução guardada num histórico de melhores soluções.

Uma nova solução poderá ser aceite conforme duas verificações possíveis. A primeira, e mais explícita, é a aceitação de uma sequência sempre que esta otimize a função objetivo, neste caso a minimização do makespan total obtido. O segundo critério é a aceitação de uma solução mesmo que não otimize a função objetivo, com base numa probabilidade pré-definida. Apesar de pontualmente provocar uma regressão na qualidade da solução obtida, com o decorrer do número de iterações essa solução volta a ser otimizada, ultrapassando por vezes valores obtidos até então, garantindo desta forma que o sistema não está confinado a um ótimo local.

Após atingido o critério de paragem predefinido, é escolhida, a partir do histórico de melhores soluções, aquela que mais otimiza a função objetivo.

No Anexo F, encontra-se o pseudocódigo para lógica seguida.

#### 4.2.3. Medidas de desempenho

O cálculo do valor de OEE pode ser utilizado para avaliar o estado atual de um sistema como também verificar se o sistema será capaz de satisfazer previsões de procura futura. Apesar do cálculo do OEE predefinido e global, a empresa não considera alguns parâmetros referidos no capítulo 2.4 para este indicador.

O cálculo do fator de *performance* é realizado de acordo com o definido na literatura. Para o fator da qualidade são considerados os retratamentos necessários e os ensaios de tratamento para protótipos, no entanto um fator associado aos produtos defeituosos não é considerado. Esta desconsideração justifica-se pela falta de um ponto de controlo de qualidade imediatamente após o tratamento térmico, e só após outras secções de processamento o que não permite com objetividade associar os defeitos detetados às linhas de TTC. Para o cálculo do fator de disponibilidade são apenas considerados para o tempo total de produção planeado os tempos de *setup* de temperatura e de limpeza por porcas e o tempo de produção efetivo, entenda-se, tempo útil de tratamento das peças.

---

No algoritmo heurístico desenvolvido não são consideradas paragens não planeadas, como tal o cálculo obtido para o fator de disponibilidade, não está totalmente em conformidade com o definido na literatura, considerando-se apenas tempos de tratamento efetivos, tempos em vazio e tempos de *setup* dependentes da sequência. Na presente dissertação, sempre que é referido o fator de disponibilidade utilizado no algoritmo, entenda-se como o calculado na Equação 12. Apesar da não conformidade com a literatura, o cálculo é realizado da mesma forma adotada pela empresa, o que torna possível uma análise e comparação mais próxima entre o valor obtido pelo algoritmo heurístico e o calculado pela empresa.

$$\begin{aligned} & \textit{fator de disponibilidade} \\ & = \frac{\textit{tempo total útil de tratamento}}{\textit{tempo total útil de tratamento} + \textit{tempos de setup} + \textit{tempos em vazio}} \end{aligned} \quad (12)$$

A minimização do *makespan* total, como calculado em 4.2.2.2, para uma mesma lista de lotes a processar, traduzir-se-á num aumento do fator de disponibilidade. Esta analogia verifica-se, pois, o algoritmo considera parâmetros de tratamento fixos e predefinidos por referência, e *lead times* fixos entre estações sendo que uma variação do tempo total utilizado só pode ser possível pela variação dos tempos de *setup* praticados ou períodos em vazio entre lotes.



## 5. RESULTADOS

O presente capítulo serve para expor e discutir os resultados obtidos com o algoritmo desenvolvido, os parâmetros utilizados para calibrar a heurística e o porquê de terem sido escolhidos.

### 5.1. Desenho fatorial $2^k$

Para analisar e definir a adequabilidade dos parâmetros considerados na heurística ao problema em análise, começou-se por fazer um estudo designado por desenho fatorial  $2^k$ .

Segundo Law (2017), uma análise fatorial  $2^k$  *factorial* serve para identificar o impacto dos fatores na solução final obtida, assim como o nível de interação entre eles. São realizadas  $2^k$  combinações de cenários possíveis utilizando um valor menor e um maior para cada critério a analisar, na Tabela 3 encontram-se os valores considerados. Ao menor valor de cada parâmetro é atribuído um símbolo negativo, e ao maior valor um símbolo positivo. Os cenários possíveis encontram-se representados na Tabela 4.

**Tabela 3: Cenários considerados para os parâmetros do algoritmo IG**

Fator	Critério	Valor menor	Valor maior
<b>A</b>	Nº máximo de iterações	10	50
<b>B</b>	Nível de destruição	3	10
<b>C</b>	Probabilidade de aceitação de solução pior [%]	0.15	0.5

Tabela 4: Cenários de combinação fatorial

Combinação de fatores	A	B	C	Fator de disponibilidade
1	-	-	-	$R_1 = 79.88$
2	+	-	-	$R_2 = 78.11$
3	-	+	-	$R_3 = 74.94$
4	+	+	-	$R_4 = 76.92$
5	-	-	+	$R_5 = 78.97$
6	+	-	+	$R_6 = 79.78$
7	-	+	+	$R_7 = 79.24$
8	+	+	+	$R_8 = 75.05$

O fator de impacto  $j$ , é obtido pela diferença entre a média do fator de disponibilidade quando o fator  $j$  é positivo, e a média do fator de disponibilidade quando o fator  $j$  é negativo. Nas equações seguintes é exemplificado o cálculo do impacto para cada fator.

$$e_A = \frac{R_2 + R_4 + R_6 + R_8}{4} - \frac{R_1 + R_3 + R_5 + R_7}{4} \quad (13)$$

$$= \frac{78.11 + 76.92 + 79.78 + 75.05}{4} - \frac{79.88 + 74.94 + 78.97 + 79.24}{4} = -0.793$$

$$e_B = \frac{R_3 + R_4 + R_7 + R_8}{4} - \frac{R_1 + R_2 + R_5 + R_6}{4} = -2.827 \quad (14)$$

$$e_C = \frac{R_5 + R_6 + R_7 + R_8}{4} - \frac{R_1 + R_2 + R_3 + R_4}{4} = 0.797 \quad (15)$$

Após analisado o impacto que cada fator tinha para os valores predefinidos inicialmente, verificou-se que os valores não correspondiam com o esperado. O valor negativo obtido para  $e_A$  indica que um maior número de iterações tem um efeito negativo na solução a obter, o que não seria de esperar. Este resultado poderá indicar que o algoritmo se encontra num ótimo local e que não foram realizadas as iterações suficientes para desconsiderar esta solução. O indicador  $e_B$  negativo significa que um valor maior para o nível de destruição não favorece a convergência para uma solução próxima da ótima e  $e_C$

indica que uma maior probabilidade de aceitar uma pior solução, otimiza a solução obtida, o que dá ainda mais fundamento à hipótese de que o algoritmo se encontra num ótimo local.

Pelo referido anteriormente, foram reconsiderados os valores atribuídos a cada parâmetro de modo a verificar se o algoritmo está num ótimo local. Para tal, o número máximo de iterações foi duplicado, o menor valor para a probabilidade de aceitação aumentou e reduziu-se ligeiramente o nível de destruição.

**Tabela 5: Impacto de cada critério na solução final**

<b>Fator</b>	<b>Valor menor</b>	<b>Valor maior</b>	<b><math>e_j</math></b>
<b>A</b>	10	100	1.513
<b>B</b>	3	6	-0.695
<b>C</b>	0.25	0.5	-0.748

Por observação da tabela anterior, verifica-se que o número máximo de iterações passou a beneficiar uma aproximação à solução ótima, validando a possibilidade de existência de pelo menos um ótimo local e de que um maior número de iterações leva a um melhor valor para a solução obtida.

Por outro lado, o indicador de impacto  $e_c$  passou para um valor negativo, significando que uma menor probabilidade de aceitação, dentro do intervalo considerado, beneficia o alcance de melhores soluções, o que refuta o indicado pela Equação 15. Esta contradição poderá indicar que o valor mínimo escolhido anteriormente era muito baixo, dado que para a nova análise o valor máximo foi mantido da análise anterior. Os *outputs* do sistema para este cenário encontram-se no Anexo C.

Considerando o último cenário abordado e após calculado o impacto individual que cada fator tem na solução, é também verificado se a alteração de um fator poderá ter implicações noutros, significando que existe uma interação entre fatores. O nível de interação entre fatores,  $e_{i,j}$ , é obtido pelo cálculo do *output* médio do algoritmo quando ambos os fatores têm o mesmo sinal, subtraído pelo *output* médio quando os fatores possuem sinais contrários. Abaixo apresentam-se as equações para o cálculo em questão.

$$e_{A,B} = \frac{R_1 + R_4 + R_5 + R_8}{4} - \frac{R_2 + R_3 + R_6 + R_7}{4} = 0.53 \quad (16)$$

$$e_{A,C} = \frac{R_1 + R_3 + R_6 + R_8}{4} - \frac{R_2 + R_4 + R_5 + R_7}{4} = -0.79 \quad (17)$$

$$e_{B,C} = \frac{R_1 + R_2 + R_7 + R_8}{4} - \frac{R_3 + R_4 + R_5 + R_6}{4} = 0.45 \quad (18)$$

Um valor positivo para o nível de interação entre fatores significa que o aumento de ambos levará a uma otimização da solução obtida para o parâmetro de disponibilidade. Comparando a grandeza obtida para os valores de disponibilidade na Tabela 6 e a grandeza dos fatores obtidos nas equações anteriores, verifica-se os parâmetros, para os valores considerados, não têm grande impacto na solução final obtida.

## 5.2. Desenho de experiências

Considerando a Tabela 5 verifica-se que o parâmetro número de iterações, tem na solução final, o dobro do impacto que os parâmetros nível de destruição ou probabilidade de aceitação de uma solução pior.

Os valores obtidos no desenho de experiências encontram-se representados na Tabela 6. Apesar da existência de um ótimo local identificado na análise anterior para uma probabilidade de aceitar uma solução pior qua a atual de 15%, quando é reconsiderada uma probabilidade de 25% e aumentamos o número de iterações, é verificada uma rápida convergência para uma solução candidata a solução ótima, pois, um aumento significativo do número de iterações, não se traduz numa diferença de *output* justificável.

Tabela 6: Evolução do fator de disponibilidade por variação do número de iterações

	Número de iterações			
	25	100	500	1000
Nível de destruição = 3	79.24 %	79.244 %	80.153%	80.647%
Probabilidade de aceitação = 25%				

Para um número de 25 iterações e uma probabilidade de aceitação de 25%, o algoritmo desenvolvido exige um esforço computacional em torno de 57 segundos, e cerca de 4 minutos e 38 segundos para 100 iterações, que não traduzem uma melhoria na qualidade da solução. Considerando uma aplicabilidade futura da heurística em contexto real e na integração com outros sistemas de tratamento de dados, pretende-se que o esforço computacional seja o menor possível sem comprometer a qualidade da solução obtida, assim sendo, o modelo foi utilizado com o nível de destruição fixo a 3 e probabilidade de aceitação fixada a 25%, para um número total de iterações igual a 25.

A possibilidade de o algoritmo estar preso a um ótimo local é baixa pois, na tabela 6, o cenário considerado possuía um tamanho de amostra igual a 25 lotes e foram realizadas 1000 iterações com um critério de destruição igual a 3, sendo muito baixa a probabilidade de algum lote não ter sido considerado e de o sistema nunca ter desconsiderado um ótimo local. Outra possível explicação para o verificado nos cenários indicados pode ser a elevada qualidade obtida na solução inicial, o que faz com que o algoritmo convirja rapidamente para uma solução próxima da solução ótima.

Nesta análise é relacionado apenas o número de iterações com a existência de um ótimo local pois os indicadores caracterizam os parâmetros de destruição e probabilidade como sendo de baixo impacto.

### 5.3. Outputs de programação

Abaixo são indicados os *outputs* obtidos pela lógica de programação implementada em *Python* sendo que, através do cenário considerado na Figura 20, foi obtido um fator de produção de 79.24% e um *makespan* de 44.19 horas.

ID	Referência	Data	Hora	Lead Time até forno
0	400 95	2022-05-15	16:25:00	0
1	500 42	2022-05-16	00:00:17	0
2	400 81	2022-05-16	00:13:00	0
3	500 17	2022-05-16	00:15:00	0
4	400 79	2022-05-16	03:00:00	0
5	500 17	2022-05-16	04:45:00	0
6	500 17	2022-05-16	08:15:00	0
7	500 42	2022-05-16	08:30:00	0
8	400 47	2022-05-16	11:55:00	0
9	400 82	2022-05-16	12:40:00	0
10	500 35	2022-05-16	19:00:00	180
11	500 17	2022-05-17	00:30:00	0
12	500 17	2022-05-17	00:30:00	0
13	500 20	2022-05-17	05:00:00	0
14	500 20	2022-05-17	13:30:00	0
15	500 18	2022-05-17	19:20:00	0
16	500 42	2022-05-17	20:00:00	480
17	500 20	2022-05-17	13:00:29	0
18	500 20	2022-05-18	13:00:00	0
19	400 73	2022-05-18	14:00:00	720
20	500 17	2022-05-18	19:00:00	360
21	500 39	2022-05-22	01:00:00	90
22	500 39	2022-05-22	00:26:04	0
23	400 84	2022-06-17	08:00:00	480
24	400 95	2022-06-17	07:49:31	180

**Figura 19: Output do cenário modelo com a lista de peças a tratamento**

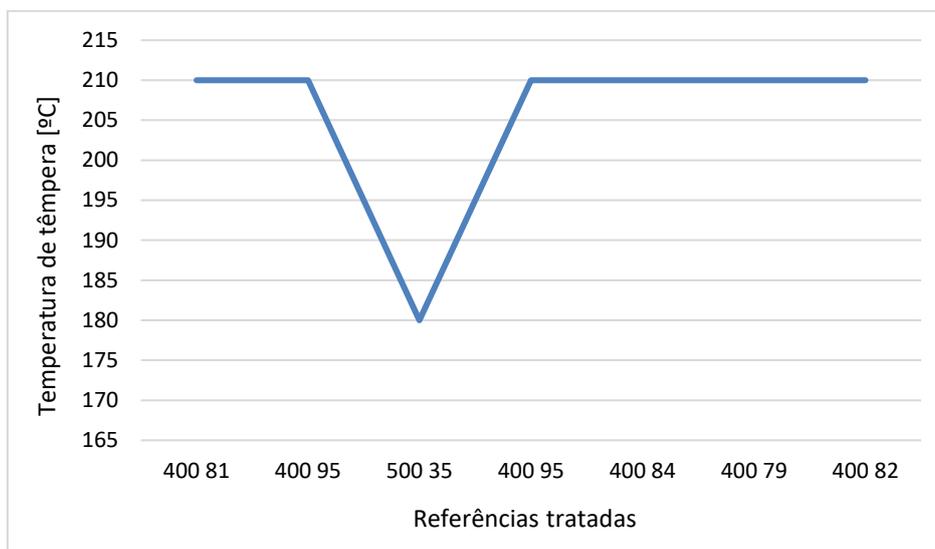
Por observação da sequência obtida para cada forno (Figura 20) e recorrendo às folhas de tratamento térmico utilizadas pela empresa, foi possível verificar que à linha 1 foram alocadas as referências que possuíam temperaturas de têmpera iguais a 260 °C e à linha 2 as referências tratadas a 180°C e 210°C.

Sequência Forno 1: ['400 47', 'Porcas', '500 17', '500 17', '500 17', 'Porcas', '500 20', '500 20', '500 20', '500 20', 'Porcas', '500 42', '500 42', '500 39', '500 42', 'Porcas', '500 17', 'Porcas', '500 39', 'Porcas', '500 18', 'Porcas', '400 73']

Sequência Forno 2: ['500 17', '500 17', 'Porcas', '400 95', '400 95', 'Porcas', '500 35', 'Porcas', '400 84', 'Porcas', '400 81', 'Porcas', '400 79', 'Porcas', '400 82']

**Figura 20: Sequência de tratamento obtida para o cenário modelo**

Tabela 7: Evolução da temperatura ao longo da sequência para a linha 2



Na Tabela 7 encontra-se representada a sequ ncia atribu da   linha 2 e apresentada a varia o da temperatura ao longo do sequenciamento. Verifica-se que entre ambas as refer ncias 400 95 o forno regista uma descida de temperatura para tratamento de uma refer ncia diferente e depois volta   temperatura anterior, isto   justificado pelas tr s primeiras refer ncias terem de ser produzidas o quanto antes para que n o existam atrasos. Como a segunda refer ncia 400 95 possui um prazo de entrega em armaz m superior,   dada prioridade   entrega atempada das outras refer ncias em detrimento de uma redu o do tempo de *setup*.

Quando verificados atrasos nos lotes a processar, essa informa o   transmitida ao utilizador, na forma de tabela que al m de indicadas as refer ncias em atraso, s o indicadas as datas de entrega inicial e a data prevista com atraso. A apresenta o destes dados permite aos respons veis pela gest o de produ o reajustar a produ o nas linhas de montagem para o stock de pe as soltas previsto em armaz m.

Na an lise foi considerado o dia 15/05/2020  s 00:00:00 horas, no entanto, para implementa o pr tica real da l gica desenvolvida,   considerado o tempo real em que   iniciado o programa.

	ref	Prazo de entrega inicial	Prazo de entrega atualizado
0	400 79	2022-05-16 03:00:00	2022-05-16 16:54:11
1	500 18	2022-05-17 19:20:00	2022-05-19 07:45:26
2	400 73	2022-05-18 14:00:00	2022-05-19 09:45:26
3	400 82	2022-05-16 12:40:00	2022-05-20 08:47:02

**Figura 21: Tabela output do modelo para as referências em atraso**

## 6. CONCLUSÕES

O objetivo do presente trabalho foi a construção de uma lógica de sequenciamento que tornasse possível a otimização das sequências de tratamento utilizadas nas linhas paralelas de tratamento térmico em contínuo.

Foi utilizado o algoritmo heurístico *Iterated Greedy*, recorrendo à linguagem de programação em *Python*, e utilizado o *makespan* como critério de otimização. Visto que no modelo desenvolvido consideram-se *lead times* entre estações fixos, e que os tempos de tratamento de cada referência estão predefinidos, o *makespan* considerado só poderá variar pela variação dos tempos de *setup* necessários.

Por análise dos parâmetros intrínsecos da heurística como o nível de destruição, critério de paragem e probabilidade de aceitação de uma pior solução, foi possível concluir que estes não possuem um impacto considerável na solução final obtida, dentro dos intervalos de análise considerados, ou por outro lado, que a solução inicial obtida é de elevada qualidade.

Com o modelo desenvolvido, verificou-se que ainda existe margem para aumentar a rentabilidade das linhas de tratamento térmico em contínuo, atingindo um valor de 79% para o fator de disponibilidade e superando assim os 69.55% desejados pela empresa para fazer face ao volume de procura futuro.

Os valores de disponibilidade obtidos na presente dissertação consideraram o processamento em dois fornos, estando o algoritmo preparado para uma implementação direta em três fornos. Para um mesmo número de intervalos de temperatura nos parâmetros de tratamento, com a implementação de um terceiro forno o fator de disponibilização poderá ainda sofrer um incremento, pois, com vista a redução dos tempos de *setup* o algoritmo irá agrupar a cada forno as referências que lhe forem compatíveis e possuam as temperaturas de tratamento mais próximas entre si.

Considerando que todas as secções de processamento do chão de fábrica acompanhariam a otimização alcançada nas linhas de tratamento térmico, que o mercado daria escoamento ao volume produzido, e que a capacidade produtiva atual instalada é de cerca de 10 000 000 de metros de corrente por ano, um crescimento do fator de 9.45% no fator de disponibilidade permitiria um acréscimo de 945 000 metros de corrente anualmente.

Após implementação de um terceiro forno as linhas de tratamento térmico passam a estar restritas a 15 000 000 metros de corrente anuais, tornando ainda mais vantajoso a otimização deste parâmetro.

### **6.1. Limitações e recomendações para trabalhos futuros**

Aquando da conclusão da presente lógica heurística, para uma mesma referência poderiam existir em simultâneo duas temperaturas de tratamento distintas, o que não estava previsto inicialmente na definição da lógica e parâmetros a considerar, por não serem as condições de funcionamento ditas normais. Esta variação pode justificar-se pela falta do aço normalmente utilizado para uma determinada referência ou pela aquisição de matéria-prima a fornecedores distintos, no entanto, seria benéfico para a empresa uma definição mais restrita dos parâmetros para as propriedades da matéria-prima a adquirir junto dos fornecedores.

Os resultados obtidos com o presente algoritmo dependem de dados de *input* introduzidos pelo utilizador tais como o tempo restante até que um lote dê entrada para tratamento no *buffer* das linhas de tratamento térmico contínuo e que referências tratar, estando desta forma expostos a erros de *input* associados ao fator humano.

Atualmente, em chão de fábrica, não existem pontos de controlo por cada secção de processamento o que torna difícil prever em que fase de processamento se encontra, quais os tempos médios entre secções e quando é que uma referência dará entrada no *buffer*, como tal, a introdução do tempo até que um lote chegue ao *buffer* apenas é possível de forma grosseira, através de alguém com um *know-how* elevado e após um levantamento manual da fase de tratamento de cada lote. Para solucionar esta imprecisão deveria no futuro ser estudada a implementação de pontos de controlo ao longo do fluxo de processamento e realizada uma análise dos valores de forma a conseguir uma maior assertividade nos tempos utilizados.

Além do referido anteriormente, o tempo incorrido desde as linhas de tratamento térmico em contínuo até ao armazém é também um fator de grande variabilidade justificada por restrições e limitações de processamento noutras secções da fábrica, como também pela existência de referências que recorrem a serviços de tratamento externos, o que torna difícil o controlo de *lead times* associados a fornecedores e transporte entre instalações fabris.

Com uma maior aposta num controlo de processos através de sistemas em nuvem e utilização de uma plataforma de *software* de *Internet of Things*, seria possível introduzir no sistema a heurística desenvolvida na presente dissertação, aumentando assim a abrangência e assertividade dos dados considerados. Além do referido, um maior registo de dados permitirá obter indicadores que considerem cenários com maior horizonte temporal de tratamento, solucionando assim uma das maiores limitações do algoritmo desenvolvido que é baixa visibilidade sobre o sistema, além dos fornos de tratamento e a dedução de conclusões gerais a partir de um cenário redutor, que poderá induzir a decisões menos acertadas. Além disso, uma digitalização do chão de fábrica despende da introdução manual dos dados, que em comunicação em tempo real com outros setores permite uma maior estabilização e controlo de fluxo.

Na lógica desenvolvida é considerado que um lote começa a ser carregado no forno de tratamento térmico no instante imediatamente a seguir a estar disponível, no entanto e associado ao fator humano, essa alocação num contexto real nunca é realizada de forma imediata e existem tempos de inutilização de forno não considerados no algoritmo.

Atualmente, quando existem *setups* de processamento, a carga das linhas é realizada com base na experiência que o operário tem e na sua previsão, considerando parâmetros de tratamento predefinidos, da disponibilidade de forno. De forma a reduzir a variabilidade e dependência do *know-how* do trabalhador, deveria ser realizado um estudo aos intervalos de carga dados pelos operários, como também, implementado algum tipo de aviso que indique a disponibilidade do forno. A carga do forno dependente da experiência do operador e sem qualquer tipo de aviso sonoro, visual ou luminoso, poderá ter um grande impacto numa sequência com elevados números de *setups*.

Além do referido anteriormente, o algoritmo não considera a possibilidade de paragens não planeadas além de *setup* e curtos intervalos em vazio, assumindo assim que, após iniciado o sequenciamento, o forno só irá parar quando todas as referências de *input* forem tratadas.

Por fim, para uma maior assertividade e conformidade do cálculo do *Overall Equipment Effectiveness* relacionado com as linhas de tratamento térmico, deveria ser analisada a viabilidade de implementação de um controlo de qualidade após tratamento térmico contínuo. Ponto de controlo este que poderá detetar defeitos com uma maior

antecedência e reduzir assim custos associados a desperdícios de movimentação ou de tratamentos desnecessários.

---

## REFERÊNCIAS BIBLIOGRÁFICAS

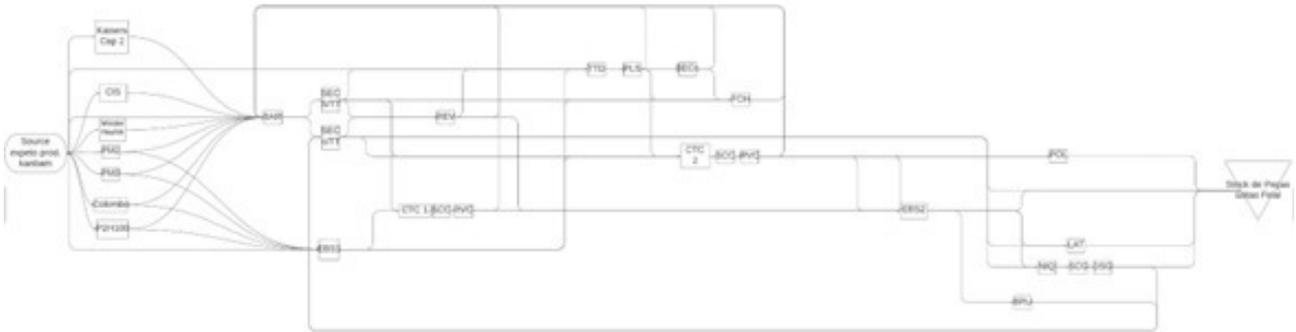
- Abdullah Make, M. R., Ab. Rashid, M. F., & Razali, M. M. (2016). A review of two-sided assembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, 89(5-8), 1743–1763. <https://doi.org/10.1007/s00170-016-9158-3>
- Allahverdi, A., Gupta, J. N. D., & Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega*, 27(2), 219–239. [https://doi.org/10.1016/s0305-0483\(98\)00042-5](https://doi.org/10.1016/s0305-0483(98)00042-5)
- Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, 246(2), 345–378. <https://doi.org/10.1016/j.ejor.2015.04.004>
- Alrabghi, A. O., Tiwari, A., & Savill, M. (2015). *Simulation-based optimisation of Complex Maintenance Systems* (PhD Thesis) School of Engineering – Cranfield University, 112. DOI:10.13140/RG.2.1.2926.8082
- Arroyo, J. E., & Leung, J. Y.-T. (2017). An effective iterated greedy algorithm for scheduling unrelated parallel batch machines with non-identical capacities and unequal ready times. *Computers & Industrial Engineering*, 105, 84–100. <https://doi.org/10.1016/j.cie.2016.12.038>
- Baker, K. R., & Trietsch, D. (2019). *Principles of sequencing and scheduling*. John Wiley & Sons, Inc.
- Binti Aminuddin, N. A., Garza-Reyes, J. A., Kumar, V., Antony, J., & Rocha-Lona, L. (2015). An analysis of managerial factors affecting the implementation and use of overall equipment effectiveness. *International Journal of Production Research*, 54(15), 4430–4447. <https://doi.org/10.1080/00207543.2015.1055849>
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization. *ACM Computing Surveys*, 35(3), 268–308. <https://doi.org/10.1145/937503.937505>
- Boysen, N., Schulze, P., & Scholl, A. (2022). Assembly line balancing: What happened in the last fifteen years? *European Journal of Operational Research*, 301(3), 797–814. <https://doi.org/10.1016/j.ejor.2021.11.043>
- Drexl, A., & Kimms, A. (1997). Lot sizing and scheduling — survey and Extensions. *European Journal of Operational Research*, 99(2), 221–235. [https://doi.org/10.1016/s0377-2217\(97\)00030-1](https://doi.org/10.1016/s0377-2217(97)00030-1)
- Eghtesadifard, M., Khalifeh, M., & Khorram, M. (2020). A systematic review of research themes and hot topics in assembly line balancing through the web of science within

- 1990–2017. *Computers & Industrial Engineering*, 139, 106182.  
<https://doi.org/10.1016/j.cie.2019.106182>
- Sidney, J. B. (1983). Sequencing and scheduling-an introduction to the mathematics of the job-shop, by Simon French, Wiley, 1982, 245 pp. *Networks*, 13(2), 310–311.  
<https://doi.org/10.1002/net.3230130218>
- Fuchigami, H. Y., & Rangel, S. (2018). A survey of Case Studies in production scheduling: Analysis and Perspectives. *Journal of Computational Science*, 25, 425–436. <https://doi.org/10.1016/j.jocs.2017.06.004>
- Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability a guide to the theory of Np-completeness*. Freeman & Company, New York.
- Karimi, B., Fatemi Ghomi, S. M. T., & Wilson, J. M. (2003). The capacitated lot sizing problem: A review of Models and algorithms. *Omega*, 31(5), 365–378.  
[https://doi.org/10.1016/s0305-0483\(03\)00059-8](https://doi.org/10.1016/s0305-0483(03)00059-8)
- Law, A. M. (2017). A tutorial on design of experiments for Simulation Modeling. 2017 *Winter Simulation Conference (WSC)*. <https://doi.org/10.1109/wsc.2017.8247814>
- Lee, Y. H., & Pinedo, M. (1997). Scheduling jobs on parallel machines with sequence-dependent setup times. *European Journal of Operational Research*, 100(3), 464–474.  
[https://doi.org/10.1016/s0377-2217\(95\)00376-2](https://doi.org/10.1016/s0377-2217(95)00376-2)
- Low, C., & Wu, T.-H. (2001). Mathematical modelling and heuristic approaches to Operation Scheduling Problems in an FMS environment. *International Journal of Production Research*, 39(4), 689–708. <https://doi.org/10.1080/00207540150504403>
- Nof, S. Y., Wilhelm, W. E., & Warnecke, H. J. (1997). *Industrial Assembly*. Chapman & Hall.
- Özcan, U., & Toklu, B. (2008). A Tabu search algorithm for two-sided assembly line balancing. *The International Journal of Advanced Manufacturing Technology*, 43(7-8), 822–829. <https://doi.org/10.1007/s00170-008-1753-5>
- Ribas, I., Companys, R., & Tort-Martorell, X. (2021). An iterated greedy algorithm for the parallel blocking flow shop scheduling problem and sequence-dependent setup times. *Expert Systems with Applications*, 184, 115535.  
<https://doi.org/10.1016/j.eswa.2021.115535>
- Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, 165(2), 479–494.  
<https://doi.org/10.1016/j.ejor.2004.04.017>

- Ruiz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3), 2033–2049. <https://doi.org/10.1016/j.ejor.2005.12.009>
- Shingō Shigeo. (2019). *A revolution in manufacturing: The smed system*. Routledge
- Wagner, H. M. (1959). An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly*, 6(2), 131–140. <https://doi.org/10.1002/nav.3800060205>
- Xu, J., Chiu, S. Y., & Glover, F. (1998). Fine-tuning a tabu search algorithm with statistical tests. *International Transactions in Operational Research*, 5(3), 233–244. <https://doi.org/10.1111/j.1475-3995.1998.tb00117.x>
- Ying, K.-C., & Cheng, H.-M. (2010). Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic. *Expert Systems with Applications*, 37(4), 2848–2852. <https://doi.org/10.1016/j.eswa.2009.09.006>



# ANEXO A





## ANEXO B

260°C	220°C	215°C	210°C	205°C	195°C	180°C
900 13	500 37	440 21	400 78	815 23	815 24	500 27
419 11	500 44	440 23	400 79		815 26	500 35
440 12		440 24	400 80			500 38
440 13		440 25	400 81			
440 14			400 82			
440 17			400 84			
400 20			400 85			
400 31			400 86			
400 34			400 87			
400 35			400 89			
400 47			400 95			
400 48			400 96			
400 49			400 97			
400 73			400 98			
400 75			400 99			
400 76						
400 77						
400 88						
500 17						
500 18						
500 19						
500 20						
500 28						
500 29						
500 34						
500 39						
500 40						
500 41						
500 42						
500 43						



**ANEXO C**

<b>Combinação de fatores</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>Fator de disponibilidade</b>
<b>1</b>	-	-	-	$R_1 = 78.96$
<b>2</b>	+	-	-	$R_2 = 80.44$
<b>3</b>	-	+	-	$R_3 = 76.99$
<b>4</b>	+	+	-	$R_4 = 80.12$
<b>5</b>	-	-	+	$R_5 = 78.26$
<b>6</b>	+	-	+	$R_6 = 78.75$
<b>7</b>	-	+	+	$R_7 = 77.78$
<b>8</b>	+	+	+	$R_8 = 78.74$



## ANEXO D

Procedimento **Fase de Destruição:**

```
lista_removidos = lista vazia para os lotes removidos
lista_geral = lista com a solução inicial
nivel_de_destruição = numero de lotes a retirar de forma aleatória
for i in range(nivel_de_destruição):
    Retiro à solução anterior um lote aleatório
    Adiciono o lote selecionado à lista_removidos

return lista parcial da solução anterior
```



## ANEXO E

Procedimento **Fase de Construção:**

```

for c in lista_removidos_destruição:
    var_alocação = [listas a considerar]
    stp_min = 99999999999 #valor suficientemente grande para ser
otimizado
    ref = lista_removidos_destruição[c]
    LT_af = lead time até forno da ref
    if ref compatível com a linha L:
        sol_parcial = solução parcial da linha L
        for s in sol_parcial:
            mkspn_ant = makespan da ref em s-1

            if mkspn_ant >= LT_af:
                stp_T_ant = setup de temperatura entre s e s-1
                stp_P_ant = setup de porcas entre s e s-1
                stp_T_seg = setup de temperatura entre s e s+1
                stp_P_seg = setup de porcas entre s e s+1
                stp_ant = max(stp_T_ant, stp_P_ant)
                stp_seg = max(stp_T_seg, stp_P_seg)
                stp = stp_ant + stp_seg
                stp = max(stp_T, stp_P)

            if stp < stp_min:
                chama o Procedimento de verificação de atrasos
                if flag_atraso = False:
                    stp_min = stp
                    var_alocação = atualiza a lisa considerada

```



## ANEXO F

```
Procedimento Critério de Aceitação:
  Solução_pior = [True, False]
  Probabilidade_aceitação = %
  Distribuição = [Probabilidade_aceitação, (1 -
Probabilidade_aceitação)]
  while critério de paragem não atingido:
    Aceitar_solução_pior = random.choices(Solução_pior,
Distribuição) [0]
    Solução_atual = Z
    Solução_nova = Y
    if Solução_nova < Solução_atual or Aceitar_solução_pior == True:
      Verifica a existência de atrasos provocados
      if não existem atrasos induzidos por essa alocação:
        Solução_atual = Solução_nova
        Historico_de_soluções.append(Solução_nova)
```



## ANEXO G

Procedimento **Verificação de Atrasos:**

```
sol_parcial = solução parcial
i = posição de alocação na lista parcial
ref = referência a alocar em i
prazo_ref = Prazo limite de entrega em armazém de ref
mkspn_ant = makespan da referência em i-1
stp_ant = setup entre a posição i e i-1
t_carga = tempo de carga ao forno de ref
LT_fa = lead time médio da ref desde a saída do forno até ao armazém
t_TCC = tempo de tratamento por têmpera de ref
t_RVC = tempo de tratamento por revenido de ref

mkspn_ant = makespan da referência em i-1
mkspn_ref = mkspn_ant + stp_ant + t_carga
lead_time_ref = mkspn_ref + t_TCC + t_RVC + LT_fa
flag_atraso = False
if lead_time_ref < prazo_ref:
    for r in range(i + 1, len(sol_parcial)):
        Calcula o makesan e lead time para a referência em r
        if flag_atraso == False:
            if não existe atraso da referência em r:
                flag_atraso = False
            else:
                flag_atraso = True
return flag_atraso
```