



Universidade do Minho
Escola de Engenharia

Natália Maria Lemos Costa

Comparação de Escalonamento num Ambiente
de Linha de Produção Híbrida com um
Ambiente de Linhas de Produção Paralelas –
Um Caso Industrial



Universidade do Minho
Escola de Engenharia

Natália Maria Lemos Costa

Comparação de Escalonamento num Ambiente
de Linha de Produção Híbrida com um
Ambiente de Linhas de Produção Paralelas –
Um Caso Industrial

Dissertação de Mestrado
Mestrado em Engenharia e Gestão Industrial

Trabalho efetuado sob a orientação da
Professora Doutora Maria Leonilde Rocha Varela

DECLARAÇÃO

Nome: Natália Maria Lemos Costa

Endereço eletrónico: natalialemos690@gmail.com Telefone: 252962093/ 919158255

Número do Bilhete de Identidade: 12455186

Título da dissertação:

Comparação de escalonamento num ambiente de Linha de Produção Híbrida com um ambiente de Linhas de Produção Paralelas – Um Caso Industrial

Orientador(es):

Professora Doutora Maria Leonilde Rocha Varela

Ano de conclusão: 2014

Designação do Mestrado: Mestrado em Engenharia e Gestão Industrial

Nos exemplares das teses de doutoramento ou dissertações de mestrado ou de outros trabalhos entregues para prestação de provas públicas nas universidades ou outros estabelecimentos de ensino, e dos quais é obrigatoriamente enviado um exemplar para depósito legal na Biblioteca Nacional e, pelo menos outro para a biblioteca da universidade respetiva, deve constar uma das seguintes declarações:

1. É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;
2. É AUTORIZADA A REPRODUÇÃO PARCIAL DESTA DISSERTAÇÃO (indicar, caso tal seja necessário, nº máximo de páginas, ilustrações, gráficos, etc.), APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;
3. DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA DISSERTAÇÃO

Universidade do Minho, ___/___/_____

Assinatura:

AGRADECIMENTOS

Aproveito a oportunidade para deixar os meus sinceros agradecimentos à Professora Maria Leonilde Varela, pela oportunidade proporcionada para desenvolver este trabalho numa área que tenho muito interesse, bem como por toda a disponibilidade, conselhos, motivação e apoio que me foram essenciais para a elaboração desta tese.

Deixo também um especial e sincero agradecimento aos meus colegas André Santos, Joana Dantas e Ana Rita Carvalho pelo auxílio que me facultaram.

Uma palavra para os meus colegas e amigos.

Para toda a minha família.

RESUMO

O escalonamento é uma das decisões mais importantes nos sistemas de controlo de produção. Nesta dissertação foram exploradas diferentes abordagens para o escalonamento de trabalhos com duas fases, nomeadamente em duas configurações diferentes de sistemas de produção orientados ao produto: linha de fabrico híbrida (*hybrid flow shop* - HFS) e linhas de fabrico paralelas (*parallel flow shop* - PFS). Foi analisado um caso prático da indústria de componentes automóveis e comparada a resolução do problema no cenário de HFS com o cenário de PFS em termos de instante máximo de conclusão dos trabalhos no sistema (*makespan* - C_{max}), assim como outras medidas de desempenho. Os resultados permitiram concluir que o cenário HFS, em geral, permite obter resultados de desempenho consideravelmente melhores que o cenário PFS, quando não existem restrições associadas. No caso de existirem tempos de transporte entre máquinas iguais a 2 minutos, o cenário PFS apresenta um melhor desempenho.

PALAVRAS-CHAVE

Escalonamento; heurísticas; linha de fabrico híbrida; linhas de fabrico paralelas; *makespan*.

ABSTRACT

Scheduling is one of the most important decisions in production control systems. This study explores different approaches for scheduling two stage jobs in two configurations of product oriented manufacturing systems, namely an hybrid flow shop (HFS) and the parallel flow shops (PFS). An industrial case is analyzed from the automotive components industry. The HFS problem resolution is compared with the PFS one in terms of makespan (C_{max}) and other performance measures. The results allowed the conclusion that the HFS generally performs considerably better than the PFS when there are no constraints. However, when there are transportation times between machines equal to 2 minutes, the PFS presents better results than the HFS.

KEYWORDS

Scheduling; heuristics; hybrid flow shop, parallel flow shop; makespan.

ÍNDICE

Agradecimentos.....	iii
Resumo.....	v
Abstract.....	vii
Índice de Figuras	xiii
Índice de Tabelas	xv
Lista de Abreviaturas, Siglas e Acrónimos.....	xvii
1. Introdução	1
1.1 Motivação.....	4
1.2 Objetivos	5
1.3 Metodologia	5
1.4 Estrutura do relatório.....	6
2. Estado da arte.....	9
2.1 Linhas de fabrico híbridas.....	9
2.2 Linhas de fabrico paralelas	10
3. Problemas de escalonamento da produção.....	13
3.1 Sistemas de Produção	16
3.2 Sistemas Orientados ao produto ou ao processo	18
3.2.1 Sistemas de Produção Orientados ao Produto	18
3.2.2 Sistemas de Produção Orientados ao Processo ou à Função	19
3.3 Características dos Trabalhos e dos Recursos	20
3.4 Medidas de Desempenho ou Critérios de Otimização	22
3.5 Resumo	23
4. Métodos e Sistemas de Escalonamento de Produção.....	25
4.1 Métodos de Escalonamento	25
4.1.1 Regras de Prioridade.....	26
4.1.2 Métodos de Pesquisa Local e Meta-Heurísticas.....	27

4.1.3	Métodos de gargalo de estrangulamento ou <i>bottleneck methods</i>	28
4.2	Sistemas de Escalonamento	29
4.2.1	Sistemas tradicionais	30
4.2.2	Sistemas de web	33
4.3	Resumo	36
5.	Caso Industrial.....	39
5.1	Linhas de fabrico paralelas	40
5.1.1	Procedimento de enumeração total	41
5.1.2	Heurística de Kedia.....	42
5.1.3	Algoritmo de Johnson	43
5.1.4	Heurística <i>General Shifting Bottleneck Routine</i> (PFS)	44
5.2	Linha de fabrico hibrida	44
5.2.1	Regras de Prioridade.....	45
5.2.2	Heurística <i>General Shifting Bottleneck Routine</i> (HFS).....	46
5.3	Resumo	47
6.	Análise dos Resultados.....	49
6.1	Problema PFS	49
6.1.1	Procedimento de enumeração total combinado com Regra de Johnson	49
6.1.2	Heurística de Kedia combinada com Regra de Johnson	50
6.1.3	Heurística <i>General Shifting Bottleneck Routine</i>	51
6.2	Problema HFS.....	51
6.2.1	Heurística <i>General Shifting Bottleneck Routine</i>	52
6.3	Resultados Globais e Discussão	53
6.4	Generalização.....	53
6.4.1	Tempo de transporte de 1 minuto	54
6.4.2	Tempo de transporte de 2 minutos	56

6.5	Resumo	57
7.	Conclusão.....	59
	Referências Bibliográficas	61
	Anexo I – Resultados	lxv

ÍNDICE DE FIGURAS

Figura 1. Enquadramento geral do Planeamento e Controlo da Produção	13
Figura 2. Medidas de desempenho típicas.....	15
Figura 3. Sistema LEKIN.....	32
Figura 4. Configuração PFS.	40
Figura 5. Configuração HFS com duas fases.	45
Figura 6. Diagrama de Gantt – Enumeração total combinado com Regra de Johnson	50
Figura 7. Diagrama de Gantt – Heurística de Kedia combinada com Regra de Johnson.....	50
Figura 8. Diagrama de Gantt – <i>General Shifting Bottleneck routine</i> (PFS)	51
Figura 9. Diagrama de Gantt – <i>General Shifting Bottleneck routine</i> (HFS).....	52
Figura 10. Comparação do <i>Makespan</i> dos diferentes cenários.....	53
Figura 11. Diferenças de <i>makespan</i> entre os dois cenários - Tempos de transporte entre máquinas de 1 minuto.....	55
Figura 12. Diferenças de <i>makespan</i> entre os dois cenários - Tempos de transporte entre máquinas de 2 minutos.....	57
Figura 13. Comparação da taxa de utilização do sistema dos diferentes cenários.....	lxv
Figura 14. Comparação das médias do <i>Makespan</i> dos diferentes cenários	lxvi

ÍNDICE DE TABELAS

Tabela 1. Tempos de processamento.....	39
Tabela 2. Algoritmo da Heurística de Kedia	42
Tabela 3. Algoritmo de Johnson	43
Tabela 4. Algoritmo da Heurística <i>Shifting Bottleneck</i>	46
Tabela 5. Teste <i>T de Student</i> para tempos de transporte entre máquinas de 1 minuto.....	55
Tabela 6. Teste <i>T de Student</i> para tempos de transporte entre máquinas de 2 minutos	56
Tabela 7. Resultados obtidos com os métodos implementados.....	lxv

LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

EDD – Earliest due date
FCFS – First Come First Served
FOFO – First Off First On
HFS - Hybrid Flow Shop
HFSP – Hybrid Flow Shop Problem
LPT – Longest Processing Time
LWKR – Least Work Remaining
MWKR – Most Work Remaining
OTED - One-Touch Exchange of Die
PFS – Parallel Flow Shop
PFSP – Parallel Flow Shop Problem
PPC – Production Planning and Control
SB – Shifting Bottleneck
SBR – General Shifting Bottleneck Routine
SMED - Single-Minute Exchange of Die
SPT – Shortest Processing Time
TSPT - Truncated SPT
TWORK – Total Work
WIP – Work in Progress

1. INTRODUÇÃO

O escalonamento é uma atividade de controlo de produção importante. Em muitas situações industriais os trabalhos a escalonar precisam de ser sequenciados antes do processamento poder começar. O escalonamento requer total conhecimento das tarefas do trabalho, limitações dos recursos, e restrições de sequência das operações. Devido à diversidade dos problemas de escalonamento, a definição do problema e sua caracterização podem beneficiar com a especificação da nomenclatura e ontologia do problema de escalonamento (Varela & do Carmo Silva, 2008).

Podemos abordar os problemas de escalonamento reconhecendo duas classes principais de ambientes de produção, nomeadamente, Sistemas de Produção Orientados ao Produto (SPOP) e Sistemas de Produção Orientados à Função (SPOF) (Silvio Carmo Silva & Alves, 2006). Os SPOP são sistemas de produção projetados para a produção de um único tipo de produto ou família de produtos similares. Os Sistemas de Produção Orientados à Função (SPOF) são sistemas de produção caracterizados pela existência de seções funcionais, tendo cada uma delas afeta uma função da produção. Neste caso a configuração do sistema de produção é do tipo linha de fabrico (*flow shop*). Comparados com os SPOF, os SPOP tendem a ter melhor produtividade que os SPOF. Tal deve-se à sua elevada dedicação ao produto, conseguindo assim também menores tempos de entrega do produto. Isto acontece porque os tempos de processamento nos SPOP tendem a ser menores do que nos SPOF.

Numa configuração linha de fabrico com múltiplos processadores, a linha tem duas ou mais máquinas equivalentes, em pelo menos uma das fases de processamento (Garey, Graham, & Johnson, 1976; Johnson, 1954). Se existir um processador ou máquina por fase de processamento, numa linha de fabrico com duas fases de processamento, o problema de encontrar a sequência para um número de trabalhos de forma a minimizar o instante máximo de conclusão dos trabalhos no sistema (*makespan - C_{max}*) tem a solução ótima em tempo polinomial (Johnson, 1954). Contudo, se o número de máquinas for superior a duas, para duas ou mais

fases de processamento, então o problema torna-se mais complexo e fortemente *NP-hard* (Garey et al., 1976). É sabido que a maioria dos problemas de linha de fabrico do mundo real são *NP-hard* (Al-Salem, 2004).

O ambiente de produção do tipo linha de fabrico híbrida (*hybrid flow shop* – HFS) pode ser considerado uma extensão da clássica linha de fabrico. (Aneke & Carrie, 1986; Rubén Ruiz & Vázquez-Rodríguez, 2010). O HFS difere de uma clássica linha de fabrico no facto de que tem duas ou mais máquinas idênticas em pelo menos uma fase de processamento. Mas, assim como nas clássicas linhas de fabrico, o fluxo dos trabalhos numa HFS é direto ou sequencial (Ribas, Leisten, & Framiñan, 2010; Rubén Ruiz & Vázquez-Rodríguez, 2010). Assim, todos os trabalhos que têm múltiplas operações são processados sem interrupções (*preemption*), seguindo o mesmo caminho ao longo do sistema de produção. Alguns autores referem-se ao HFS como *flexible flow shop* ou *flexible flow line* (Quadt & Kuhn, 2007).

O problema de escalonamento numa configuração HFS pode também ser visto como uma generalização de um problema de escalonamento em ambiente PFS com uma única fase de fabrico para cada uma das linhas individuais que integram o ambiente PFS, e tem sido provado ser um problema de escalonamento *NP-hard* (Cheng, Karuno, & Kise, 2001).

A configuração em linhas de fabrico ou de produção paralelas (*parallel flow shop* - PFS) pode ser definido simplesmente como a replicação de várias clássicas linhas de fabrico, i.e., em vez de ter apenas uma linha clássica ou uma linha de fabrico simples (*permutation flow shop*). Podemos ter várias linhas em paralelo, e todas elas integrando um único sistema de produção (Al-Salem, 2004). A configuração HFS oferece uma maior flexibilidade (Vairaktarakis & Elhafi, 2000) do que outras, mas exige geralmente uma maior ou mais complexa manipulação dos trabalhos. Como alternativa ao HFS, Vairaktarakis & Elhafi (2000) introduziram o conceito de *parallel flow lines*, e estudaram os casos de duas fases, com F_1, \dots, F_m linhas de fabrico, cada uma consistindo numa série de duas máquinas M_{1i} e M_{2i} ($i = 1, \dots, m$). Abordaram também, o que chamaram de problema em configuração do tipo *Parallel Hybrid Flow Line* com duas fases, onde m *parallel hybrid flow lines* são definidas tendo máquinas idênticas em cada fase, uma em cada linha, e partilhando o conjunto de m máquinas idênticas na segunda fase.

A maioria dos problemas de escalonamento são *NP-hard*, portanto as heurísticas são a primeira forma de resolver estes problemas em tempo útil. As regras de prioridade devem ser consideradas um tipo de heurísticas simples. Os métodos heurísticos são métodos que têm demonstrado grande aplicabilidade nos últimos anos. Isto deve-se ao facto de muitos deles serem intuitivos e fáceis de implementar (Kenneth R Baker & Baker, 1974; Ribeiro & Varela, 2003).

Procedimentos genéricos importantes como as meta-heurísticas envolvendo computação biologicamente inspirada e outros fenómenos naturais, têm sido usados para desenvolver heurísticas altamente eficazes e eficientes para problemas de escalonamento complexos (Cao & Chen, 2003; Samarghandi & ElMekkawy, 2012; Varela, Barbosa, & Putnik, 2012).

O escalonamento em configurações da tipologia das que são reportadas neste estudo e outros exigem o uso de uma grande variedade de métodos e algoritmos de escalonamento, muitos deles heurísticos. Estes métodos e algoritmos têm sido desenvolvidos e relatados por vários autores incluindo Baker (1974), Lenstra (1990), Blazewicz (2001), Brucker (2007), Conway (2012), e Pinedo (2012). Muitos destes algoritmos estão disponíveis através de sistemas de escalonamento integrados. Exemplos de tais sistemas disponíveis e gratuitos podem ser obtidos através da Internet, são os sistemas de escalonamento LEKIN® (Pinedo, 2012) e Lisa® (Andresen, Bräsel, Engelhardt, & Werner, 2010) entre outros.

Neste trabalho foi utilizado o sistema de escalonamento LEKIN® para resolver problemas de escalonamento em configuração HFS, assim como para uma das heurísticas utilizadas na resolução do problema em configuração PFS.

Neste estudo procurou-se obter boas soluções para a medida de desempenho *makespan*, num caso industrial do mundo real, modelado como HFS e também como PFS.

1.1 Motivação

A motivação para este trabalho surgiu da necessidade de aprofundamento e aplicação de várias alternativas de métodos para a resolução do problema de escalonamento explorado inicialmente por J. Costa & Varela (2012), assim como a obtenção de melhores resultados de desempenho no sistema de produção em causa.

É necessária a definição das configurações do sistema em que se fará a abordagem ao problema, assim como os métodos de escalonamento adequados para cada situação. Estes métodos além de eficazes devem ser eficientes, pois os problemas que ocorrem neste tipo de indústria necessitam de uma resposta em tempo útil.

Apesar dos muitos estudos realizados nesta área, a aplicação de métodos de escalonamento a um caso prático torna-se interessante dado a utilização de dados reais permitir analisar melhor como se comportam os sistemas na realidade. Além disso, através da variação de determinadas restrições ou parâmetros nos problemas podemos também verificar as alterações no sistema real. É nossa convicção de que é possível encontrar outros métodos que permitam obter uma solução ótima, portanto ainda melhor do que as que foram encontradas neste trabalho, porém os métodos que foram usados neste trabalho permitiram obter melhorias consideráveis relativamente aos que foram usados no estudo anterior acima referido, podendo, desta forma, os resultados obtidos serem considerados uma mais-valia importante, não só para o sistema de produção aqui equacionado como também para futuras outras implementações em sistemas de produção com configurações idênticas.

Do estudo efetuado pode constatar-se que este tipo de problema apresenta um nível de complexidade que o permite enquadrar na classe geral dos problemas do tipo *NP-hard*, muito embora a instância de problema que foi foco de análise neste trabalho, dada a sua dimensão, não o seja, como se pode constatar pela descrição do problema que se apresenta a seguir.

1.2 Objetivos

Este estudo baseia-se num caso industrial da indústria automóvel relatado por J. Costa & Varela (2012). O caso em estudo envolve um sistema de produção com duas fases de processamento que inclui duas operações distintas, nomeadamente, termoformagem e prensagem de componentes automóveis.

Foram estudadas duas configurações de sistemas de produção diferentes que podem ser consideradas extensões do sistema clássico de linha de fabrico (Garey *et al.*, 1976; Johnson, 1954). Assim foram identificados dois problemas de escalonamento da produção: um problema de linhas paralelas (*parallel flow shop problem* - PFSP) e um problema de linhas flexíveis ou linhas híbridas (*hybrid flow shop problema* - HFSP) (Aneke & Carrie, 1986; Rubén Ruiz & Vázquez-Rodríguez, 2010).

Pretende-se efetuar o escalonamento da produção, e efetuar a seleção do melhor método de escalonamento, de entre um conjunto de métodos alternativos disponíveis, tendo como critério de otimização principal o *makespan* (instante máximo de conclusão dos trabalhos no sistema). Desta forma os principais objetivos do trabalho são:

- Analisar um caso industrial da indústria de componentes automóveis;
- Explorar diferentes abordagens para o escalonamento de trabalhos de duas fases em duas configurações de sistemas de produção orientados ao produto;
- Comparar a resolução dos problemas em termos de *makespan* e outras medidas de desempenho.

1.3 Metodologia

O estudo efetuado neste trabalho foi realizado utilizando a estratégia “caso de estudo” (*case study*). Este tipo de estratégias são muito utilizadas em abordagens dedutivas, procurando responder a perguntas de investigação do tipo: “quem”, “o quê”, “onde”, “quanto”. Permitem

recolher dados quantitativos que podem ser analisados através de estatística, e sugerir razões possíveis para relações entre variáveis e para produzir modelos explicativos dessas relações (Saunders, Lewis, & Thornhill, 2009).

Inicialmente foi realizada uma revisão da literatura focada em configurações de produção do tipo linhas de produção, as suas características, os métodos de escalonamento e as medidas de desempenho. Foram ainda analisados alguns os problemas de escalonamento neste tipo de configurações e seu estado de arte. Nesta fase, foram utilizados os três tipos de referências bibliográficas, primárias, secundárias e terciárias. As fontes primárias foram a consulta de teses e dissertações, as secundárias a consulta de livros e artigos, e as terciárias sendo a forma de encontrar as outras fontes.

Seguidamente foi analisado o artigo que nos serve de base para este estudo (J. Costa & Varela, 2012), e realizados alguns *brainstormings* com ideias para melhorar as soluções obtidas para este caso industrial. A partir destes *brainstormings* foram testadas algumas configurações de produção possíveis para a resolução deste problema e implementados alguns métodos de escalonamento apropriados a cada configuração.

Após a seleção dos métodos que iriam ser utilizados e a obtenção de resultados a partir dos mesmos, foram analisados os resultados, realizada a generalização dos resultados obtidos e retiradas as conclusões deste estudo.

1.4 Estrutura do relatório

Esta dissertação está organizada em sete capítulos:

No primeiro capítulo é realizada uma introdução e enquadramento ao trabalho desenvolvido, assim como a motivação que levou à realização do mesmo. São ainda definidos os objetivos que se pretendem atingir com esta dissertação de mestrado.

No segundo capítulo é efetuada uma revisão do estado da arte do escalonamento em ambiente de linha de produção híbrida, e em ambiente de linhas de produção paralelas.

No terceiro capítulo é analisada a integração do problema de escalonamento no planeamento da produção e analisada a sua importância. Efetua-se a contextualização do problema de escalonamento com base nas características que o compõem, dando particular importância à sua classificação quanto à dedicação ao produto.

No quarto capítulo são apresentados alguns métodos de escalonamento da produção, assim como sistemas que são capazes de efetuar o escalonamento quer em plataformas ou sítios web, ou em sistemas tradicionais.

No quinto capítulo é descrito o caso de estudo, inclusive o tipo de problemas a resolver e a descrição dos métodos e estratégias de resolução dos mesmos.

O sexto capítulo apresenta os resultados obtidos e a sua análise.

Por último, no sétimo capítulo são apresentadas as conclusões do trabalho realizado e identificados possíveis trabalhos futuros.

2. ESTADO DA ARTE

2.1 Linhas de fabrico híbridas

O estudo do escalonamento em configuração HFS atraiu muitos investigadores desde o trabalho de S. M. Johnson (1954).

Os primeiros trabalhos de investigação sobre o escalonamento em configuração HFS apareceram na década de 1970, com Salvador (1973). Poucos anos mais tarde, D. S. Johnson & Garey (1979) mostraram que o problema HFS cujo critério de otimização/desempenho é o *makespan* trata-se de um problema NP-completo (ou *NP-Complete*), tendo sido comprovada essa natureza de complexidade ao longo dos anos de investigação.

A partir desse momento foram propostos um grande número de heurísticas e algoritmos de aproximação para configurações HFS diferentes. Durante a última década, a pesquisa centrou-se em problemas com restrições de tempos de preparação (*setups*) nas máquinas, de elegibilidade de máquinas, restrições de precedência nos postos de trabalho, entre outros, tendo em conta o preenchimento de lacunas entre a teoria e a prática (Ribas et al., 2010). Os estudos sobre problemas de escalonamento em linhas paralelas flexíveis ou híbridas são relativamente recentes. As principais investigações neste tipo de problemas são HFS com duas fases e pretendem otimizar a medida de desempenho *makespan*. No entanto, alguns trabalhos têm sido realizados com o objetivo de otimização de medidas de desempenho como o *lateness* e o *tardiness*. As mais recentes revisões abrangentes sobre o escalonamento em HFS datam de 1999 (Linn & Zhang, 1999; Vignier et al., 1999).

Entre as publicações recentes encontram-se as de R. Ruiz & Maroto (2005) que fizeram uma comparação de 25 métodos, que vão desde o clássico algoritmo de Johnson, regras de prioridade às mais recentes meta-heurísticas. Naderi, Ruiz, & Zandieh (2010) propuseram algoritmos avançados tendo em conta a minimização do *makespan* em problemas *hybrid flexible flow shops*. Ribas et al. (2010) fizeram uma revisão do estado da arte em problemas de escalonamento em ambiente de HFS.

2.2 Linhas de fabrico paralelas

O primeiro estudo sobre problemas de escalonamento em máquinas paralelas surgiu na década de 1950 com McNaughton (1959) e seguidamente por Hu (1961). Este tipo de problemas têm recebido um contínuo interesse por parte dos investigadores desde então, devido à sua relevância para sistemas de computadores e sistemas de produção, e assim a literatura continua a aumentar. Nos estudos realizados, diferentes restrições foram tidas em conta e avaliadas diversas medidas de desempenho. Mais recentemente foi efetuado por Cheng & Sin (1990) um estudo intensivo sobre os principais resultados da investigação teórica sobre o escalonamento neste tipo de configuração de sistema, tal como o seu estado de arte, tratando-se de um tipo de sistema de produção que não só aparece com bastante frequência nesta sua forma de base em ambientes de produção industrial, como também inseridos em sistemas de produção mais complexos, nomeadamente em forma de linha de produção.

Por outro lado, a minimização do *makespan* é um dos critérios mais estudados na literatura de escalonamento e também no escalonamento de máquinas paralelas idênticas. Por exemplo, Min & Cheng (1999) propuseram um tipo de algoritmo genético para minimizar o *makespan* no problema de escalonamento de máquinas idênticas, Zhang & Van de Velde (2012) desenvolveram um algoritmo de aproximação para linhas paralelas com duas e três fases com o objetivo de minimizar o *makespan* (C_{max}); outro exemplo é o algoritmo de aproximação eficiente e facilmente implementável para o problema de linhas paralelas, tendo como objetivo a otimização do *makespan*, que foi apresentado por Nowicki & Smutnicki (1998).

Muitos outros estudos sobre o problema PFS foram realizados por vários autores. Mais alguns exemplos disso são: 1) Um estudo multi-simulação para analisar a eficácia de um algoritmo heurístico proposto para pequenos e grandes problemas efetuado por Al-Salem (2004); 2) Zhang & Van de Velde (2012) desenvolveram um algoritmo de aproximação para PFS com duas ou três fases de processamento, tendo como objetivo minimizar o *makespan* (C_{max}) e 3). Cao & Chen (2003) desenvolveram um modelo de programação matemática para atribuição de componentes combinados e a calendarização dos trabalhos.

3. PROBLEMAS DE ESCALONAMENTO DA PRODUÇÃO

A concorrência em ambiente industrial assenta principalmente em requisitos como prazos de entrega e na satisfação das exigências do cliente. De forma a responder a estas exigências, torna-se primordial que os sistemas de produção, se adaptem à inovação e permitam a rápida produção para entrega ao cliente de produtos com qualidade, permitindo também proveitos económicos, técnicos e sociais para o produtor.

O planeamento constitui uma das mais importantes atividades da gestão de operações, fornecendo as bases para a tomada de decisões mais detalhadas do planeamento e controlo de produção (PPC – *Production Planning and Control*). O PPC é parte integrante do sistema de informação do sistema produtivo, fornecendo informação para gerir eficientemente os fluxos de materiais a utilizar, as pessoas e os equipamentos; coordenar as atividades internas com as dos fornecedores, e comunicar com os clientes sobre as necessidades do mercado. Segundo Vollmann, William, & Whybark (1992) o PPC engloba quatro níveis típicos, planeamento da produção, planeamento diretor da produção, planeamento das necessidades de materiais e controlo fabril, tal como mostra a figura 1:



Figura 1. Enquadramento geral do Planeamento e Controlo da Produção

Efetuar o escalonamento da produção é definir numa escala de tempo os instantes de início e conclusão do processamento dos trabalhos ou tarefas nos diversos recursos do sistema produtivo. O escalonamento da produção é das tarefas de mais baixo nível no quadro do PPC de um sistema produtivo pois está diretamente ligada ao espaço fabril e detém um carácter de curto prazo (Carvalho, 2000).

Frequentemente o escalonamento é visto como a ação integrada de afetação e sequenciamento, porém em muitas situações este reduz-se ao sequenciamento.

Um problema de escalonamento provém da necessidade de executar um número de tarefas com determinados recursos. Portanto, para resolver um problema de escalonamento é necessário conhecer e caracterizar detalhadamente aspetos relevantes, como os trabalhos ou tarefas, os processadores, os meios auxiliares, as configurações dos recursos disponíveis, restrições e os critérios de otimização. Neste processo torna-se fundamental a perceção da forma como estes elementos se relacionam, e restrições inerentes à ordem de execução de cada tarefa. Podemos afirmar que o escalonamento permite planear o uso eficiente dos recursos de produção existentes, com vista a atingir o objetivo de otimização definido. A natureza do objetivo de otimização pode ser distinguida em duas classes (S. Carmo Silva, 2010):

- Tem por base a função custo do programa de produção;
- Não tem em conta a função custo do programa - Neste tipo de objetivos incluem-se medidas como as apresentadas na fig.2.

Tempo de percurso médio:	$F_{med} = \bar{F} = \frac{1}{n} \sum_1^n F_i$
Atraso médio:	$A_{med} = \bar{A} = \frac{1}{n} \sum_1^n A_i$
Tempo de percurso máximo:	$F_{max} = \max_{1 \leq i \leq n} \{F_i\}$
Atraso máximo:	$A_{max} = \max_{1 \leq i \leq n} \{A_i\}$
Número de entidades atrasadas:	$N_a = \sum_i^n \partial\{A_i\}$
	Onde $\partial\{x\} = 1$ se $x > 0$
	e $\partial\{x\} = 0$ caso contrário

Figura 2. Medidas de desempenho típicas

Devido ao amplo estudo nas últimas décadas deste tipo de problemas, surgiu a necessidade de desenvolver sistemas de representação que descrevessem de forma sucinta os modelos que representam. Desta forma, existem várias nomenclaturas utilizadas para a classificação de problemas de escalonamento propostas por vários autores tais como Graham, Conway, French, Brucker, Blazewicz, Pinedo. Nesta dissertação utilizaremos a nomenclatura de Pinedo (2012), que define os problemas de escalonamento da produção em três classes, na forma $\alpha | \beta | \gamma$, tal como Brucker e Blazewicz. Onde α descreve como estão dispostos os recursos afetos à produção, β descreve as restrições do processo e γ descreve o critério de otimização (Blazewicz, 2001; Pinedo, 2012).

3.1 Sistemas de Produção

A complexidade do escalonamento depende de vários fatores, nomeadamente, do tipo de sistema de produção em que o problema ocorre.

Assumindo a igualdade de condições e restrições a influenciar um problema, pode afirmar-se que o esforço requerido para o escalonamento é normalmente menor em sistemas do tipo processador único, crescendo gradualmente à medida que se progride de um sistema desta natureza para sistemas mais “complexos”, como é o exemplo de sistema flexível que produza uma grande variedade de produtos (Varela, 2007). Contudo existem situações em que tal não se verifica, e tais diferenças entre sistemas de escalonamento podem ser muito ténues, desaparecer completamente, ou até mesmo inverterm-se. Tudo isto, devido à complexidade do problema em questão.

Os sistemas de produção podem ser caracterizados por diversos fatores, tais como número de recursos e/ou de máquinas, suas características e configuração, nível de automação, tipo de sistema de manipulação de material, entre outros (Pinedo, 2005). As diferentes características dos sistemas de produção podem originar diferentes formas de os classificar.

A classificação quanto à implantação é uma das mais utilizadas, assumindo-se que a maioria dos sistemas de produção têm configurações que se podem enquadrar em arranjos fundamentais, nomeadamente, as oficinas de produção, as linhas de produção e as células de produção (Alves, 1999) e estes tipos de implantações ou configurações que os sistemas de produção podem assumir no âmbito dos problemas de escalonamento da produção podem ser expressos através de uma variável α (Pinedo, 2012), que pode assumir diferentes valores, conforme o sistema de produção considerado em cada problema, conforme se descreve abaixo.

- **Oficinas de produção – Do inglês, *job shop (Jm)***, a organização neste tipo de sistema de produção é feita tendo em conta o processo a realizar, dispondo assim, cada oficina de equipamentos funções semelhantes. Este tipo de implantação é, geralmente,

utilizado em situações em que existe uma grande variabilidade de produtos, havendo a necessidade de recursos com algum grau de flexibilidade suficiente para serem capazes de executar um leque alargado de trabalhos, e nem sempre com a mesma sequência de operações.

Apesar da flexibilidade que confere à produção, este tipo de implantação tem a desvantagem de ter um número elevado de trabalhos em processamento (*work in progress*), assim como tempos de preparação (*setup*) que podem ser elevados.

- **Linhas de produção – Do inglês *flow shop (Fm)***, tratam-se de um tipo de sistema de produção em que os equipamentos são, em geral, altamente especializados, sendo implementados com a finalidade de produzir em série uma pequena gama de produtos, ou um único produto, tendo um único percurso ao longo do sistema de produção.

Este tipo de implantação permite geralmente uma elevada produtividade e elevada taxa de produção, com baixos custos, porém não confere grande flexibilidade de produção de outros produtos além dos implementados.

- **Células de produção – Do inglês *manufacturin cell* ou simplesmente *cell***, é o tipo de implantação em que a organização física dos recursos utilizados é tipicamente numa configuração simples, em linha ou em U, para o fabrico de produtos geralmente em pequenas séries, pois estes sistemas de produção dedicam-se geralmente a um único ou a um conjunto restrito de produtos pertencentes a uma mesma família de produtos, por intervalo de tempo de planeamento considerado. A produção tanto pode ser considerada para armazém como para encomenda, sendo esta última filosofia de produção geralmente mais comum neste tipo de sistema.

O facto de muitas vezes os recursos alternarem entre diferentes tarefas exige que se tenha em particular atenção os tempos de *setup*, utilizando-se muitas vezes técnicas de SMED (*Single-Minute Exchange of Die*) e OTED (*One-Touch Exchange of Die*) para a redução dos mesmos.

Numa classificação mais pormenorizada, e segundo a classificação de Pinedo (2012) dos problemas de escalonamento, poderemos então sintetizar para α , as seguintes configurações de sistemas de produção:

- Máquina Única (1 ou vazio);
- Máquinas em Paralelo Idênticas (Pm);
- Máquinas em Paralelo com Velocidades Diferentes (Qm);
- Máquinas em Paralelo Diferentes (Rm);
- Linha de Fabrico (Fm);
- Linha de Fabrico Híbrida (HFc);
- Oficina de Fabrico (Jm);
- Oficina de Fabrico Flexível (FJc);
- Oficina Aberta ou Sistema Aberto (Om)

3.2 Sistemas Orientados ao produto ou ao processo

Os arranjos fundamentais dos sistemas de produção podem dividir-se em duas grandes categorias, nomeadamente atendendo à sua dedicação ao produto denominando-se de Sistemas de Produção Orientados ao Produto (SPOP), ou dedicados ao processo ou ao serviço, tendo então a denominação de Sistemas de Produção Orientados à Função (SPOF). Tendo em conta estas categorias, as linhas e as células de produção enquadram-se geralmente incluídos na categoria de Sistemas de Produção Orientados ao Produto, enquanto que as oficinas de produção se enquadram geralmente em Sistemas de Produção orientados ao processo ou à função.

3.2.1 Sistemas de Produção Orientados ao Produto

As famílias de produtos manufaturados nos Sistemas de Produção Orientados ao Produto (SPOP) nem sempre são coincidentes com famílias comerciais. São agrupamentos de diferentes artigos

que podem e devem ser produzidos conjuntamente, não só por poderem partilhar os mesmos meios e processos de produção mas também, por muito frequentemente terem as mesmas sequências operatórias.

Devido aos requisitos de capacidade, na prática os sistemas de produção requerem mais do que um único processador ou máquina por fase de processamento. Assim os SPOP podem ter em cada fase de processamento um conjunto de máquinas replicadas ou máquinas paralelas i.e. máquinas equivalentes que frequentemente podem ser consideradas idênticas. O propósito da duplicação em diferentes fases de processamento é aumentar o *throughput*, para balancear a capacidade de produção ao longo da fábrica e, portanto, eliminar ou reduzir os efeitos negativos dos gargalos na eficiência geral do sistema de produção.

A afetação a um dado produto por um período limitado é uma característica fundamental dos SPOP, período esse, ao fim do qual o sistema é adaptado ou reconfigurado com possibilidade de supressão ou acréscimo de recursos de produção. Tal adaptação é dependente de eventuais novas necessidades de produção associadas ao mesmo, ou a um novo produto no próximo período de produção (Alves, 1999).

3.2.2 Sistemas de Produção Orientados ao Processo ou à Função

Os Sistemas de Produção Orientados à Função (SPOF) são sistemas caracterizados pela existência de secções funcionais em cada uma das quais se realiza uma função da produção, denominando-se este tipo de sistema por organização ou implantação funcional. Estes sistemas permitem geralmente produzir uma grande variedade de artigos em pequenos lotes ou mesmo fabricação unitária.

O fluxo da produção segue a sequência de operações necessárias implicando a deslocação dos lotes entre secções, e por vezes o retrocesso destes a secções por onde já tinham passado, no caso de oficina mais geral. Nestes casos o controlo do fluxo de produção é um processo complexo, pois existe um elevado número de *work in progress* (WIP) o que torna a determinação do tempo total de produção ou *makespan* numa tarefa normalmente difícil, assim como a entrega dos

produtos ao cliente na data prevista. Uma forma de atenuar estas dificuldades consiste em manter um stock de produtos acabados e de WIP, o que em contrapartida pode gerar elevados custos para as empresas.

3.3 Características dos Trabalhos e dos Recursos

Outro parâmetro que interfere na diversidade e complexidade de resolução dos problemas de escalonamento são as restrições do processo. Segundo Pinedo (2012), para β são possíveis as seguintes características:

- **Data de Lançamento** (r_j) - Representa a impossibilidade da tarefa j começar a ser processada antes da data r_j . Se não existir esta restrição em β , então todas as tarefas podem começar a ser executadas imediatamente, assim que chegam ao sistema de produção.
- **Interrupções** ($prmp$) - Representa a possibilidade de interrupção do processamento de uma tarefa antes de esta estar concluída. O trabalho interrompido não é perdido, permitindo a conclusão da tarefa à posteriori.
- **Relações de Precedência** ($prec$) - Representa a existência de relações de precedência entre a execução das tarefas. Uma tarefa que tenha uma relação de precedência sobre outra terá de ser processada completamente antes que a tarefa que a sucede.
- **Tempos de Setup dependentes da Sequência** (sjk) - Representa a existência de tempos de preparação, dependentes da sequência em que as tarefas são processadas. Assim s_{jk} , representa o tempo de preparação necessário entre a execução das tarefas j e k .
- **Família de Tarefas** (fm/s) - Representa a existência de famílias de tarefas. Dentro de cada família, as tarefas podem ter tempos de processamento diferentes mas não necessitam de tempo de preparação entre a execução das mesmas.

- **Produção por Lote** ($batch(b)$) - Representa a possibilidade de uma máquina executar mais que uma tarefa simultaneamente. Assim a máquina pode executar b tarefas simultaneamente, que apenas serão concluídas quando a tarefa com o tempo de processamento mais longo for concluída.
- **Indisponibilidade das máquinas** ($brkdown$) - Representa a indisponibilidade das máquinas para a realização de qualquer tarefa. Estes instantes ou períodos de indisponibilidade são por vezes também referidos como restrições de disponibilidade da máquina.
- **Permutações** ($prmu$) - Representa a impossibilidade de alterar a sequência de tarefas após execução na primeira máquina. Este tipo de restrição pode existir em configurações do tipo linha de fabrico.
- **Sem Espera** (nwt) - Representa a proibição de esperas entre duas máquinas em configurações do tipo linha de fabrico. Isto implica que uma tarefa apenas pode começar a ser executada, quando estiverem reunidas as condições para poder circular através da linha sem qualquer tempo de espera.
- **Recirculação** ($rcrc$) - Representa a possibilidade de uma tarefa poder visitar uma máquina mais do que uma vez. Esta característica apenas ocorre em configurações do tipo oficinas de fabrico simples ou híbridas.

Existem outras restrições específicas para certas configurações, tais como, as restrições de elegibilidade de uma máquina (M) que podem, por exemplo, existir em configurações de máquinas idênticas em paralelo (Pm), e o bloqueamento ($block$), um fenómeno que pode ocorrer em linhas de fabrico (Fm).

Assim, para caracterizar β é necessário conhecer as especificidades do problema de escalonamento a ser tratado. Ao contrário de α , o campo β pode conter mais do que uma variável ou parâmetro se o problema estiver sujeito a várias restrições, nomeadamente de entre o conjunto acima apresentado. Caso não existam restrições ao problema o mesmo encontra-se vazio.

3.4 Medidas de Desempenho ou Critérios de Otimização

A solução de um problema de escalonamento tem relação direta ou indireta com variáveis como os prazos de entrega da produção, a quantidade de trabalhos em curso, a utilização dos equipamentos, entre outros. Encontrar um bom critério de otimização, que traduza o objetivo a atingir, pode ser uma tarefa difícil no escalonamento. Muitas vezes a melhoria de um critério traduz-se na perda de eficiência por parte de outro critério. Assim para selecionar o critério de otimização é necessário conhecer o objetivo do problema de escalonamento. Para γ são possíveis variadas hipóteses, de entre as quais se destacam as seguintes:

- **Makespan** (C_{max}) – Representa o tempo desde do lançamento de uma ordem de produção até à sua conclusão. É equivalente ao tempo de conclusão do último trabalho que deixa o sistema. Em geral minimizar o *makespan* implica também otimizar a utilização dos recursos.
- **Atraso Máximo** (L_{max}) – Mede o pior caso de transgressão às datas de entrega acordadas. É definido pela tarefa que tenha o pior desempenho relativamente à sua data de conclusão, tendo em conta, a data de entrega definida, cujo prazo foi ultrapassado.
- **Soma Ponderada dos Atrasos Positivos** ($w_j T_j$) – Define a soma ponderada, das tarefas que não vão cumprir as suas datas de entrega. Isto é, tarefas cuja data de conclusão é maior que a data de entrega devida.
- **Número Ponderado de Tarefas Atrasadas** ($w_j U_j$) - Define uma soma ponderada de todos os trabalhos que não vão cumprir as suas datas de entrega.
- **Soma Ponderada dos Tempos de Conclusão** ($w_j C_j$) – Dá uma indicação dos custos de armazenamento ou inventário, incorridos pelo escalonamento. A soma dos tempos de conclusão é muitas vezes chamado de *flow time*. E a soma ponderada dos tempos de conclusão de *flow time* ponderado ou pesado.

3.5 Resumo

Neste o capítulo foi abordado o problema de escalonamento da produção. O escalonamento considera, geralmente, as funções de alocação e sequenciamento de cada trabalho ou tarefa, por um ou mais intervalos de tempo, a uma ou mais máquinas, incluindo eventuais recursos auxiliares para a execução dos trabalhos considerados. O objetivo do escalonamento é, idealmente, o de encontrar o valor ótimo para uma dada medida de desempenho ou mais, ou seja, minimizar ou maximizar, um dado critério ou conjunto de critérios de otimização.

O problema de escalonamento pode ser dividido em duas fases distintas, sendo elas, a afetação dos trabalhos ou das tarefas aos recursos ou simplesmente às máquinas disponíveis, e a sequenciação dos(das) mesmas de forma a utilizar os recursos de produção de forma eficiente, otimizando os tempos de produção totais e, se possível, cumprindo os prazos de entrega dos trabalhos.

O escalonamento da produção é uma função importante no contexto da gestão da produção industrial no qual podem existir problemas relativamente simples, até problemas com elevada complexidade decorrentes da dinâmica subjacente aos ambientes de produção em que ocorrem. Desta forma foram apresentadas, neste capítulo, as classes em que os problemas de escalonamento se podem dividir (Pinedo, 2012) e respetiva nomenclatura.

Foi ainda apresentada a classificação que tem por base os arranjos fundamentais dos sistemas de produção típicos, atendendo à sua dedicação ao produto, ou ao processo ou função. Assim, abordamos os sistemas orientados ao produto e os sistemas orientados à função e respetivas características de cada um deles.

4. MÉTODOS E SISTEMAS DE ESCALONAMENTO DE PRODUÇÃO

Os métodos de escalonamento têm uma aplicação bastante alargada à resolução de uma grande variedade de problemas de escalonamento, quer na prática industrial de produção de bens, quer também para os serviços (Varela, 2007).

Alguns problemas de planeamento e escalonamento da produção podem ser formulados como programas lineares e são de fácil resolução, sendo assim facilmente resolvidos através da utilização de algoritmos eficientes. Outros problemas podem ainda ser resolvidos por algoritmos de tempo polinomial. Contudo, existem muitos problemas intrinsecamente muito difíceis, problemas *NP-hard*, que não podem ser formulados na forma de programas lineares e para os quais não existem regras simples ou algoritmos que gerem soluções ótimas em tempo útil. Neste contexto é possível a utilização de métodos de resolução rápida e eficiente como são os métodos heurísticos. Estes podem ser métodos simples de pesquisa probabilística, como é o caso dos métodos de pesquisa aleatória, e diversos métodos heurísticos de aplicação direta baseados em regras de prioridade, ou mesmo outros procedimentos mais elaborados, que por exemplo, também se baseiam em regras de sequenciamento ou de prioridade, como é o caso do método de Johnson para linhas simples com duas máquinas apenas e sem restrições adicionais.

Torna-se então pertinente classificar os métodos e enquadrá-los nas classes de problemas que resolvem.

Neste capítulo abordaremos alguns dos métodos desenvolvidos para resolver problemas do tipo *NP-hard*.

4.1 Métodos de Escalonamento

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

As heurísticas tendem a ser procedimentos genéricos facilmente adaptáveis a uma grande variedade de problemas de escalonamento. Além disso, permitem, por vezes, obter soluções muito próximas da solução ótima, com base num número de iterações reduzido.

Todos os métodos a seguir apresentados, embora não garantam uma solução ótima, visam encontrar soluções razoavelmente boas em tempo relativamente curto e em tempo útil, para a resolução dos problemas.

4.1.1 Regras de Prioridade

A pesquisa de regras de prioridade tem sido alvo de variados estudos durante décadas, tendo sido estudadas na literatura muitas regras diferentes. Estas regras podem ser distinguidas e classificadas tendo em conta alguns critérios, como poderemos ver no Capítulo 4.

A vantagem da aplicação deste tipo de regras é a possibilidade de encontrar um bom escalonamento em tempo razoável quando se pretende otimizar uma única medida de desempenho tal como o *makespan*, o tempo total de conclusão ou, por exemplo, o atraso máximo dos trabalhos. De entre muitas outras regras de prioridade destacaremos as seguintes:

- **LWKR** (*Least Work Remaining*) - São selecionadas para processamento as tarefas cujo tempo total de processamento das operações por executar seja o menor.
- **MWKR** (*Most Work Remaining*) - São selecionadas para processamento as tarefas cujo tempo total de processamento das operações por executar seja o maior.
- **RANDOM** - Seleciona trabalhos aleatoriamente. Não há objetivo em melhorar qualquer medida de desempenho específica.
- **TWORK** (*Total Work*) - Seleciona para processamento as tarefas cujo tempo total de processamento de todas as operações seja o menor.
- **EDD** (*Earliest Due Date*) - Dá prioridade às tarefas com menor tempo até à data de entrega.

- **TSPT** (*Truncated SPT*) - A prioridade é atribuída com base na regra *SPT* até que pelo menos uma das tarefas da fila atinja uma espera determinada, W . Em que, W pode variar entre zero e um tempo estipulado.

Outras regras como *SPT*, *LPT* e *FCFS* serão abordadas no capítulo 4.

No entanto, frequentemente no mundo real existe a necessidade de otimizar mais do que um critério de desempenho. Por exemplo, um objetivo realista pode ser uma combinação de vários objetivos básicos em função do tempo, ou em função do conjunto de trabalhos em espera para processamento.

Para tal existem regras de prioridade mais elaboradas, que contemplam vários parâmetros diferentes, para assim tratar funções objetivo mais complexas. Estas são designadas de regras de sequenciamento compostas. Uma regra de prioridade composta é uma expressão de classificação que combina uma série de regras de prioridade elementares.

4.1.2 Métodos de Pesquisa Local e Meta-Heurísticas

Os métodos de pesquisa local são uma importante classe de algoritmos que embora não garantam a solução ótima procuram a melhor solução para a função objetivo na sua vizinhança. Dois escalonamentos são vizinhos se existir uma regra que permita transformar uma sequência na outra e vice-versa. Este método realiza a cada iteração uma pesquisa dentro da vizinhança e avalia as várias soluções obtidas, aceitando ou rejeitando as soluções encontradas na vizinhança mediante a comparação com um critério de aceitação/rejeição pré-definido.

A técnica de *hill-climbing* é uma estratégia de intensificação genérica, usada na Programação da Produção por Wilkerson e Irwin em 1971 (Varela, 2007). Esta técnica tenta encontrar primeiramente uma solução inicial com recurso a qualquer outro método de escalonamento, e de seguida explora todos os caminhos possíveis para alterar a sequência gerada. Desta forma esta técnica mantém as soluções na vizinhança da solução corrente e avalia cada sequência resultante. O processo de pesquisa da melhor solução termina quando não forem possíveis mais melhorias.

No entanto se existirem melhorias possíveis às soluções encontradas seleciona-se a melhor solução encontrada até ao instante, e continua-se a pesquisa a partir desta. Este processo repete-se sucessivamente até não se puderem efetuar melhorias.

Os métodos baseados na pesquisa de vizinhança são adequados para a obtenção de boas soluções de forma rápida, simples e flexível. Subjacente a estes métodos existe um conjunto de procedimentos:

- Um mecanismo de exploração de vizinhanças;
- A geração das vizinhanças;
- Um método para seleção de uma (nova) semente;
- A geração da sequência através de um procedimento de pesquisa local ou outra heurística.

É relativamente rápido e fácil programar e procurar um local ótimo, sendo que, se o processo for iniciado a partir de uma boa semente heurística este pode ser extremamente preciso. A sua principal desvantagem está no facto de utilizar exclusivamente procedimentos de intensificação em vez de diversificação, pelo que se deverá tentar balancear vantagens de ambos tipos de abordagens.

Existem extensões ao conceito de *hill-climbing* que permitem considerar alguns métodos de pesquisa local estendida, nomeadamente baseados nas técnicas de *simulated annealing*, pesquisa tabu, *ejection chains* e algoritmos genéticos, entre outros (Varela, 2007).

4.1.3 Métodos de gargalo de estrangulamento ou bottleneck methods

Os métodos de gargalo de estrangulamento ou *bottleneck methods* são outro exemplo de métodos ou procedimentos heurísticos e podem dividir-se em decisões do tipo *Myopic* para a sequenciação, com base em regras do tipo *OPT-like*, e/ou *Bottleneck Dynamics*.

Os *shifting bottleneck methods* são métodos bastante utilizados no escalonamento em configurações do tipo oficinas, nomeadamente para problemas do tipo $Jm \parallel C_{max}$.

Estes métodos mostram o relacionamento entre o conceito de estrangulamento e conceitos combinatórios como o caminho crítico (mais longo) e o atraso máximo. O caminho crítico indica a localização e o momento de um gargalo. O atraso máximo dá a indicação do valor que o *makespan* aumenta, se uma máquina for adicionada ao conjunto de máquinas já programadas. Embora sejam limitados ao objetivo *makespan* e a sistemas de produção de complexidade moderada, têm um grande potencial pois permitem estimar custos associados a cada tarefa e aos recursos de um sistema de produção (*Bottleneck Dynamics*). Desta forma é possível a resolução de vários problemas pois é possível, por exemplo, estimar a tarefa à qual está associada um maior rácio benefício/custo, e assim balancear o sistema, assim como decidir sobre a espera de processadores mantendo-os inativos até à chegada de trabalhos urgentes ou prioritários, suscetíveis de estar prestes a chegar ao sistema de produção.

Além disso, existem muitos outros tipos de métodos que podem ser usados para efetuar o escalonamento da produção, quer heurísticos quer otimizantes, porém, o foco deste trabalho recai essencialmente no uso do métodos heurísticos e em especial alguns que existem disponíveis para serem usados através de sistemas gratuitos de escalonamento da produção, como é o caso do Sistema LEKIN® de Pinedo (2012), que foi usado para a realização deste trabalho, entre outros, que são brevemente apresentados a seguir.

4.2 Sistemas de Escalonamento

Durante as duas últimas décadas, centenas de sistemas de programação têm sido desenvolvidos na indústria e na academia de vários países, quer sistemas genéricos ou de uso mais geral, quer mais específicos.

Os sistemas genéricos comerciais são projetados para aplicação numa ampla variedade de configurações ou tipos de sistemas ou ambientes de produção, com a necessidade de apenas pequenas personalizações ou adaptações necessárias por parte do utilizador. Apesar de serem

sistemas genéricos, muitas vezes estes sistemas podem então ser especificados para um determinado setor de atividade ou contexto de produção.

Além dos sistemas comerciais existem protótipos académicos desenvolvidos para investigação e ensino, cuja conceção é completamente diferente dos sistemas comerciais.

Estes sistemas tendem também a ser essencialmente sistemas de uso geral, incluindo normalmente diferentes tipos de abordagens ao escalonamento da produção, do que resultam diferentes tipos de sistemas. Entre eles existem sistemas baseados em inteligência artificial, sistemas baseados em redes neuronais e algoritmos genéticos, e sistemas baseados em técnicas de investigação operacional (Varela, 2007). A aplicação destes sistemas pode ser efetuada em vários sectores industriais, e podem incluir vários métodos para a resolução de um conjunto vasto de problemas de diferentes graus de complexidade e abrangência.

Além disso, os sistemas de escalonamento da produção podem ainda ser definidos em termos de sistemas tradicionais ou sistemas web.

4.2.1 Sistemas tradicionais

Consideram-se sistemas tradicionais, os sistemas que se baseiam em tecnologias convencionais existentes e que são usados ainda na atualidade, para o apoio à tomada de decisão no escalonamento.

De entre estes sistemas destacaremos o sistema LEKIN® e LISA® usados para configurações do tipo processadores paralelos, linhas e oficinas flexíveis (LEKIN®) e processador único (LISA®).

- **LEKIN®**

O sistema LEKIN® foi desenvolvido com o intuito de escalonamento de trabalhos em configurações do tipo processadores paralelos, linhas e oficinas flexíveis. Além destes, permite abordar operações de montagem, isto é, uma unidade que foi realizada num processador e outra

unidade que foi realizada num segundo processador poderão ambas ir para um terceiro processador para serem montadas como uma unidade. Assim sendo pode ser usado como ferramenta de ensino e suporte à investigação, incluído vários tipos de métodos e algoritmos para diversos ambientes de produção, tais como: máquina única (*single machine*), máquinas paralelas (*parallel machines*), linha de fabrico (*flow shop*) e oficina de fabrico (*job shop*), além de, linha de fabrico flexível ou híbrida (*flexible or hybrid flow shop*), e ainda oficina de fabrico flexível (*flexible job shop*).

Este sistema permite ainda a inclusão de tempos de preparação (*setup*) dependentes da sequência de processamento em todos os processadores, e possui um calendário que permite restringir o número de movimentos para cada processador e a definição de períodos de manutenção e feriados. As suas limitações são a execução de até 18 trabalhos, em até 10 centros de trabalho, e até 4 processadores em cada centro de trabalho.

Estão disponíveis para utilização métodos como as regras de prioridade, heurísticas carregadas na base de dados do sistema e heurísticas definidas pelo utilizador para resolver problemas atendendo a vários critérios, incluindo, *makespan* ou C_{max} , *maximum tardiness*, *total number of late jobs*, *total flow time*, *total tardiness*, *total weighted flow time* e *total weighted tardiness* (Pinedo, 2012).

O sistema permite ainda visualizar as soluções obtidas através de diagramas de Gantt, assim como guardar e comparar diferentes soluções obtidas pela execução de diferentes métodos ou heurísticas, e permite ainda a importação e exportação de algoritmos.

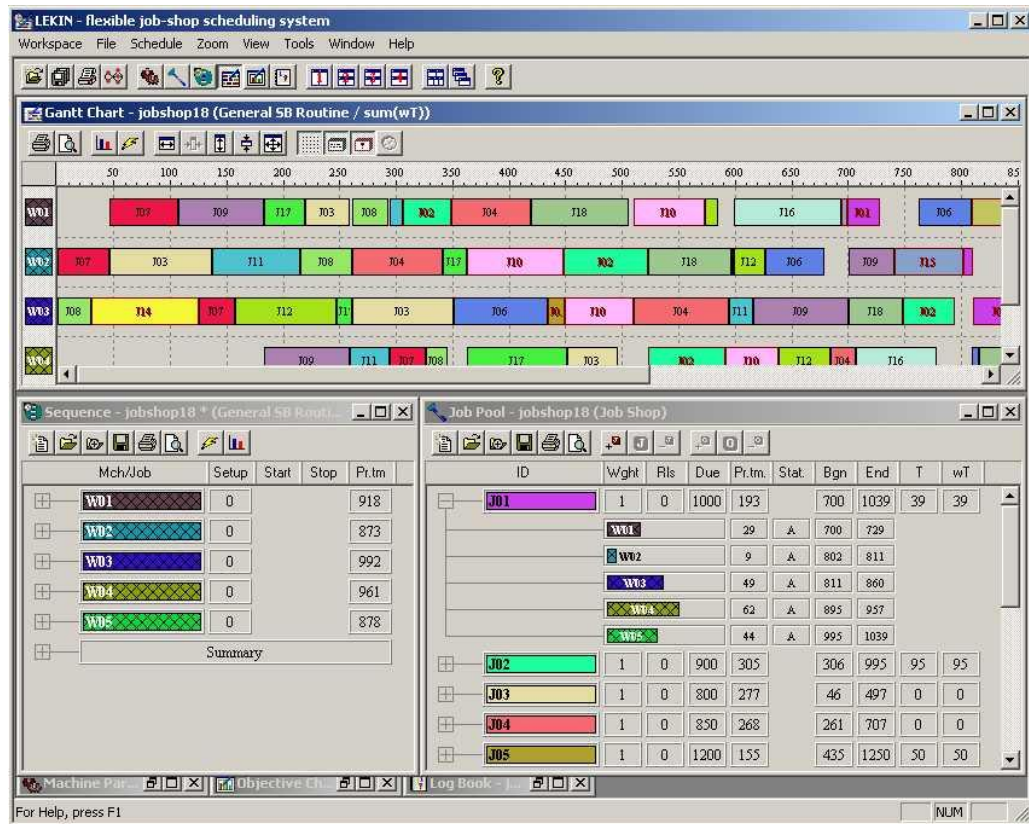


Figura 3. Sistema LEKIN

○ **LISA®, Library of Scheduling Algorithms**

O sistema LISA®, Library of Scheduling Algorithms, é um sistema para entrada, edição e resolução de problemas determinísticos de escalonamento, particularmente para a resolução de problemas de processador único ou máquina única. Este sistema pode ser usado para determinar a complexidade dos problemas permitindo visualizar e manipular as soluções obtidas, podendo ser a base de desenvolvimento de métodos adequados à resolução de determinados problemas. Os métodos disponibilizados para a resolução dos problemas são implementados externamente e podem ser métodos heurísticos, assim como, métodos exatos. Entre as cerca de três dezenas de métodos disponibilizados incluem-se regras de prioridade e métodos do tipo *branch-and-bound*,

shifting bottleneck, beam search, neighborhood search: simulated annealing, threshold accepting, tabu search entre outros procedimentos heurísticos e exatos. Para além destes métodos, permite também testar métodos desenvolvidos pelo utilizador.

Existem muitos outros sistemas tais como, o sistema de escalonamento desenvolvido por Lin e Lee, o Sistema de escalonamento desenvolvido por Kazerooni *et al*, desenvolvidos também na década de 90. Outros sistemas de escalonamento mais antigos como o YAMS, SCORE, PATRIARCH, MASCOT, OPAL e o ISIS, entre muitos outros, foram já desenvolvidos nas décadas de 80 para tratar também da resolução de problemas de escalonamento e muitos outros continuam a surgir atualmente, incluindo sistemas baseados na web.

4.2.2 Sistemas de web

Os sistemas de web são sistemas que surgiram na sequência dos sistemas tradicionais, porém, têm a vantagem de poderem ser facilmente usados e partilhados em rede. Nas duas últimas décadas foram desenvolvidos alguns sistemas desta natureza tais como os que serão seguidamente apresentados.

- **NEOS Server**

O NEOS Server é resultado da colaboração entre o *Optimization Technology Center*, da Northwestern University e do Argonne National Laboratory. Trata-se de um sistema web que permite resolver problemas de otimização e que tem uma abordagem simples para a resolução de uma grande variedade de problemas de escalonamento entre outros. Disponibiliza 60 programas através de várias interfaces disponíveis em <http://www.neos-server.org/neos/>.

○ **BBN's Vishnu**

É um sistema web que possui um mecanismo de escalonamento poderoso, assim como, um módulo facilmente adaptável para a realização de diversas formas de escalonamento, desde otimização de operações, planeamento estratégico e logístico, e o escalonamento de processos. O sistema automático do Vishnu ("schedulers") utiliza pesquisa genética para encontrar programas otimizados. É um sistema fácil de integrar dado que as entradas e saídas, assim como as especificações dos problemas são todas expressas através de *xml schemas*. Trata-se de um sistema de escalonamento de ampla aplicação e com reduzidos custos de desenvolvimento e implementação. As principais características deste sistema são:

- **Otimização** – programa de pesquisa genética que permite encontrar uma sequência e alocação ótimos, tendo em conta as restrições dos problemas.
- **Integração** - entradas, saídas e especificações do problema são todos expressos através de esquemas XML no programa Vishnu reconfigurável.
- **Colaboração** - múltiplos programas Vishnu podem colaborar, quer na forma de processos autónomos em Java ou numa estrutura de enquadramento de agentes.
- **Modo de servidor web** - todas as interações podem ser efetuadas através da web, como sejam definir problemas de escalonamento, ver programas e controlar quando o programa automático deverá decorrer.
- **Desenvolvimento rápido de protótipos** - permite o desenvolvimento de protótipos rápidos, sendo possível ao mesmo tempo realizar todas as outras operações, nomeadamente editar programas, dados e regras de escalonamento e personalizar a visualização de dados.

- **FortMP**

O FortMP é um sistema web de programação matemática concebido para resolver problemas de programação linear, programação quadrática, programação inteira através de algoritmos que incluem o SIMPLEX, PRIMAL e DUAL. É projetado em quase todos os processadores e é disponibilizado o seu código-fonte. Este sistema web é atualizado regularmente e aceita dados de problemas em formato MPSX, permitindo aos utilizadores usar os próprios métodos/algoritmos para a entrada de dados de problemas de otimização.

- **e-OCEA**

Trata-se de um portal web para investigação em escalonamento, incluindo métodos, referências e conjuntos de dados trazidos pela comunidade de escalonamento. Este projeto é da responsabilidade de um grupo de investigadores que trabalha em escalonamento no Laboratório de Ciências da Computação da Universidade de Tours.

Alguns componentes do sistema estão já completamente funcionais, embora outros continuam ainda em fase de desenvolvimento e melhoramento. O objetivo do portal é fornecer gratuitamente uma plataforma para escalonamento, ajudar no desenvolvimento de novos algoritmos e conduzir fóruns de discussão através da Internet.

- **LekiNET**

O LekiNET é um protótipo de um ambiente de escalonamento na Internet, resultante da migração de um sistema de escalonamento para configurações do tipo oficina de fabrico flexível, contudo o LekiNET valoriza custos económicos, e agentes de escalonamento para a resolução de problemas. Trata-se de um sistema colaborativo formado por um grupo de agentes de escalonamento na Internet que partilha recursos computacionais para a resolução de problemas.

○ **Riot – Remote Interactive Optimization Testbed**

Este sistema web permite ao utilizador experimentar diferentes ambientes de produção, características de problemas e combinações de objetivos, testando o desempenho dos sistemas de produção com recurso a vários métodos. Estes métodos podem ser ótimos ou heurísticos. Geralmente os métodos são procedimentos simples ou regras de prioridade, mas permitem diferentes combinações de métodos e objetivos a testar.

4.3 Resumo

O estudo académico do escalonamento ainda apresenta diferenças significativas dos problemas encontrados na realidade industrial.

Apesar de existirem métodos exatos para a resolução dos problemas de escalonamento estes são, em geral, pouco apropriados para problemas com funções de complexidade não polinomiais. Desta forma muitas vezes são utilizados os métodos heurísticos, que embora não garantam as soluções ótimas requerem menos tempo de processamento para a sua resolução.

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

Neste capítulo apresentámos alguns métodos para a resolução de problemas de escalonamento, nomeadamente os métodos utilizados para a resolução dos problemas em estudo nesta dissertação.

Porém, tendo em conta os objetivos pretendidos com o escalonamento da produção é necessário por vezes encontrar sistemas capazes que resolvam os problemas de escalonamento.

Os sistemas de escalonamento têm diferentes abordagens, que vão desde abordagens tradicionais a sistemas web, e ambos os tipos de sistemas podem consistir em ferramentas de trabalho

importantes para dar apoio na decisão de escalonamento da produção, tal como foi no caso da realização deste trabalho que se baseou no uso do sistema LEKIN ® de Pinedo (2012).

Existem sistemas que atuam no âmbito da investigação operacional (IO), onde se têm desenvolvido métodos muito diversificados, outros que atuam no domínio da inteligência artificial (IA), e ainda outros com regras baseadas no bom senso e heurísticas, e também baseados em simulação.

A utilização destes sistemas e métodos permitem um melhor suporte à tomada de decisão no escalonamento da produção para melhor coordenar as tarefas e as máquinas de forma a cumprir os prazos de realização e entrega dos trabalhos, o que se torna particularmente crítico quando se lida com problemas oriundos do contexto industrial real, tal como aconteceu neste estudo, que foi baseado na resolução de um problema de escalonamento da produção do ramo automóvel.

5. CASO INDUSTRIAL

O problema de escalonamento considerado neste trabalho relaciona-se com um processo de produção com duas fases, envolvendo operações de termoformagem e prensagem numa indústria de componentes automóveis (J. Costa & Varela, 2012), cujos tempos de processamento se encontram na tabela seguinte.

Tabela 1. Tempos de processamento

Trabalho	Aquecedor de moldes	Prensa	Total
1	320 Min	600 Min	920 Min
2	320 Min	210 Min	530 Min
3	320 Min	400 Min	720 Min
4	230 Min	200 Min	430 Min
5	280 Min	267 Min	547 Min
6	320 Min	167 Min	487 Min
7	230 Min	267 Min	497 Min
8	280 Min	147 Min	427 Min
9	280 Min	150 Min	430 Min
10	300 Min	200 Min	500 Min
11	230 Min	300 Min	530 Min
12	230 Min	147 Min	377 Min

A primeira operação baseia-se no aquecimento do molde e a segunda é uma operação de corte com recurso a uma prensa. Ambas as operações são executadas em 12 trabalhos ou tarefas. Para a realização de tais tarefas existem três aquecedores de moldes e três prensas. Cada um dos trabalhos requer uma operação de aquecimento dos moldes e uma operação de corte com prensa. Os trabalhos são processados primeiramente no aquecedor de moldes antes de serem processados na prensa de corte. Dois ambientes de SPOP foram configurados de forma a resolver o problema de escalonamento destes trabalhos. Num deles foi considerada a configuração de PFS (*Parallel Flow Shop*) e no outro a configuração de uma HFS (*Hybrid Flow Shop*). Em cada configuração ou ambiente de produção foram usadas algumas heurísticas e meta-heurísticas,

entre os métodos a seguir descritos para resolver o problema de escalonamento das tarefas consideradas.

Os tempos de processamento dos 12 trabalhos ou tarefas para ambas as operações de aquecimento e prensagem encontram-se na Tabela 1. Os trabalhos são executados sem interrupções (*preemption*). As restrições de precedência entre as operações de aquecimento e prensagem devem ser satisfeitas. Neste estudo industrial foram aplicadas diferentes abordagens para o escalonamento das 12 tarefas com duas operações, com base nas configurações de produção PFS e HFS anteriormente descritas.

Deste trabalho resultou um artigo apresentado na 6ª Conferência Internacional Nabic14, e publicado (Costa, Varela, & Carmo-Silva, 2014) acessível através do IEEE Xplore Digital Library (<http://ieeexplore.ieee.org/Xplore/home.jsp>).

5.1 Linhas de fabrico paralelas

O problema de escalonamento em estudo, quando aplicada a configuração de PFS é do tipo Fm || C_{max} e compreende três linhas paralelas (PFS) com duas fases de processamento cada, como ilustra na Figura 4.

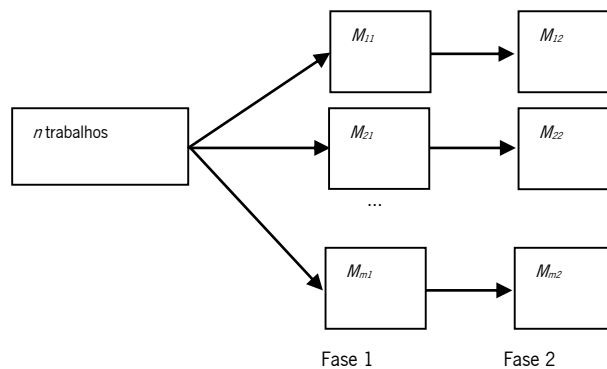


Figura 4. Configuração PFS.

Para resolver o problema de escalonamento em configuração PFS foram consideradas três estratégias diferentes, nomeadamente o procedimento de enumeração total combinado com o algoritmo de Johnson, a heurística de Kedia combinada com o algoritmo de Johnson, e a heurística *general shifting bottleneck routine* (SBR) através do sistema LEKIN® de Pinedo (2012). Estes e outros métodos estão também disponíveis em forma de *web-based systems* como descrito em Varela, Putnik, & Cruz-Cunha (2012); e Vieira, Varela, & Putnik, (2012).

5.1.1 Procedimento de enumeração total

Este procedimento (enumeração total) consiste simplesmente em explorar todas as soluções possíveis de afetação das tarefas para os 12 trabalhos do problema considerado, tendo em vista a afetação destas 12 tarefas a 3 linhas simples (do tipo *flow shop*). Foram então consideradas três linhas, existentes em paralelo, tendo-se estipulado que a cada uma das linhas deveria ficar atribuído um terço do total dos 12 trabalhos, i.e. 4 trabalhos por linha de fabrico. Neste caso aplicando um procedimento de enumeração total implementado foram então geradas todas as ${}^{12}C_4$ combinações possíveis para este problema de escalonamento. Foi feita a afetação tendo em conta o tempo de processamento total para cada trabalho, e seleccionada a melhor solução obtida em termos de tempo de produção total (*makespan*).

Para o subconjunto de tarefas que foram alocadas a cada uma das três linhas de fabrico, cada uma delas integrando duas máquinas, através da enumeração total de soluções possíveis anteriormente expostas, seguidamente foi aplicado o algoritmo de Johnson (1954) de forma a obter a sequência ótima das tarefas que passam pelo aquecedor de moldes e na prensa, em cada uma das 3 linhas de fabrico.

5.1.2 Heurística de Kedia

A heurística de Kedia pretende minimizar o C_{max} (Pinedo, 2012). Este método tem aplicação em problemas de máquinas paralelas com lotes independentes sem interrupção. Embora não garantindo a minimização do C_{max} , é considerada uma heurística muito eficiente, obtendo frequentemente a solução ótima do problema (K.R. Baker, 1974).

Esta heurística usa a regra de prioridade *Longest Processing Time* (LPT) como mecanismo de afetação dos trabalhos. O seu procedimento de implementação encontra-se descrito na tabela 2.

Tabela 2. Algoritmo da Heurística de Kedia

1º Passo - Ordenar os trabalhos por ordem decrescente do tempo de processamento, ou seja, através da regra de prioridade *Longest Processing Time* (LPT).

2º Passo - Afetar os trabalhos, ordenadamente, começando pelo primeiro da lista, afetando-o à máquina com a mínima carga já afetada.

A heurística de Kedia (K.R. Baker, 1974) foi usada para a alocação das 12 tarefas às três linhas paralelas. A heurística de Kedia é aplicada ao sistema de máquinas paralelas sucessivamente, alocando os trabalhos à máquina com menor carga de trabalho baseando-se na regra *Longest Processing Time* (LPT). Foi usado o tempo total de processamento de cada trabalho, relativo a ambas as operações de aquecimento e prensagem destes. A alocação dos trabalhos a cada linha de fabrico foi então obtida, numa primeira abordagem, através da aplicação da heurística de Kedia, usando-se seguidamente a regra de Johnson para obter a melhor sequência de fabrico dos trabalhos alocados em cada uma das três linhas de fabrico.

5.1.3 Algoritmo de Johnson

O algoritmo de Johnson resolve problemas do tipo $F2 || C_{max}$, permitindo obter a melhor sequência ordenada numa linha de produção com duas máquinas (Pinedo, 2012). Este método não garante soluções ótimas, mas sim a melhor das sequências ordenadas, visto que o conjunto de todas as sequências ordenadas possíveis não contém obrigatoriamente a sequência em que o C_{max} é ótimo (Carmo-Silva, 2010).

Originalmente, obtidos por Johnson (1954), os resultados deste algoritmo são fundamentos normalizados na teoria da programação da produção. A implementação deste método faz-se da seguinte forma:

A regra de Johnson determina que a entidade i precede a entidade j numa sequência ótima caso,

$$\min \{t_{i1}, t_{i2}\} \leq \min \{t_{j1}, t_{j2}\}$$

e segue os passos descritos na tabela 3:

Tabela 3. Algoritmo de Johnson

- **1º Passo** - Encontrar $\min \{t_{i1}, t_{i2}\}$
 - **2º Passo** - Se o menor tempo de processamento requer a 1ª máquina, colocar a entidade respetiva na primeira posição disponível para o trabalho em questão na sequência. Ir para o passo 3.
 - **2º Passo: fase b** - Se o menor tempo de processamento requer a 2ª máquina, colocar a entidade respetiva na última posição disponível na sequência dos trabalhos. Ir para o passo 3.
-

-
- **3º Passo** - Retirar a entidade atribuída e voltar ao passo 1 até que todas as entidades sejam atribuídas.
-

O algoritmo de Johnson foi utilizado na configuração PFS para obter as melhores sequências, após terem sido aplicados outros métodos para a afetação dos trabalhos. Num dos casos, a primeira abordagem para a afetação dos trabalhos às três linhas foi efetuada com recurso ao procedimento de enumeração total anteriormente descrito, no outro caso foi utilizada a heurística de Kedia para realizar a afetação.

5.1.4 Heurística *General Shifting Bottleneck Routine* (PFS)

No caso da configuração em tipo de PFS primeiramente a alocação foi realizada considerando três linhas paralelas para a alocação das tarefas e tendo em conta o tempo total de processamento de cada tarefa, como se descreveu anteriormente, mas agora recorrendo à heurística SBR do sistema LEKIN® (Pinedo, 2012). Seguidamente aplicou-se a mesma heurística SBR a cada linha, de forma a obter a melhor sequência das tarefas, isto considerando o tempo de processamento de cada tarefa para cada uma das operações.

Os resultados obtidos são apresentados e analisados no capítulo seguinte.

5.2 Linha de fabrico híbrida

A outra configuração considerada neste caso industrial, para a resolução de um problema do tipo $HFc | prec | C_{max}$ foi a aqui designada por HFS. As tarefas podem ser processadas em qualquer máquina disponível de cada fase para processar a operação correspondente, com o único requisito de satisfazer a precedência de trabalhos da fase 1 para a fase 2, como representado na Figura 5. Para resolver o problema de escalonamento na configuração do tipo HFS usamos o software

LEKIN® novamente, e testamos três regras de prioridade, nomeadamente FCFS, LPT e SPT; e a heurística *General Shifting Bottleneck Routine* (SBR).

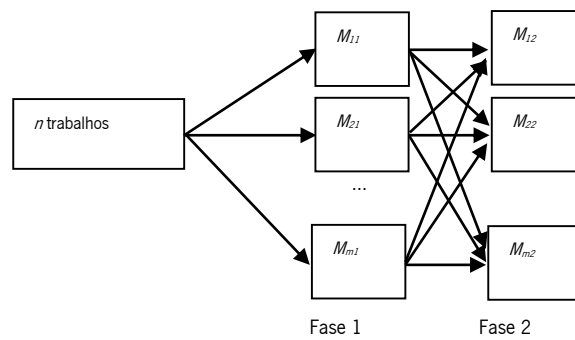


Figura 5. Configuração HFS com duas fases.

5.2.1 Regras de Prioridade

Estas são regras que estabelecem a prioridade com que as tarefas são processadas numa máquina. Quando uma máquina termina o processamento de uma tarefa, é a regra de prioridade que seleciona a tarefa a ser processada seguidamente, por prioridade mais elevada. Estas regras podem ser simples ou combinadas, locais ou globais (K.R. Baker, 1974). De entre as centenas de regras de prioridade existentes interessa fazer a distinção entre regras estáticas e dinâmicas. As regras de prioridade estáticas são regras em que a posição das tarefas dentro da sequência não é alterada com o tempo, nas regras dinâmicas o valor da prioridade difere com o tempo, ou seja, as tarefas são sequenciadas sempre que uma decisão deve ser tomada.

De entre a grande variedade de regras de prioridade destacamos as seguintes, pelo facto de que foram as regras utilizadas neste trabalho:

-
- **SPT** (*Shortest Processing Time*) - São selecionadas para processamento primeiramente as tarefas que demorem menos tempo a ser processadas nessa máquina;
 - **LPT** (*Longest Processing Time*) - São selecionadas para processamento primeiramente as tarefas que demorem maior tempo a ser processadas nessa máquina;
 - **FCFS** (*First Come First Served*) - É selecionada para processamento primeiramente a tarefa que está disponível mais cedo para processamento. Esta regra garante que nenhuma tarefa fica em espera infinitamente.

5.2.2 Heurística *General Shifting Bottleneck Routine* (HFS)

Trata-se de uma heurística muito utilizada para resolver problemas de minimização do *makespan* em configurações do tipo oficina de fabrico.

A heurística *General Shifting Bottleneck Routine* (SBR) é uma heurística que sequencia as máquinas uma por uma sucessivamente, identificando a máquina que é o gargalo de entre as máquinas ainda não alocadas (Pinedo, 2012). Depois de uma nova máquina ser alocada, todas as sequências anteriormente estabelecidas são reotimizadas localmente. Embora a heurística apresente bons resultados, continua a ser necessário resolver problemas da classe $I|r_j|L_{max}$ para cada uma das máquinas individualmente. O procedimento de implementação da heurística pode ser verificado a seguir, na Tabela 4:

Tabela 4. Algoritmo da Heurística Shifting Bottleneck

1º Passo - M = Todas as máquinas no problema;

2º Passo - M_0 = Máquinas onde as tarefas já foram sequenciadas;

Inicialmente $M_0 = \emptyset$

3º Passo - Escolher entre $M - M_0$ a máquina *bottleneck*. Determinar qual vai ser o próximo *bottleneck*, resolver um problema do tipo $1/r_j/L_{max}$, para cada uma das máquinas, e escolher aquela que obteve o atraso máximo;

4º Passo - Para a máquina escolhida como *bottleneck*, conectar os arcos disjuntivos, de maneira a minimizar o atraso máximo;

5º Passo - Para todas as máquinas pertencentes a M_0 , analisar a sequência das tarefas novamente;

6º Passo - Até todas as máquinas pertencerem a M_0 , voltar ao 3º passo.

A heurística *General Shifting Bottleneck Routine* (SBR) foi utilizada em ambas as configurações PFS e HFS, recorrendo ao software LEKIN®.

5.3 Resumo

Existem diversas heurísticas para resolver o problema de minimização do *makespan*, sendo que existem métodos que são aplicáveis a qualquer ambiente de produção, e outros métodos específicos para determinado ambiente.

Neste capítulo foi apresentado o caso industrial em estudo e as suas especificidades em cada um dos ambientes SPOP considerados. Foram apresentados os métodos e sistema utilizados para a resolução do problema de escalonamento nos dois ambientes de produção, HFS e PFS, e descritos os procedimentos de aplicação dos métodos implementados.

Na configuração PFS foram utilizados três estratégias, nomeadamente o procedimento de enumeração total combinado com o algoritmo de Johnson, a heurística de Kedia combinada com o algoritmo de Johnson, e a heurística *general shifting bottleneck routine* (SBR) através do sistema LEKIN® de Pinedo (2012).

Na configuração HFS utilizamos o software LEKIN® e foram testadas três regras de prioridade, nomeadamente FCFS, LPT e SPT; e a heurística *General Shifting Bottleneck Routine* (SBR).

6. ANÁLISE DOS RESULTADOS

Neste capítulo tratamos da análise dos resultados obtidos através da resolução dos problemas de escalonamento em estudo nesta dissertação. As medidas de desempenho consideradas nestes problemas foram o *makespan* (C_{max}) e a taxa de utilização do sistema. Além disso, os valores médios de tempo de percurso dos trabalhos no sistema de produção correspondente também foram analisados.

6.1 Problema PFS

Na configuração PFS usamos as três estratégias de escalonamento descritas anteriormente. Os principais resultados obtidos para cada estratégia usada são apresentados seguidamente.

6.1.1 Procedimento de enumeração total combinado com Regra de Johnson

Para o procedimento de enumeração total o *makespan* mínimo do sistema foi de 1550 minutos, (Figura.6), correspondendo ao C_{max} da linha de fabrico 1. Na linha de fabrico 2 o C_{max} foi de 1330 min. e na linha de fabrico 3 foi de 1487 min.

O *makespan*, em média, neste cenário, foi de 1456 min., sendo que a taxa de utilização média do sistema foi de 69%.

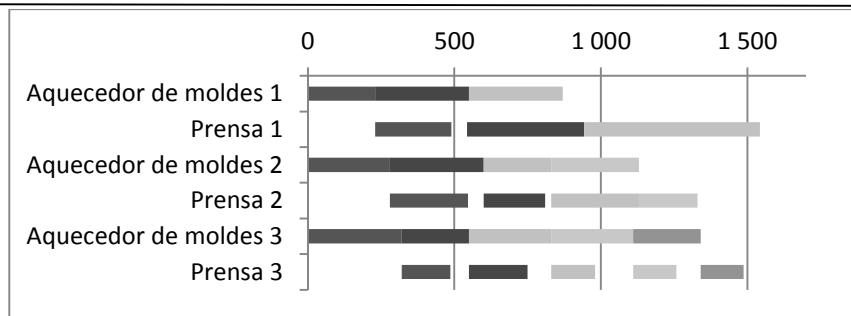


Figura 6. Diagrama de Gantt – Enumeração total combinado com Regra de Johnson

6.1.2 Heurística de Kedia combinada com Regra de Johnson

O *makespan* mínimo do sistema utilizando a heurística de Kedia foi de 1467 minutos (Figura 7), correspondendo também ao C_{max} da linha de fabrico 1. Na linha de fabrico 2 o C_{max} teve um valor de 1297 min., e na linha de fabrico 3 o valor de 1260 min.

Neste cenário o *makespan* médio foi de 1341 min., e a taxa de utilização do sistema teve um valor de 73%.

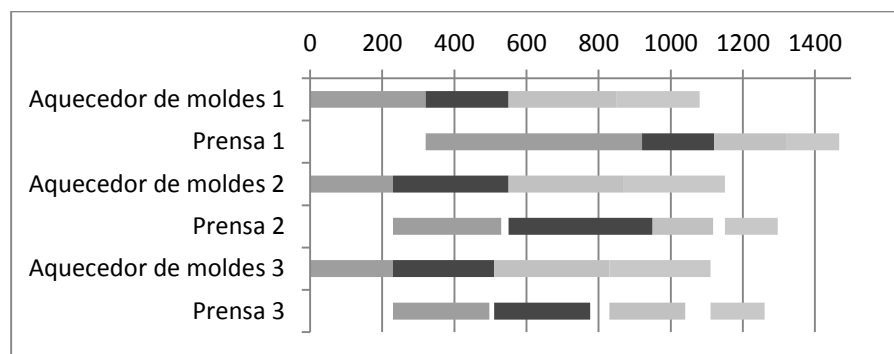


Figura 7. Diagrama de Gantt – Heurística de Kedia combinada com Regra de Johnson

6.1.3 Heurística *General Shifting Bottleneck Routine*

O $makespan$ mínimo do sistema utilizando a heurística SBR foi de 1444 minutos (Figura 8), sendo este o C_{max} da linha de fabrico 3. Na linha de fabrico 1 o C_{max} foi de 1367 min., e na linha de fabrico 2 teve um valor de 1387 min.

Neste cenário o $makespan$ médio foi de 1349 min., sendo que taxa de utilização do sistema foi de 74%.

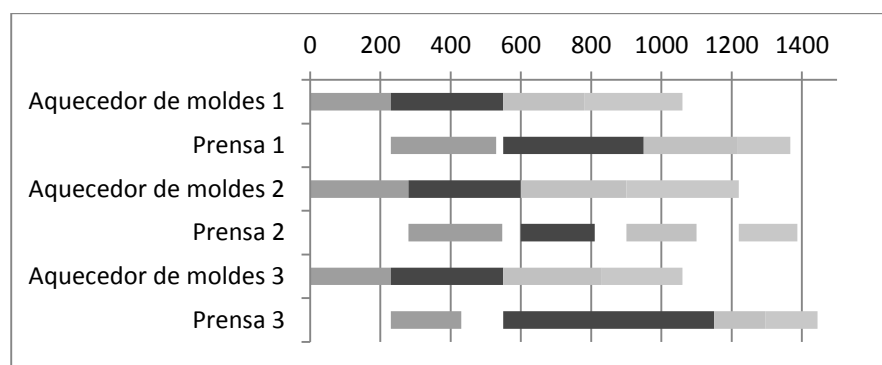


Figura 8. Diagrama de Gantt – *General Shifting Bottleneck routine* (PFS)

6.2 Problema HFS

Neste cenário foram utilizadas algumas regras de prioridade entre as quais a LPT, a SPT e a FCFS, e a heurística *General Shifting Bottleneck Routine* (SBR). Porém os resultados das regras de prioridade não foram relevantes (ver tabela 7). Desta forma apresentaremos apenas os melhores resultados, que foram obtidos através da aplicação da heurística *General Shifting Bottleneck Routine* (SBR).

6.2.1 Heurística *General Shifting Bottleneck Routine*

Usando a configuração HFS os melhores resultados de desempenho foram obtidos usando a heurística SBR. Neste cenário obtivemos um *makespan* de 1317 minutos.

No aquecedor de moldes 1 o C_{max} foi de 1150 min. (Figura 9) e a taxa de utilização do sistema foi de 87%. No aquecedor de moldes 2 o C_{max} teve um valor de 1150 min. (Figura 9), e a taxa de utilização teve um valor de 87% também. No aquecedor de moldes 3 o C_{max} foi de 1040 min. (Figura 9), e a taxa de utilização teve um valor de 79%.

A prensa 1 obteve um C_{max} de 1308 min. (Figura 9), e uma taxa de utilização de 80%. Na prensa 2 o C_{max} foi de 1317 min. (Figura 9), e a taxa de utilização de 73%. A prensa 3 o C_{max} foi de 1300 min. (Figura 9), e a taxa de utilização de 80%.

Neste cenário o *makespan* médio foi de 1211 min., sendo que taxa de utilização do sistema foi de 81%.

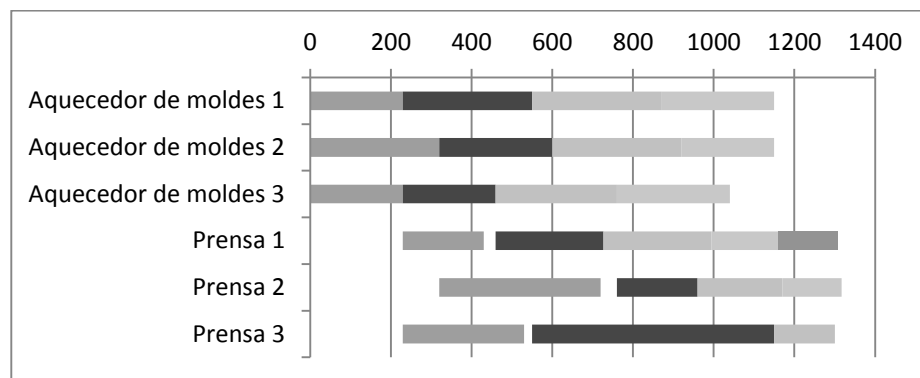


Figura 9. Diagrama de Gantt – *General Shifting Bottleneck routine* (HFS)

6.3 Resultados Globais e Discussão

Considerando os resultados obtidos (Figura 13, Figura 14, Tabela 7), para ambos os cenários PFS e HFS, podemos concluir que o cenário com melhor C_{max} foi o HFS que atingiu um valor de 1317 min. (Figura 9). Este cenário não é apenas aquele com melhor desempenho a nível de *makespan* quando comparado com os outros cenários, é também o cenário com os melhores valores para todas as outras medidas de desempenho consideradas, ou seja, a taxa de utilização global e o *makespan* médio (Figura 10).

Os resultados podem ser explicados pelo facto de que o HFS consistir numa generalização do PFS com uma configuração mais flexível, permitindo um melhor uso dos recursos disponíveis, facto que tem um impacto positivo na melhoria do *makespan* e na correspondente utilização do sistema.

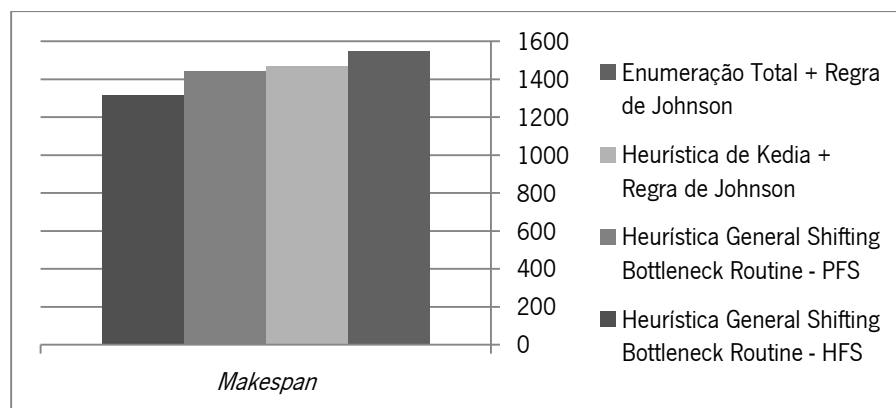


Figura 10. Comparação do *Makespan* dos diferentes cenários

6.4 Generalização

Como os resultados mostraram, a heurística *general shifting bottleneck routine* (SBR) disponível através do sistema de escalonamento LEKIN®, foi a estratégia que permitiu obter melhores resultados para ambas as configurações de produção (PFS e HFS), para o problema industrial

considerado. Contudo, pretendemos generalizar o estudo por forma a tentar chegar a conclusões mais gerais para as duas configurações de produção visadas, fazendo variar os tempos de transporte. Sendo assim, no ambiente PFS foi usada a heurística de Kedia porque esta distribui equitativamente as tarefas, e no ambiente HFS utilizou-se a heurística *general shifting bottleneck routine* (SBR) porque foi a estratégia onde se verificaram melhores resultados. Foram usados 30 conjuntos de tarefas para efetuar a generalização do problema, em cada cenário, por forma a obter uma distribuição normal dos resultados. O conjunto de tarefas menor tem 4 trabalhos para executar, sendo que o conjunto de tarefas maior tem dimensão de 33 trabalhos para concluir, tendo então os 30 conjuntos de tarefas considerados variado de dimensão 4 até 33 trabalhos. A generalização foi efetuada em dois passos, usando duas restrições distintas. Numa situação um tempo de transporte entre máquinas de 1 minuto foi considerado, e na outra um tempo de transporte entre máquinas de 2 minutos.

Para avaliar a significância dos resultados obtidos, calculamos a diferença de *makespan* entre os conjuntos de tarefas com a mesma extensão (portanto, com o mesmo número de trabalhos). Foi calculada a diferença do *makespan* entre cenários de mesma dimensão, e calculada a diferença (distância) para o melhor cenário da respetiva dimensão. Seguidamente usamos o Teste T (Tabelas 5 e 6) com as hipóteses:

$$H_0: \mu_{HFS} = \mu_{PFS}$$

$$H_1: \mu_{HFS} \neq \mu_{PFS}$$

6.4.1 Tempo de transporte de 1 minuto

Para este tempo de transporte (de 1 minuto), não é possível assumir a igualdade das médias, como sugere a hipótese nula H_0 , que considera a inexistência de diferença entre o desempenho das duas configurações, sendo rejeitada a hipótese nula com 95% de nível de confiança ($p\text{-value} < \alpha$). Como se pode verificar na tabela 5.

Tabela 5. Teste T de Student para tempos de transporte entre máquinas de 1 minuto

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Distância para o melhor dos 2	Equal variances assumed	36,47	,000	-3,36	58	,001	-,500	,149	-,798	-,202
	Equal variances not assumed			-3,36	35,16	,002	-,500	,149	-,802	-,198

Usando a análise estatística para este caso, a HFS continua com um desempenho melhor do que a PFS, como mostra a Figura 11.

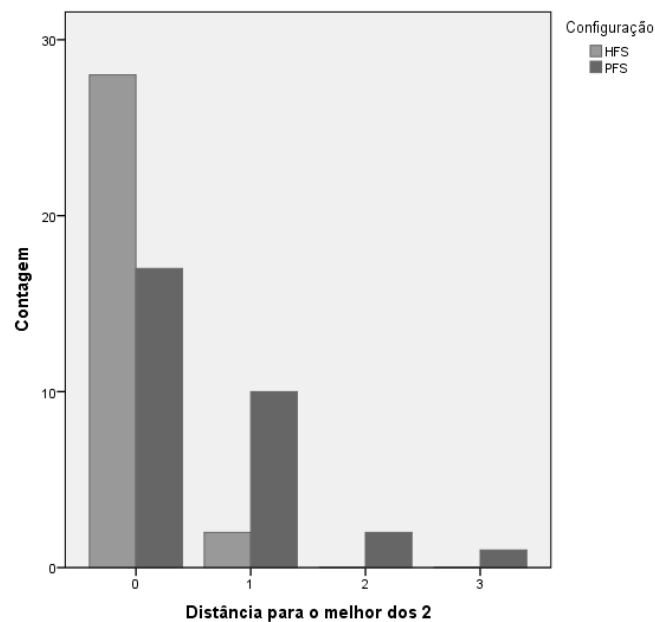


Figura 11. Diferenças de makespan entre os dois cenários - Tempos de transporte entre máquinas de 1 minuto

6.4.2 Tempo de transporte de 2 minutos

Analisando os resultados, para este tempo de transporte (de 2 minutos), não é possível assumir a igualdade das médias, como sugere a hipótese nula H_0 , que considera a inexistência de diferença entre o desempenho das duas configurações, sendo rejeitada a hipótese nula com 95% de nível de confiança ($p\text{-value} < \alpha$).

Tabela 6. Teste T de Student para tempos de transporte entre máquinas de 2 minutos

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
Distância para o melhor dos 2	Equal variances assumed	14,64	,000	3,63	58	,001	,500	,137	,225	,775
	Equal variances not assumed			3,63	52,15	,001	,500	,137	,224	,776

Para este caso a PFS tem um desempenho melhor do que a HFS, como se pode verificar através da análise estatística (Figura 12).

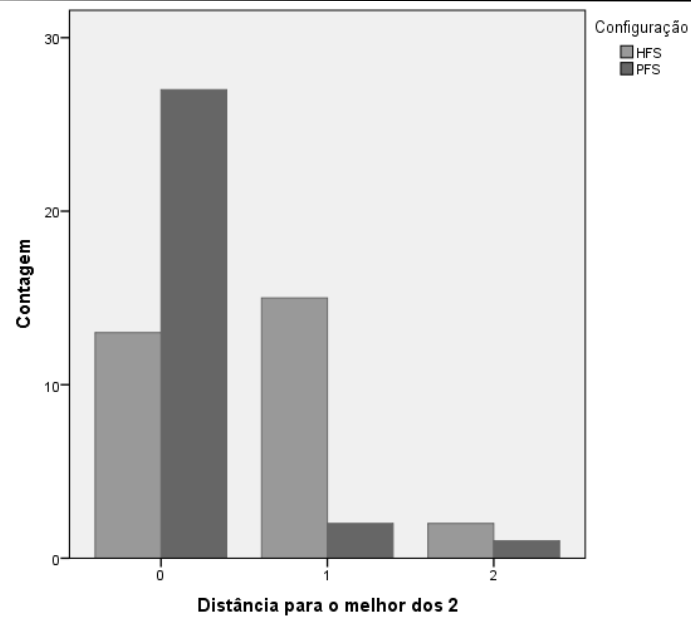


Figura 12. Diferenças de makespan entre os dois cenários - Tempos de transporte entre máquinas de 2 minutos

6.5 Resumo

Os resultados obtidos do escalonamento na configuração HFS foram significativamente melhores do que os resultados obtidos na configuração PFS, tanto em termos de *makespan* como de utilização geral do sistema.

Na configuração PFS foram aplicadas três estratégias para o escalonamento, nomeadamente o procedimento de enumeração total combinado com a regra de Johnson, a heurística de Kedia combinada com a regra de Johnson, e a heurística *general shifting bottleneck routine* disponível no sistema de escalonamento LEKIN®. Neste cenário os resultados obtidos foram consideravelmente melhores quando utilizamos a heurística SBR, tendo o *makespan* atingido um valor de 1444 minutos e a taxa de utilização teve um valor de 74%.

Para efetuar o escalonamento na configuração HFS foram utilizados os métodos FCFS, LPT, SPT e a heurística *general shifting bottleneck routine* disponível no sistema de escalonamento LEKIN®

de Pinedo (2012). O melhor resultado obtido neste ambiente de produção foi obtido usando a heurística *general shifting bottleneck routine* (SBR), tendo o *makespan* um valor de 1317 minutos e a taxa de utilização do sistema atingido o valor de 81%.

Assim a heurística SBR foi consistente em obter os melhores resultados em ambos os cenários. Os melhores resultados foram obtidos no cenário HFS utilizando a heurística *general shifting bottleneck routine* (SBR).

Por forma a chegar a conclusões mais gerais selecionamos um método para cada uma das configurações de produção visadas. Para a configuração PFS utilizamos a heurística de Kedia combinada com o algoritmo de Johnson e para a configuração HFS utilizamos a heurística *general shifting bottleneck routine*. Foram utilizados 30 conjuntos de tarefas e adicionadas restrições de tempos de transporte, uma de um minuto, e outra de dois minutos.

Seguidamente foi realizada, com recurso ao SPSS, o *Teste T* por forma a verificar se existem diferenças significativas de desempenho entre os cenários, considerando as restrições impostas. Foi possível verificar que utilizando qualquer das restrições impostas não é possível assumir a inexistência de diferença de desempenho entre as duas configurações.

Tendo em conta as medidas de desempenho e utilizando os tempos de transporte entre as máquinas de 1 minuto, a configuração HFS continua a obter os melhores resultados, porém se forem considerados tempos de transporte entre as máquinas de 2 minutos, a PFS torna-se a configuração com melhores resultados para o problema considerado.

7. CONCLUSÃO

Neste trabalho foi realizado um estudo industrial de análise de configurações de sistemas de produção, considerando diferentes combinações de métodos de escalonamento para a resolução de um problema real da indústria de componentes automóveis, fabricados em duas fases, i.e., integrando aquecedores de moldes numa primeira fase e prensas numa segunda fase. Foram testadas duas configurações ou ambientes de produção que se incluem na categoria de Sistemas de Produção Orientados ao Produto (SPOP), nomeadamente, linhas de fabrico paralelas (PFS) e uma linha de fabrico híbrida (HFS). A configuração PFS foi configurada com três linhas de fabrico idênticas e independentes.

Este estudo mostrou que a configuração HFS tem um melhor desempenho que a configuração PFS, tanto em termos de *makespan* como de utilização geral do sistema. Este resultado foi obtido usando a heurística *general shifting bottleneck routine* (SBR) disponível no sistema de escalonamento LEKIN® de Pinedo (2012). A mesma heurística foi consistente em obter os melhores resultados, não só para a configuração HFS, mas também para a configuração PFS. Nesta configuração (PFS) duas outras estratégias de programação foram testadas, nomeadamente o procedimento de enumeração total combinado com a regra de Johnson e uma adaptação da heurística de Kedia combinada com a regra de Johnson. Os resultados obtidos com estas duas estratégias em configuração PFS foram consideravelmente piores do que os resultados obtidos usando a heurística SBR para o problema industrial analisado.

A principal conclusão que pode ser extraída deste estudo é a recomendação da utilização da configuração HFS para o ambiente industrial estudado, e que o escalonamento deve ser efetuado com a heurística SBR ou algoritmos mais eficientes, se disponíveis.

De acordo com os resultados obtidos, considerando não existirem restrições no sistema, a diferença de desempenho entre HFS e PFS é tão expressiva que poderemos dizer que existe uma forte probabilidade em outros ambientes de produção reais, idênticos ao estudado, se estas configurações forem consideradas para o funcionamento do sistema, que o sistema de produção

deve ser operado, preferencialmente, na forma de configuração HFS recorrendo ao uso de algoritmos de escalonamento eficientes, nomeadamente o SBR. Contudo, se forem consideradas algumas restrições os resultados podem ser consideravelmente diferentes. Por exemplo, considerando tempos de transporte entre as máquinas de 1 minuto, a configuração HFS continua a obter os melhores resultados, porém se forem considerados tempos de transporte entre as máquinas de 2 minutos, a PFS torna-se a configuração com melhores resultados para o problema considerado.

Como trabalho futuro, está a ser elaborada uma versão estendida do artigo (Costa, Varela, & Carmo-Silva, 2014) para publicação numa revista internacional. Além disso propõe-se a extensão deste estudo, aplicando outros métodos de escalonamento incluindo regras *EDD*, assim como, algumas meta-heurísticas, nomeadamente métodos baseados em *simulated annealing*, técnicas *tabu search* e/ou baseados em algoritmos genéticos.

REFERÊNCIAS BIBLIOGRÁFICAS

- Al-Salem, A. (2004). A heuristic to minimize makespan in proportional parallel flow shops. *International Journal of Computing & Information Sciences*, 2(2), 98.
- Alves, A. C. (1999). Metodologia para a concepção de sistemas de produção orientados ao produto.
- Andresen, M., Bräsel, H., Engelhardt, F., & Werner, F. (2010). *LiSA-a Library of Scheduling Algorithms: Handbook for Version 3.0*: Univ., Fak. für Mathematik.
- Aneke, N., & Carrie, A. (1986). A design technique for the layout of multi-product flowlines. *International Journal of Production Research*, 24(3), 471-481.
- Baker, K. R., & Baker, K. R. (1974). *Introduction to sequencing and scheduling* (Vol. 15): Wiley New York.
- Blazewicz, J. (2001). *Scheduling computer and manufacturing processes*: Springer.
- Brucker, P., & Brucker, P. (2007). *Scheduling algorithms* (Vol. 3): Springer.
- Cao, D., & Chen, M. (2003). Parallel flowshop scheduling using Tabu search. *International journal of production research*, 41(13), 3059-3073.
- Carvalho, D. (2000). Textos sobre Gestão Industrial - Programação da Produção. 2014, from http://pessoais.dps.uminho.pt/jdac/apontamentos/Cap03_Program.pdf
- Cheng, J., Karuno, Y., & Kise, H. (2001). A shifting bottleneck approach for a parallel-machine flowshop scheduling problem. *Journal of the Operations Research Society of Japan-Keiei Kagaku*, 44(2), 140-156.
- Cheng, T., & Sin, C. (1990). A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research*, 47(3), 271-292.
- Conway, R. W., Maxwell, W. L., & Miller, L. W. (2012). *Theory of scheduling*: Courier Dover Publications.
- Costa, J., & Varela, L. (2012). Decision System for Supporting the Implementation of a Manufacturing Section on an Automotive Factory in Portugal.
- Costa, N., Varela, M., & Carmo-Silva, S. (2014). *Scheduling in product oriented manufacturing systems*. Paper presented at the Nature and Biologically Inspired Computing (NaBIC), 2014 Sixth World Congress on.
- Garey, M. R., Graham, R. L., & Johnson, D. S. (1976). *Some NP-complete geometric problems*. Paper presented at the Proceedings of the eighth annual ACM symposium on Theory of computing.
- Hu, T. C. (1961). Parallel sequencing and assembly line problems. *Operations research*, 9(6), 841-848.
- Johnson, D. S., & Garey, M. R. (1979). Computers and Intractability-A Guide to the Theory of NP-Completeness. *Freeman&Co, San Francisco*.
- Johnson, S. M. (1954). Optimal two-and three-stage production schedules with setup times included. *Naval research logistics quarterly*, 1(1), 61-68.
- Lenstra, J. K., Shmoys, D. B., & Tardos, É. (1990). Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46(1-3), 259-271.
- Linn, R., & Zhang, W. (1999). Hybrid flow shop scheduling: a survey. *Computers & Industrial Engineering*, 37(1), 57-61.
- McNaughton, R. (1959). Scheduling with deadlines and loss functions. *Management Science*, 6(1), 1-12.

-
- Min, L., & Cheng, W. (1999). A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines. *Artificial Intelligence in Engineering*, 13(4), 399-403.
- Naderi, B., Ruiz, R., & Zandieh, M. (2010). Algorithms for a realistic variant of flowshop scheduling. *Comput. Oper. Res.*, 37(2), 236-246. doi: 10.1016/j.cor.2009.04.017
- Nowicki, E., & Smutnicki, C. (1998). The flow shop with parallel machines: A tabu search approach. *European Journal of Operational Research*, 106(2), 226-253.
- Pinedo, M. (2005). Planning and scheduling in manufacturing and services.
- Pinedo, M. (2012). *Scheduling: theory, algorithms, and systems*: Springer.
- Quadt, D., & Kuhn, H. (2007). A taxonomy of flexible flow line scheduling procedures. *European Journal of Operational Research*, 178(3), 686-698.
- Ribas, I., Leisten, R., & Framiñan, J. M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective (Vol. 37, pp. 1439-1454).
- Ribeiro, R. A., & Varela, L. R. (2003). Fuzzy optimization using simulated annealing: An example set *Fuzzy Sets Based Heuristics for Optimization* (pp. 159-180): Springer.
- Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, 165(2), 479-494. doi: 10.1016/j.ejor.2004.04.017
- Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205(1), 1-18. doi: <http://dx.doi.org/10.1016/j.ejor.2009.09.024>
- Salvador, M. S. (1973). *A solution to a special class of flow shop scheduling problems*. Paper presented at the Symposium on the theory of scheduling and its applications.
- Samarghandi, H., & ElMekkawy, T. Y. (2012). A meta-heuristic approach for solving the no-wait flow-shop problem. *International Journal of Production Research*, 50(24), 7313-7326. doi: 10.1080/00207543.2011.648277
- Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research Methods for Business Students* (P. Education Ed. Fifth edition ed.).
- Silva, S. C. (2010). Textos de Gestão da Produção V01-10. Departamento de Produção e Sistemas, Escola de Engenharia, Universidade do Minho.
- Silva, S. C., & Alves, A. C. (2006). Detailed design of product oriented manufacturing systems.
- Vairaktarakis, G., & Elhafi, M. (2000). The use of flowlines to simplify routing complexity in two-stage flowshops. *IIE Transactions*, 32(8), 687-699.
- Varela, M. L. R. (2007). Uma contribuição para o escalonamento da produção baseado em métodos globalmente distribuídos.
- Varela, M. L. R., Barbosa, R., & Putnik, G. D. (2012). Experimental Platform for Collaborative Inter and Intra Cellular Fuzzy Scheduling in an Ubiquitous Manufacturing System *Virtual and Networked Organizations, Emergent Technologies and Tools* (pp. 220-229): Springer.
- Varela, M. L. R., & do Carmo Silva, S. (2008). An ontology for a model of manufacturing scheduling problems to be solved on the web *Innovation in Manufacturing Networks* (pp. 197-204): Springer.
- Varela, M. L. R., Putnik, G. D., & Cruz-Cunha, M. M. (2012). Web-based Technologies Integration for Distributed Manufacturing Scheduling in a Virtual Enterprise. *International Journal of Web Portals (IJWP)*, 4(2), 19-34.
- Vieira, G., Varela, M. L. R., & Putnik, G. D. (2012). Technologies integration for distributed manufacturing scheduling in a virtual enterprise *Virtual and Networked Organizations, Emergent Technologies and Tools* (pp. 337-347): Springer.
-

-
- Vignier, A., Billaut, J., & Proust, C. (1999). Hybrid flowshop scheduling problems: State of the art. *Rairo-Recherche Operationnelle-Operations Research*, 33(2), 117-183.
- Vollmann, T. E., William, L. B., & Whybark, D. C. (1992). *Manufacturing planning and control systems* (third edition ed.): Richard D. Irwin, Inc. .
- Zhang, X., & van de Velde, S. (2012). Approximation algorithms for the parallel flow shop problem. *European Journal of Operational Research*, 216(3), 544-552. doi: <http://dx.doi.org/10.1016/j.ejor.2011.08.007>

ANEXO I – RESULTADOS

Tabela 7. Resultados obtidos com os métodos implementados

	Método	Makespan (min.)	Taxa de Utilização do Sistema
Configuração PFS	Enumeração Total +Regra de Johnson	1550	69%
	Heurística de Kedia + Regra de Johnson	1467	73%
	Heurística <i>General Shifting Bottleneck Routine</i>	1444	74%
Configuração HFS	Heurística <i>General Shifting Bottleneck Routine</i>	1317	81%
	FCFS	1437	74%
	LPT	1394	76%
	SPT	1750	61%

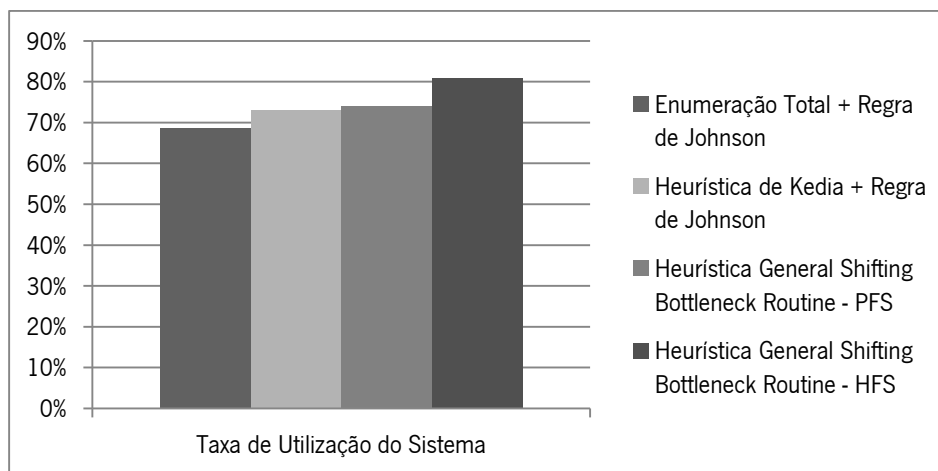


Figura 13. Comparação da taxa de utilização do sistema dos diferentes cenários

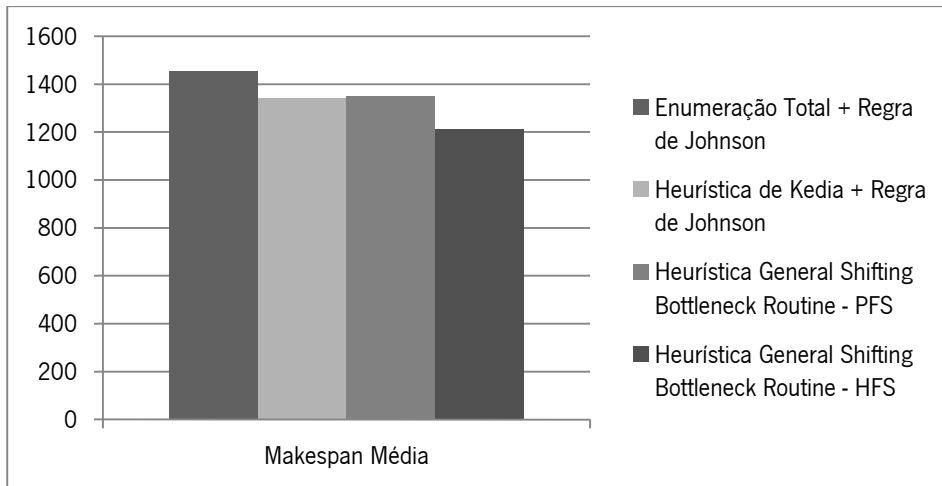


Figura 14. Comparação das médias do Makespan dos diferentes cenários