

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Balanceamento e Sequenciamento de Linhas de Produção na Indústria do Calçado

João Pedro Tavares Vieira Basto

MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE
COMPUTADORES

Orientador: José Soeiro Ferreira

Co-orientador: Rui Diogo Rebelo

24 de Julho de 2016

Balanceamento e Sequenciamento de Linhas de Produção na Indústria do Calçado

João Pedro Tavares Vieira Basto

MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE
COMPUTADORES

Resumo

Nos últimos anos, o paradigma da indústria do calçado portuguesa alterou-se drasticamente: em vez de produção massificada de baixo custo para grandes empresas internacionais, hoje em dia os clientes são principalmente pequenas cadeias de retalho, cujos pedidos são de pequenas quantidades.

Para lidar com estas alterações, tornou-se fundamental para a indústria do calçado desenvolver a rapidez e a flexibilidade do processo produtivo, de modo a ser capaz de responder a qualquer encomenda, por mais pequena que fosse. É neste sentido que o *cluster* do calçado definiu como uma das suas principais linhas de ação a melhoria dos bens de equipamento e processos de modo a reforçar a flexibilidade e produtividade do setor.

O caso em estudo nesta dissertação envolve a criação de um novo método de sequenciamento da produção para uma fábrica real de calçado do grupo Kyaia, precisamente com o objetivo de diminuir o tempo total de produção da mesma para um conjunto de ordens de fabrico a realizar, o que consequentemente aumenta a produtividade e flexibilidade da mesma.

Assim, foi desenvolvido um modelo matemático do problema, que permite a sua resolução exata para pequenas instâncias, e uma meta-heurística - ACI (Algoritmo Competitivo Imperialista) - que resolve o problema para instâncias de maior dimensão.

A análise dos resultados computacionais obtidos permite concluir que a meta-heurística desenvolvida foi capaz de obter um menor tempo total de produção que o método atualmente em vigor na fábrica. Além disso, foi ainda possível tirar conclusões adicionais relativas a possíveis melhorias no balanceamento da linha de produção.

Abstract

In the last years, the paradigm of the portuguese footwear industry has changed drastically: instead of low cost mass production for big international companies, nowadays the clients consist mainly of small retail chains, whose orders are of small quantity.

To deal with these changes, it became fundamental for the footwear industry to develop the speed and flexibility of the productive process, in order to be able to deliver any order, no matter how small it was. It's in this sense that the footwear cluster defined as one of its main lines of action the improvement of equipment goods and processes with the goal of reinforcing the flexibility and productivity of the sector.

The case study in this dissertation involves the creation of a new method for the production sequencing of a real footwear factory of Kyaia group, precisely with the goal of reducing the total production time for a set of production orders, which consequently increases the productivity and flexibility of the factory.

Therefore, a mathematical model of the problem, which allows its exact resolution for small instances, and a meta-heuristics - ICA (Imperialist Competitive Algorithm) - which solves the problem for bigger instances were developed.

The analysis of the computational results allows the conclusion that the meta-heuristics developed was able to obtain a smaller total production time than the method currently used in the factory. Besides, it was also possible to take additional conclusions relative to possible improvements on the assembly line balancing.

Agradecimentos

Em primeiro lugar, quero agradecer ao Professor Doutor José Soeiro Ferreira, por ter confiado em mim ao dar-me a oportunidade de participar neste projeto e por toda a disponibilidade e apoio prestado ao longo da execução do trabalho. Além disso, gostaria também de agradecer ao Engenheiro Rui Diogo Rebelo por me ter acolhido no projeto e pelo interesse e dedicação no meu trabalho.

À Parisa Sadeghi, pela colaboração ao longo deste projeto e por toda a disponibilidade.

Ao INESC TEC, pela oportunidade e pelo bom ambiente de trabalho que me proporcionou.

Aos meus amigos, que estão sempre lá para mim e me dão força e motivação para continuar, não só neste projeto, mas em tudo na vida.

À minha família porque sem o esforço e apoio deles nunca teria chegado até aqui.

João Pedro Tavares Vieira Basto

*“A pessimist sees the difficulty in every opportunity;
an optimist sees the opportunity in every difficulty.”*

Winston Churchill

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.1.1	A Indústria do Calçado	1
1.1.2	INESC-TEC	2
1.2	Objetivos	4
1.3	Motivação	4
1.4	Estrutura da dissertação	5
2	Caso em estudo	7
2.1	Características do sistema de produção	7
2.1.1	Matéria-prima	7
2.1.2	Armazém	8
2.1.3	Linha de produção	8
2.1.4	Postos de trabalho	9
2.1.5	Gamas operatórias	9
2.1.6	Operações	9
2.1.7	Considerações finais	10
2.2	Método de sequenciamento atual	10
2.3	Método de sequenciamento a implementar	11
3	Revisão bibliográfica	15
3.1	Balanceamento e sequenciamento de linhas de produção	15
3.1.1	Linhas de produção	15
3.1.2	Balanceamento de linhas de produção	16
3.1.3	Sequenciamento de linhas de produção	20
3.1.4	Integração do balanceamento com o sequenciamento de linhas de produção	24
3.2	Métodos de otimização	25
3.2.1	<i>Branch-and-bound</i>	26
3.2.2	Algoritmo competitivo imperialista	26
4	Modelo de otimização	31
4.1	Parâmetros	31
4.2	Variáveis de decisão	31
4.3	Restrições	32
4.4	Função objetivo	34
4.5	Programação do modelo	34

5	Algoritmo competitivo imperialista	35
5.1	Leitura de dados	35
5.2	Aplicação do Algoritmo Competitivo Imperialista	36
5.2.1	Codificação da solução	37
5.2.2	Geração de soluções iniciais	38
5.2.3	Cálculo da função objetivo	39
5.2.4	Assimilação com o imperialista	39
5.2.5	Mutação das soluções	41
6	Resultados computacionais	45
6.1	Modelo em CPLEX	46
6.2	Algoritmo competitivo imperialista	46
6.2.1	Problemas de teste	46
6.2.2	Problema real	47
7	Conclusões e trabalho futuro	53
7.1	Conclusões	53
7.2	Trabalho futuro	54
A	Diagramas de Gantt	55
B	Diagrama de Gantt para nova configuração	61
	Referências	66

Lista de Figuras

2.1	Esquema do <i>layout</i> da fábrica	8
2.2	Armazém da fábrica	9
2.3	Linha de produção	10
2.4	Gama operatória linear	11
2.5	Gama operatória em grafo	12
2.6	Visão sistémica do método de sequenciamento a implementar	13
3.1	Grafo de precedência, adaptado de (1)	15
3.2	Principais métodos de otimização, adaptado de (2)	25
3.3	Fluxograma do algoritmo competitivo imperialista	27
5.1	Gama operatória 1	37
5.2	Gama operatória 2	37
5.3	Matriz exemplo da sequência de operações por lote e dos postos atribuídos	39
5.4	Vetor exemplo da sequência global de operações	40
5.5	Exemplo da assimilação na sequência global de operações	40
5.6	Exemplo da assimilação na matriz da sequência de operações por lote e dos postos atribuídos	42
5.7	Exemplo da mutação na sequência global de operações	43
5.8	Exemplo da mutação na sequência de operações de um lote e nos postos atribuídos	43
6.1	Evolução do <i>makespan</i> ao longo do tempo	48
6.2	Evolução do <i>stock</i> no armazém ao longo do tempo	49
6.3	Taxa de ocupação média por posto	51
A.1	Diagrama de Gantt e respetivo caminho crítico para a corrida 1 do ACI	56
A.2	Diagrama de Gantt e respetivo caminho crítico para a corrida 2 do ACI	57
A.3	Diagrama de Gantt e respetivo caminho crítico para a corrida 3 do ACI	58
A.4	Diagrama de Gantt e respetivo caminho crítico para a corrida 4 do ACI	59
A.5	Diagrama de Gantt e respetivo caminho crítico para a corrida 5 do ACI	60
B.1	Diagrama de Gantt e respetivo caminho crítico para configuração da linha com mais um posto	62

Lista de Tabelas

3.1	Versões do SALBP, adaptado de (1)	17
5.1	Tempos de duração das operações e postos admissíveis	38
6.1	Dimensão dos problemas de teste e do caso em estudo	45
6.2	Resultados da execução do modelo em CPLEX	46
6.3	Resultados da execução do ACI	47
6.4	Níveis médios de <i>stock</i> para as diferentes soluções geradas	49
6.5	Taxa de ocupação média dos postos de trabalho para cada método de sequenciamento	50

Abreviaturas e Símbolos

ACI Algoritmo Competitivo Imperialista

Capítulo 1

Introdução

1.1 Enquadramento

1.1.1 A Indústria do Calçado

Atualmente, a indústria do calçado portuguesa assume um papel de relevância internacional, como facilmente se pode verificar por alguns dados estatísticos referentes aos últimos anos.

Entre 2007 e 2012, o preço médio de exportação do calçado português subiu dos 17,7 € para os 22,7 €, um aumento de cerca de 28%, e é já o 2º calçado mais caro do mundo, sendo apenas batido pelo italiano (3). Ao nível das exportações, é importante referir que em 2013 estas ultrapassaram os 1700 milhões de euros, cerca de 1% do PIB português, e 95% do calçado produzido foi exportado, ilustrando a forte reputação internacional do produto (4).

Para entender o sucesso desta indústria, é importante fazer uma pequena análise da história recente da mesma, indo de encontro às causas e eventos que levaram ao contexto atual do calçado português.

Há cerca de 30 anos, a indústria do calçado ocupava-se sobretudo da produção massificada de baixo custo para grandes empresas internacionais. As exportações nacionais eram devidas maioritariamente a grandes unidades produtivas instaladas no país pertencentes a empresas estrangeiras. Porém, a liberalização do comércio internacional permitiu a essas mesmas empresas deslocarem a sua produção para países asiáticos, como a China, fazendo deste continente o maior exportador de calçado, quer em volume, quer em valor, ainda nos dias de hoje (3).

O novo contexto competitivo resultante do aparecimento dos produtores asiáticos cujos preços de mão-de-obra eram muito reduzidos levou a indústria do calçado portuguesa a uma mudança estratégica. Assumindo a incapacidade de competir no preço com a indústria asiática, a indústria portuguesa mudou o seu foco competitivo. Por um lado, apostou na rapidez e flexibilidade de resposta, de modo a tornar-se capaz de responder a uma qualquer encomenda, por mais pequena que fosse. Por outro lado, reforçou o investimento na qualidade e desenvolvimento do produto, procurando acrescentar maior valor ao mesmo.

Estas mudanças levaram também a uma alteração do tipo de clientes da indústria: em vez de grandes marcas internacionais passaram a ser principalmente pequenas cadeias de retalho, cujos pedidos são de pequenas quantidades.

Para garantir o sucesso internacional da indústria, as empresas têm apostado na criação de marcas próprias, com preocupação em relação ao design e à internacionalização e promoção comercial.

Como se pode verificar pelos resultados estatísticos acima mencionados, a estratégia seguida pelas empresas nacionais foi bem sucedida, tornando-se a indústria do calçado portuguesa numa das mais caras do mundo atualmente.

Um dos grandes pilares que permitiu o desenvolvimento da indústria foi a inovação tecnológica, fundamental para o desenvolvimento da rapidez e flexibilidade do processo produtivo. A importância desta área foi amplamente reconhecida, sendo precisamente uma das três iniciativas estratégicas do *cluster* do calçado para o futuro a inovação. Uma das quatro grandes linhas de ação que definem a inovação no calçado é a melhoria dos bens de equipamento e processos de modo a reforçar a flexibilidade e produtividade do setor (3), que é precisamente a área na qual esta dissertação se insere.

Facilmente se observa que a inovação, especialmente nas áreas dos novos materiais e da melhoria dos equipamentos e processos, apenas pode ser possível com o recurso a soluções tecnológicas. Foi no sentido de dar à indústria estas soluções que se estabeleceu uma parceria entre o *cluster* do calçado e o INESC-TEC, entidade acolhedora desta dissertação. Esta parceria trouxe bastantes frutos no desenvolvimento tecnológico e inovação do setor, sendo relevante descrever detalhadamente o INESC-TEC e os seus contributos nesta área.

1.1.2 INESC-TEC

O INESC-TEC – Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência – é um Laboratório Associado que teve origem no pólo do Porto do INESC, criado em maio de 1985. O INESC-TEC, anteriormente designado por INESC Porto, foi constituído em 1998, tendo como associados fundadores o INESC, a Universidade do Porto e a Faculdade de Engenharia da Universidade do Porto. Atualmente, o INESC-TEC conta com 12 centros de I&D e 1 unidade de I&D associada e tem como associados a Universidade do Porto e o Instituto Politécnico do Porto. A Faculdade de Engenharia da Universidade do Porto foi absorvida pela Universidade do Porto, na sequência da transformação desta em Fundação, não sendo por isso um associado identificado autonomamente.

O INESC-TEC é uma instituição criada para constituir uma interface entre o mundo académico e o mundo empresarial da indústria e dos serviços, bem como a administração pública, no âmbito das Tecnologias de Informação, Telecomunicações e Eletrónica.

Foi esta vontade de criar conhecimento científico e tecnológico com aplicação prática na indústria e economia portuguesas que fizeram do INESC-TEC uma das Instituições com maior destaque pelas suas parcerias estratégicas em setores críticos da atividade económica.

Um dos setores parceiros do INESC-TEC é precisamente a indústria do calçado, com a qual já trabalha desde 1995. Em 1996, ainda no início desta parceria, o INESC-TEC instalou o LOGICS-TORE - um sistema altamente inovador de armazenamento e distribuição automática para linhas de costura - no pólo de Paredes de Coura do Grupo Kyaia, maior fabricante português de calçado (5).

A parceria com a Kyaia manteve-se até aos dias de hoje, destacando-se o projeto "*High Speed Shoe Factory*", que teve como objetivo conceber, desenvolver e implementar um novo modelo de fábrica de calçado para resposta ágil em 24 horas (1-2 dias) orientado para a produção unitária, par a par, capaz de responder sem stocks, às vendas pela Internet, às pequenas encomendas e reposições de produtos em loja e ao fabrico rápido das amostras e testes de novos produtos para as novas coleções (6). Em novembro de 2014, o INESC-TEC e a Kyaia assinaram um contrato-programa, que formaliza a aliança que une as duas instituições, assegurando a continuidade da parceria. Durante os próximos três anos, o Grupo Kyaia vai financiar projetos de investigação conjuntos que deem origem a novos processos, produtos e serviços, bem como apoiar a formação avançada, nomeadamente ao nível de mestrado e doutoramento (5).

O INESC-TEC é também parceiro do CTCP, Centro Tecnológico do Calçado de Portugal, com o qual já desenvolveu diversos projetos. Um exemplo é o sistema *ShoeBiz@PT*, criado em 2006, que disponibilizou uma infra-estrutura de comunicação segura e fiável de documentos electrónicos, permitindo o envio/recepção de documentos de negócio (encomendas, requisições, notas de entrega e ordens de produção, entre outros) (7). O sistema permitiu facilitar o processo de integração eletrónica de PMEs na indústria do calçado.

Os principais projetos do INESC-TEC na área do calçado destacam-se pela flexibilidade e rapidez de produção que a sua implementação proporciona. Para além do "*High Speed Shoe Factory*", já referido anteriormente, é também relevante mencionar o "*One Step Production Process*", que integra num único sistema todas as fases produtivas, desde ao corte à montagem (8), e o "*Modular multi-ring system for flexible supply of workstation*", que diz respeito a um sistema logístico inovador com dois anéis – um a nível exterior que faz interface com as máquinas e operadores, e um anel interior que permite fazer ultrapassagens, levando cada sapato a visitar unicamente os postos onde vão ser executadas operações (9). Ambos os projetos foram implementados na Kyaia, comprovando o reconhecimento por parte da indústria da qualidade do trabalho desenvolvido pelo INESC-TEC.

Em paralelo, foram publicados vários trabalhos científicos na área dos quais se destacam um artigo referente à implementação de tecnologia RFID na indústria do calçado e como lidar com os desafios da mesma (10) e outros dois artigos ligados à criação de sistemas de simulação para avaliar a performance de linhas de produção de calçado (11) (12). Além disso, (13) remete para a forma como a investigação operacional foi aplicada à indústria do calçado.

1.2 Objetivos

O objetivo principal desta dissertação passa pela criação de um novo método de sequenciamento da produção para uma fábrica real de calçado do grupo Kyaia.

Face a um conjunto de ordens de fabrico, nas quais devem ser realizadas uma série de operações na fábrica, é necessário definir qual a sequência segundo a qual devem ser feitas estas operações, bem como os postos de trabalho que as devem efetuar. No sentido de avaliar a qualidade deste sequenciamento de produção, considera-se o valor do *makespan*, isto é, o tempo total para concluir todas as ordens de fabrico, sendo o sequenciamento melhor quanto menor for este valor. Pelo facto de também ser necessário escolher o posto onde as operações devem ser realizadas, a situação a abordar integra também o balanceamento da linha de produção para além do seu sequenciamento.

O método de sequenciamento a desenvolver passará pela criação de um modelo matemático do problema, bem como de uma meta-heurística que seja capaz de dar boas soluções para instâncias grandes do problema.

O modelo matemático do problema será programado em CPLEX, quer para validar a correspondência do modelo com o caso em estudo, quer para resolver pequenas instâncias do problema.

O objetivo de resolver por uma meta-heurística passa pela necessidade de obter boas soluções para o caso em estudo num tempo reduzido, de modo a ser possível fazer um re-sequenciamento da produção de forma rápida, no caso de surgir algum imprevisto, como uma operação demorar mais tempo do que era suposto ou um posto de trabalho ter que ser fechado. A meta-heurística escolhida foi o algoritmo competitivo imperialista.

Será também necessário criar um simulador para obter os resultados da aplicação do método de sequenciamento em vigor na fábrica ao problema em estudo.

Por fim, o método de sequenciamento desenvolvido nesta dissertação será comparado com o procedimento em vigor atualmente na fábrica, para comparar os resultados, avaliar a viabilidade da sua aplicação no caso real e retirar conclusões.

1.3 Motivação

Como já foi referido anteriormente, a flexibilidade na produção, quer em volume quer em variedade, e a rapidez no tempo de resposta foram fatores-chave no sucesso internacional da indústria do calçado. Além disso, a aposta na melhoria dos processos é uma das linhas de ação da inovação no setor.

O balanceamento de linhas de produção consiste no ato de distribuir operações pelos postos de trabalho, otimizando determinados objetivos e respeitando as restrições existentes.

Para uma fábrica com número de postos de trabalho fixo, um bom balanceamento da linha de produção permite distribuir a carga das várias operações o mais uniformemente possível pelos postos de trabalho, minimizando o tempo de ciclo do sistema. Pela Lei de Little, é possível afirmar que um tempo de ciclo menor conduz a um tempo total de processamento de um produto menor.

Desta forma, o balanceamento de linhas permite aumentar a rapidez de resposta de um sistema produtivo.

Por outro lado, a procura é cada vez mais constituída por pedidos de pequenas quantidades e grande variedade dos modelos, o que criou a necessidade nas fábricas de produzirem simultaneamente diversos modelos do produto em pequenas quantidades. Deste modo, torna-se crucial que as soluções de balanceamento de linhas sejam capazes de responder eficiente e eficazmente a este novo paradigma da produção na indústria do calçado.

Do mesmo modo, nestas linhas *mixed-model* é crucial garantir um sequenciamento eficiente dos modelos a produzir. Tendo em conta que existem variações nos tempos de processamento de cada estação para diferentes modelos do calçado, a sequência definida torna-se fundamental para garantir a minimização de sobrecargas e de tempos ociosos nas estações de trabalho, bem como do *makespan*.

É interessante realçar os baixos custos (ou até mesmo nulos) que as melhorias nos métodos de balanceamento e/ou sequenciamento de linhas implicam e as vantagens competitivas que os mesmos trazem a uma fábrica. Sem fazer qualquer investimento em novos equipamentos ou num novo *layout*, soluções que geralmente são caras, uma fábrica pode alterar os seus métodos de balanceamento e/ou sequenciamento da linha de produção, conseguindo aumentar a sua produtividade e diminuir o tempo de ciclo do sistema e o tempo total de produção de um determinado produto.

É esta particularidade do balanceamento e sequenciamento de linhas, e também o facto de poderem ser aplicados a qualquer indústria manufatureira e não só à do calçado, que constituem a motivação deste trabalho.

1.4 Estrutura da dissertação

Esta dissertação está dividida em sete capítulos, de acordo com a sequência do trabalho desenvolvido ao longo do projeto.

Neste primeiro capítulo é feita uma introdução à indústria do calçado e são apresentados os objetivos que se pretendem alcançar com este trabalho e as motivações que levaram à sua realização.

No capítulo 2, são apresentadas as características do caso em estudo, bem como se descreve mais pormenorizadamente o trabalho a realizar e os seus objetivos.

No capítulo 3, é apresentada uma revisão bibliográfica sobre os temas fundamentais à obtenção do conhecimento necessário para a execução do projeto desta dissertação. São descritos conceitos sobre linhas de produção e o seu balanceamento e sequenciamento. Além disso, abordam-se os diversos métodos de otimização utilizados para resolver problemas nesta área.

O capítulo 4 refere-se ao modelo matemático desenvolvido para representar o problema a resolver.

O capítulo 5 descreve a meta-heurística desenvolvida para resolver o problema, abordando as especificidades da aplicação da mesma ao caso em estudo.

O capítulo 6 contém os resultados da aplicação quer do modelo matemático, quer da meta-heurística. Além disso, os resultados são discutidos e comparados com os da heurística em vigor na fábrica.

Por fim, no capítulo 7 são apresentadas as conclusões do trabalho desenvolvido e levantadas as questões sobre as quais seria interessante realizar trabalho futuro.

Capítulo 2

Caso em estudo

O caso em estudo trata-se da criação de um novo método de sequenciamento da produção para uma fábrica real de calçado. Nesta fábrica, são realizadas as operações de costura do calçado, isto é, recebendo como *inputs* as diferentes peças dos sapatos já cortadas, estas são costuradas dando origem à gáspea, o *output* do processo produtivo, que é enviada para outra fábrica onde é feita a montagem da gáspea com a sola, concluindo a produção do sapato.

Tratando-se de uma fábrica real, foi fundamental obter um conhecimento aprofundado das características da mesma. Por um lado, porque o sistema de produção da mesma é único, especialmente desenvolvido para a fábrica e, portanto, também substancialmente diferente dos que são tratados nos artigos académicos referentes a linhas de produção. Por outro lado, só com a compreensão completa do funcionamento do sistema é que é possível construir um método de sequenciamento da produção com garantias de que está adequado e adaptado à fábrica, podendo ser aplicado na mesma.

Neste capítulo, começar-se-á então por descrever o sistema de produção em causa e as principais características do mesmo. De seguida, apresentar-se-á a solução de sequenciamento atualmente utilizada pela empresa e, por fim, definir-se-á a estrutura base e os objetivos do novo método de sequenciamento da produção a desenvolver.

2.1 Características do sistema de produção

A fábrica em questão é constituída por duas linhas de costura independentes. Cada uma destas linhas é composta por um armazém e duas linhas de produção. Cada linha de produção é constituída por um conjunto de postos de trabalho e dois tapetes rolantes que transportam as caixas de fabrico entre os postos e o armazém. Este *layout* da fábrica é ilustrado pelo esquema da figura 2.1.

2.1.1 Matéria-prima

A matéria-prima, isto é, os componentes dos sapatos a serem costurados, deslocam-se pela fábrica através de caixas. Cada uma destas caixas leva os materiais necessários para produzir diversos pares de calçado, todos do mesmo modelo, cor e tamanho. O número de pares de calçado

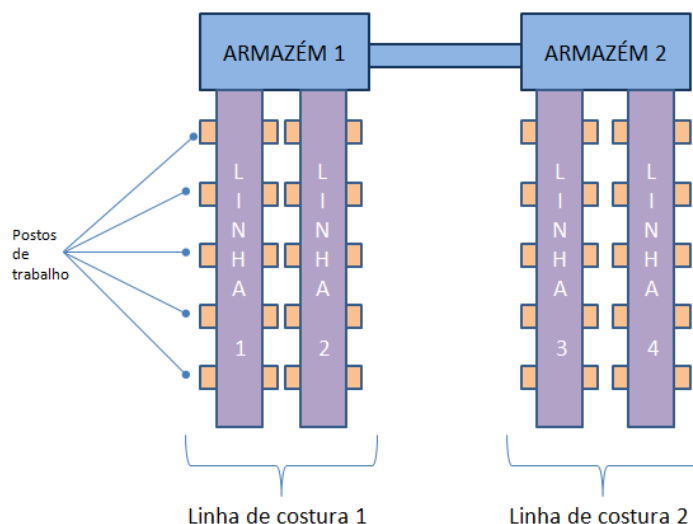


Figura 2.1: Esquema do *layout* da fábrica

transportado numa caixa é variável, sendo frequentemente de dez pares, mas podendo estar entre os cinco e os catorze pares. Além disso, é importante referir que cada caixa tem um grau de prioridade associado, sendo que quão mais prioritária for uma caixa, mais importante é produzi-la rapidamente.

2.1.2 Armazém

Entre a realização das diferentes operações numa dada caixa, esta é colocada no armazém, ilustrado na figura 2.2. O armazém funciona por gravidade e tem dezasseis células, cada uma com capacidade para quatro caixas, podendo assim albergar sessenta e quatro caixas simultaneamente.

Embora as linhas de costura sejam independentes, existe um tapete transportador que permite a passagem de caixas de um armazém para o outro.

2.1.3 Linha de produção

Ao longo da linha de produção, representada na figura 2.3, estão dispostos os postos de trabalho. O transporte de uma caixa do armazém para um posto é assegurado por um tapete rolante. Chegando ao posto de trabalho, a caixa é depositada num *buffer*, isto é, um espaço onde fica à espera que o posto de trabalho esteja livre para começar a ser trabalhada. Depois de fazer a operação numa caixa, o operador do posto coloca a mesma num outro tapete rolante que transporta a caixa de volta para o armazém.

No caso do posto onde se vai realizar a operação seguinte da caixa já estar disponível, existe um mecanismo de envio rápido. Este mecanismo implica que a caixa vá até ao fim da linha, mas não suba para o armazém, e volte a entrar no tapete rolante.



Figura 2.2: Armazém da fábrica

2.1.4 Postos de trabalho

Cada posto de trabalho é constituído por um operador e uma máquina. Cada operador apenas consegue efetuar um determinado conjunto de operações, conjunto esse que é definido pela sua competência. O mesmo se pode dizer da máquina, isto é, as características da máquina determinam o conjunto de operações que esta pode efetuar.

Desta forma, cada posto está apto para efetuar apenas as operações que tanto o seu operador como a sua máquina são capazes de realizar.

2.1.5 Gamas operatórias

A cada caixa a ser produzida está associada uma gama operatória. A gama operatória define quais as operações que devem ser realizadas para a caixa e ainda as relações de precedência entre as mesmas, isto é, a ordem pela qual devem ser efetuadas.

No caso da gama operatória ser linear, as operações apenas podem ser executadas numa determinada sequência. A figura 2.4 representa um exemplo de uma gama linear. Neste caso, a sequência de operações é obrigatoriamente: 1-2-3-4.

No caso da gama operatória ser em grafo, existem diversas possíveis sequências para realizar as operações. A figura 2.5 representa um exemplo de uma gama em grafo. Neste caso, a sequência de operações pode ser 1-2-3-4 ou 1-3-2-4.

No caso desta fábrica, alguns modelos de sapatos apresentam gamas lineares e outros gamas em grafo.

2.1.6 Operações

Cada operação tem associada um tempo de operação, isto é, o tempo que demora a ser realizada. Além disso, como já foi referido em 2.1.4, apenas alguns postos de trabalho são admissíveis



Figura 2.3: Linha de produção

para realizar uma dada operação.

2.1.7 Considerações finais

Da análise às características do sistema de produção em estudo, a conclusão mais importante que se pode retirar é que este tem um modo de funcionamento único, consideravelmente diferente dos sistemas de produção mais comuns.

A principal diferença na linha é o facto de uma dada caixa poder ser transportada entre quaisquer dois postos, não obstante de ter de passar pelo armazém. Ao contrário das linhas tradicionais, em que as caixas seguem todas o mesmo caminho pela linha, passando sequencialmente pelos postos de trabalho pela mesma ordem, nesta linha, o caminho seguido por cada caixa é independente.

Além disso, é importante referir que ficou claro que o método de sequenciamento a implementar deve não só definir a sequência de operações a efetuar em cada posto, mas também definir em que posto deve ser feita uma dada operação, no caso desta poder ser efetuada em diversos postos. Assim, para além do sequenciamento puro da produção, o método também afetará o balanceamento da linha, ao definir as tarefas a executar em cada posto.

2.2 Método de sequenciamento atual

O método de sequenciamento atualmente aplicado pela empresa é uma heurística que, a cada instante do tempo, verifica se existem postos sem caixas à espera e, em caso afirmativo, decide qual caixa enviar para esse posto, com base num conjunto de critérios de decisão.

O primeiro critério é o limite de *stock* da operação. Este critério define que, se já existir um número de caixas em espera para realizar uma dada operação superior a um limite previamente definido, então não se pode fazer o despacho para um posto de uma caixa que vá efetuar essa operação. Este critério tem como finalidade impedir que se acumule um elevado número de caixas

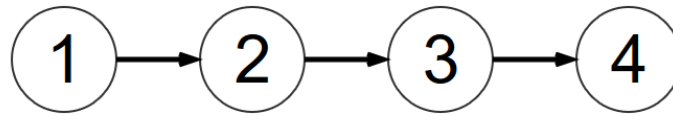


Figura 2.4: Gama operatória linear

para fazer uma dada operação, pois tal geraria um *bottleneck* na produção, tornando o tempo total de produção das caixas maior, o que é indesejável.

Caso exista mais que uma caixa disponível para ser enviada para um posto livre e cujo limite de *stock* da operação não seja violado, são então seguidos uma série de critérios auxiliares para decidir que caixa enviar.

Os critérios são aplicados sequencialmente, começando pelo mais importante, até que haja um "desempate", isto é, até que uma caixa tenha melhores indicadores que as restantes para o critério em causa.

Os critérios de seleção são então os seguintes, já ordenados pela respetiva importância: maior prioridade da caixa, posição no armazém mais próxima do tapete rolante, menor número de caixas em *stock* para a operação subsequente à próxima operação da caixa, menor número de operações por realizar na gama operatória.

Como se pode verificar, o método de sequenciamento aplicado utiliza apenas a informação do estado atual das caixas e da linha de produção para decidir qual a próxima caixa a ser enviada para um dado posto da linha. Ao criar um método de sequenciamento de produção que tenha uma visão global do horizonte de planeamento, isto é, que sabendo as características da linha e as caixas a produzir, efetua o sequenciamento completo da produção *a priori*, poder-se-á chegar a soluções de sequenciamento de maior qualidade, fruto do maior nível de informação utilizado.

Foi esta constatação relevante e o sentimento de capacidade de melhorar a situação que levaram à concepção do novo método de sequenciamento a implementar, analisado na secção seguinte.

2.3 Método de sequenciamento a implementar

Tendo analisado as características da fábrica e o funcionamento do processo produtivo da mesma, é possível definir os traços gerais do método de sequenciamento a implementar.

Este pode ser visto como um sistema que recebe um conjunto de *inputs*, realiza uma série de operações e devolve um *output*, visão que está representada na figura 2.6.

Como *inputs*, o sistema deve receber todas as informações referentes à linha de produção necessárias para proceder ao sequenciamento das operações. Assim, deve ser fornecida ao sistema a lista de postos que estão disponíveis para produzir. É também necessário transmitir a lista de caixas a serem produzidas, bem como os dados referentes às mesmas: o número de pares de sapatos na caixa, a gama operatória e o conjunto de operações já realizadas (no caso da caixa já ter começado a ser trabalhada na fábrica). Para cada gama operatória, é necessário saber as relações

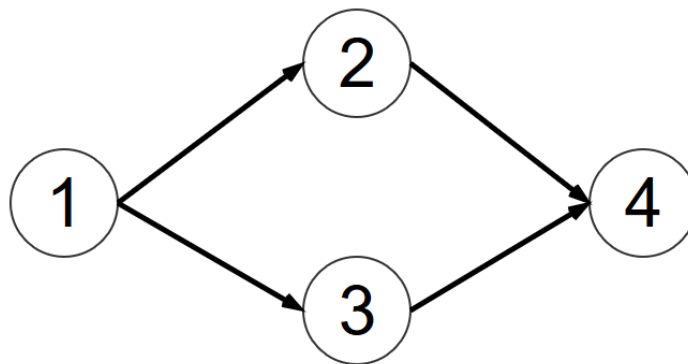


Figura 2.5: Gama operatória em grafo

de precedência entre as diferentes operações, bem como o tempo de duração de cada operação e o conjunto de postos que estão disponíveis para a realizar.

Partindo destes *inputs*, o sistema deve determinar para cada uma das operações a serem feitas qual o posto que as irá efetuar, bem como a sequência das operações a serem realizadas em cada posto. O objetivo do sistema é encontrar uma solução de sequenciamento que minimize o *makespan* da produção, isto é, o tempo total para produzir todas as caixas em fabrico. O processo pelo qual o sistema chega à solução para o problema é discutido posteriormente, nos capítulos 4 e 5.

Tendo obtido uma solução para o problema, o sistema deve gerar um *output* que consiste na informação referente à solução encontrada. Esta informação consiste em transmitir, para cada posto, as operações de cada caixa que serão lá efetuadas, bem como o instante de tempo de início das mesmas. Com estes dados, o sequenciamento da produção para a fábrica fica completamente definido.

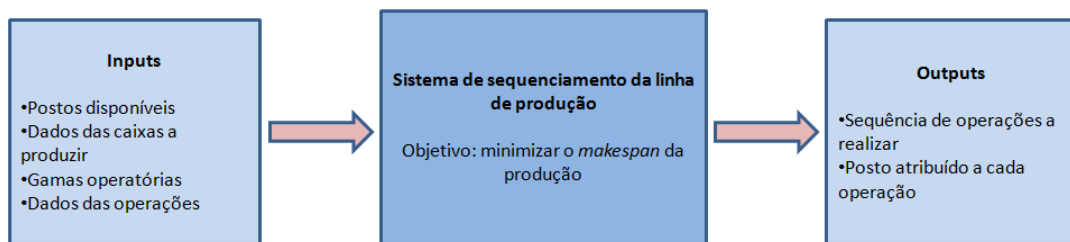


Figura 2.6: Visão sistêmica do método de sequenciamento a implementar

Capítulo 3

Revisão bibliográfica

3.1 Balanceamento e sequenciamento de linhas de produção

3.1.1 Linhas de produção

Uma linha de produção é um processo de manufatura no qual operações vão sendo executadas sequencialmente ao produto a ser fabricado, à medida que este passa pelas diferentes estações de trabalho até o produto final estar completo. Pelo facto da movimentação das peças ser feita automaticamente por tapetes rolantes, o processo produtivo é mais rápido e menos trabalhoso do que quando são trabalhadores que movimentam as peças para um produto estacionário. O uso das linhas de produção é muito frequente na indústria, sendo o caso mais conhecido o da indústria automóvel.

O conceito moderno de linhas de produção foi criado por Henry Ford, em 1913, e aplicado ao processo produtivo do automóvel Model T. A aplicação das linhas de produção permitiu que o Model T fosse fabricado em 93 minutos, oito vezes menos o tempo previamente atingido, e com uma taxa de produção de 20 carros por hora (14).

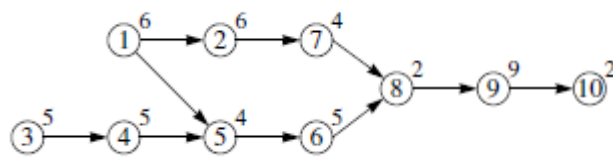


Figura 3.1: Grafo de precedência, adaptado de (1)

O fabrico de um produto numa linha de produção requer que o trabalho de processamento completo seja dividido num conjunto de operações elementares chamadas tarefas. Cada tarefa tem um tempo de duração associado e requer certos equipamentos e/ou capacidades dos trabalhadores. Além disso, as tarefas devem ser feitas segundo uma determinada sequência e têm relações de precedência entre si, isto é, há tarefas que apenas podem ser executadas após outras determinadas tarefas. Estas relações de precedência podem ser resumidas e representadas por um grafo de

precedência, como na figura 3.1. Nestes grafos, cada nó representa uma tarefa, os pesos dos nós representam o tempo de duração da tarefa e os arcos as relações de precedência. Na figura 3.1, por exemplo, a tarefa 5 só pode ser iniciada após as tarefas 1 e 4 (precedentes diretos) e 3 (precedente indireto) serem executadas. Por outro lado, a tarefa 5 tem de estar concluída antes das tarefas 6, 8, 9 e 10 começarem.

3.1.2 Balanceamento de linhas de produção

O balanceamento de linhas de produção consiste em alocar cada uma das tarefas a ser realizada a uma estação de trabalho de modo a que as relações de precedência e ainda outras restrições da linha sejam respeitadas. A soma dos tempos das tarefas atribuídas a uma estação designa-se por tempo de estação. Quando o tempo de ciclo da linha é dado, nenhuma das estações de trabalho pode ter um tempo de estação maior que o tempo de ciclo para que este seja exequível.

Tendo em conta o grande investimento de capital que a implementação de uma linha envolve, o balanceamento de linhas de produção é importante para garantir que o sistema produtivo é o mais eficiente possível.

No que toca à classificação dos problemas de balanceamento de linhas, diversos autores propuseram diferentes classificações para estes mesmos. (15) propõe que os problemas sejam categorizados em dependência de cinco fatores: número de modelos, controlo da linha, frequência, nível de automação e linha de negócio. (16) remete para uma classificação baseada no número de modelos, no tempo das tarefas, isto é, se estes são determinísticos ou estocásticos, e em se os problemas são simples ou generalizados. Por fim, (17) começa por dividir os problemas de balanceamento em simples e generalizados, sub-dividindo estes de acordo com os seus objetivos, restrições e características específicas.

Neste documento, apresenta-se uma classificação baseada nesta última, englobando ainda alguns tópicos abordados nas outras duas acima referidas.

3.1.2.1 SALBP

O SALBP (*Simple Assembly Line Balancing Problem*), ou em português, problema de balanceamento de linhas de produção simples, tem como principais características (17):

- Produção em massa de um único produto;
- Processo de produção completamente pré-definido;
- Linha ritmada com tempo de ciclo fixo c ;
- Tempos de tarefa t_j determinísticos;
- Únicas restrições nas alocações das operações são as relações de precedência;
- *Layout* em série com m estações de trabalho;
- Todas as estações têm o mesmo equipamento em relação a máquinas e operadores;

		Tempo de ciclo	
		Dado	Minimizar
Número de estações	Dado	SALBP-F	SALBP-2
	Minimizar	SALBP-1	SALBP-E

Tabela 3.1: Versões do SALBP, adaptado de (1)

- O objetivo é maximizar a eficiência da linha $E = t_{sum}/(m.c)$, com n tarefas e tempo total de tarefas $t_{sum} = \sum_{j=1}^n t_j$.

A mudança do objetivo levou à formulação de várias versões do problema, representadas na tabela 3.1.

O SALBP-1 representa o problema SALBP original, com tempo de ciclo fixo e com o objetivo de minimizar o número de estações. O SALBP-2 é o problema "dual" do SALBP-1, isto é, com número de estações fixo e o objetivo de minimizar o tempo de ciclo. O SALBP-F consiste em determinar se existe um balanceamento de linha possível para um número de estações e um tempo de ciclo fixos. Por fim, o SALBP-E é a versão mais geral do problema e tem como objetivo minimizar simultaneamente o número de estações e o tempo de ciclo, de modo a maximizar a eficiência da linha. As versões do SALBP podem ainda incluir um objetivo secundário que consiste em equilibrar as cargas das estações de trabalho, isto é, suavizar as diferenças entre o tempo total das tarefas atribuídas a cada uma das estações de trabalho (17).

3.1.2.2 GALBP

O SALBP parte de princípios que restringem muito as características das linhas de produção, afastando-se do que se verifica em ambientes industriais reais. Assim, os investigadores intensificaram recentemente o seu trabalho para identificar, formular e resolver problemas mais próximos dos reais. Esses problemas são designados por GALBP (*Generalized Assembly Line Balancing Problems*), ou em português, problemas de balanceamento de linhas de produção generalizados. As principais características que caracterizam os GALBP serão explicitadas seguidamente.

Características da linha de produção

- **Ritmo de produção das estações de trabalho:** Em relação ao ritmo de produção, uma linha pode ser designada como *paced* ou *unpaced*. Nas linhas *paced*, todas as estações de trabalho têm o mesmo tempo de ciclo e a linha apresenta uma taxa de produção constante (inversa ao tempo de ciclo).

Nas linhas *unpaced*, todas as estações trabalham à sua própria velocidade, não existindo um tempo de ciclo comum. Neste caso, as peças de trabalho poderão ter de esperar antes de entrarem na estação seguinte e/ou as estações podem ter tempos ociosos quando têm de esperar pela próxima peça. Para resolver parcialmente este problema, são colocados *buffers* entre as estações (linhas de produção *buffered*), sendo necessário decidir o dimensionamento e posicionamento desses mesmos *buffers*.

- **Número de modelos em produção:** Quando apenas um modelo é produzido, trata-se de uma linha *single-model*.

Quando vários modelos são produzidos, pode-se tratar de uma linha *multi-model* ou *mixed-model*. Nas linhas *mixed-model*, modelos diferentes são produzidos numa sequência onde se misturam, sendo necessário definir esse sequenciamento (ver secção 3.1.3). Nas linhas *multi-model*, os modelos são produzidos numa sequência de lotes (cada um apenas com um modelo ou grupo de modelos semelhantes) com operações de *setup* intermédias. Neste caso, para além do sequenciamento, é preciso também fazer o dimensionamento dos lotes.

- **Tempos de execução das tarefas:** Os tempos de execução são considerados determinísticos quando a sua variabilidade esperada é suficientemente pequena para poderem ser tratados como constantes. Por outro lado, quando há variações consideráveis, principalmente devido à instabilidade humana em relação ao ritmo de trabalho, capacidades e motivação, os tempos de execução têm de ser estocásticos. Além disso, os tempos também podem sofrer reduções sistemáticas como efeito da aprendizagem ou da melhoria contínua do processo produtivo.
- **Layout da linha:** Tradicionalmente, uma linha de produção está organizada em série, estando as estações posicionadas ao longo de uma cadeia transportadora em linha reta.

Contudo, existem também linhas em U, onde as estações podem trabalhar simultaneamente em dois troços da linha que estejam de frente um para o outro (estações *crossover*). As linhas U, para além de benefícios no enriquecimento do trabalho e na ocupação de uma menor área pela linha, podem permitir um melhor balanceamento pois aumentam o número de combinações tarefa/estação possíveis.

É também possível encontrar linhas em que existem estações paralelas, isto é, as peças são distribuídas por diversas estações que efetuam as mesmas tarefas. Estas estações permitem a redução do tempo de ciclo se certas tarefas tiverem tempos maiores que o tempo de ciclo desejado.

Por fim, um *layout* também importante de mencionar é o das linhas de produção de produtos pesados, onde muitas vezes a linha tem dois segmentos paralelos, havendo pares de estações a trabalhar simultaneamente, em lados opostos do produto a processar.

- **Tarefas do processo produtivo:** No que toca ao processo produtivo, por vezes, este pode permitir escolhas alternativas, isto é, uma determinada tarefa pode ser realizada por equipamentos ou tecnologias diferentes, sendo necessário tomar a decisão de selecionar a alternativa do processo a implementar. Além disso, há também restrições relativas aos operadores/equipamento (só operadores/equipamento com um certo grau de capacidade podem realizar determinadas tarefas) e ao posicionamento da peça, no caso de peças pesadas. Por fim, é importante referir outras restrições que surgem de incompatibilidades entre tarefas, como distâncias mínimas e máximas entre estações responsáveis por um grupo de tarefas.

Possíveis funções objetivo

As funções objetivo mais comuns nos problemas de balanceamento já foram definidas para o caso do SALBP, podendo também ser aplicadas a qualquer GALBP. Assim, qualquer um dos seguintes objetivos é aplicável a um problema de balanceamento:

- Maximizar a eficiência da linha $E = t_{sum}/(m.c)$, com n tarefas e tempo total de tarefas $t_{sum} = \sum_{j=1}^n t_j$.
- Encontrar uma solução factível para um número de estações e um tempo de ciclo dados.
- Minimizar o tempo de ciclo para um número de estações dado.
- Minimizar o número de estações para um tempo de ciclo dado.
- Minimizar simultaneamente o tempo de ciclo e o número de estações.

Por vezes, os problemas de balanceamento são multi-objetivo, sendo usual aliar a um dos objetivos referidos anteriormente o de suavizar as diferenças de carga entre as diversas estações (também conhecido como balanceamento vertical).

Existem também objetivos orientados aos custos, como minimizar o investimento financeiro relativo a estações de trabalho, e aos lucros, como maximizar a margem de lucro por turno.

Quando se utilizam modelos estocásticos para os tempos das tarefas, surgem objetivos orientados ao tempo. Esses objetivos podem ser, por exemplo, minimizar a probabilidade de exceder o tempo de ciclo em qualquer estação ou minimizar as variações estocásticas dos tempos das estações.

Por fim, para o caso específico das linhas *mixed-model*, é necessário fazer o balanceamento horizontal, que consiste em suavizar as diferenças dos tempos de processamento entre os diversos modelos, para cada uma das estações de trabalho. Este objetivo é fundamental para evitar ineficiências como sobrecargas ou tempos ociosos.

3.1.2.3 Análise a métodos de resolução já utilizados

Chen et al. (18) propuseram uma solução para o balanceamento de uma linha de costura na indústria do calçado. Na sua abordagem, começaram por definir uma heurística construtiva para gerar uma solução inicial. Depois, é executado um algoritmo de *branch and bound* para encontrar a solução ótima. O algoritmo utiliza a solução inicial para evitar ter que explorar alguns sub-ramos, tornando-se assim mais rápida a sua execução. O método foi testado com uma série de problemas de teste e chegou-se à conclusão que o *branch and bound* melhorava significativamente a solução inicial dada pela heurística, especialmente para modelos com muitas tarefas e grafos de precedência com diversos ramos.

Simaria e Vilarinho (19) desenvolveram uma abordagem para o balanceamento de linhas *mixed-model* com estações de trabalho paralelas. O método começa por determinar uma solução inicial através de uma versão da heurística *ranked positional weight*. Depois, é utilizado um

algoritmo baseado em *simulated annealing* para encontrar uma solução que minimize o número de estações para um dado tempo de ciclo. Quando o critério de paragem for alcançado (o que neste caso acontece quando a temperatura toma o valor 10), é corrido um segundo algoritmo de *simulated annealing* que tem como objetivo equilibrar a quantidade de trabalho entre as estações (balanceamento vertical) e dentro das mesmas (balanceamento horizontal).

Os mesmos autores (20) resolveram de forma semelhante um problema de balanceamento onde o número de estações era constante e se queria minimizar o tempo de ciclo. O método desenvolvido começa por gerar uma solução inicial através de uma heurística construtiva que usa três regras prioritárias: número máximo de sucessores, máximo tempo de processamento médio e máximo peso posicional. Depois, são executadas duas fases de um algoritmo genético: na primeira fase, o algoritmo minimiza o tempo de ciclo da linha, na segunda fase, faz o balanceamento horizontal do sistema.

Bagher et al. (21) utilizaram um algoritmo imperialista competitivo para minimizar o número de estações e o tempo ocioso e a probabilidade de não-conclusão de uma tarefa de cada estação. A linha de produção em causa era uma linha em U com tempos de processamento estocásticos. O algoritmo foi testado para problemas de diferente dimensão e os resultados indicaram que este tinha maior taxa de convergência que o algoritmo genético proposto por Baykasoğlu e Özbakir (22), que era o melhor algoritmo proposto até ao momento para resolver um problema com aquelas características.

3.1.3 Sequenciamento de linhas de produção

Nas linhas de produção do tipo *mixed-model*, que produzem diversos modelos de um produto simultânea e continuamente, é importante determinar a sequência de modelos a produzir de forma a garantir o uso eficiente da linha.

O número total de possíveis sequências diferentes de modelos a produzir, definido como N , é dado por (23):

$$N = \frac{(\sum_{m=1}^M d_m)!}{\prod_{m=1}^M (d_m!)} \quad (3.1)$$

onde M é o número de modelos diferentes, m é o tipo de modelo e d_m é a procura do modelo m . À medida que o problema cresce em tamanho, o número de soluções admissíveis cresce exponencialmente.

O problema base de sequenciamento de linhas de produção, bem como as características que definem as variações mais comuns do mesmo, são apresentados nas próximas secções. São também explorados métodos de resolução já utilizados para problemas de sequenciamento de linhas.

3.1.3.1 Definição do problema de sequenciamento

No processo de balanceamento de linhas de produção *mixed-model*, o tempo de ciclo definido é usualmente determinado de modo a que se verifique em média para todos os modelos, isto é,

com base na procura e nos tempos de processamento para cada modelo, define-se um tempo de ciclo médio que é, na verdade, menor do que o verificado para alguns modelos e maior do que o verificado para outros (17).

Se diversos modelos com tempos de processamento maiores para uma determinada estação forem produzidos consecutivamente, o trabalhador dessa estação não será capaz de produzi-los no tempo de ciclo médio definido, gerando-se uma situação de sobrecarga. Para resolver esta situação, será necessária uma destas quatro reações (24):

- Parar a linha de produção até que todas as estações tenham terminado as tarefas no produto a ser trabalhado.
- Trabalhadores utilitários ajudam o operador da estação a terminar o trabalho antes que a peça chegue ao fim da estação.
- O produto não é processado na estação de trabalho nem nas estações subsequentes, sendo removido no fim da linha de produção, para ser processado numa estação especial de acabamento de produtos.
- O operador da estação acelera a velocidade de produção, aumentando a probabilidade de produzir uma peça defeituosa.

Para evitar estas compensações, que têm elevados custos, o sequenciamento de linhas de produção tem como objetivo encontrar sequências onde os modelos com tempos de processamento maiores alternam com modelos com tempos de processamento menores, em cada uma das estações. Para este efeito, o sequenciamento da linha tem como *input* as características das estações de trabalho e da linha de produção, bem como os objetivos a alcançar com o sequenciamento.

Características das estações de trabalho

- **Limites das estações:** No que toca aos limites das estações, estas podem-se classificar como fechadas ou abertas. Nas estações fechadas, o trabalhador não pode sair da sua estação de trabalho, o que acontece, por exemplo, quando a sua área de trabalho exige condições ambientais específicas. Nas estações abertas, o trabalhador pode mover-se para lá da sua estação até limites específicos, que poderão ser, por exemplo, o alcance máximo do seu equipamento de trabalho.
- **Reação à sobrecarga:** Quando o operador de uma estação não consegue acabar as tarefas designadas antes do produto sair da estação, há uma sobrecarga. Existem várias reações à sobrecarga possíveis, como parar a linha de produção até que as tarefas relativas ao produto estejam concluídas, retirar o produto da linha de produção para ser concluído fora da linha ou acelerar a velocidade de processamento do produto (24).
- **Tempos de processamento:** Os tempos de processamento podem ser determinísticos para tarefas simples ou automáticas. Contudo, para tarefas com grande participação humana,

deve-se considerar tempos de processamento estocásticos, fruto da variabilidade humana em relação ao ritmo de trabalho, capacidade, fadiga e motivação. Além disso, também se podem considerar reduções sistemáticas dos tempos, fruto de um efeito de aprendizagem ou da melhoria contínua do processo produtivo. (17).

- **Trabalho paralelo:** Trabalho paralelo permite aos operadores de uma estação começarem a processar um produto antes da estação anterior concluir as tarefas relativas a esse mesmo produto. Para se poder realizar, as estações de trabalho devem ser abertas.
- **Setups:** Tempos e custos de *setup* são considerados se a mudança do modelo a ser processado numa determinada estação implicar tempos e custos extra.
- **Estações paralelas:** No caso de haver tarefas com tempos de processamento muito maiores que os das restantes tarefas, são frequentemente instaladas estações paralelas, que alternadamente recebem os produtos a ser processados da estação anterior. Este paralelismo permite reduzir o tempo de ciclo do sistema produtivo (17).

Características da linha de produção

- **Número e homogeneidade das estações:** Tendo em conta que as linhas de produção são compostas por mais do que uma estação, estas podem combinar estações com diferentes características, sendo neste caso designadas por linhas híbridas (24).
- **Layout da linha:** Normalmente, a linha de produção é desenhada numa configuração em série, onde as estações formam uma linha reta. Contudo, existem outras configurações possíveis, como é o caso das linhas em forma de U, nas quais estações de trabalho podem fazer tarefas em dois segmentos da linha diferentes que estejam na sua área (estações de *crossover*), permitindo um maior número de combinações de tarefa/estação (17).
- **Disciplina de lançamento:** O lançamento de peças na linha pode ser feito a uma taxa fixa (*fixed rate launching*), isto é, o intervalo de tempo entre duas peças serem lançadas no sistema é constante, ou então pode ser variável (*variable rate launching*). Se a taxa de lançamento for variável, esta pode ser adaptada de modo a prevenir tempos ociosos e sobrecargas, aumentando a flexibilidade da linha (24).
- **Velocidade de retorno do trabalhador:** Quando um trabalhador acaba as tarefas designadas relativas a um dado produto, ele retorna para o ponto de chegada da nova peça a ser processada na sua estação. Este tempo pode ser menosprezado, tratado como um tempo fixo ou calculado usando velocidades de retorno finitas (24).
- **Buffers intermédios:** As estações de trabalho podem ter *buffers* intermédios que permitem alguma independência das estações a jusante na linha de produção (23). Estas linhas são denominadas de linhas *buffered* (17).

Objetivos do sequenciamento

Os problemas de sequenciamento de linhas de produção são usualmente problemas multi-objetivo. Os principais tipos de funções objetivos para sequenciamento são (23):

- **Manter uma taxa constante de uso de peças:** Nos sistemas *just-in-time* é necessário manter um abastecimento contínuo e estável de peças. Isto pode ser conseguido minimizando a diferença entre as taxas de consumo das peças que se verificam ao longo do tempo e as taxas médias de consumo esperadas (25).
- **Minimizar a variação das taxas de produção:** Minimizar a variação das taxas de produção é um objetivo equivalente a manter uma taxa constante de uso de peças, desde que os modelos do produto sejam compostos por um *mix* de peças aproximadamente igual.
- **Minimizar a sobrecarga de trabalho:** Consiste em minimizar o tempo para lá do disponível que seria gasto nas estações se não houvesse qualquer tipo de reação à sobrecarga. Este objetivo também é designado por minimizar o trabalho utilitário total, quando a reação à sobrecarga é designar um trabalhador utilitário para cumprir as tarefas em falta.
- **Minimizar o tempo/custo de *setup*:** Os *setups* ocorrem quando o tipo de modelo a ser processado por uma dada estação muda. Os custos desta mudança normalmente dependem da combinação estação/tipo de modelo atual/tipo de modelo seguinte.
- **Minimizar o comprimento da linha:** Indiretamente este objetivo visa minimizar o custo de investimento no sistema de transporte da linha de produção, que é proporcional ao comprimento da mesma.
- **Nivelar a carga das estações:** Este objetivo pretende garantir que, para cada estação, a taxa de utilização se mantém estável ao longo do tempo. Tal permite reduzir a congestão da linha e, assim, aumentar a capacidade da mesma.
- **Minimizar o *makespan*:** O *makespan* é o intervalo de tempo entre o início da produção do primeiro produto e o fim da produção do último.
- **Minimizar a duração das paragens da linha:** No caso da reação à sobrecarga ser parar a linha até todas as tarefas a serem realizadas estarem concluídas, é útil minimizar estes tempos de paragem para que a linha seja eficiente.

3.1.3.2 Análise a métodos de resolução já utilizados

Pela revisão bibliográfica elaborada, verificou-se que a maioria dos métodos de resolução são heurísticas e meta-heurísticas, existindo poucos artigos relativos a métodos exatos de resolução do problema de sequenciamento (24).

Rahimi-Vahed e Mirzaei (26) criaram um algoritmo híbrido baseado nas meta-heurísticas *shuffled frog-leaping algorithm* e *bacteria optimization* para resolver um problema de sequenciamento

multi-objetivo numa linha de produção *mixed-model*. Os objetivos definidos foram a minimização do trabalho utilitário total, da variação da taxa total de produção e dos custos de *setup*. O algoritmo foi testado com diversas instâncias do problema de sequenciamento e os resultados obtidos indicaram que este tem melhor *performance* que os algoritmos genéticos multi-objetivo, especialmente para problemas de grande dimensão.

Akgündüz e Tunalı (25) desenvolveram um algoritmo genético multi-objetivo para resolver um problema de sequenciamento numa linha *mixed-model*, onde usaram um mecanismo de controlo adaptativo de parâmetros para melhorar o funcionamento do algoritmo. Este mecanismo adaptativo decide a probabilidade de mutações e de preservação das elites a partir do *feedback* obtido durante a execução do algoritmo. Os resultados experimentais mostraram que a versão adaptativa proposta do algoritmo gera mais e melhores soluções para o problema do que a versão não-adaptativa.

Javadi e Rahimi-Vahed (27) propuseram um método baseado em *fuzzy goal programming* para resolver um problema de sequenciamento com três objetivos: minimização do trabalho utilitário total, da variação da taxa total de produção e dos custos de *setup*. O método foi testado com problemas de teste e foi possível chegar à sequência ótima para todos os problemas.

Bard et al. (28) desenvolveram um modelo que envolvia dois objetivos: minimizar o comprimento da linha de produção e manter a taxa de utilização de cada uma das peças constante. Para solucionar o problema, utilizaram a soma ponderada das duas funções objetivo para obter um único objetivo e aplicaram depois um método de pesquisa tabu.

McMullen e Frazier (29) consideraram um modelo semelhante no qual se procurou minimizar os custos de *setup* e manter a taxa de utilização das peças constante. Ao contrário de (28), onde se aplicou pesquisa tabu, neste caso foi utilizada uma meta-heurística baseada em *simulated annealing* para resolver o problema. As soluções obtidas para problemas de teste mais pequenos ficaram próximas do ótimo.

3.1.4 Integração do balanceamento com o sequenciamento de linhas de produção

O sequenciamento a curto-prazo dos modelos a produzir numa linha de produção depende fortemente do balanceamento a médio-longo prazo definido para essa mesma linha.

O balanceamento define que tarefas são alocadas a uma dada estação de trabalho, determinando assim o conteúdo de trabalho e o uso de material de produção para cada estação. A solução criada constitui o *input* para o problema de sequenciamento.

Tendo em conta esta relação de dependência entre os dois problemas, diversos autores propuseram uma abordagem que visasse resolver os dois simultânea e integradamente.

Kim et al.(30) propôs um algoritmo coevolucionário para resolver o problema combinado com o objetivo de minimizar o trabalho utilitário total.

Özcan et al. (31) desenvolveu um algoritmo genético para o balanceamento e sequenciamento de uma linha de produção em U com tempos de processamento estocásticos.

Lian et al. (32) propôs um método de resolução baseado num algoritmo competitivo imperialista modificado para o problema integrado numa linha de produção em U. A modificação

introduzida no algoritmo foi a inclusão de um novo tipo de país, o país independente. Após a realização de testes, os autores concluíram que o algoritmo criado tinha melhor performance que a versão original e que a maioria das meta-heurísticas desenvolvidas até ao momento.

3.2 Métodos de otimização

Tendo feito uma revisão dos métodos utilizados mais regularmente para resolver problemas de balanceamento e sequenciamento de linhas de produção, é importante analisar sucintamente os métodos mais utilizados para resolver problemas de otimização combinatória.

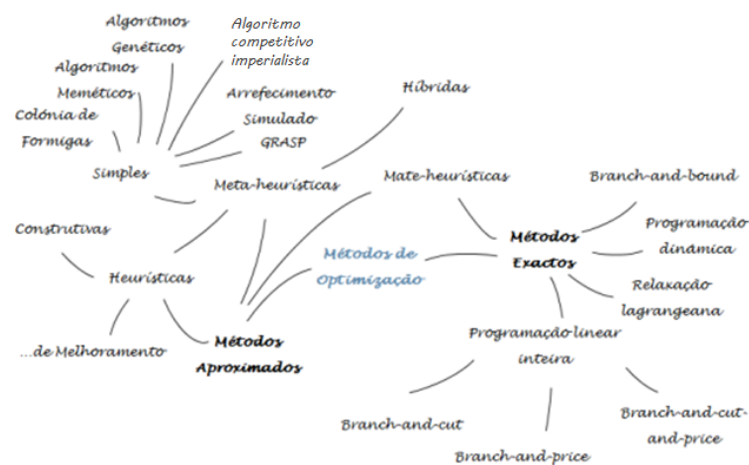


Figura 3.2: Principais métodos de otimização, adaptado de (2)

Como se pode ver pela figura 3.2, os métodos de otimização podem ser divididos em métodos exatos e métodos aproximados.

Os métodos exatos garantem que se encontra uma solução ótima para o problema, porém o seu tempo de execução cresce exponencialmente com o aumento da instância do problema, o que faz com que, frequentemente, apenas pequenas instâncias se possam resolver por estes métodos. O *branch-and-bound* e a programação dinâmica são exemplos destes métodos.

Os métodos aproximados, como as heurísticas e as meta-heurísticas, fazem uma pesquisa no espaço de soluções, procurando boas soluções para o problema. Ao contrário dos métodos exatos, os aproximados não são capazes de garantir a optimalidade da solução, porém são capazes de chegar a boas soluções num espaço de tempo limitado. Assim, existe um *trade-off* entre eficácia (capacidade de chegar à solução ótima) e eficiência (capacidade de chegar a uma boa solução num dado espaço de tempo) no que toca à utilização de um método em detrimento do outro.

Por fim, existe ainda a hipótese de combinar os dois tipos de métodos, algo que se designa como sendo mate-heurísticas. Nestes métodos, um método exato e um aproximado são usados conjuntamente, trocando informação entre si, o que permite frequentemente melhor desempenho no que diz respeito à qualidade da solução e tempo para obtê-la (33).

Nas secções seguintes, descrever-se-á com mais pormenor um método exato, o *branch-and-bound*, e um método aproximado, o algoritmo competitivo imperialista, visto que são os métodos utilizados na resolução do caso em estudo.

3.2.1 *Branch-and-bound*

Este método de otimização é utilizado para resolver problemas de programação linear inteira. O *branch-and-bound* começa por fazer uma relaxação linear do problema, isto é, retira as restrições que obrigam as variáveis a ser inteiras, transformando as mesmas em variáveis contínuas. Partindo desta relaxação, o algoritmo faz uma enumeração inteligente das soluções candidatas a soluções ótimas inteiras, particionando sucessivamente o espaço de soluções e cortando a árvore de pesquisa. Os cortes na árvore de pesquisa são feitos considerando os limites inferiores e superiores para o valor da função objetivo na solução ótima que são calculados ao longo da enumeração.

Estes limites tornam o algoritmo de *branch-and-bound* mais eficiente ao permitir descartar nós da árvore de pesquisa ainda não completamente explorados, pela certeza de que nunca originarão soluções melhores do que as que já foram obtidas. Por outro lado, permitem medir a distância (em termos de valor da função objetivo) a que se está da solução ótima.

3.2.2 Algoritmo competitivo imperialista

O algoritmo competitivo imperialista é uma meta-heurística populacional que se inspirou no processo sócio-político de competição imperialista. Tal como noutras meta-heurísticas populacionais, este algoritmo mantém uma população de soluções para o problema ao longo do processo de otimização.

De acordo com a notação própria do algoritmo, cada solução corresponde a um país, o seu valor da função objetivo é o custo e a sua aptidão (vulgarmente designada por *fitness*) é o poder do país.

O algoritmo competitivo imperialista segue o fluxograma representado na figura 3.3. Os quatro principais passos do algoritmo são os seguintes:

- **Criação da população inicial:** Cada indivíduo da população é um país, que é gerado aleatoriamente ou através do uso de regras heurísticas.
- **Construção inicial dos impérios:** Os países da população inicial são classificados como sendo imperialistas ou colónias, com base no seu valor da função objetivo. Inicialmente, alguns dos melhores países são seleccionados como imperialistas e os restantes formam as colónias. A cada colónia é depois atribuído um imperialista e o conjunto de um imperialista com as suas colónias constitui um império. O número de indivíduos N_{pop} e o número de imperialistas N_{imp} da população inicial são parâmetros do algoritmo.
- **Assimilação e mutação:** A assimilação é feita em cada império a cada iteração do algoritmo. Refere-se ao processo pelo qual o imperialista num império tenta influenciar todas as

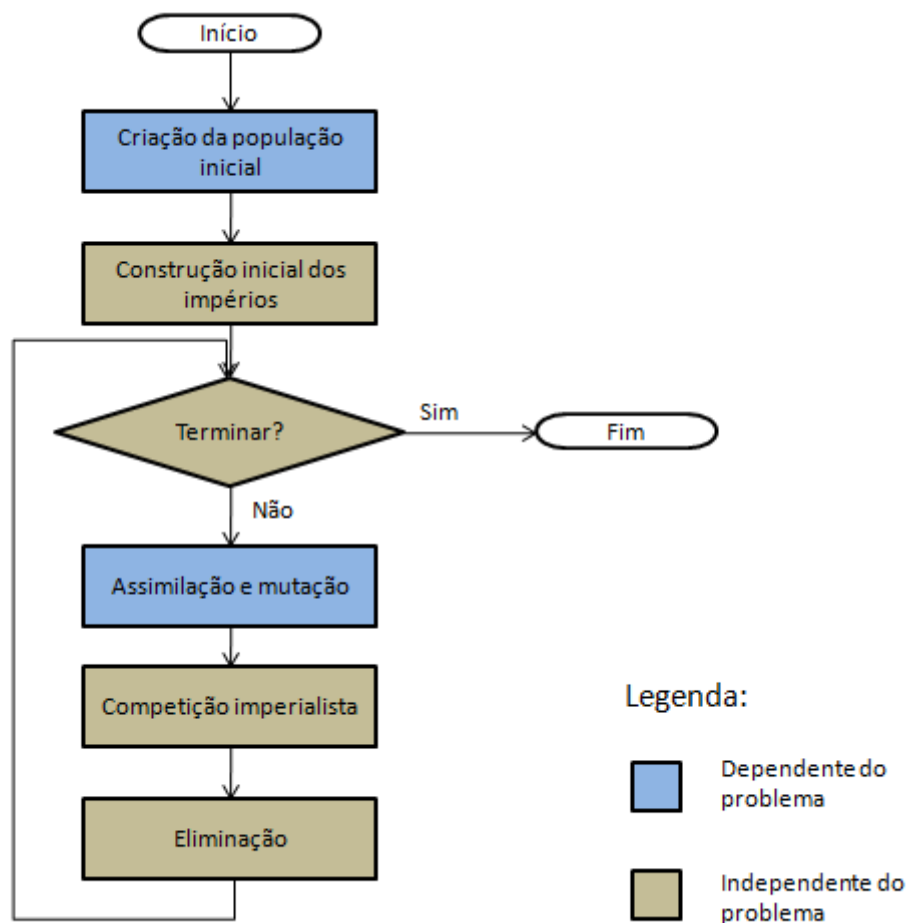


Figura 3.3: Fluxograma do algoritmo competitivo imperialista

suas colónias a serem mais semelhantes a si mesmo. Após a assimilação, é feita uma mutação a cada uma das colónias para aumentar a diversidade da população. Se depois destes dois processos, uma colónia for melhor que o seu imperialista, esta torna-se no imperialista e vice-versa.

- **Competição imperialista:** Nesta fase, a colónia mais fraca do império mais fraco é libertada e todos os impérios competem para conquistar essa colónia. Cada império tem uma probabilidade de conquista que é proporcional ao seu poder total. A competição imperialista torna alguns impérios mais fortes e outros mais fracos.
- **Eliminação:** A competição imperialista poderá levar a impérios sem colónias, visto que estas foram conquistadas por outros impérios. Neste caso, o império sem colónias é eliminado da população.

Como se pode verificar pela figura 3.3, algumas fases do algoritmo são independentes do problema em causa, enquanto outras são definidas caso a caso, consoante o problema a resolver. De seguida, descrever-se-á com maior pormenor as fases independentes do problema. As fases

dependentes do caso em estudo são explicadas no capítulo 5, visto que remetem já à abordagem que foi feita para a resolução do mesmo.

3.2.2.1 Construção inicial dos impérios

Após a criação da população inicial, alguns dos melhores países são selecionados para serem os imperialistas. O tamanho da população inicial é N_{pop} , o número de imperialistas é N_{imp} e o número de colónias é N_{col} . Para distribuir as colónias pelos imperialistas, é necessário calcular o custo normalizado de cada imperialista usando a seguinte fórmula:

$$C_n = \max_{i=1, \dots, N_{imp}} c_i - c_n \quad (3.2)$$

onde c_n é o custo do n -ésimo imperialista e C_n é o seu custo normalizado. As colónias são distribuídas pelos imperialistas com base no seu poder normalizado. O poder normalizado p_n de cada imperialista é definido como:

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (3.3)$$

O número de colónias do império n , NC_n , é então dado por:

$$NC_n = \text{round}(p_n \times N_{col}) \quad (3.4)$$

Por fim, cada colónia é atribuída aleatoriamente a um imperialista até se perfazer o número de colónias de cada império.

3.2.2.2 Competição imperialista

No algoritmo competitivo imperialista, todos os impérios competem para conquistar mais colónias para além daquelas que possuem. A competição imperialista resulta gradualmente num aumento do poder dos impérios mais fortes e numa diminuição do poder dos impérios mais fracos. Para modelar esta competição, a colónia mais fraca do império mais fraco é libertada do seu imperialista atual. Os impérios competem então para conquistar esta colónia, sendo que impérios com maior poder têm maior probabilidade de conquista.

O poder total de um império é determinado pelo poder do seu imperialista e das suas colónias. O custo total do império n é dado pela expressão:

$$CT_n = c_n(\text{imperialista}_n) + \alpha \times \text{media}(c_n(\text{colonias do imperio}_n)) \quad (3.5)$$

onde CT_n é o custo total do império n e α é um número entre 0 e 1 que reflete o peso das colônias no custo total do império. O valor de α é um parâmetro do algoritmo. O custo total normalizado do império n , CTN_n , é dado por:

$$CTN_n = \max_{i=1, \dots, N_{imp}} CT_i - CT_n \quad (3.6)$$

Finalmente, a probabilidade de conquista para o império n é dada por:

$$P_{q_n} = \frac{CTN_n}{\sum_{i=1}^{N_{imp}} CTN_i} \quad (3.7)$$

3.2.2.3 Critério de paragem

O critério de paragem define quando o algoritmo deve parar de ser executado. Existem diversas hipóteses para este critério, sendo as mais comuns o número de iterações, o tempo de execução do algoritmo e a convergência da solução.

Capítulo 4

Modelo de otimização

A criação de modelos de otimização contribui para uma melhor compreensão de problemas em estudo, é ponto de partida para a sua resolução segundo métodos exatos, já analisados no capítulo 3, e também pode ajudar em abordagens por métodos aproximados, métodos que efetivamente serão utilizados no capítulo 5. Os modelos de otimização também poderão servir para a validação de soluções obtidas por métodos aproximados.

Assim, para a resolução do caso em estudo, começou-se por criar um modelo matemático do problema de otimização associado ao caso de estudo descrito anteriormente. Apresenta-se seguidamente o referido modelo, indicando com clareza os seus parâmetros, variáveis de decisão, restrições e função objetivo escolhida.

4.1 Parâmetros

Os parâmetros do modelo são os seguintes:

C: número de caixas

P: número de postos

N_i : número de operações da caixa i . $i \in \{1, \dots, C\}$

T_{trans} : tempo de transporte de uma caixa entre o armazém e um posto, ou vice-versa

t_{ik} : tempo de duração da operação k da caixa i . $i \in \{1, \dots, C\}$, $k \in \{1, \dots, N_i\}$

λ_{ikl} : parâmetro binário: 1 se a operação k da caixa i é feita após a operação l da caixa i , 0 no caso contrário. $i \in \{1, \dots, C\}$, $k \in \{1, \dots, N_i\}$, $l \in \{1, \dots, N_i\} \setminus k$

ρ_{ikp} : parâmetro binário: 1 se a operação k da caixa i pode ser executada no posto p , 0 no caso contrário. $i \in \{1, \dots, C\}$, $k \in \{1, \dots, N_i\}$, $p \in \{1, \dots, P\}$

M: número "muito grande"(ex.: 100000)

4.2 Variáveis de decisão

As variáveis de decisão do modelo são as seguintes:

s_{ik} : tempo de início da operação k da caixa i . $i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}$

α_{ikp} : variável binária: 1 se a operação k da caixa i é executada no posto p , 0 no caso contrário.
 $i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, p \in \{1, \dots, P\}$

$\delta_{ik\tilde{i}\tilde{k}}$: variável binária: 1 se a operação k da caixa i é feita após a operação \tilde{k} da caixa \tilde{i} , 0 no caso contrário. $i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{i} \in \{1, \dots, C\} : \tilde{i} \geq i, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : (\tilde{i} > i \vee \tilde{k} > k)$

$\phi_{ik\tilde{i}\tilde{k}p}$: variável binária: 1 se a operação k da caixa i e a operação \tilde{k} da caixa \tilde{i} são realizadas no posto p , 0 no caso contrário. $i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{i} \in \{1, \dots, C\} : \tilde{i} \geq i, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : (\tilde{i} > i \vee \tilde{k} > k), p \in \{1, \dots, P\}$

$\omega_{ik\tilde{i}\tilde{k}}$: variável binária: 1 se a operação k da caixa i é realizada no mesmo posto que a operação \tilde{k} da caixa \tilde{i} , 0 no caso contrário. $i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{i} \in \{1, \dots, C\} : \tilde{i} \geq i, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : (\tilde{i} > i \vee \tilde{k} > k)$

T : variável que representa o *makespan*

4.3 Restrições

As restrições do problema, bem como a sua formulação matemática, são as seguintes:

- Um posto só pode realizar uma operação de cada vez, modelada pelas equações 4.1 e 4.2.

$$s_{ik} + (1 - \delta_{ik\tilde{i}\tilde{k}}) * M + (1 - \omega_{ik\tilde{i}\tilde{k}}) * M \geq s_{\tilde{i}\tilde{k}} + t_{\tilde{i}\tilde{k}}, \quad (4.1)$$

$$\forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{i} \in \{1, \dots, C\} : \tilde{i} \geq i, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : (\tilde{i} > i \vee \tilde{k} > k)$$

$$s_{\tilde{i}\tilde{k}} + \delta_{ik\tilde{i}\tilde{k}} * M + (1 - \omega_{ik\tilde{i}\tilde{k}}) * M \geq s_{ik} + t_{ik}, \quad (4.2)$$

$$\forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{i} \in \{1, \dots, C\} : \tilde{i} \geq i, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : (\tilde{i} > i \vee \tilde{k} > k)$$

- Um posto só pode processar uma operação para a qual esteja apto, modelada pela equação 4.3.

$$\alpha_{ikp} \leq \rho_{ikp}, \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, p \in \{1, \dots, P\} \quad (4.3)$$

- Uma operação de uma caixa é realizada num e num só posto, modelada pela equação 4.4.

$$\sum_{p=1}^P \alpha_{ikp} = 1, \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\} \quad (4.4)$$

- As equações 4.5 e 4.6 são restrições auxiliares para definir o valor de $\phi_{ik\tilde{i}k}$.

$$\begin{aligned} \phi_{ik\tilde{i}k} &\geq \alpha_{ikp} + \alpha_{\tilde{i}kp} - 1, \\ \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{i} \in \{1, \dots, C\} : \tilde{i} \geq i, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : (\tilde{i} > i \vee \tilde{k} > k), p \in \{1, \dots, P\} \end{aligned} \quad (4.5)$$

$$\begin{aligned} \phi_{ik\tilde{i}k} &\leq \frac{\alpha_{ikp} + \alpha_{\tilde{i}kp}}{2}, \\ \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{i} \in \{1, \dots, C\} : \tilde{i} \geq i, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : (\tilde{i} > i \vee \tilde{k} > k), p \in \{1, \dots, P\} \end{aligned} \quad (4.6)$$

- As equações 4.7 e 4.8 são restrições auxiliares para definir o valor de $\omega_{ik\tilde{i}k}$.

$$\omega_{ik\tilde{i}k} * M \geq \sum_{p=1}^P \phi_{ik\tilde{i}kp}, \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{i} \in \{1, \dots, C\} : \tilde{i} \geq i, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : (\tilde{i} > i \vee \tilde{k} > k) \quad (4.7)$$

$$\omega_{ik\tilde{i}k} \leq \sum_{p=1}^P \phi_{ik\tilde{i}kp}, \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{i} \in \{1, \dots, C\} : \tilde{i} \geq i, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : (\tilde{i} > i \vee \tilde{k} > k) \quad (4.8)$$

- Para uma dada caixa, sempre que há restrições de precedência entre duas operações, estas devem ser respeitadas, modelada pela equação 4.9.

$$s_{ik} + (1 - \lambda_{ikl}) * M \geq s_{il} + t_{il}, \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, l \in \{1, \dots, N_i\} \setminus k \quad (4.9)$$

- Cada caixa apenas pode realizar uma operação de cada vez e existe um tempo de transporte entre duas operações consecutivas de $2 * T_{trans}$, modelada pelas equações 4.10 e 4.11.

$$\begin{aligned} s_{ik} + (1 - \delta_{ik\tilde{i}k}) * M &\geq s_{\tilde{i}k} + t_{\tilde{i}k} + 2 * T_{trans}, \\ \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : \tilde{k} > k \end{aligned} \quad (4.10)$$

$$\begin{aligned} s_{\tilde{i}k} + \delta_{ik\tilde{i}k} * M &\geq s_{ik} + t_{ik} + 2 * T_{trans}, \\ \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\}, \tilde{k} \in \{1, \dots, N_{\tilde{i}}\} : \tilde{k} > k \end{aligned} \quad (4.11)$$

- Os tempos de início das tarefas não podem ser menores que o tempo de transporte de uma caixa do armazém para um posto, modelada pela equação 4.12.

$$s_{ik} \geq T_{trans}, \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\} \quad (4.12)$$

- O *makespan* é maior ou igual ao tempo de conclusão de todas as operações, modelada pela equação 4.13.

$$T \geq s_{ik} + t_{ik}, \forall i \in \{1, \dots, C\}, k \in \{1, \dots, N_i\} \quad (4.13)$$

4.4 Função objetivo

O objetivo deste modelo é minimizar o *makespan*, isto é, o tempo total de produção, como se pode verificar na equação 4.14.

$$\min \quad T \quad (4.14)$$

4.5 Programação do modelo

O modelo matemático desenvolvido foi programado no *software* ILOG CPLEX Optimization Studio, versão 12.6, da IBM. Partindo do modelo e dos parâmetros do problema a resolver, este *software* aplica um método de otimização exato para chegar à solução ótima do problema. No caso em estudo em específico, que se trata de um problema de programação linear inteira, o CPLEX utiliza o algoritmo *branch-and-bound* (já abordado no capítulo 3) para resolver o problema de forma exata.

Para problemas de grande dimensão, como é o do caso em estudo, o CPLEX pode demorar muito tempo a chegar à solução ou até mesmo não conseguir resolver o problema por estar limitado pela memória RAM do computador onde está a ser executado. Ainda assim, a implementação do modelo em CPLEX e o teste do mesmo com instâncias de menor dimensão do problema tem diversas vantagens para a resolução do problema. Por um lado, permite validar o modelo matemático desenvolvido, através da análise das soluções obtidas. Por outro lado, permite validar o algoritmo competitivo imperialista desenvolvido e avaliar a *performance* do mesmo, comparando as soluções obtidas e o tempo de execução do algoritmo com o modelo em CPLEX, para instâncias pequenas do problemas.

Os resultados da execução do modelo em CPLEX estão expostos na secção 6.1 do capítulo 6.

Capítulo 5

Algoritmo competitivo imperialista

Tendo em conta que para problemas de grande dimensão, como é o caso do problema real da fábrica, os métodos exatos de otimização demoram muito tempo a chegar à solução ótima do problema, foi necessário definir uma abordagem que permitisse a obtenção de uma boa solução para o problema num intervalo de tempo viável para o caso real.

Face à revisão bibliográfica realizada, presente no capítulo 3, a implementação de uma meta-heurística surge como uma abordagem muito adequada para a obtenção de uma boa solução num menor intervalo de tempo.

Das diferentes meta-heurísticas existentes, optou-se pela utilização do algoritmo competitivo imperialista (ACI), já abordado no capítulo 3. Esta é uma meta-heurística recente (foi criada em 2007) e, da revisão bibliográfica feita, ainda não foi aplicada num problema com as características do caso em estudo. Ainda assim, já foi aplicada com sucesso para resolver problemas nesta área, como se indicou na revisão bibliográfica (3.1.2.3 e 3.1.4).

Desta forma, a aplicação do ACI neste problema poderá constituir um contributo interessante no estudo da viabilidade da utilização desta meta-heurística para resolver problemas de balanceamento e sequenciamento de linhas de produção.

Após a definição da meta-heurística a aplicar, estabeleceu-se a estrutura do programa de otimização a criar. O programa de otimização do sequenciamento da produção divide-se em dois blocos fundamentais: leitura de dados e aplicação do ACI. Na fase de leitura de dados, é recolhida toda a informação necessária sobre o funcionamento da linha de produção, que servirá de entrada para a segunda fase do programa, a aplicação do ACI.

Nas seguintes secções, o funcionamento do programa será discutido mais pormenorizada-mente, no que toca quer à leitura dos dados, quer à adaptação do funcionamento geral do ACI, já discutido no capítulo 3, para o caso específico do problema em estudo.

5.1 Leitura de dados

O programa começa por ler a lista de postos disponíveis para a execução de operações, a qual é fornecida através do ficheiro `postos.txt`.

De seguida, são lidos os dados referentes às operações a realizar, nomeadamente: o código identificador da operação, o seu tempo de duração e a lista de postos que estão habilitados a executá-la. Estes dados são lidos do ficheiro `operacoes.txt`.

No que toca à leitura das gamas operatórias, esta é feita de acordo com um algoritmo um pouco mais complexo, sendo interessante abordar o funcionamento do mesmo. As gamas operatórias são lidas a partir do ficheiro `gamas.txt`. Neste ficheiro, cada linha representa uma operação de uma dada gama e as operações da qual esta é precedente imediata.

Assim, a título de exemplo, a gama operatória 1, cujo formato está representado na figura 5.1, seria representada no ficheiro `gamas.txt` como:

```
1, 1, 2, 3
1, 2, 4
1, 3, 4
1, 4
```

O algoritmo então seguido para ler o ficheiro é:

Para todas as linhas de texto do ficheiro:

1. Ler a gama.
2. Se é a primeira vez que a gama é lida, criar a estrutura de dados que a representa no programa.
3. Ler operação da gama.
4. Se é a primeira vez que a operação da gama é lida, criar a estrutura de dados que a representa no programa.
5. Para todas as operações sucessoras:
 - 5.1. Ler operação da gama.
 - 5.2. Se é a primeira vez que a operação da gama é lida, criar a estrutura de dados que a representa no programa.
 - 5.3. Criar as relações de precedência entre a operação lida e a operação predecessora.

Por fim, são lidos os lotes a fabricar. Para cada um deles, é lido o seu código identificador, o tamanho, o código identificador da sua gama operatória e a sua prioridade. Além disso, são também lidas as operações do lote que já foram realizadas, no caso deste já ter começado a ser produzido.

5.2 Aplicação do Algoritmo Competitivo Imperialista

A aplicação do ACI ao caso em estudo só foi possível através da adaptação das características gerais do mesmo às características específicas do problema. Em relação aos parâmetros do ACI, definiu-se N_{pop} como 100, N_{imp} como 7 e α como 0,6. Além disso, foi necessário definir uma codificação das soluções, um mecanismo de geração de soluções iniciais, o método de cálculo da função objetivo e os operadores de assimilação e mutação das soluções.

Antes de explicar estas características do ACI, e para permitir uma melhor compreensão das mesmas, é importante definir um pequeno problema de exemplo, que servirá de base para a ilustração de diversos aspetos do programa.

Este problema consiste numa fábrica onde existem 4 postos de trabalho, identificados pelos números de 1 a 4. Nesta fábrica serão produzidos dois lotes: o lote 1 e o lote 2, cada um constituído por 2 pares de sapatos e com igual prioridade. O lote 1 deve cumprir a gama operatória 1 (figura 5.1), já o lote 2 deve realizar a gama operatória 2 (figura 5.2).

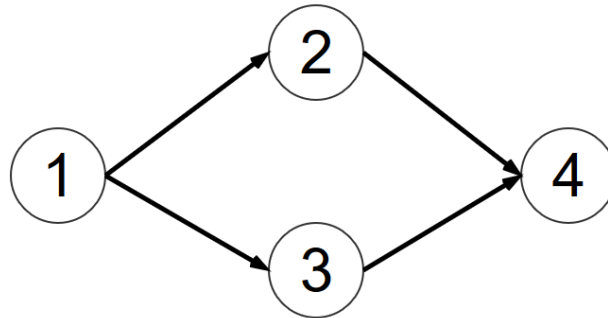


Figura 5.1: Gama operatória 1

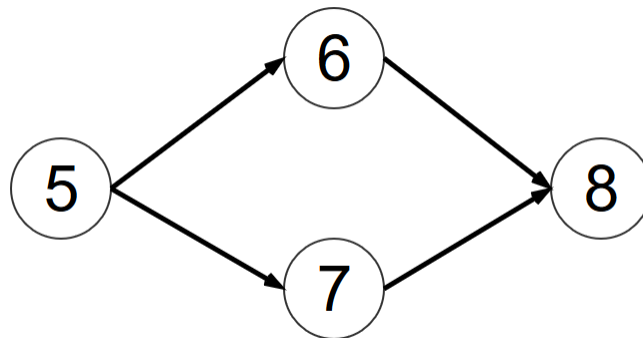


Figura 5.2: Gama operatória 2

Os tempos de duração das operações em segundos, bem como a lista de postos capazes de realizá-las, estão indicados na tabela 5.1.

Os dados até agora mencionados são os dados que o programa de otimização lê, para depois proceder à aplicação do ACI.

5.2.1 Codificação da solução

O primeiro passo para aplicar o ACI consistiu em definir uma codificação para a representação de uma solução.

Assim, definiu-se a seguinte codificação: cada solução possui uma matriz onde se define a sequência das operações a realizar para cada lote e o posto onde estas serão executadas. A figura 5.3 representa esta mesma matriz para uma possível solução do problema exemplo.

Operação	Tempo de duração (s)	Postos admissíveis
1	70.0	3
2	40.0	1, 2
3	20.0	2
4	20.0	1, 2, 3, 4
5	50.0	2
6	60.0	1, 3
7	30.0	1
8	20.0	1, 2, 3, 4

Tabela 5.1: Tempos de duração das operações e postos admissíveis

Cada conjunto de duas linhas consecutivas da matriz refere-se a um dado lote. Neste caso, as duas primeiras linhas referem-se ao lote 1 e as outras duas ao lote 2. As primeira e terceira linhas definem a sequência de operações a realizar para o respetivo lote. As células das segunda e quarta linhas definem o posto que irá executar a operação referida na célula imediatamente acima.

Além disso, cada solução possui ainda um vetor que define a sequência global das operações. Neste vetor, cada elemento indica apenas o lote a que se refere a operação a executar, visto que a sequência das operações para um determinado lote já é definida pela matriz anteriormente exposta. A figura 5.4 representa este mesmo vetor para uma possível solução do problema exemplo.

Neste caso, a primeira operação a ser calendarizada é a primeira operação do lote 1, seguindo-se as duas primeiras operações do lote 2, e assim por diante.

5.2.2 Geração de soluções iniciais

Para gerar uma solução inicial, foi seguido o seguinte algoritmo:

1. Calcular o número total de operações para cada lote.
2. Enquanto o vetor da sequência global de operações não estiver preenchido:
 - 2.1. Escolher aleatoriamente um lote que apareça no vetor um número de vezes menor que o seu número total de operações.
 - 2.2. Acrescentar um novo elemento com este lote ao vetor.
3. Para todos os lotes:
 - 3.1. Para um número de iterações igual ao número de operações do lote:
 - 3.1.1. Verificar quais operações podem ser calendarizadas, isto é, quais operações não têm predecessores diretos por calendarizar.
 - 3.1.2. Escolher uma dessas operações aleatoriamente e calendarizá-la, isto é, incluí-la na matriz da sequência de operações por lote e de postos.
 - 3.1.3. Escolher aleatoriamente um posto da lista de postos admissíveis para realizar essa operação. Colocar este posto na matriz da sequência de operações por lote e de postos, na célula imediatamente abaixo da respetiva operação.

1	2	3	4
3	2	2	2
5	7	6	8
2	1	1	3

Figura 5.3: Matriz exemplo da sequência de operações por lote e dos postos atribuídos

5.2.3 Cálculo da função objetivo

Para cada solução gerada, é necessário calcular o respetivo valor da função objetivo ou, como normalmente é designado na terminologia do ACI, o custo da solução. Este cálculo é feito segundo este algoritmo:

1. Para um número de iterações igual ao número total de operações dos lotes:

1.1. Ler a operação seguinte a calendarizar, cruzando a informação do vetor da sequência global de operações com a matriz da sequência de operações por lote e de postos.

1.2. Se o instante de tempo em que o posto atribuído à operação fica disponível for maior que o instante de tempo em que a operação anterior do lote ficou concluída mais o tempo de transporte do lote de um posto para o armazém mais o tempo de transporte do lote do armazém para um posto:

1.2.1. Definir o tempo de início da operação como sendo o instante de tempo em que o posto que lhe foi atribuído fica disponível.

1.2.2. Atualizar o instante de tempo em que o posto atribuído a esta operação fica disponível para o instante de tempo em que a operação fica concluída.

1.3. Senão:

1.3.1. Definir o tempo de início da operação como sendo o instante de tempo em que a operação anterior do lote ficou concluída mais o tempo de transporte do lote de um posto para o armazém mais o tempo de transporte do lote do armazém para um posto.

1.3.2. Atualizar o instante de tempo em que o posto atribuído a esta operação fica disponível para o instante de tempo em que a operação fica concluída.

1.4. Encontrar o tempo de conclusão para todas as operações máximo e defini-lo como sendo o custo da solução.

5.2.4 Assimilação com o imperialista

A assimilação de uma colónia com o seu imperialista foi feita através de um operador de *crossover* de dois pontos. A assimilação seguiu o seguinte algoritmo:



Figura 5.4: Vetor exemplo da sequência global de operações

1. Escolher aleatoriamente dois elementos da matriz da sequência global de operações do imperialista, que serão os pontos de *crossover*.

2. Copiar elementos do imperialista anteriores ao ponto de *crossover* de menor índice e posteriores ao ponto de *crossover* de maior índice para a nova colônia e apagá-los da antiga colônia.

3. Copiar elementos restantes da antiga colônia para a nova colônia.

4. Para um número de iterações igual ao número de lotes:

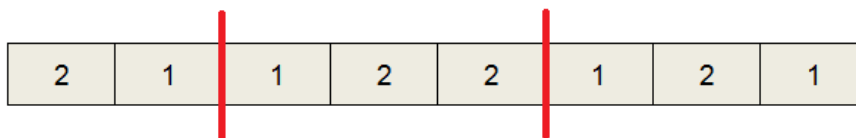
4.1. Escolher aleatoriamente dois elementos da matriz da sequência de operações por lote do imperialista, da linha referente ao lote atual, que serão os pontos de *crossover*.

4.2. Copiar elementos anteriores ao ponto de *crossover* de menor índice e posteriores ao ponto de *crossover* de maior índice para a nova colônia e apagá-los da antiga colônia.

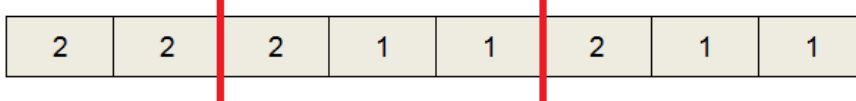
4.3. Copiar elementos restantes da antiga colônia para a nova colônia.

A figura 5.5 exemplifica o processo de assimilação para a sequência global de operações. Já a figura 5.6 demonstra a assimilação na matriz da sequência de operações por lote e dos postos atribuídos.

Imperialista:



Colônia:



Antiga colônia após remover elementos que serão copiados do imperialista:



Nova colônia:



Figura 5.5: Exemplo da assimilação na sequência global de operações

5.2.5 Mutação das soluções

A mutação das soluções foi conseguida através da troca de operações na sequência das mesmas, garantindo sempre que a mutação não gera uma solução que não obedece às relações de precedência entre as operações. Além disso, é também feita uma mutação nos postos de trabalho, sendo atribuído um novo posto de trabalho aleatoriamente para uma operação, também ela escolhida de forma aleatória. O algoritmo seguido para a mutação foi o seguinte:

1. Escolher aleatoriamente dois elementos da matriz da sequência global de operações e trocá-los.

2. Escolher aleatoriamente uma operação de um lote e atribuir-lhe um novo posto, escolhido de forma aleatória entre os postos admissíveis.

3. Escolher aleatoriamente um lote para se mudar a sua sequência de operações.

4. Guardar a sequência de operações desse mesmo lote num vetor auxiliar.

5. Escolher aleatoriamente dois elementos do vetor auxiliar e trocá-los.

6. Criar um vetor que guardará os elementos que se tentou agendar mas não foi possível devido às restrições de precedência.

7. Enquanto a sequência de operações do lote não estiver completa:

- 7.1. Se tiver elementos no vetor das operações que se tentou agendar mas não foi possível devido às restrições de precedência, verificar para todos eles se é possível colocá-los na nova sequência de operações (isto é, verificar se todos os seus predecessores diretos já foram agendados). Se for, então agendar a operação.

- 7.2. Se no passo 7.1. não foi possível agendar nenhuma operação, então ler o próximo elemento do vetor auxiliar. Se todos os seus predecessores diretos já foram agendados, então agendar esta operação. Senão, guardá-lo no vetor das operações que se tentou agendar mas não foi possível devido às restrições de precedência.

Na figura 5.7, é apresentado um exemplo da mutação na sequência global de operações, onde o quarto e o sexto elementos da sequência foram trocados. Na figura 5.8, é apresentado um exemplo da mutação na sequência de operações de um lote, sendo que para o primeiro lote, os primeiro e terceiro elementos foram trocados. Como esta troca não cumpriria as restrições de precedência, o algoritmo reparou a solução gerada, criando a solução apresentada na figura, que cumpre as restrições de precedência. Além disso, foi também efetuada uma mutação no posto atribuído à operação 8 do segundo lote, mudando do posto 1 para o posto 3.

Imperialista:

1	3	2	4
3	2	2	4
5	7	6	8
2	1	1	1

Colônia:

1	3	2	4
3	2	1	4
5	6	7	8
2	1	1	1

Nova colônia:

1	3	2	4
3	2	2	4
5	6	7	8
2	1	1	1

Figura 5.6: Exemplo da assimilação na matriz da sequência de operações por lote e dos postos atribuídos

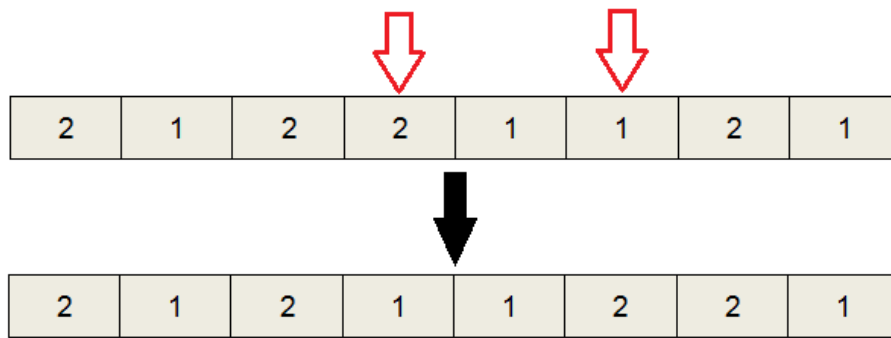


Figura 5.7: Exemplo da mutação na sequência global de operações

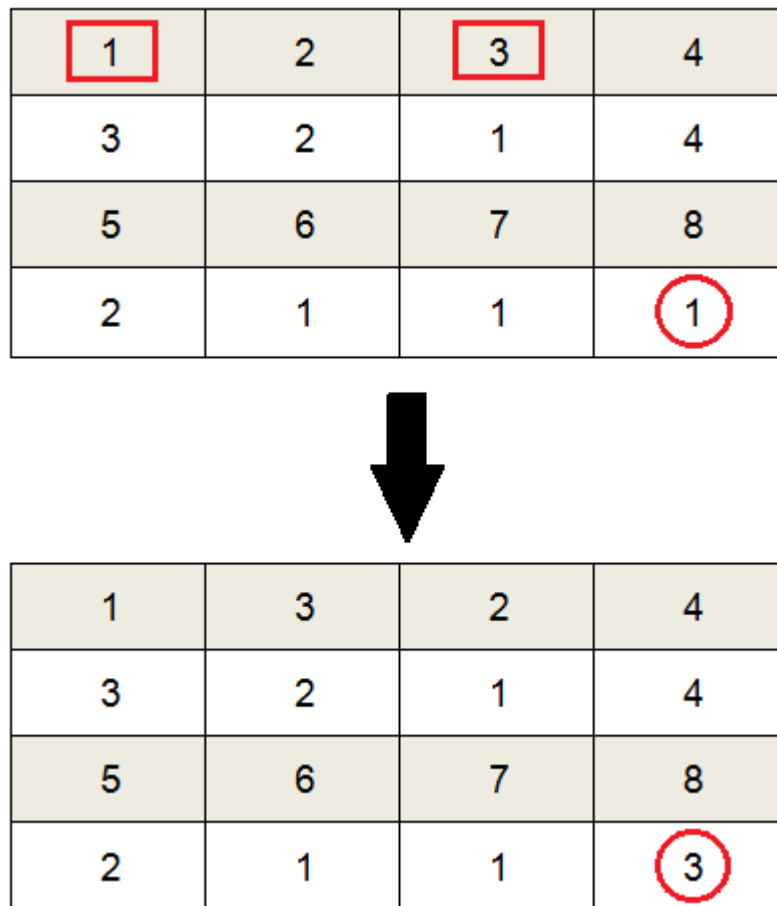


Figura 5.8: Exemplo da mutação na sequência de operações de um lote e nos postos atribuídos

Capítulo 6

Resultados computacionais

Este capítulo apresenta os resultados obtidos, quer pelo modelo de otimização implementado em CPLEX, quer pelo ACI desenvolvido, para um conjunto de problemas de teste e para o problema real.

É importante referir que os problemas de teste utilizados foram criados de modo a terem dimensões diferentes entre si, permitindo analisar o desempenho dos métodos de otimização consoante o tamanho do problema. Este conjunto de problemas de teste foi criado de forma aleatória, isto é, os seus dados foram gerados automaticamente num programa desenvolvido para este efeito. Ainda assim, houve a preocupação de garantir que os problemas gerados pelo programa estavam de acordo com a realidade do caso em estudo. Nomeadamente, garantiu-se que todos os problemas gerados têm simultaneamente caixas com gamas lineares e em grafo e operações que podem ser executadas em diversos postos, mantendo as características do caso em estudo.

A dimensão dos problemas de teste criados, bem como do problema do caso em estudo, está representada na tabela 6.1.

Tabela 6.1: Dimensão dos problemas de teste e do caso em estudo

Problema	Nº de caixas	Nº médio de operações por caixa	Nº de postos
1	5	5	5
2	7	7	7
3	9	9	9
4	11	11	11
5	13	13	13
Caso em estudo	97	9,42	60

Por fim, é importante referir que os resultados apresentados foram obtidos para a execução do modelo em CPLEX e do ACI num computador Acer Aspire E5-521G, com 4 GB de memória RAM e processador AMD A8-6410 (*quad-core* com velocidade 2 GHz).

Tabela 6.2: Resultados da execução do modelo em CPLEX

Problema	Resultados da execução do modelo	
	Solução ótima	Tempo médio de execução do programa (segundos)
1	534	0,562
2	754	1,922
3	970	15,488
4	1202	88,394
5	1385	687,222
Caso em estudo	-	Impossível de executar

6.1 Modelo em CPLEX

A tabela 6.2 indica os resultados da execução do modelo em CPLEX para os diferentes problemas de teste e para o problema real. Para cada problema, o modelo em CPLEX foi executado 5 vezes, de modo a obter um tempo médio de execução. Como se pode constatar, o tempo de execução do modelo cresce exponencialmente com o aumento do problema.

No que toca à utilização deste modelo em CPLEX para o caso em estudo, os resultados indicam que tal não seria viável. Por um lado, para o problema real, não foi possível chegar à solução do problema. Tal deveu-se ao facto de que o problema era de tal modo grande que a memória RAM do computador utilizado não era suficiente para guardar todas as variáveis geradas pela execução do algoritmo. Por outro lado, ainda que se procurasse ultrapassar este obstáculo com a utilização de um computador com maior memória RAM, o crescimento exponencial do tempo de execução do programa com o aumento das dimensões do problema inviabiliza a sua utilização no caso real.

Para a fábrica em questão, é importante que o sequenciamento da produção consiga ser feito num tempo reduzido, nomeadamente, um tempo inferior a 5 minutos, visto que, deste modo, é possível fazer um re-sequenciamento da produção de forma rápida, no caso de surgir algum imprevisto, como uma operação demorar mais tempo do que era suposto ou um posto ter que ser fechado (por exemplo, pelo operador do posto ter ficado doente).

Assim, a utilização do modelo em CPLEX para o caso em estudo não é viável. Porém, os resultados obtidos para os problemas de teste são dados importantes para avaliar o desempenho do ACI.

6.2 Algoritmo competitivo imperialista

6.2.1 Problemas de teste

A tabela 6.3 representa os resultados da execução do ACI para os diferentes problemas de teste e a respetiva comparação com a solução ótima.

O critério de paragem do ACI é o tempo de execução, sendo que este foi sendo adaptado conforme as dimensões do problema.

Tabela 6.3: Resultados da execução do ACI

Dados do problema		Resultados da execução do modelo						
Problema	Solução ótima	Tempo de execução do algoritmo (segundos)	Valor médio da solução	Melhor valor da solução	Pior valor da solução	GAP médio	Melhor GAP	Pior GAP
1	534	3	534	534	534	0%	0%	0%
2	754	3	754	754	754	0%	0%	0%
3	970	60	970	970	970	0%	0%	0%
4	1202	120	1204,2	1202	1213	0,183%	0%	0,915%
5	1385	240	1425,8	1401	1442	2,946%	1,155%	4,116%

A análise dos resultados da execução do ACI permite tirar algumas conclusões importantes.

Por um lado, o facto da solução obtida para as instâncias de menor dimensão do problema ser igual à solução ótima valida o correto funcionamento do ACI, isto é, comprova que o ACI procura efetivamente resolver o problema definido no modelo matemático e respeita as restrições do mesmo, sendo esse o motivo pelo qual chega ao mesmo valor que o da solução ótima.

Por outro lado, é também possível afirmar que o ACI tem um bom desempenho, quer no que toca à sua eficácia, quer no que toca à sua eficiência. Para as 3 menores instâncias, o ACI foi capaz de chegar à solução ótima em pouco tempo. Para as 2 maiores instâncias, o GAP médio obtido é menor que 5%, e o tempo de execução está dentro do que é aceitável para o caso prático (onde se deseja um tempo máximo de execução de 300 segundos).

6.2.2 Problema real

Após a análise com os problemas de teste, que permitiu validar o correto funcionamento e o bom desempenho do ACI, o mesmo foi testado para o problema real.

Tendo em conta que para este problema não foi possível executar o modelo em CPLEX, face às dimensões do mesmo, tornou-se importante encontrar um método que permitisse avaliar a qualidade das soluções obtidas pelo ACI.

Neste sentido, foi desenvolvido um programa de simulação que efetua o sequenciamento e balanceamento das operações de acordo com a heurística implementada atualmente na fábrica. Desta forma, é possível obter o valor do *makespan* do problema real para esta heurística e comparar com a solução obtida pelo ACI, permitindo retirar conclusões sobre o desempenho da mesma e a viabilidade da sua utilização em ambiente fabril.

Além disso, obtiveram-se também uma série de indicadores-chave de desempenho para além do *makespan*, para obter uma visão mais rica das consequências da aplicação de cada um dos métodos de balanceamento e sequenciamento da produção na fábrica.

Os resultados obtidos, bem como a sua comparação e discussão, encontram-se apresentados de seguida.

6.2.2.1 Makespan

Para avaliar o *makespan*, o ACI foi executado 5 vezes e registou-se o valor da melhor solução ao longo do tempo para cada uma das corridas do ACI.

Na figura 6.1 encontra-se representada a evolução do *makespan* para cada uma das corridas, bem como o valor obtido pela heurística. Ao contrário do ACI, a heurística cria apenas uma solução e não a melhora ao longo do tempo, daí que o seu valor ao longo do tempo seja constante.

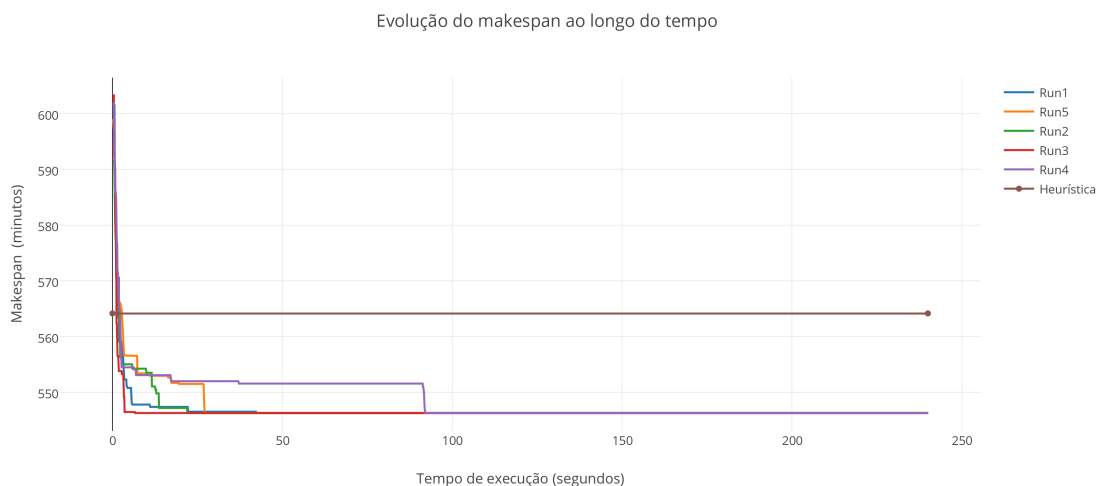


Figura 6.1: Evolução do *makespan* ao longo do tempo

Partindo deste gráfico, e analisando as soluções obtidas, podem-se tecer algumas considerações relevantes:

- O valor obtido pela heurística para o *makespan* é de 564,16 minutos.
- Embora todas as corridas do ACI tenham chegado a soluções finais diferentes, estas têm todas o mesmo valor de *makespan*: 546,26 minutos. A razão pela qual as soluções têm todas o mesmo valor prende-se com o facto de todas terem um caminho crítico constituído pelo mesmo conjunto de operações, algo que é discutido mais profundamente na secção 6.2.2.4.
- O ACI foi assim capaz de gerar soluções que reduzem em, aproximadamente, 3,173% o *makespan* obtido pela heurística.
- O ACI converge rapidamente para uma solução. A corrida onde a solução demorou mais tempo a chegar ao seu valor final foi a corrida 4, onde demorou 92,067 segundos. Esta rápida convergência é fundamental, face à necessidade de obter uma solução num intervalo de tempo menor que 5 minutos para o caso real.

6.2.2.2 Stock no armazém

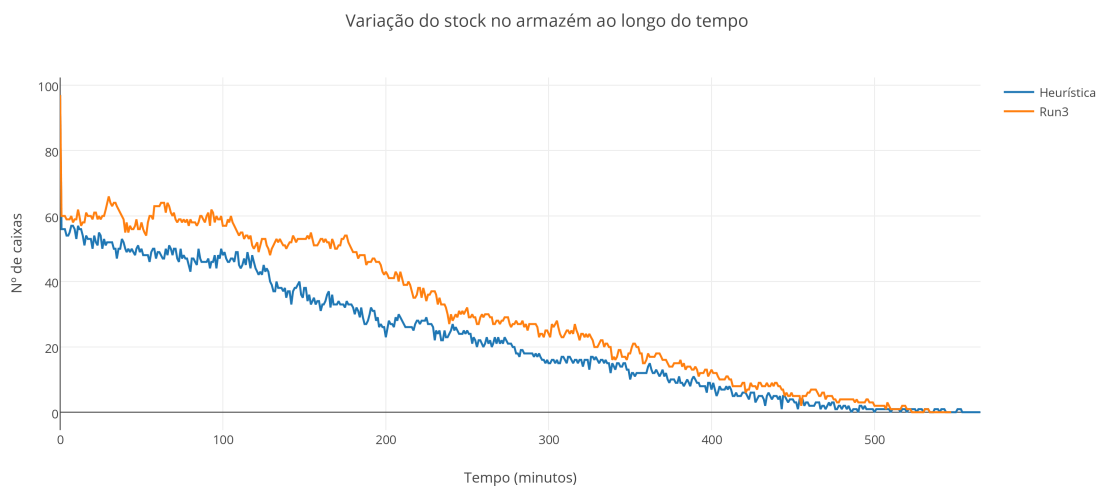
Para cada uma das soluções geradas pelas 5 corridas do ACI, bem como para a heurística, foi obtido o nível de *stock* no armazém. Admitiu-se que todas as caixas se encontravam inicialmente no armazém e que, quando uma caixa estivesse completamente trabalhada, esta era retirada da linha. Os níveis de *stock* médios obtidos estão representados na tabela 6.4.

Tabela 6.4: Níveis médios de *stock* para as diferentes soluções geradas

Método de sequenciamento	Nível de <i>stock</i> médio
ACI (Corrida 1)	31,12
ACI (Corrida 2)	31,13
ACI (Corrida 3)	30,81
ACI (Corrida 4)	30,76
ACI (Corrida 5)	30,55
Heurística	22,51

A heurística apresenta níveis de *stock* substancialmente mais baixos que o ACI, o que poderá ser importante caso o armazém tenha baixa capacidade de armazenamento. Tal deve-se à forma como a heurística efetua o despacho das caixas: sempre que há um posto livre disponível para receber alguma caixa que esteja no armazém, esta é imediatamente enviada para lá, o que reduz os níveis de *stock* no armazém, mas pode levar a um maior *makespan*.

A título de exemplo, e para melhor ilustrar a variação do *stock* ao longo do tempo, foi criada a figura 6.2, que representa os níveis de *stock* ao longo do tempo para a solução dada pela heurística e para a solução dada por uma das corridas do ACI.

Figura 6.2: Evolução do *stock* no armazém ao longo do tempo

6.2.2.3 Taxa de ocupação dos postos

Tendo em conta que o ACI devolve soluções com menor *makespan* que a heurística, a taxa de ocupação média dos postos de trabalho é maior para as soluções do ACI, como se pode observar pela tabela 6.5.

Além disso, foi também analisada a taxa de ocupação média por posto, como se pode ver na figura 6.3.

Tabela 6.5: Taxa de ocupação média dos postos de trabalho para cada método de sequenciamento

Método de sequenciamento	Taxa de ocupação média dos postos
ACI	31,75%
Heurística	30,77%

Como se pode verificar, existe um grande desequilíbrio entre os diversos postos de trabalho. Tendo em conta que o ACI tenta distribuir as operações a serem feitas pelos diversos postos de trabalho de forma equilibrada, pois dessa forma atingem-se valores mais baixos para o *makespan*, a razão deste desequilíbrio é sobretudo o balanceamento prévio da linha. Por um lado, existem postos de trabalho abertos que não são capazes de efetuar nenhuma operação das que têm de ser realizadas, o que leva a taxas de ocupação nulas. Por outro lado, existem postos de trabalhos abertos que são os únicos capazes de executar um dado conjunto de operações, o que leva a uma maior taxa de ocupação desses mesmos postos.

Assim, para melhor equilibrar a taxa de ocupação dos postos, poder-se-ia treinar os postos de trabalho menos ocupados para serem capazes de fazer as operações efetuadas pelos postos mais ocupados.

Este aumento da polivalência dos postos seria benéfico não só para equilibrar as taxas de ocupação dos postos, mas também para diminuir o *makespan* obtido, na medida em que aumenta o espaço de soluções admissíveis para o sequenciamento e balanceamento da produção.

6.2.2.4 Caminho crítico

O caminho crítico pode ser definido como sendo o conjunto de operações que se se atrasarem causarão necessariamente um aumento do *makespan*. De igual modo se pode afirmar que as operações pertencentes ao caminho crítico são as que impedem o *makespan* de ser menor.

A título de exemplo, e para melhor explicar a importância do conhecimento do caminho crítico, admita-se que a fábrica tinha a hipótese de reduzir o tempo de uma qualquer operação de modo a reduzir o *makespan*. Caso fosse escolhida uma operação que não pertence ao caminho crítico, o *makespan* manter-se-ia inalterado. Desta forma, saber o caminho crítico permitirá à fábrica obter mais informação sobre de como modo poderá reduzir o seu *makespan*.

Assim, foi feito um estudo do caminho crítico para 5 corridas do ACI. Para cada uma destas corridas, gerou-se o caminho crítico das mesmas e, além disso, fez-se o diagrama de Gantt para cada uma delas, colocando a vermelho as operações do caminho crítico. Estes diagramas podem ser consultados no anexo A.

Analisando quer o diagrama de Gantt, quer os dados do caminho crítico para cada uma das corridas do ACI, foi possível tirar as seguintes conclusões:

- Para todos os sequenciamentos gerados pelo ACI, o caminho crítico é constituído por 58 operações feitas no 11º posto seguidas de um conjunto de operações da 77ª caixa.
- Todas as 58 operações feitas no 11º posto apenas podem ser feitas neste posto.



Figura 6.3: Taxa de ocupação média por posto

- Tendo em conta esta presença dominante do 11º posto nos caminhos críticos das diferentes soluções, pode-se concluir que, caso houvesse mais postos aptos a fazer as operações que apenas este posto consegue realizar, poder-se-ia distribuir melhor estas operações, diminuindo o *makespan*.

Com uma observação mais atenta dos diagramas de Gantt, é possível também definir um limite inferior (ou *lower bound*) para o *makespan*.

Defina-se $T_{makespan}$ como o *makespan*, C_{11} como o conjunto de caixas que executam operações no 11º posto e T_{trans} o tempo de transporte de uma caixa do armazém para o posto. Além disso, defina-se TO_{11} como o tempo total das operações feitas no 11º posto e T_i como o tempo total gasto em transportes e operações pela caixa i após ser trabalhada no 11º posto. Então, pode-se afirmar que:

$$T_{makespan} \geq T_{trans} + TO_{11} + \min_{\forall i \in C_{11}} T_i \quad (6.1)$$

Assim, foram calculados os valores presentes na equação 6.1 e encontrou-se um limite inferior para o *makespan*: 489,11 minutos. Desta forma, a solução obtida pelo ACI está, no máximo, a aproximadamente 11,7% do ótimo.

6.2.2.5 Possível criação de um novo posto

Tendo concluído que todas as operações do 11º posto estão no caminho crítico e tendo em conta que estas operações apenas podem ser feitas por este posto, pode-se afirmar que o 11º posto constitui um *bottleneck* do sistema produtivo.

Deste modo, a criação de um novo posto, igual ao 11º, poderia permitir a redução do *makespan*, na medida em que distribuiria as operações a realizar pelo 11º posto e pelo novo posto criado.

Para testar essa hipótese, alteraram-se os dados do problema, criando um novo posto, igual ao 11º, e verificando qual o valor do *makespan* para esta nova configuração obtido pelo ACI.

Para esta nova configuração, o ACI foi capaz de obter um *makespan* de 482,68 minutos, ou seja, houve uma redução de aproximadamente 11,64% do valor obtido para a configuração em vigor na fábrica (546,26 minutos).

A melhoria proporcionada por esta nova configuração é também visível no diagrama de Gantt obtido, representado no anexo [B](#).

Capítulo 7

Conclusões e trabalho futuro

Neste capítulo, são descritas as conclusões referentes ao trabalho desenvolvido ao longo do projeto que deu origem a esta dissertação. Para além de se ilustrar as contribuições dadas por este trabalho e avaliar se os objetivos inicialmente propostos foram satisfeitos, são também dadas algumas sugestões para trabalho futuro.

7.1 Conclusões

O principal objetivo deste projeto de dissertação prendia-se com a criação de um novo método de sequenciamento para um problema de produção complicado de uma fábrica real de calçado. Por fim, o resultado dessa abordagem seria comparado com os resultados obtidos pelo método em vigor na fábrica.

Começou-se por criar um modelo de otimização matemática para o problema real e programou-se o mesmo em CPLEX. A análise dos resultados obtidos permitiu validar que o modelo correspondia de facto ao caso em estudo como descrito, sendo ainda capaz da resolução exata do problema para pequenas instâncias do mesmo.

Para atacar maiores instâncias do problema, aplicou-se uma meta-heurística, o ACI. Para além de validar o correto funcionamento do mesmo, os resultados indicam que o ACI é capaz de obter boas soluções num curto espaço de tempo.

Por fim, de modo a conseguir alcançar por completo os objetivos traçados para este projeto, foi criado de raiz um simulador para obter os resultados da aplicação do método de sequenciamento em vigor na fábrica ao problema em estudo.

A análise dos resultados obtidos levou à conclusão de que o ACI desenvolvido gera uma solução de sequenciamento 3,173% melhor que a da heurística usada atualmente na fábrica, no que toca ao *makespan* da produção.

É também importante referir que a análise dos resultados possibilitou ainda tirar algumas conclusões em relação ao atual balanceamento da linha e a possíveis melhorias para o mesmo. O balanceamento da linha poderia ser melhorado, visto que há um grande desequilíbrio na taxa de

ocupação dos postos, existindo mesmo postos que não são capazes de realizar nenhuma das operações necessárias para as caixas na fábrica. Além disso, a análise do caminho crítico permitiu concluir que todas as operações do 11º posto estão no caminho crítico. Desta forma, e tendo em conta que as operações do 11º posto apenas podem ser feitas por ele, pode-se concluir que ele constitui um *bottleneck* do sistema produtivo e que a criação de um novo posto igual ao 11º levaria à redução do *makespan*.

Assim, e em jeito de conclusão, pode-se afirmar que os objetivos propostos para este projeto de dissertação foram cumpridos, tendo sido ainda possível ir para lá dos mesmos e localizar possíveis melhorias a implementar.

7.2 Trabalho futuro

Para esta dissertação, considerou-se que o tempo de transporte de uma caixa do armazém para um posto era constante. Porém, na situação real, o tempo de transporte depende da localização do posto na fábrica. Como trabalho futuro, seria interessante utilizar os valores reais dos tempos de transporte para cada posto, fazendo as devidas adaptações quer no modelo matemático quer no ACI.

Além disso, o tempo de transporte de uma caixa do armazém para um posto depende também da posição no armazém da caixa. Assim, a integração do problema já definido nesta dissertação com o problema da decisão do local no armazém onde as diversas caixas devem ser mantidas entre operações poderia constituir uma melhoria ao projeto.

Por fim, em relação ao ACI desenvolvido, avaliar o seu desempenho para outros conjuntos de ordens de fabrico e investigar o efeito da mudança dos seus parâmetros poderá contribuir para melhor comprovar a adequabilidade do mesmo ao problema.

Anexo A

Diagramas de Gantt

Neste anexo encontram-se os diagramas de Gantt para 5 diferentes soluções geradas pelo ACI para o caso em estudo. As operações pertencentes ao caminho crítico estão indicadas a vermelho.

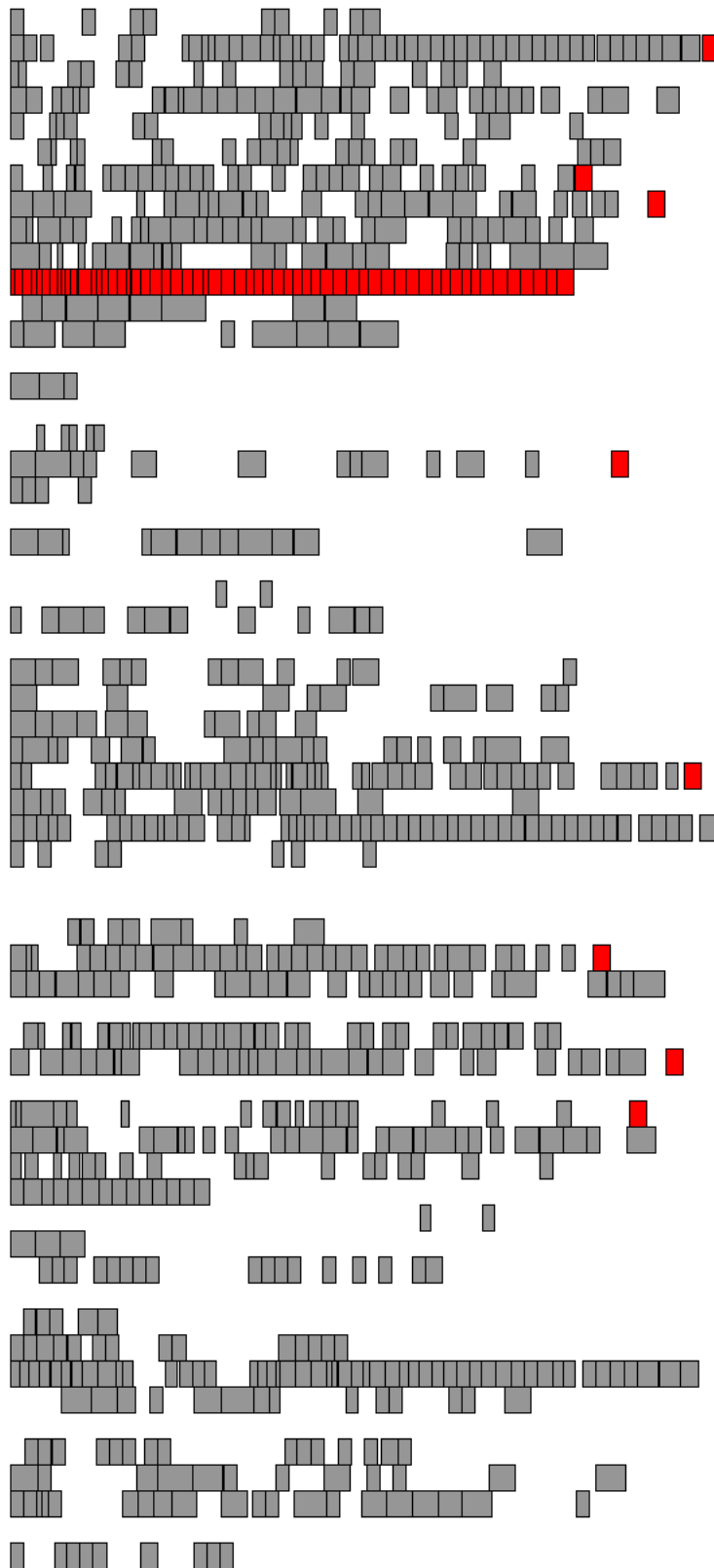


Figura A.1: Diagrama de Gantt e respetivo caminho crítico para a corrida 1 do ACI

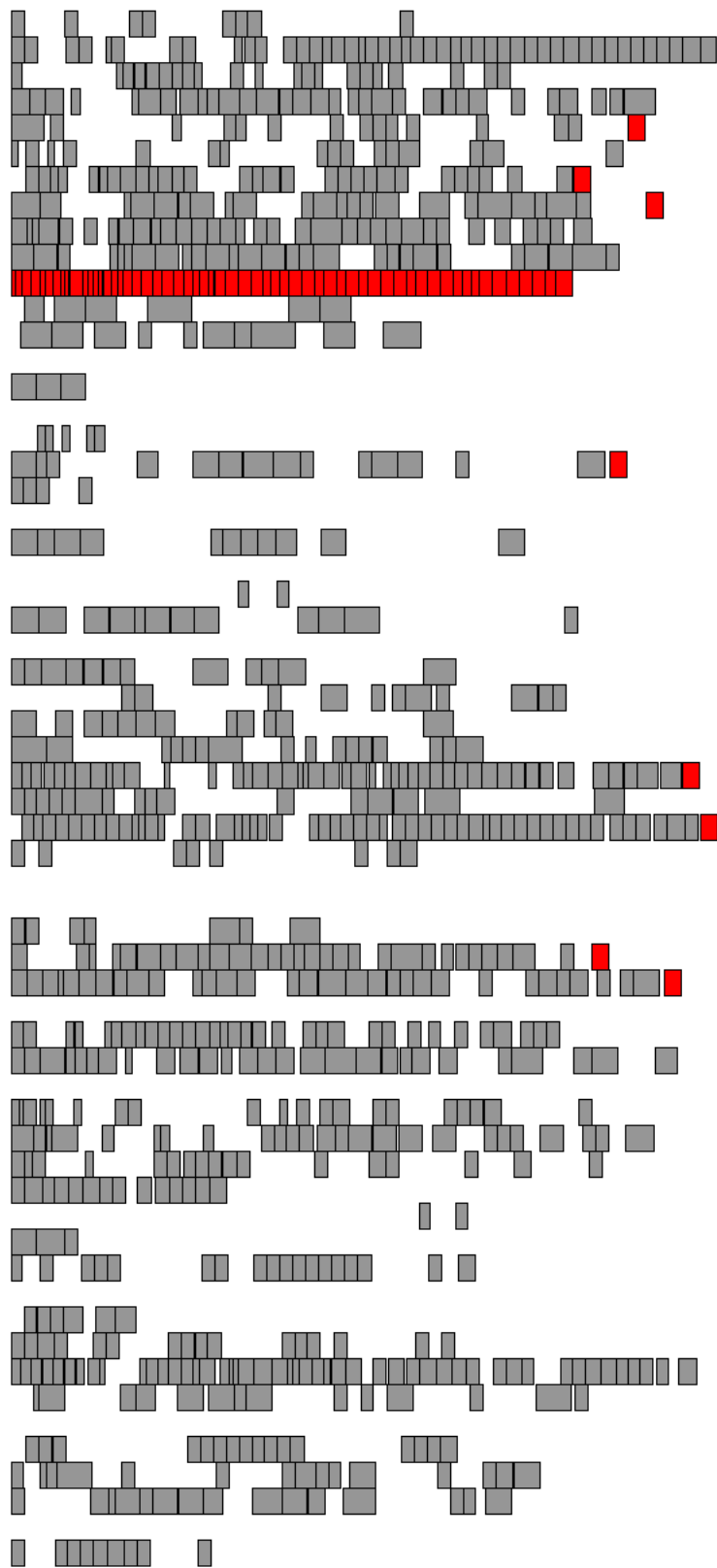


Figura A.2: Diagrama de Gantt e respetivo caminho crítico para a corrida 2 do ACI

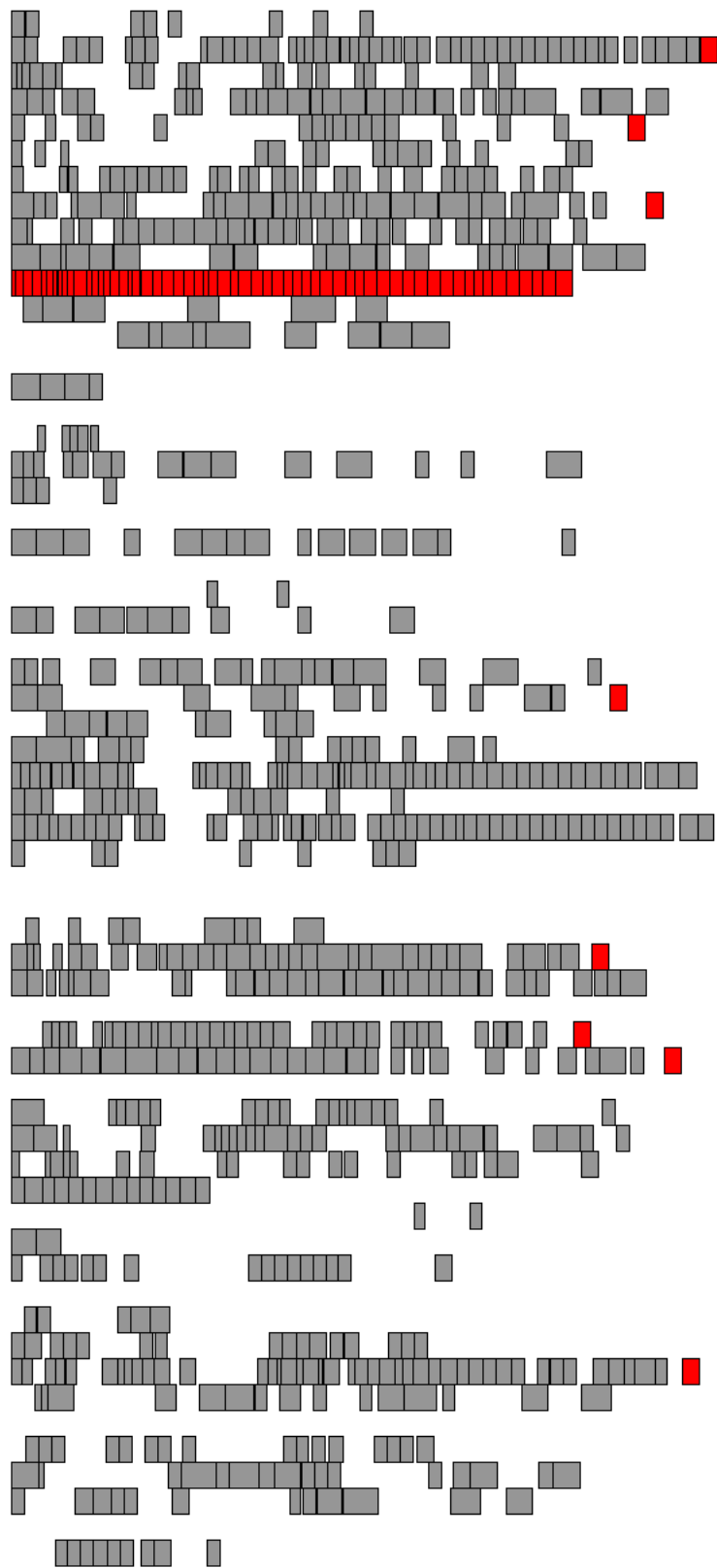


Figura A.4: Diagrama de Gantt e respetivo caminho crítico para a corrida 4 do ACI

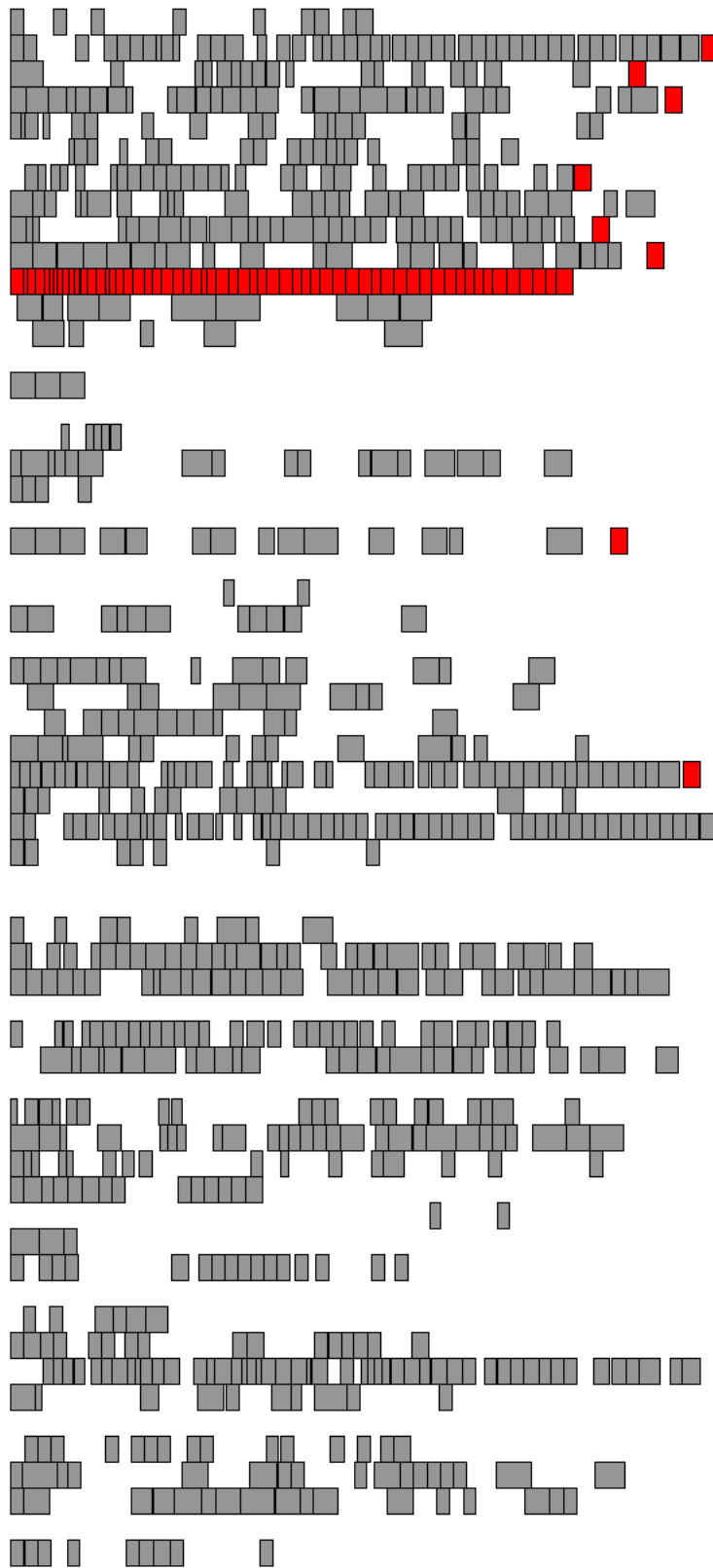


Figura A.5: Diagrama de Gantt e respetivo caminho crítico para a corrida 5 do ACI

Anexo B

Diagrama de Gantt para nova configuração

Neste anexo encontra-se o diagrama de Gantt para uma solução gerada pelo ACI para uma nova configuração da linha, em que se incluiu mais um posto, com as mesmas competências que o 11º. As operações pertencentes ao caminho crítico estão indicadas a vermelho.

Referências

- [1] Armin Scholl and Christian Becker. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168(3):666 – 693, 2006.
- [2] Ana Maria Moreira Rodrigues. *Sectores e Rotas na Recolha de Resíduos Sólidos Urbanos*. PhD thesis, Universidade do Porto, 2014.
- [3] APICCAPS. *Footure 2020: Plano Estratégico - Cluster do Calçado*. APICCAPS (Associação Portuguesa dos Industriais de Calçado, Componentes e Artigos de Pele e seus Sucedâneos, Julho 2013. ISBN: 978-989-96477-4-9.
- [4] APICCAPS. Evolução da indústria portuguesa do calçado, 2015. Disponível em http://www.apiccaps.pt/c/document_library/get_file?uuid=4dcf096f-fdbf-4bdc-95d7-f8a31cccd7e0&groupId=10136, último acesso em 2015/11/08.
- [5] INESC-TEC. INESC-TEC assina protocolo com maior fabricante português de calçado, Novembro 2014. Disponível em <https://www.inesctec.pt/cese/noticias-eventos/noticias/inesc-tec-assina-protocolo-com-maior-fabricante-portugues-de-calcado/>, último acesso em 2015/11/15.
- [6] Relatório Técnico Final do Projeto nº 21600 High Speed Shoe Factory, Julho 2014.
- [7] INESC-TEC. UESP desenvolve sistema para PMEs na Indústria do Calçado, 2007. Disponível em <http://bipz.inescporto.pt/index.php?id=200&no=68&lng=pt>, último acesso em 2015/11/15.
- [8] INESC-TEC. INESC Porto desenvolve tecnologia para revitalizar indústria do calçado, Maio 2009. Disponível em <https://www.inesctec.pt/cese/noticias-eventos/nos-na-imprensa/inesc-porto-desenvolve-tecnologia-para-revitalizar-industria-do-calcado/>, último acesso em 2015/11/15.
- [9] INESC-TEC. INESC-TEC patenteia tecnologia que revoluciona indústria do calçado, Abril 2012. Disponível em <https://www.inesctec.pt/cese/noticias-eventos/nos-na-imprensa/inesc-tec-patenteia-tecnologia-que-revoluciona-industria-do-calcado/>, último acesso em 2015/11/15.

- [//www.inesctec.pt/cese/noticias-eventos/noticias/inesc-tec-patenteia-tecnologia-que-revoluciona-industria-do-calcado/](http://www.inesctec.pt/cese/noticias-eventos/noticias/inesc-tec-patenteia-tecnologia-que-revoluciona-industria-do-calcado/), último acesso em 2015/11/15.
- [10] Ana C Barros, Rui Rebelo, Pedro França, João Sousa, José Rios, Ana Gomes, Paulo Monteiro, and Guirish Vaguela. Rfid implementation in the footwear supply chain: From production line to retail store and back. In *Advances in Sustainable and Competitive Manufacturing Systems*, pages 1035–1047. Springer, 2013.
- [11] Maria Teresa Costa and José Soeiro Ferreira. A simulation analysis of sequencing rules in a flexible flowline. *European Journal of Operational Research*, 119(2):440–450, 1999.
- [12] Juan A. Marques and José Soeiro Ferreira. A line balancing simulation system for the footwear industry. *CISTI 2006 - 1ª Conf. Ibérica de Sistemas e Tecnologias de Informação*, 2:3–18, 2006.
- [13] José Soeiro Ferreira and Rui Diogo Rebelo. A investigação operacional na indústria do calçado. *Antunes, C. (et al) (eds), A Investigação Operacional em Portugal: novos desafios, novas ideias - Homenagem ao Professor Luís Valadares Tavares*, pages 281–298, 2016.
- [14] Robert W. Doom. *Michigan Yesterday Today*. Voyageur Press, 2009. ISBN: 978-0760333853.
- [15] Nils Boysen, Malte Fließner, and Armin Scholl. Assembly line balancing: Which model to use when?, 2006.
- [16] Nuchsa Kriengkarakot and Nalin Pianthong. The assembly line balancing problem: Review articles. *KKU Engineering Journal*, 34(2):133 – 140, 2007.
- [17] Christian Becker and Armin Scholl. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168(3):694 – 715, 2006.
- [18] James C Chen, Hutama Parwananta, Tran Dinh Duy Thao, Kung-Jeng Wang, Maria Anityasari, and Chun-Chieh Chen. Branch and bound method for solving line balancing problem in footwear sewing line.
- [19] Pedro M Vilarinho and Ana Sofia Simaria. A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations. *International Journal of Production Research*, 40(6):1405–1420, 2002.
- [20] Ana Sofia Simaria and Pedro M Vilarinho. A genetic algorithm based approach to the mixed-model assembly line balancing problem of type ii. *Computers & Industrial Engineering*, 47(4):391–407, 2004.

- [21] M Bagher, M Zandieh, and H Farsijani. Balancing of stochastic u-type assembly lines: an imperialist competitive algorithm. *The International Journal of Advanced Manufacturing Technology*, 54(1-4):271–285, 2011.
- [22] Adil Baykasoğlu and Lale Özbakır. Stochastic u-line balancing using genetic algorithms. *The International Journal of Advanced Manufacturing Technology*, 32(1-2):139–147, 2007.
- [23] Onur Serkan Akgündüz and Semra Tunalı. A review of the current applications of genetic algorithms in mixed-model assembly line sequencing. *International Journal of Production Research*, 49(15):4483–4503, 2011.
- [24] Nils Boysen, Malte Fließner, and Armin Scholl. Sequencing mixed-model assembly lines: Survey, classification and model critique. *Jena Research Papers in Business and Economics - Working and Discussion Papers 02/2007*, Friedrich-Schiller-University Jena, School of Economics and Business Administration, March 2007.
- [25] Onur Serkan Akgündüz and Semra Tunalı. An adaptive genetic algorithm approach for the mixed-model assembly line sequencing problem. *International Journal of Production Research*, 48(17):5157–5179, 2010.
- [26] Alireza Rahimi-Vahed and Ali Hossein Mirzaei. A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem. *Computers & Industrial Engineering*, 53(4):642–666, 2007.
- [27] Babak Javadi, Alireza Rahimi-Vahed, Masoud Rabbani, and Mostafa Dangchi. Solving a multi-objective mixed-model assembly line sequencing problem by a fuzzy goal programming approach. *The International Journal of Advanced Manufacturing Technology*, 39(9-10):975–982, 2008.
- [28] JF Bard, A Shtub, and SB Joshi. Sequencing mixed-model assembly lines to level parts usage and minimize line length. *The International Journal of Production Research*, 32(10):2431–2454, 1994.
- [29] Patrick R McMullen and Gregory V Frazier. A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line. *Iie Transactions*, 32(8):679–686, 2000.
- [30] Yeo Keun Kim, Jae Yun Kim, and Yeongho Kim. A coevolutionary algorithm for balancing and sequencing in mixed model assembly lines. *Applied Intelligence*, 13(3):247–258, 2000.
- [31] Uğur Özcan, Talip Kellegöz, and Bilal Toklu. A genetic algorithm for the stochastic mixed-model u-line balancing and sequencing problem. *International Journal of Production Research*, 49(6):1605–1626, 2011.

- [32] Kunlei Lian, Chaoyong Zhang, Liang Gao, and Xinyu Shao. A modified colonial competitive algorithm for the mixed-model u-line balancing and sequencing problem. *International Journal of Production Research*, 50(18):5117–5131, 2012.
- [33] Jakob Puchinger and Günther R Raidl. *Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification*. Springer, 2005.