

**Faculdade de Engenharia da Universidade do Porto**



**FEUP**

**Um sistema flexível para o escalonamento de operações industriais**

(A flexible scheduling system for industrial operation)

João Faria Amorim

VERSÃO FINAL

Dissertação/Relatório de Projecto realizada(o) no âmbito do  
Mestrado Integrado em Engenharia Electrotécnica e de Computadores  
Major Automação

Orientador: Professor Jorge Pinho de Sousa

Co-orientador: Eng. Luís Guardão

Julho de 2009



# Resumo

Nesta dissertação descreve-se uma abordagem para resolver problemas de escalonamento de Job-Shop, numa perspectiva multi-critério. Na prática, os problemas de escalonamento (scheduling) são de grande complexidade, pelo seu carácter combinatório e pela necessidade de considerar objectivos múltiplos, em geral conflituosos.

Resulta daqui a necessidade de dispor de Sistemas de Apoio à Decisão que permitam ao planeador avaliar várias soluções alternativas e escolher uma delas de maneira mais fundamentada.

Existem diversos sistemas disponíveis comercialmente para apoiar o escalonamento de operações. Neste trabalho é feita uma referência especial a um destes sistemas no qual se enquadram os desenvolvimentos deste projecto.

Foi desenvolvida uma aplicação informática, com a filosofia de um Sistema de Apoio à Decisão, baseada numa abordagem com duas fases fundamentais, Sequenciar e Escalonar as tarefas.

O sequenciamento de tarefas é feito através de regras de prioridade do tipo das usadas nos problemas de uma única máquina. Com base nesse sequenciamento, as operações das tarefas são escalonadas nas máquinas, através de um algoritmo simples e eficiente.

A solução inicial é depois sujeita a um procedimento sistemático de melhoramento, baseado na troca das tarefas, do qual resulta um conjunto de soluções não dominadas. Estas soluções são avaliadas através de vários indicadores e apresentadas ao planeador para que ele escolha uma das soluções.

A abordagem é ilustrada por meio de diversos exemplos. Esses exemplos têm graus de complexidade diversos, permitindo mostrar o potencial desta abordagem, em particular se for integrada num sistema mais vasto de apoio à decisão.



# Abstract

This dissertation describes an approach to solve the Job-Shop scheduling problem in a multi-criteria perspective. In practice scheduling problems have a great complexity due to their combinatorial nature and to the existence of multiple, often conflicting objectives.

There is therefore a need for more advanced Decision Support Systems that should allow the planner to evaluate several alternative solutions and eventually choose one of them in a sound way.

Commercially there are several systems available to support operations scheduling. In this work a special reference is made to one such system that has been used to support the project developments and assess its achievements.

A software application was therefore developed, following the philosophy of a Decision Support System, and based on one approach with two fundamental phases, sequencing and scheduling the jobs.

Sequencing the jobs is performed through a set of “dispatching rules” similar to those used in single machine problems. Based on the resulting sequence, the operations of the jobs are scheduled in the machines by a simple and efficient process.

The initial solution is then subject to a systematic improvement procedure based on “swap” movements, from which a set of non-dominated solutions results. These solutions are evaluated through several indicators and they are presented to the planner so that he/she can choose one of those solutions.

Finally the approach is illustrated by using some examples. These examples have different degrees of complexity, allowing us to show the potential of this approach, in particularly if integrated in a broader decision support system.



# Índice

Resumo .....	iii
Abstract .....	v
Índice .....	vii
Lista de figuras .....	xi
Lista de tabelas .....	xiii
Abreviaturas e Símbolos .....	xv
<b>Capítulo 1 .....</b>	<b>1</b>
Introdução.....	1
1.1. Instituição de acolhimento .....	1
1.2. Objectivos e âmbito do projecto .....	2
1.3. Estrutura do relatório .....	3
<b>Capítulo 2 .....</b>	<b>5</b>
Job-Shop scheduling - contexto .....	5
2.2. Sistemas de Produção .....	5
2.3. Planear a Produção .....	6
2.3.1. Planeamento hierárquico.....	6
2.3.2. <i>Materials Requirements Planning</i> - MRP .....	9
2.3.3. <i>Manufacturing Resources Planning</i> - MRP II .....	9
2.3.4. Planeamento das capacidades (CRP -Capacity Requirements Planning) .....	10
2.4. Modelos de organização da produção.....	10
2.4.1. Produção contínua - Flow-Shop .....	11
2.4.2. Produção descontínua - <i>Job-Shop</i> .....	12
2.5. Produção para stock, por encomenda e pontos de desacoplamento.....	12
2.5.1. Produção para stock.....	13
2.5.2. Montagem por encomenda .....	13

2.5.3. Produção por encomenda .....	14
2.5.4. Engenharia por encomenda .....	14
2.6. Escalonamento - <i>Scheduling</i> .....	14
2.6.1. Problemas com uma única máquina .....	16
2.6.2. Problemas com máquinas paralelas .....	17
2.6.3. Problemas multi-operação .....	17
2.7. Escalonamento de uma única máquina ( <i>single-machine</i> ) .....	19
2.7.1. Critérios de optimalidade .....	20
2.8. O problema em estudo .....	21
<b>Capítulo 3 .....</b>	<b>23</b>
Enquadramento do projecto .....	23
3.1. Exemplos de escalonadores .....	23
3.1.1. Preactor® .....	23
3.1.2. LEKIN® .....	24
3.1.3. Lacunas .....	26
3.2. Iزارo Grey .....	26
3.2.1. Operar o Iزارo Grey .....	28
3.2.2. Vantagens da solução Iزارo Grey .....	29
3.2.3. Aplicações .....	29
3.3. Contextualização do trabalho .....	31
<b>Capítulo 4 .....</b>	<b>33</b>
A abordagem “sequenciar/escalonar” para o problema de Job-Shop .....	33
4.1. Principais técnicas utilizadas .....	33
4.1.1. Regras de Prioridade ( <i>priority rules</i> ) .....	34
4.1.2. Heurísticas de melhoramento .....	34
4.1.3. Multi-objectivo .....	35
4.2. Abordagem adoptada .....	36
4.2.1. Leitura de Dados ( <i>Read Data</i> ) .....	36
4.2.2. Sequenciamento ( <i>Sort</i> ) .....	38
4.2.3. Escalonamento ( <i>Scheduling</i> ) .....	39
4.2.4. Diagrama de Gantt .....	40
4.2.5. Avaliação ( <i>Evaluate</i> ) .....	41
4.2.6. Trocas ( <i>Swap</i> ) .....	42
4.3. Exemplo .....	42
4.4. Aplicação desenvolvida .....	45
4.5. Exemplo 1 .....	52
4.6. Exemplo 2 .....	60
<b>Capítulo 5 .....</b>	<b>63</b>

Conclusões e desenvolvimentos futuros .....	63
<b>Referências.....</b>	<b>65</b>



# Lista de figuras

Figura 1: Níveis de Planeamento.....	7
Figura 2: Relação entre o nível de detalhe e o horizonte temporal de cada nível.....	8
Figura 3: Esquema geral de um exemplo de produção em Flow-Shop.....	11
Figura 4: Exemplo de uma produção Job-Shop .....	12
Figura 5: Modelos de Negócio.....	13
Figura 6: Modelo "Flow-Shop" básico .....	18
Figura 7: Exemplos de interfaces do Preactor .....	24
Figura 8: Menu inicial do LEKIN® .....	24
Figura 9: Vista geral de um exemplo de utilização do escalonador LEKIN®.....	25
Figura 10: Arquitectura do motor de escalonamento da ferramenta Iزارo Grey. (Pinho de Sousa e Guardão, 2008) .....	27
Figura 11: Diagrama de Gantt resultante do escalonamento utilizando o Iزارo Grey .....	28
Figura 12: Algumas referências de empresas que utilizam a aplicação Iزارo APS.....	30
Figura 13: Estrutura da aplicação.....	37
Figura 14: Esquema da relação entre as tabelas da Base de Dados usado.....	38
Figura 15: Exemplo de um Diagrama de Gantt .....	41
Figura 16: Instante Inicial .....	43

Figura 17: Primeira operação da primeira tarefa escalonada .....	44
Figura 18: Resultado do escalonamento da primeira tarefa .....	44
Figura 19: Primeira operação da tarefa 3 (vermelho).....	44
Figura 20: Espera da 2ª operação da tarefa 3 (vermelho).....	45
Figura 21: Escalonamento final .....	45
Figura 22: Vista da página inicial da aplicação desenvolvida .....	46
Figura 23: Janela de aviso no caso de ser seleccionado apenas um objectivo. ....	46
Figura 24: Página inicial onde aparece a atribuição de pesos aos objectivos .....	47
Figura 25: Página principal da aplicação .....	48
Figura 26: Vista geral da janela com um exemplo de um conjunto de soluções .....	49
Figura 27: Vista detalhada de uma das sequências do conjunto de soluções resultante .....	50
Figura 28: Resultado do escalonamento das tarefas apresentadas na Tabela 3.....	51
Figura 29: Diagrama de Gantt relativo ao exemplo 1 .....	54
Figura 30: Soluções não dominadas (com 2 tarefas em atraso) .....	59
Figura 31: Resultados obtidos para uma instância retirada de um caso real .....	60
Figura 32: Gráfico das soluções obtidas .....	62

## Lista de tabelas

Tabela 1: Tempos de processamento .....	43
Tabela 2: Máquinas onde são realizadas as operações .....	43
Tabela 3: Dados .....	50
Tabela 4: Dados relativos ao primeiro exemplo com 6 tarefas .....	52
Tabela 5: Extracto da matriz de setups para o exemplo 1 .....	53
Tabela 6: Ocupação dos recursos no exemplo 1 .....	54
Tabela 7: Avaliação dos resultados .....	55
Tabela 8: Dados relativos ao início e fim das operações .....	56
Tabela 9: Transcrição dos dados relativos à utilização dos recursos .....	57
Tabela 10: Primeiro conjunto de soluções .....	57
Tabela 11: Segundo conjunto de soluções .....	58
Tabela 12: Resultado do processo de trocas para o segundo exemplo .....	61



# Abreviaturas e Símbolos

Lista de abreviaturas (ordenadas por ordem alfabética)

DEEC	Departamento de Engenharia Electrotécnica e de Computadores
EDD	<i>Earliest Due Date</i>
FEUP	Faculdade de Engenharia da Universidade do Porto
INESC Porto	Instituto de Engenharia de Sistemas e Computadores do Porto
MIEEC	Mestrado Integrado em Engenharia Electrotécnica e de Computadores
OFs	Ordens de Fabrico
SAD	Sistema de Apoio à Decisão
SPT	Shortest Processing Time
VB	Visual Basic
WIP	Work-In-Process



# Capítulo 1

## Introdução

Neste capítulo é feito o enquadramento do projecto que deu origem a esta dissertação e são definidos os objectivos e o âmbito do trabalho realizado.

É ainda apresentada a estrutura deste documento.

### 1.1. Instituição de acolhimento

Este projecto foi desenvolvido no INESC Porto. O INESC Porto - Instituto de Engenharia de Sistemas e Computadores do Porto é uma associação privada sem fins lucrativos reconhecida como instituição de utilidade pública, tendo adquirido em 2002 o estatuto de Laboratório Associado.

Desenvolve actividades de investigação e desenvolvimento, consultoria, formação avançada e transferência de tecnologia nas áreas de Telecomunicações e Multimédia, Sistemas de Energia, Sistemas de Produção, Sistemas de Informação e Comunicação e Optoelectrónica (<http://www.inescporto.pt/>).

O INESC Porto é uma instituição criada para constituir uma interface entre o mundo académico e o mundo empresarial da indústria e dos serviços, bem como a administração pública, no âmbito das Tecnologias de Informação, Telecomunicações e Electrónica, dedicando-se a actividades de investigação científica e desenvolvimento tecnológico, transferência de tecnologia, consultoria e formação avançada.

O Instituto procura pautar a sua acção por critérios de inovação, de internacionalização e de impacto no tecido económico e social, sobretudo pelo estabelecimento de um conjunto de parcerias estratégicas que garantam a sua estabilidade institucional e sustentabilidade económica.

O INESC Porto está dividido em diversas unidades de acordo com as suas várias áreas de intervenção, que são:

- Unidade de Engenharia e Sistemas de Produção (UESP);
- Unidade de Inovação e Transferência de Tecnologia (UITT);
- Unidade de Optoelectrónica e Sistemas Electrónicos (UOSE);
- Unidade de Sistemas de Energia (USE);
- Unidade de Sistemas de informação e Comunicação (USIC);
- Unidade de Telecomunicações e Multimédia (UTM).

Este projecto foi desenvolvido na UESP, a Unidade de Engenharia de Sistemas de Produção, que desenvolve a sua actividade nas áreas de Produção (sistemas de informação avançados de apoio à gestão industrial, gestão da qualidade, gestão da manutenção; sistemas de planeamento e controlo da produção; racionalização e optimização dos processos produtivos; automação; sistemas de apoio à decisão) e Logística (sistemas de gestão de cadeias de fornecimento; planeamento de sistemas logísticos; integração e optimização de estruturas logísticas).

A unidade dedica-se ainda ao Negócio Electrónico, Tele-trabalho, Negócio Electrónico entre Empresas e Engenharia Empresarial (análise e optimização de processos no âmbito da gestão industrial; gestão de projectos de inovação empresarial; análise de requisitos técnicos e organizacionais; desenho e análise de redes de cooperação empresarial) (<http://www.inescporto.pt/>).

### 1.2. Objectivos e âmbito do projecto

Em Gestão de Operações, problemas de escalonamento complexos têm assumido cada vez mais importância, em particular quando se considera um sistema de planeamento mais abrangente. Contudo, versões mais elementares destes problemas (como o tradicional modelo Job-Shop) são em geral incapazes de modelar a complexidade de situações reais e a sua natureza multi-objectivo.

Neste trabalho, procuramos lidar com os requisitos de flexibilidade dos problemas reais de escalonamento. Assim o trabalho desenvolvido irá basicamente consistir na modelação multi-objectivo dos problemas de Job-Shop tradicionais, e ainda no design de algoritmos para a abordagem a esses problemas. Este trabalho enquadra-se num projecto mais vasto em que a UESP está envolvida, no desenvolvimento de ferramentas de escalonamento que interagem

com sistemas mais alargados de gestão (do tipo ERP - Enterprise Resource Planning) usados em ambientes industriais.

No contexto deste trabalho, o INESC Porto tem como parceiro a Softi9, empresa portuguesa pertencente ao Grupo I68. A aplicação informática que enquadra os desenvolvimentos deste projecto é designada por Iزارo Grey, e é uma ferramenta de optimização de produção industrial.

A aplicação, que sequencia e optimiza a produção industrial, foi desenvolvida pelo INESC Porto. Esta aplicação permite programar as ordens de fabrico em função dos critérios específicos de cada empresa. No decorrer do desenvolvimento do Iزارo Grey, foram sendo acrescentadas diversas funcionalidades ao escalonador, (tratando, por exemplo, das mudanças de turno, fins de semana, etc). Com estas restrições todas, tornou-se difícil proceder a testes sobre os aspectos mais essenciais da abordagem adoptada, baseada em duas fases (primeiro, Sequenciar e depois, Escalonar as tarefas).

Como tal, surgiu a necessidade de desenvolver um SAD (Sistema de Apoio à Decisão) que, utilizando a abordagem “Sequenciar/Escalonar”, não considerasse as restrições adicionais que existem no Iزارo Grey, permitindo assim avaliar de uma forma dirigida as especificidades dessa abordagem.

Para além disso, o sistema criado tem a capacidade de, após ser gerada uma primeira solução, desencadear um processo de melhorias, do qual resulta um conjunto de soluções que se espera correspondam a bons compromissos entre múltiplos objectivos.

### **1.3. Estrutura do relatório**

Para além deste capítulo introdutório, esta dissertação é constituído pelos capítulos que a seguir se referem.

No Capítulo 2 é feita uma breve introdução aos temas abordados neste projecto, nomeadamente o Job-Shop e o Scheduling. Ainda neste capítulo é definido o âmbito da aplicação desenvolvida, bem como o projecto onde esta se insere.

No Capítulo 3 é descrita a estruturação adoptada para o problema e para o qual foi desenvolvida esta aplicação. A abordagem adoptada reflecte o trabalho e as discussões realizadas no início do projecto.

Com o Capítulo 4 pretende-se mostrar a interface da aplicação desenvolvida, bem como fornecer uma descrição de como o planeador poderá utilizar as funcionalidades disponibilizadas. É ainda ilustrado o funcionamento da aplicação desenvolvida neste trabalho. e apresentados os resultados obtidos num conjunto de pequenos exemplos.

## 4 Introdução

Este documento termina com um capítulo de conclusões e desenvolvimentos futuros, onde se avaliam os resultados obtidos e se dão indicações sobre possíveis extensões ao trabalho realizado.

# Capítulo 2

## Job-Shop scheduling - contexto

Neste capítulo iremos fazer uma breve introdução aos vários sistemas de produção e ao porquê da necessidade da função de escalonamento. Para tal, vão-se apresentar diversos sistemas de produção, bem como evoluções que foram sofrendo ao longo dos tempos.

Irá ser feita a apresentação dos modelos de uma única máquina, e em que medida a abordagem desse tipo de problemas poderá ser utilizada nos problemas de escalonamento de Job-Shop.

### 2.2. Sistemas de Produção

No passado, a preocupação das empresas era produzir em massa, em elevadas quantidades, com o pensamento de que os produtos iriam ser vendidos mais dia, menos dia. Nessa altura, a produção em série era utilizada na maioria das empresas (Sistema de “Push”).

No entanto os mercados foram mudando, a concorrência foi aumentando, o que fez com que as preocupações das empresas se fossem alterando. Daí resultou a necessidade de reduzir os lotes de fabrico de modo a aumentar a personalização dos produtos. É então que surge um novo paradigma de produção denominado “costumização em massa” (mass customization), associado a uma produção unitária (Sistema “Pull”), em que é um pedido de um cliente que desencadeia o início de fabrico de um produto. (Chase, Jacobs, Aquilano 2006)

Com isto, o panorama actual a nível empresarial pode ser definido da seguinte forma:

- elevada diversidade e complexidade de produtos;
- elevada personalização de produtos;

- procura muito variável e imprevisível;
- redução dos tempos de entrega;
- ciclo de vida dos produtos muito reduzido.

Nestas circunstâncias as empresas vêem-se forçadas a reestruturar toda a sua produção para poderem fornecer uma grande diversidade de produtos, com elevados padrões de qualidade e baixos custos. Com a concorrência actual, todas as exigências da procura e os produtos a terem um ciclo de vida muito reduzido, o factor “tempo” tem uma importância muito elevada, sendo necessário reduzir de forma significativa todos os tempos, desde a concepção até à entrega dos produtos. (Chase, Jacobs, Aquilano 2006)

## 2.3. Planear a Produção

De modo a poderem acompanhar esta evolução, terem sucesso e lucro, as empresas tiveram de adoptar medidas estratégicas que lhes dêem uma vantagem face à concorrência. Como tal deverão aproveitar todas as potencialidades dos recursos disponíveis, sendo por isso necessário fazer uma gestão muito cuidada desses, planeando e controlando a produção no sentido de assegurar que todas as partes da empresa estejam focadas nos mesmos objectivos.

É neste sentido que o planeamento toma uma nova importância, e se torna fundamental na gestão do percurso para o sucesso.

### 2.3.1. Planeamento hierárquico

Os sistemas produtivos envolvem uma grande quantidade de decisões que devem ser tomadas a nível de planeamento, nomeadamente nos casos em que existe um número elevado de recursos que devem ser geridos de forma a garantir não só os prazos de entrega dos produtos, bem como a minimização dos custos

Com o elevado número de decisões a serem tomadas, o planeamento é abordado de forma hierárquica dividindo-se em geral as decisões de acordo com os problemas que envolvem. Assim sendo, ao serem tomadas decisões de um determinado nível hierárquico, irão surgir restrições para os níveis inferiores. A tomada de decisões nos níveis inferiores implica, de certa forma, uma avaliação das decisões tomadas pelos níveis superiores (Chase, Jacobs, Aquilano 2006).

A divisão hierárquica do planeamento é feita de acordo com espaço temporal, a ordem de importância das decisões e o seu nível de detalhe. Como tal, e apesar de existir uma grande dependência entre os diversos níveis, o planeamento é, em geral, dividido em 3 níveis hierárquicos:

- Planeamento Estratégico;
- Planeamento Tático;
- Planeamento Operacional.

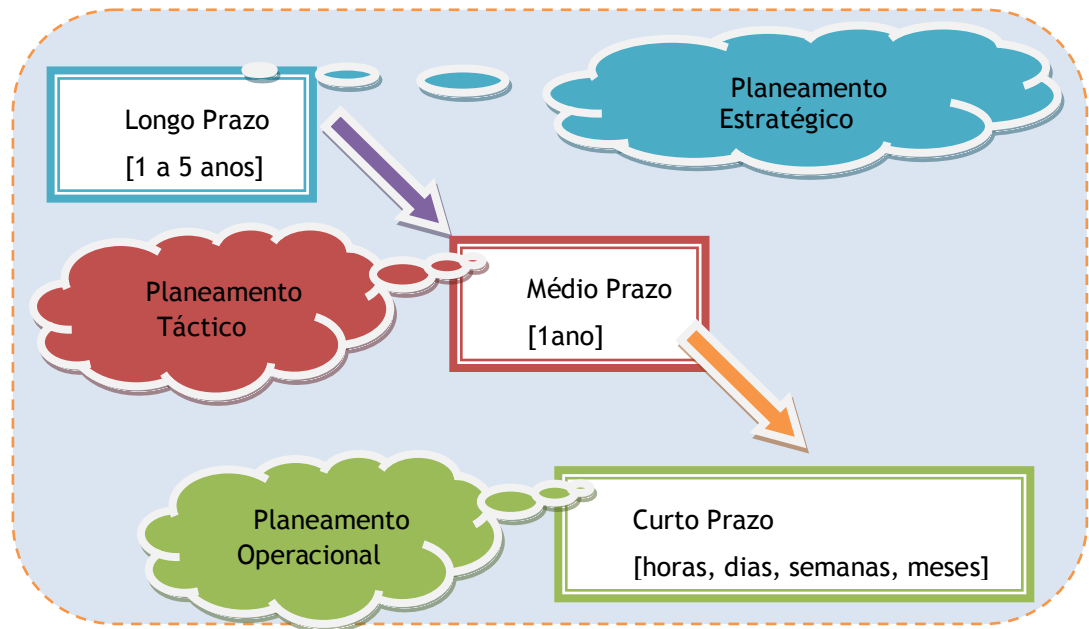


Figura 1: Níveis de Planeamento

No nível superior da hierarquia está o **Planeamento Estratégico** onde são definidas as estratégias a seguir pela empresa, de modo a torná-la competitiva e a obter o sucesso necessário para sobreviver nos dias de hoje. De entre as diferentes decisões tomadas a este nível, visto introduzirem grandes alterações a nível estrutural e provocarem efeitos de longo prazo, destacam-se pelo seu grau de importância, as que se referem a:

- definição de objectivos a cumprir;
- sistemas de produção e distribuição - escolha, alteração ou implementação;
- instalações da Empresa - localização e dimensionamento;
- produtos - desenvolvimento de novos produtos;
- logística - possibilidade de introdução de novos sistemas;
- equipamentos - estudo de aquisição de novos equipamentos.

No nível intermédio da hierarquia, onde entram as decisões de médio prazo, está-se no domínio do **Planeamento Tático**, onde as decisões dos gestores se concentram essencialmente na:

## 8 Job-Shop scheduling - contexto

- utilização dos recursos produtivos
  - alocação das capacidades às necessidades de cada família de produtos;
- utilização dos recursos de armazenamento e de distribuição
  - criação de stocks sazonais;
  - diversas alternativas de transporte dos produtos;
- mão-de-obra disponível
  - recurso a horas de trabalho extra ou a subcontratação.

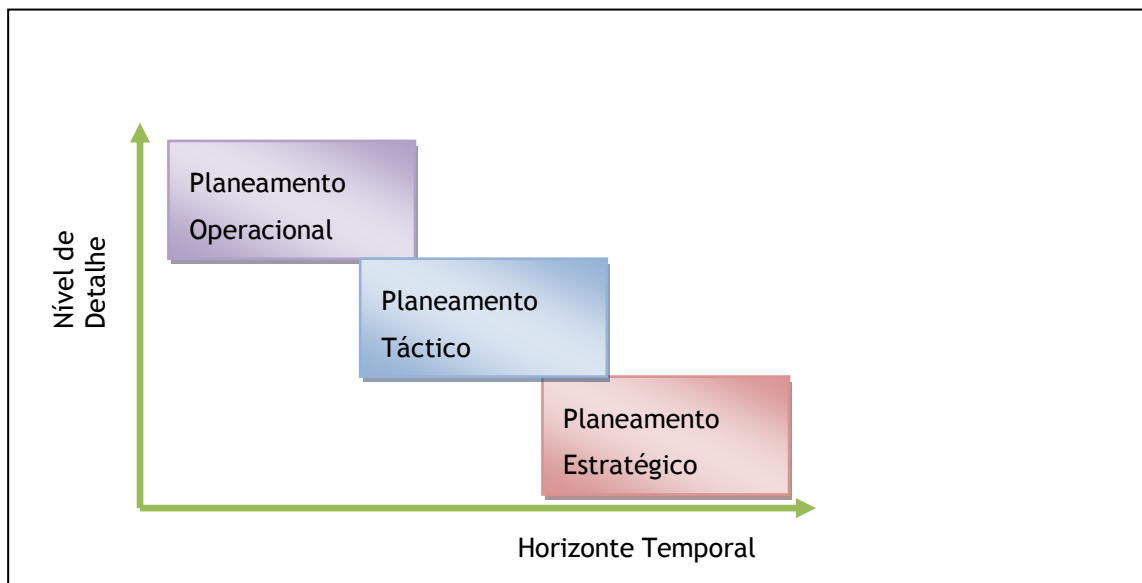


Figura 2: Relação entre o nível de detalhe e o horizonte temporal de cada nível

É neste nível intermédio que se situam as decisões da empresa de carácter comercial, com a definição do **MPS** (*Master Production Scheduling*), plano director de produção. Neste plano são feitas as previsões das necessidades, utilizando-se para tal, os dados relativos às encomendas já existentes e às previsões de encomendas futuras (Chase, Jacobs, Aquilano 2006). De acordo com os recursos disponíveis o **MPS** tem como objectivos:

- satisfazer as necessidades dos clientes quer utilizando produto em *stock*, quer desencadeando novas ordens de fabrico;
- maximizar a utilização dos recursos, da mão-de-obra e dos materiais;
- evitar a criação de *stock* desnecessário.

Por fim, as decisões a **curto prazo** são tomadas no nível de **Planeamento Operacional**. Neste nível é necessário separar toda a informação e decisões provenientes das camadas superiores, e tomar decisões centrando-se em:

- alocar as tarefas aos recursos, definindo assim uma sequência de diferentes actividades que deverão ser realizadas;
- dimensionar lotes de artigos e volumes de produção;
- gerir os stocks;
- gerir frotas.

É neste nível que é definido como tudo é produzido, por quem, onde, com que recursos, sendo por isso, o nível onde existe o maior detalhe.

### 2.3.2. *Materials Requirements Planning* - MRP

Satisfazer os pedidos dos clientes nos prazos estipulados é fundamental para o sucesso de uma empresa industrial. O planeamento e a programação da produção são os pontos-chave da organização empresarial. O objectivo principal é entregar o produto, na quantidade, no momento e no local correctos (Chase, Jacobs, Aquilano 2006).

Para fazer esta gestão do tempo, é necessário ter um elevado nível de controlo sobre todos os materiais e tempos necessários para realizar as tarefas, sendo por isso usado o **MRP** que, recorrendo às previsões de procura de cada artigo, conhecendo a lista de materiais (*Bill of materials -BOF*) e o tempo de processamento de cada tarefa, determina os materiais necessários em cada período e desencadeia as ordens de fabrico e de aprovisionamento.

O **MRP** tem como objectivo garantir a entrega dos materiais no momento em que eles são necessários e na quantidade pretendida, procurando a minimização dos stocks de materiais nos pontos de produção, maximizando a eficiência de produção, introduzindo melhorias no serviço ao cliente.

Como tal, o **MRP** deve interagir com o nível superior, recebendo informação do **Planeamento Tático** e, com o auxílio dos dados das previsões geradas pelo **MPS**, identifica as necessidades de cada componente, dos seus tempos de produção e dos prazos de entrega, gerando as ordens de fabrico, ou as ordens de compra (em caso de falta de componentes no armazém).

### 2.3.3. *Manufacturing Resources Planning* - MRP II

Para colmatar as principais limitações do sistema MRP tradicional, surge o conceito de **MRP II**.

O **MRP II** é uma evolução do **MRP**, que leva em conta também decisões de capacidade, ou seja, acrescentando informação de como produzir e, como tal, vem colmatar falhas do **MRP** no que diz respeito à gestão dos recursos existentes para a produção, nomeadamente da mão-de-obra.

Com esta evolução as aplicações passaram a ter em conta não só aspectos relacionados com a gestão dos materiais e das matérias-primas, como também todos os aspectos relacionados com a gestão de todos os recursos, desde as ferramentas à mão-de-obra.

Este novo conceito é de extrema importância na comunicação entre os diversos departamentos da empresa, principalmente entre o comercial e o da produção. Qualquer que seja o modelo de produção da empresa (*make-to-order*, *make-to-stock*, etc), a previsão da procura é a base para o planeamento. Como tal, é necessário dispor-se de um bom conjunto de dados resultante do diálogo entre os departamentos, de modo a facilitar a execução correcta do planeamento.

#### 2.3.4. Planeamento das capacidades (CRP -Capacity Requirements Planning)

O **CRP** pretende realizar o planeamento das capacidades necessárias dos recursos existentes, para que sejam cumpridos os requisitos do **MRP**, fazendo com que os materiais adequados, nas quantidades certas, sejam entregues da altura devida, ao recurso correcto.

Isto faz com que o **CRP** seja responsável por monitorizar a frequência com que são iniciadas as execuções das ordens de fabrico, verificando os tempos de processamento de cada tarefa e se há disponibilidade nos recursos, de modo a realizar uma utilização eficiente das capacidades de todos os recursos existentes na fábrica.

Com este controlo do **CRP** é possível detectar problemas existentes na utilização das capacidades, o que obriga ao seu ajuste e a uma revisão do Plano Director de Produção de forma a melhorar os planos de execução das tarefas.

## 2.4. Modelos de organização da produção

A forma como os processos produtivos são organizados depende muito dos produtos a desenvolver, da sua natureza, das quantidades envolvidas na produção, e acima de tudo, depende das exigências tecnológicas dos recursos.

Como tal, há necessidade de distinguir e identificar de forma clara os diferentes tipos de produção, no sentido de promover melhorias produtivas e proporcionar a escolha dos métodos de gestão mais adequados (Chase, Jacobs, Aquilano 2006).

Na base da separação dos vários modelos de produção estão os seguintes critérios:

- Espaço físico (implantação física do processo produtivo);
- Produto (grau de continuidade e quantidades produzidas);
- Clientes (relação com os clientes).

Usando estes critérios e generalizando-os, relativamente ao tipo de produção, os processos de fabrico são divididos em dois grandes grupos:

- Produção Contínua (Flow-Shop);
- Produção Discreta (Job-Shop).

#### 2.4.1. Produção contínua - Flow-Shop

A utilização de uma produção do tipo *contínua* está, normalmente, associada a sistemas de produção onde:

- as quantidades de produto são bastante grandes;
- existe uma produtividade elevada;
- existe a baixa qualificação dos operários envolvidos;
- existe uma gestão do processo de reduzida complexidade;
- existe flexibilidade muito reduzida.

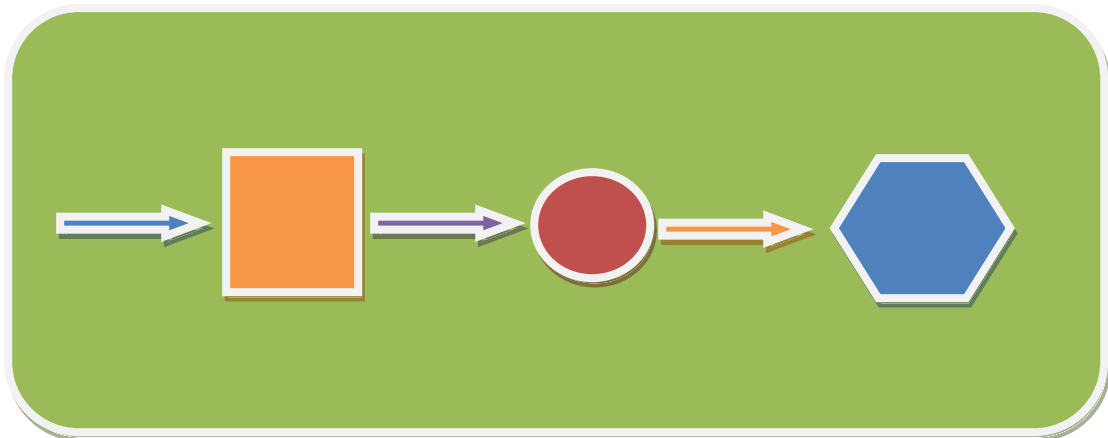


Figura 3: Esquema geral de um exemplo de produção em Flow-Shop

Com a utilização do *Flow-Shop* existe um fluxo do produto através da linha de produção. Como resultado do elevado grau de automatização dos processos, obtém-se uma linha de produção totalmente adaptada ao produto. Por este facto, a flexibilidade é em geral reduzida, pois qualquer pequena alteração, necessária a um produto específico, demora demasiado tempo, o que terá um impacto muito negativo em caso de necessidade de alterações constantes.

### 2.4.2. Produção descontínua - *Job-Shop*

Este sistema de produção foi desenvolvido para que, produzindo em pequenas quantidades, seja possível responder a alterações de produto com a maior agilidade possível.

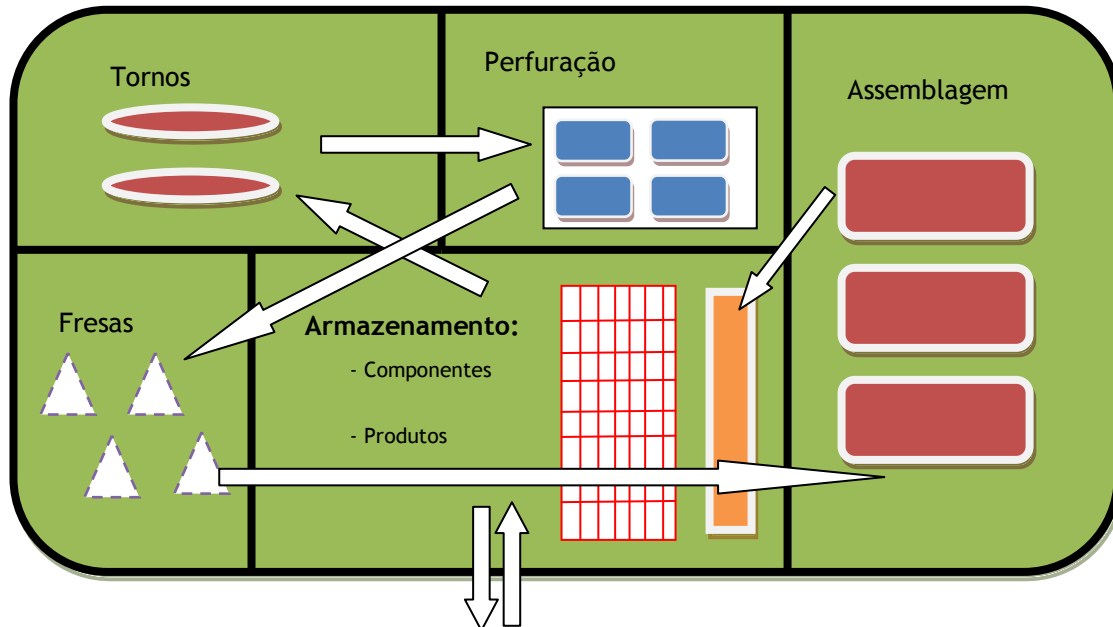


Figura 4: Exemplo de uma produção *Job-Shop*

Esta flexibilidade vai de encontro às necessidades dos clientes que, cada vez mais, querem um produto personalizado. Com isto, um sistema de produção *Job-Shop* é capaz de responder de forma eficaz a encomendas com elevado grau de especificidade, sendo por isso a organização natural no mundo competitivo dos dias de hoje, onde a concorrência faz com que as empresas necessitem de ser flexíveis para terem lucro.

## 2.5. Produção para stock, por encomenda e pontos de desacoplamento

Para além da classificação relativa à organização do fluxo de produção, existem outras formas de distinção tendo em conta relação comercial com o cliente e a forma como o negócio é realizado. Essa separação pode ser visualizada na Figura 5, onde é usada a seguinte notação (Azevedo, 2008):

- produção para *stock* (*Make-to-Stock* - MTS)
- montagem por encomenda (*Assemble-To-Order* - ATO)
- produção por encomenda (*Make-to-order* - MTO)
- engenharia por encomenda (*Engineer-To-Order* - ETO)

Os triângulos vermelhos, invertidos, representam os “pontos de desacoplamento” do processo produtivo, isto é, até onde vai a produção para stock (produção realizada

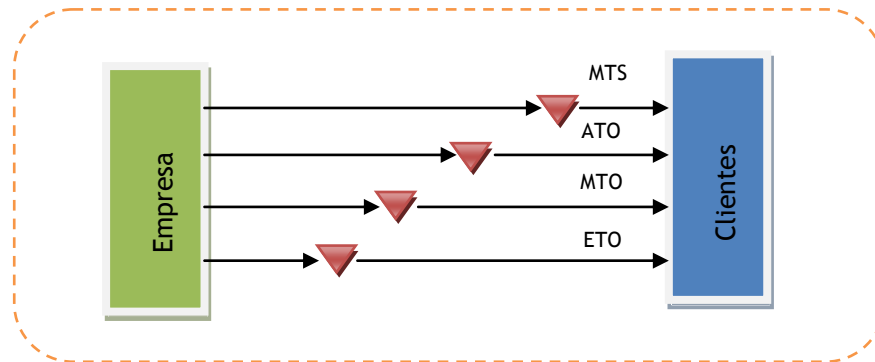


Figura 5: Modelos de Negócio

independentemente das encomendas específicas). Quanto mais à direita estiver este ponto, menor é a capacidade de personalização dada ao cliente, mas em contrapartida o tempo de entrega é menor. Por outro lado, quanto mais à esquerda, maior é a possibilidade de escolha do cliente, chegando, no limite a ser desenhado e criado especificamente o produto para determinado cliente (Azevedo, 2008).

### 2.5.1. Produção para stock

A produção para stock é um tipo de negócio vulgarmente utilizado com produtos com baixo nível de customização, e é utilizado quando é bastante difícil de produzir em tempo útil o produto após a recepção do pedido de encomenda. Como tal, são produzidas quantidades de produto, que é armazenado e fica à espera do pedido de entrega por parte do cliente.

Os níveis de stock definidos para este tipo de produção são calculados com base em previsões de procura. Um caso muito concreto deste tipo de produção é o da venda de produtos cuja procura é bastante sazonal, como é o caso dos gelados, que são produzidos de forma contínua, criando elevado número de stocks durante o Inverno de modo a que no Verão seja possível satisfazer os pedidos dos clientes. E mais uma vez aqui se vê o ponto de desacoplamento muito perto do cliente, permitindo-lhe apenas escolher entre as ofertas de gelados disponíveis no catálogo e já produzidos.

Com a produção para stock existe a vantagem de se conseguir produzir grandes quantidades de produto, mas por outro lado, este tipo de organização tem a grande desvantagem de não ser possível a personalização dos produtos, em função do cliente.

### 2.5.2. Montagem por encomenda

A grande diferença entre este tipo de produção e o anterior está no ponto em que é gerado o stock. São produzidos para stock quantidades de subconjuntos normalizados, e

quanto surge uma encomenda, por parte do cliente, é efectuada a montagem do produto final recorrendo à junção desses subconjuntos.

Aqui existe já a possibilidade de alguma personalização por parte do cliente, uma vez que o produto ainda não se encontra finalizado e é possível adequar a montagem do produto de acordo com as especificações do cliente, de entre as opções disponíveis. Mas continua a existir um elevado stock, desta vez não de produto final, mas de subconjuntos prontos a serem montados e formar o produto final

### 2.5.3. Produção por encomenda

Neste modelo o início de produção ocorre no momento em que são recebidas as encomendas do cliente, sendo apenas satisfeitas as encomendas efectuadas, não havendo por isso a criação de stocks. Sabe-se à partida a BOF (“bill of materials” ou lista de materiais) e existe um projecto bem definido para cada produto. Quando é efectuada uma encomenda, o processo de produção é desencadeado.

Uma vez que os volumes de stock são mínimos, consegue-se reduzir imenso os custos, e por outro lado aumentar de forma significativa a personalização dos produtos.

### 2.5.4. Engenharia por encomenda

Este modelo é em tudo semelhante ao anterior, mas não é conhecida a BOF, o produto é de tal forma possível de ser personalizado que por cada encomenda, é criado um projecto específico. Esta organização permite um elevado grau de personalização, permitindo ao cliente usufruir de um produto único, com as características que ele próprio escolheu, sendo o produto desenhado unicamente para ele. Isto obriga a uma grande flexibilidade por parte da empresa.

## 2.6. Escalonamento - *Scheduling*

No fundo, a gestão da produção pode ser considerada como sendo um conjunto de processos de planeamento e controlo que decorrem ao longo de várias etapas de escalonamento com maior ou menor detalhe.

O objectivo do escalonamento é, “garantir que as tarefas adequadas são executadas no momento exacto, no recurso certo” (Pinedo, 2008). Com isto, pretende-se que a capacidade de cada recurso seja utilizada de forma eficiente, de modo a que se cumpram as ordens de fabrico e melhorando o desempenho da empresa.

Para isso é necessário fazer a distribuição das tarefas pelos recursos disponíveis. Este processo de afectar as operações das tarefas a cada recurso é a actividade de escalonamento (Scheduling).

“Os modelos vitais de escalonamento são os recursos e as tarefas” (Blazewicz, Pesch, Schmidt, & Weglarz, 2007).

No estudo do escalonamento devemos ter presentes alguns conceitos fundamentais:

- **Recurso (*Resource*)**
  - capacidades funcionais e qualitativas
    - tipo de recurso
    - características em que opera
    - ferramentas que utiliza
  - capacidades quantitativas
    - tempos de *Setup*
    - dimensões das filas
- **Ordem de fabrico**
  - indica o que produzir, onde produzir e em que quantidades
- **Tarefa (*Job*)**
  - conjunto de **Operações** que levam à concretização de um produto
- **Operação**
  - conjunto de acções realizadas num recurso que possuem um tempo de processamento, uma característica e um tempo de setup bem definidos.

Assim sendo, antes de se proceder ao escalonamento, é necessário verificar a disponibilidade dos recursos e garantir a disponibilidade de toda a informação relevante.

Existe uma grande diversidade de problemas de escalonamento, associados normalmente a problemas de produção, mas um exemplo muito concreto onde isso não acontece é nas filas de espera das urgências de um hospital.

*“Of all modern medicine’s advances, the best may be the cure for the waiting room.”*

(Chase, Jacob, & Aquilano, 2006)

Esta observação evidencia que uma boa estratégia de sequenciamento/escalonamento poderá ser a solução para muitos problemas quando se tem várias tarefas para realizar. Por aqui se vê a diversidade de problemas de escalonamento que existem nas mais diversas áreas, em ambientes industriais ou em serviços.

Esta diversidade levou ao estabelecimento de notação específica para os problemas de escalonamento, como por exemplo, a notação  $\alpha|\beta|\gamma$  (ver, por exemplo, Madureira, 1995), onde:

- $\alpha$  - indica o tipo de problema:
  - Máquina única (*single machine*);
  - Máquinas Paralelas
    - Open-Shop
    - Flow-Shop
    - Job-Shop
- $\beta$  - indica as características da tarefa (independente / restrições de precedência), por omissão (se o campo  $\beta$  estiver vazio) assume-se que todas as tarefas estão disponíveis para processamento ao mesmo tempo, não é permitida a interrupção (“*preemption*”) e não existem restrições de precedência;
- $\gamma$  - identifica a função objectivo; referem-se como exemplo os problemas de minimização do número de tarefas em atraso  $\sum U_j$  ou da soma dos atrasos  $\sum L_j$ .

### 2.6.1. Problemas com uma única máquina

Na diversidade de problemas relacionados com escalonamento, existem os problemas de escalonamento de **uma única máquina** (*single machine*) em que se pretende escalonar  $n$  tarefas. Nestes problemas, considera-se, em geral, que todas as tarefas podem iniciar a sua produção no instante inicial ( $t=0$ ) e que cada uma delas tem um tempo de processamento e prazo de entrega bem definidos

Um dos casos mais comuns onde este tipo de problemas está presente é, por exemplo, quando se pretende minimizar o atraso máximo, considerando-se  $n$  tarefas, cada uma,  $j$ , com um tempo de processamento,  $p_j$ , e um prazo de entrega,  $d_j$ , bem definidos. Todas as tarefas são processadas numa única máquina, que só pode realizar uma tarefa de cada vez.

Apesar de o âmbito deste projecto não ser o dos problemas com uma única máquina, voltaremos a este assunto mais tarde, por constituir uma componente da abordagem adoptada.

### 2.6.2. Problemas com máquinas paralelas

Qualquer problema de escalonamento está sempre associado a um problema de sequenciamento e a uma afectação dos recursos. Se estiverem disponíveis recursos (máquinas) funcionalmente idênticos, trata-se de problemas com **máquinas paralelas**, muitas vezes de elevada complexidade pois é necessário para além de sequenciar as tarefas, como no caso de um única máquina, afectar as tarefas aos recursos.

Estes problemas de máquinas paralelas são caracterizados (ver por exemplo, Madureira, 1995) por possuírem  $m$  máquinas e  $n$  tarefas a serem processadas, cada uma com um tempo de processamento  $p_j$ . As tarefas só podem ser processadas numa máquina, e cada máquina só poderá processar uma tarefa de cada vez. Assume-se ainda que a máquina  $i$  processa a tarefa  $j$  com a velocidade  $s_{ij}$ .

Os problemas de máquinas paralelas podem ser divididos em três grandes classes:

- **Máquinas idênticas:**
  - o tempo de processamento de uma tarefa é igual em todas as máquinas, isto é, as máquinas têm a mesma “velocidade”;
- **Máquinas uniformes:**
  - a cada máquina  $i$  está associada uma “velocidade” (por exemplo, se uma máquina é duas vezes mais rápida que outra, então todas as tarefas são, nessa máquina, processadas em metade do tempo);
- **Máquinas não relacionadas:**
  - não há nenhuma relação entre as velocidades de processamento (isto é, os  $s_{ij}$  são específicos das tarefas e das máquinas).

### 2.6.3. Problemas multi-operação

Na formulação deste tipo de problemas é considerada informação relativa a:

- Tarefas ( $J_j$ );
- Operação  $i$  da tarefa  $j$  ( $O_{ij}$ ).

Estamos então perante um problema de escalonamento de **tarefas** constituídas por várias **operações** que são realizadas em diversas máquinas. Os problemas podem ser de vários tipos: *Flow-Shop*, *Open-Shop* ou *Job-Shop*.

### 2.6.3.1 - Flow-Shop

Nos problemas designados por *Flow-Shop*, todas as tarefas têm uma sequência idêntica para a realização das operações, ao longo das várias máquinas. Este modelo tem uma *estrutura de precedência linear* (Madureira, 1995).

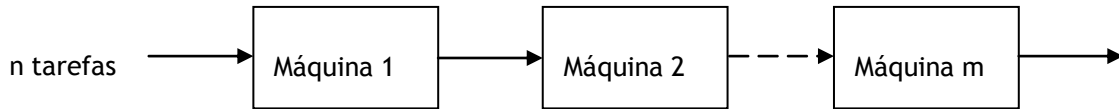


Figura 6: Modelo "Flow-Shop" básico

Neste modelo temos  $j$  tarefas todas divididas em  $m$  operações, distribuídas, uma por cada uma das  $m$  máquinas. Como o próprio nome indica, existe um fluxo de operações, das tarefas, ao longo das máquinas. Esse fluxo é unidireccional, portanto a organização das máquinas segue a ordem natural definida pela ordem das operações que essas máquinas realizam, isto é, se a operação  $j$  precede a operação  $k$ , então a máquina onde é realizada a operação  $j$ , deve preceder a máquina onde é realizada a operação  $k$ .

Poderão existir tarefas que não possuam a totalidade das  $m$  operações. Nesse caso, a operação que não seja necessário realizar para a tarefa em questão será considerada como tendo um tempo de processamento nulo. Qualquer uma das operações, uma vez iniciada, não poderá ser interrompida. Outra das características destes problemas é o facto de todas as operações estarem disponíveis para ser lançadas em produção no instante inicial. É de referir ainda que os tempos de setup são incluídos no tempo de processamento, uma vez que não dependem da sequência das operações.

Estes problemas são NP-hard, sendo que à medida que o número de tarefas e de máquinas aumenta, a complexidade do problema aumenta exponencialmente.

No caso de 2 máquinas, o problema pode ser resolvido à optimalidade através do algoritmo de Johnson. Baker e Trietsch (2009) apresentam este, da seguinte forma:

1. Encontrar o tempo de processamento mínimo de entre as tarefas ainda não escalonadas
2. Se o mínimo obtido no ponto 1 ocorre na
  - a. máquina 1, colocar a tarefa na primeira posição disponível na sequência (situações de empate podem ser resolvidas de forma aleatória). Seguir para o ponto 3.
  - b. máquina 2, colocar a tarefa associada na última posição da sequência (situações de empate podem ser resolvidas de forma aleatória). Seguir para o ponto 3.

3. Retirar a tarefa sequenciada da lista de tarefas a sequenciar e recomeçar no ponto 1. Repetir o processo até todas as tarefas estarem sequenciadas.

### 2.6.3.2 - *Open-Shop*

No modelo *Open-Shop* não há restrições quanto à ordem de realização das operações de cada tarefa nas diferentes máquinas ou na ordem em que dada máquina processa as diferentes tarefas, a ordem das operações nas diferentes máquinas é irrelevante.

Este tipo de modelo dificilmente ocorre em casos reais, pois a ordem de processamento das operações muito raramente é irrelevante.

### 2.6.3.3 - *Job-Shop*

Os princípios gerais de funcionamento do *Job-Shop* são semelhantes aos descritos para o *Flow-Shop*, menos na questão do fluxo de trabalho que não é unidireccional. Ou seja, o problema de *Job-Shop*, consiste num conjunto de tarefas constituídas por um conjunto de operações que são realizadas em determinadas máquinas, sendo a ordem dessas operações específica das tarefas.

A contrário do modelo *Flow-Shop*, a operação  $k$  de uma tarefa não tem necessariamente que ser realizada na máquina  $k$ , assim sendo é necessária uma numeração diferente para identificar as máquinas e outra para identificar as operações. Então, para este tipo de problemas, as operações são identificadas através de 3 índices,  $(i, j, k)$ , em que  $i$  é a tarefa,  $j$  é a operação dessa tarefa e  $k$  é a máquina onde essa operação é processada.

Sendo o *Job-Shop* o objecto de estudo desta dissertação, será tratado mais detalhadamente em secções posteriores.

## 2.7. Escalonamento de uma única máquina (*single-machine*)

Apesar de o tema do projecto não ser escalonamento de uma única máquina, esta secção centra-se neste tema, pois é fundamental compreender a forma como é realizado o escalonamento de uma única máquina para que se possa compreender problemas mais complexos, que muitas vezes, não são mais do que extensões do conceito de escalonamento de uma única máquina. Mesmo em problemas com várias máquinas poderá ser útil utilizar os procedimentos de escalonamento de uma única máquina (“regras de prioridade”). Por exemplo, quando existe um conjunto de máquinas, poderá existir uma máquina, cuja capacidade seja crítica, designada por *bottleneck*. Neste caso um escalonamento de uma única máquina será fundamental para reduzir o *bottleneck* ao mínimo.

Se existir um conjunto de tarefas disponíveis ao mesmo tempo para serem processadas num só recurso, estamos perante um problema de escalonamento elementar. Por outro lado, se o contexto for determinístico, considera-se que os tempos de processamento e as datas de entrega definidos previamente são independentes da sequência de processamento definida. Como se trata de uma máquina só, a própria sequência determina o escalonamento das tarefas no recurso.

Caracterizaremos o problema de escalonamento de única máquina, considerando que temos  $n$  tarefas para escalonar, da seguinte forma:

- tarefas,  $j=1, 2, 3, \dots, n$ , independentes e com tempos de processamento  $p_j$ ;
- todas disponíveis para escalonamento no instante inicial;
- tempos de setup independentes da sequência de escalonamento definida;
- sabe-se previamente as diferentes características, como por exemplo, o prazo de entrega;
- uma tarefa não pode ser interrompida, uma vez iniciado o processamento, este deverá continuar até estar terminado.

### **Restrições adicionais**

Existem ainda outras restrições que poderão ser consideradas, daí resultando diferentes tipos de problemas, como por exemplo:

- se a tarefa  $j$  tiver uma data de lançamento (**release date**)  $r_j$ , e uma data de entrega absoluta (**deadline**)  $d_j$ ;
- se as tarefas puderem ser interrompidas durante o seu processamento;
- se existirem relações de precedência.

#### 2.7.1. Critérios de optimalidade

Pode-se descrever por completo um “calendário” de escalonamento (scheduling) utilizando os tempos de início de cada tarefa e de cada operação.

Discretizando o horizonte temporal considera-se assim que o período de tempo  $t$  começa em  $t-1$  e termina em  $t$  e os períodos de tempo serão  $t=1, 2, 3, \dots, t$ . Por exemplo, se a tarefa  $j$  tem um início de processamento em  $t_j=4$ , então a tarefa  $j$  tem início no instante  $t=4$ .

Para uma solução admissível, é útil determinar num dado problema, alguns valores associados aos critérios de avaliação mais utilizados na prática (Madureira, 1995):

- **Tempo de conclusão** ( $C_j$  - *completion time*) - Indica o instante em que é dado como concluído o processamento

- $C_j = t_j + p_j$
- **Atraso ( $L_j$  - *Lateness*)** - Indica o valor da diferença entre o tempo de conclusão e o prazo de entrega. Toma valores negativos se a tarefa terminar antes do prazo de entrega
  - $L_j = C_j - d_j$
- **Atraso Positivo ( $T_j$  - *Tardiness*)** - É semelhante ao atraso anterior, mas se a tarefa terminar antes do prazo, o seu valor é nulo. Apenas representa o valor de atraso relativo à data de entrega. Normalmente utiliza-se o **Atraso Positivo** como sendo apenas o **atraso**
  - $T_j = \text{Max}\{0, L_j\}$ ;
- **Penalização unitária dos atrasos ( $U_j$  - *Unit Penalty*)** - Essencial para determinar o número de tarefas em atraso
  - $U_j = 0$  - se a tarefa terminou antes do prazo de entrega;
  - $U_j = 1$  - se a tarefa foi concluída depois do prazo de entrega.

## 2.8. O problema em estudo

Este trabalho tem como objecto de estudo o problema de escalonamento de Job-Shop, tendo sido desenvolvida uma aplicação informática do tipo Sistema de Apoio à Decisão (SAD) que permita ao planeador valorizar mais alguns de entre diversos objectivos., na procura de uma solução de escalonamento.

Neste contexto, optou-se por fazer planeamento de curto prazo utilizando o chamado escalonamento “para a frente” (Forwarding Scheduling), já que se trata da forma usual na prática. Foi considerado neste estudo tarefas em que não haja recirculação, ou seja, uma tarefa não pode ter mais do que uma operação realizada no mesmo recurso, fazendo com que o percurso dessa tarefa não passe duas vezes pelo mesmo recurso, evitando então a recirculação. Considera-se ainda que não devem existir recursos paralelos, ou seja, apenas existe um recurso para a realização de cada operação.

Para além das restrições mencionadas consideram-se os seguintes objectivos:

- Minimização do atraso total (Total Tardiness);
- Minimização do número de tarefas atrasadas (Number of Tardy Jobs);
- Minimização do atraso máximo (Largest Tardy);
- Minimização dos tempos de espera (Waiting Times);

- Minimização dos tempos de Setup (Setup Times);
- Maximização da utilização dos recursos (Capacity Utilization).

Estes objectivos foram escolhidos por serem talvez os que poderão influenciar as decisões do planeador.

Para além disso, estes objectivos, quando considerados em conjunto, formando um quadro de decisão “multi-critério”, fornecem ao decisor um conjunto de dados que lhe permite ter uma visão mais concreta do comportamento de determinada solução de escalonamento. Ao poder confrontar os valores de mais do que um objectivo ao mesmo tempo e com a ajuda do sistema de apoio à decisão (SAD) desenvolvido neste trabalho, ele pode comparar várias soluções e restringir as escolhas aos melhores resultados.

Para além disso, na aplicação desenvolvida são efectuadas diversas trocas entre tarefas de modo a gerar novas soluções, tendo a vista a melhoria das soluções de escalonamento anteriormente obtidas.

Como referido, a aplicação desenvolvida no contexto deste trabalho tem em vista testar e validar uma nova abordagem ao problema do “job-shop” baseada no princípio “sequenciar/escalonar” e numa perspectiva multi-critério. Procura-se, assim, desenvolver contributos para o melhoramento do sistema Izaro Grey, já que nesse sistema, não estão disponíveis instrumentos para uma verdadeira comparação de soluções que correspondam a diferentes compromissos entre os objectivos.

# Capítulo 3

## Enquadramento do projecto

Existem várias soluções no mercado a nível de escalonamento de operações (Scheduling). A generalidade das aplicações informáticas de apoio à decisão, disponíveis no mercado, apresentam abordagens muito semelhantes. A título de exemplo vamos, neste Capítulo, fazer uma breve referência a duas delas.

Em seguida descreve-se detalhadamente algumas características do sistema Izaro-Grey, sistema em que se enquadra o trabalho realizado neste projecto.

### 3.1. Exemplos de escalonadores

#### 3.1.1. Preactor®

O Preactor® é um software APS - Advanced Planning and Scheduling System, desenvolvido pela empresa inglesa The CIMulation Centre com a finalidade de melhorar as actividades de gestão de processos produtivos.

Trata-se de uma ferramenta especializada em programação da produção. O Preactor® faz o equilíbrio entre a capacidade dos recursos e procura.

O Preactor® permite:

- gerar e comparar programações ou programas de produção;
- reprogramar operações e ordens de produção;
- calcular prazos de execução;
- estimar datas de entrega;
- avaliar a utilização de recursos;

- acompanhar o desenvolvimento da produção.

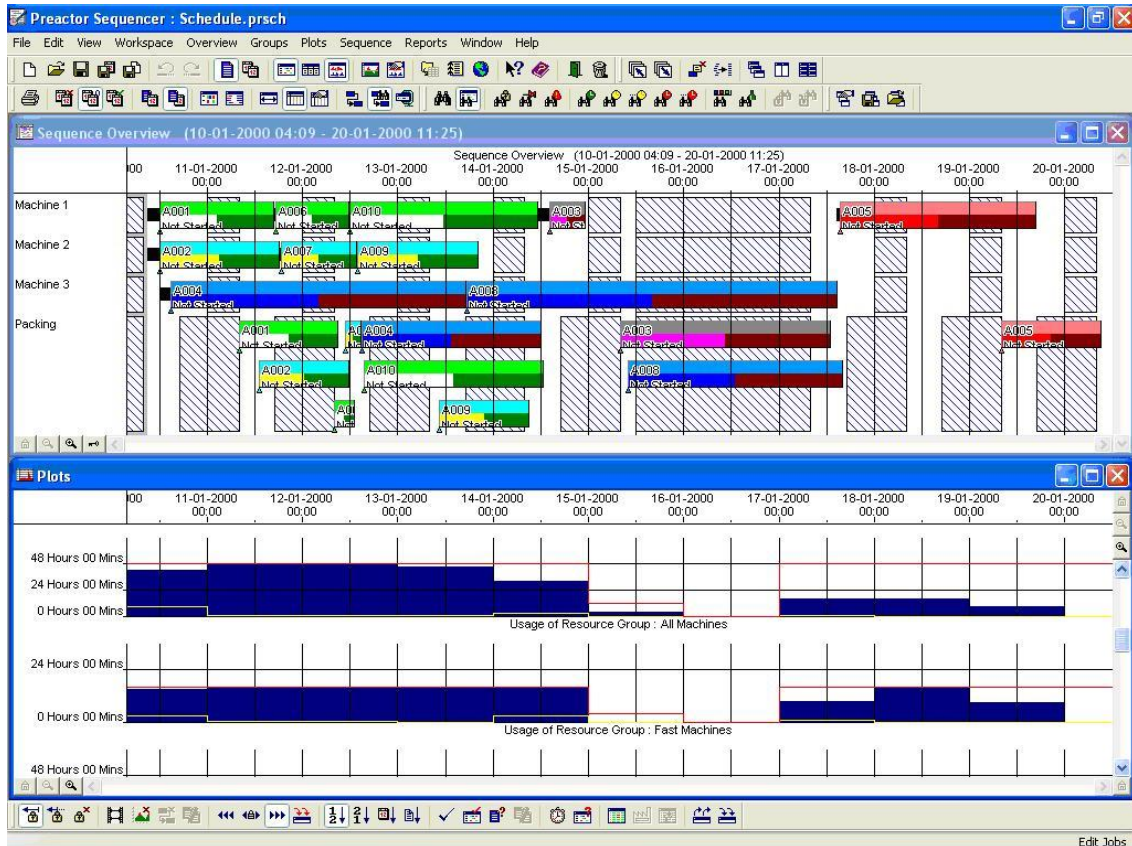


Figura 7: Exemplos de interface do Preactor

Esta ferramenta apresenta diversas capacidades gráficas de entre as quais se destacam, o diagrama de gantt com a distribuição das tarefas e a utilização dos recursos, apresentadas na Figura 7.

### 3.1.2. LEKIN®

O LEKIN® é um sistema de escalonamento desenvolvido na Stern School of Business, New York University. A maioria dos módulos que constituem o sistema foram desenvolvidos por estudantes da Columbia University. O LEKIN® foi criado como um ferramenta educacional, com o principal objectivo de fazer uma introdução à teoria do escalonamento e suas aplicações. O projecto foi dirigido pelo Professor Michael L. Pinedo, pelo Professor Xiuli Chao e pelo Professor Joseph Leung. O seu desenvolvimento foi apoiado pela National Science Foundation dos Estados Unidos da América.

O LEKIN® é um escalonador que possui:

- ambientes diferentes:

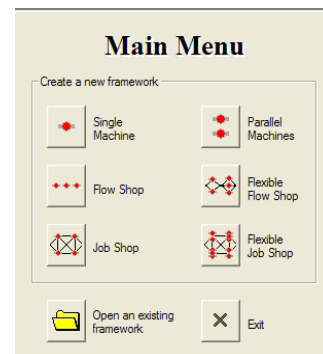


Figura 8: Menu inicial do LEKIN®

- *Single Machine;*
  - *Parallel Machine;*
  - *Flow-Shop;*
  - *Flexible Flow-Shop;*
  - *Job-Shop;*
  - *Flexible Job-Shop;*
- várias Regras e Heurísticas de despacho;
  - Gantt Chart com função clica e arrasta (drag-and-drop);
  - ferramentas gráficas para análise comparativa de diferentes escalonamentos;
  - importação e exportação de algoritmos;
  - facilidade de introdução dos dados do problema por parte do utilizador.

Existe uma versão comercial deste produto, designado por *industrial version*, que possui:

- calendário de tempo utilizado;
- as férias e os turnos de trabalho podem ser especificados;
- permite a recirculação das tarefas.

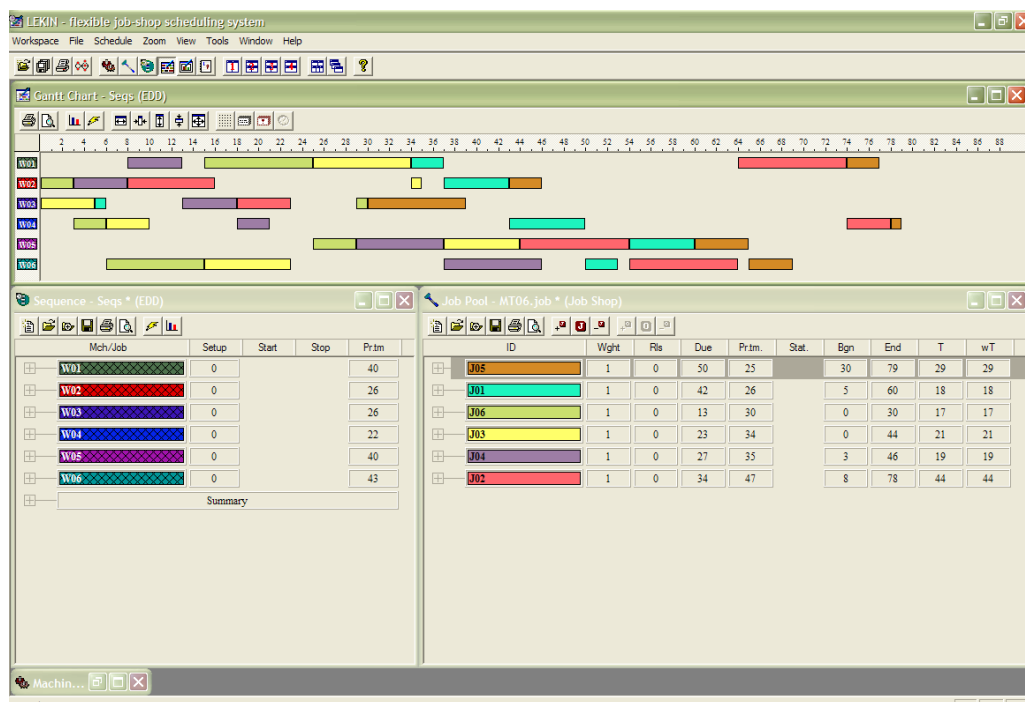


Figura 9: Vista geral de um exemplo de utilização do escalonador LEKIN®

Na Figura 9, podemos ter uma visão geral sobre o LEKIN®, versão estudante, disponível para descarregar gratuitamente através da página de internet.

### 3.1.3. Lacunas

Os dois exemplos apresentados anteriormente possuem diversas características fundamentais para melhoria da produção e aumento da competitividade de uma empresa. Possuem também capacidades gráficas, que fornecem ao planeador uma visão geral sobre os dados quer relativos às tarefas e suas operações, quer relativos aos recursos e sua utilização.

Ambas as aplicações permitem executar diferentes programações tendo em conta diferentes regras de sequenciamento, e comparar essas programações. Mas apenas utilizam algumas heurísticas para melhorar a solução segundo um único critério. Na prática isto constitui uma grande lacuna no que respeita à dificuldade presente nos problemas reais. Essa lacuna é uma das grandes preocupações de qualquer planeador, devido à natureza multi-critério dos problemas reais. Uma abordagem multi-critério é por isso essencial na prática.

## 3.2. Iزارo Grey

Como referido, este projecto está intimamente ligado ao sistema Iزارo-Grey. O sistema Iزارo Grey, procura realizar a função de “escalonamento” (ou programação das operações) com base em informação disponibilizada por um ERP (isto é, um sistema de tratamento integrado da informação da empresa). Nesta informação estão envolvidos dados (relativamente estáveis no tempo) sobre os produtos, a sua constituição e operações a realizar, os recursos e capacidades disponíveis, e ainda dados sobre a procura, firme ou estimada (provavelmente sob a forma de um “plano director da produção”) (Pinho de Sousa e Guardão, 2008).

É fundamental que o sistema de “escalonamento” esteja interligado com o sistema de planeamento de modo a que todos os dados estejam intactos na passagem do ERP para o escalonador de modo a que os resultados sejam consistentes e implementáveis.

O motor de optimização de escalonamento desenvolvido pelo INESC Porto, para a aplicação Iزارo Grey, tem por base uma abordagem heurística assente no princípio “sequenciar/escalonar” (ver Figura 10).

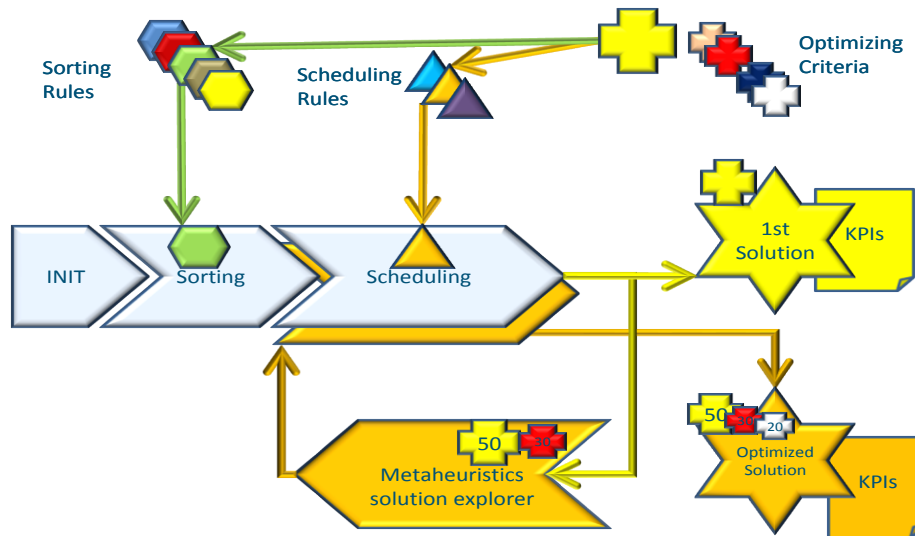


Figura 10: Arquitectura do motor de escalonamento da ferramenta Iزارo Grey. (Pinho de Sousa e Guardão, 2008)

O “algoritmo” utilizado no motor de escalonamento começa por construir uma solução “admissível” e de seguida tenta obter melhorias, por exemplo, trocando a ordem pela qual são realizadas as operações. Estas trocas são realizadas de uma forma sistemática, na expectativa de que as soluções obtidas sejam melhores que as anteriores.

Pinho de Sousa e Guardão (2008) referem que se deve ter em conta que “‘melhores soluções’, num contexto em que se consideram múltiplos critérios, pode significar apenas a optimização segundo um dos critérios, mas não necessariamente segundo todos os considerados. Daí a necessidade de envolver o planeador no sentido de, quando lhe são apresentadas várias soluções geradas pelo sistema (que correspondem a diferentes compromissos entre os objectivos), ele possa escolher uma para ser levada à prática.”

No Iزارo Grey, o processo de decisão é fortemente apoiado pela forma como as soluções são apresentadas. Cada solução é representada num diagrama de Gantt que, com os valores dos critérios, permite ao planeador, avaliar, de uma forma implícita e segundo perspectivas múltiplas, a qualidade da solução em análise.

Um diagrama de Gantt, como o da Figura 11 representa “matricialmente” o “calendário” de realização das operações sobre um eixo de tempo, em que cada linha está associada a um recurso (máquina, posto de trabalho) e em que cada operação é representada por uma barra, de comprimento proporcional à sua duração.

Pinho de Sousa e Guardão (2008), referem que “As poderosas interfaces gráficas disponibilizadas pelo Iزارo Grey são fundamentais no processo de planeamento e, com os

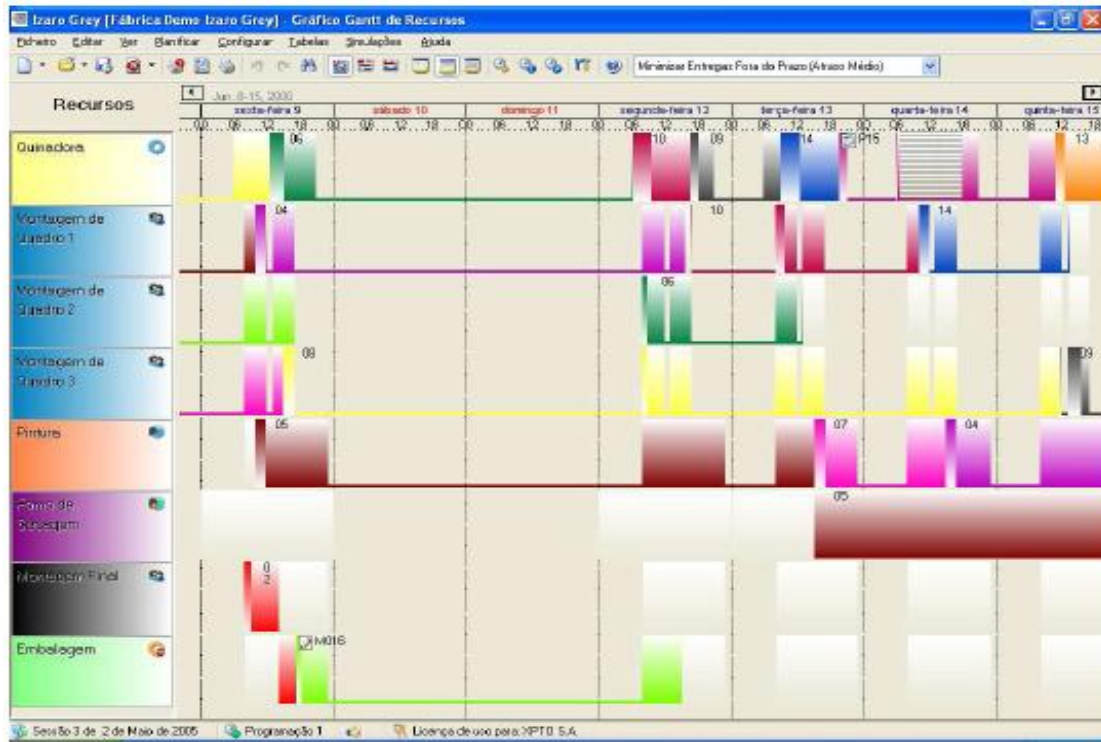


Figura 11: Diagrama de Gantt resultante do escalonamento utilizando o IZARO GREY

algoritmos que o suportam e o papel crucial do planeador, transformam esta aplicação informática (o 'escalonador') num verdadeiro Sistema de Apoio à Decisão.”

### 3.2.1. Operar o IZARO GREY

Quando se inicia a sessão de trabalho, é dado ao planeador a possibilidade de escolher um critério como o mais importante, por exemplo, minimizar os tempos de preparação. Após essa escolha, o IZARO GREY, utiliza uma regra específica para ordenar as Ordens de Fabrico. De seguida procede-se ao escalonamento das Ordens de Fabrico, isto é, as Ordens de Fabrico (OFs) são “programadas no tempo, considerando todas as restrições associadas aos recursos (horários, alterações de capacidade, tipo de recursos, etc) e às precedências (gamas operatórias, ordens de fabrico 'ligadas', disponibilidade de materiais), dando assim origem a uma primeira solução (ou plano). A avaliação da qualidade desta solução é feita através de um conjunto de indicadores (que reflectem, directa ou indirectamente, os critérios utilizados) com a visualização do plano através de um diagrama de Gantt.” (Pinho de Sousa e Guardão, 2008).

No seguimento deste processo, o planeador, se assim o entender, pode então “desencadear uma optimização da solução, baseada num processo sistemático de trocas de OFs ou de operações (com o recurso a técnicas heurísticas ou meta-heurísticas), com o qual se procuram as soluções que representem melhores 'compromissos' entre os vários objectivos para os quais se identificou a importância relativa.” (Pinho de Sousa e Guardão, 2008).

Por de trás da interface gráfica existe este procedimento sequencial (ordenação / escalonamento / optimização multi-critério), tornando o sistema desenvolvido numa ferramenta flexível e interactiva para apoiar, de facto, e em tempo útil, a obtenção de soluções de escalonamento, satisfatórias e realistas.

### 3.2.2. Vantagens da solução Iزارo Grey

A propósito do valor acrescentado desta aplicação no escalonamento das operações, Pinho de Sousa e Guardão (2008) referem que “Esta solução de escalonamento constitui-se como uma ferramenta de apoio à decisão, que valoriza a actividade do planeador, e permite às empresas, entre outros benefícios, obter melhorias no cumprimento de prazos e uma maior eficiência na utilização dos seus recursos produtivos, através, por exemplo, de uma minimização dos tempos de preparação (‘set-up’).”

Dizem ainda que “As vantagens deste sistema, relativamente aos existentes no mercado, centram-se na possibilidade de se obter soluções de escalonamento que representam bons compromissos entre múltiplos objectivos, por vezes conflituosos e até de carácter subjectivo. A qualidade destas soluções é mensurável e comparável através de um conjunto de indicadores quantitativos.”

Com um mundo tão competitivo e com tanta concorrência, (Pinho de Sousa e Guardão, 2008) atribuem o sucesso deste sistema em grande parte “a uma preocupação permanente durante a sua concepção e desenvolvimento, de ir ao encontro das necessidades concretas das empresas industriais.”

### 3.2.3. Aplicações

O *Iزارo Grey* encontra-se implementado em diversas empresas, de diversos sectores. A empresa *Soft i9* disponibiliza na sua página da internet ([www.softi9.com](http://www.softi9.com)) informação onde refere algumas dessas empresas. (ver Figura 12).

Um desses exemplos é a *Tetra Pak Tubex Portugal*, uma unidade da multinacional *Tetra Pak Tubex*, localizada em Carnaxide sendo a maior unidade fabril do mundo de Palhinhas, com uma capacidade de produção anual de 10 mil milhões de unidades. A *Tetra Pak Tubex*, optou pelas soluções de planeamento e recolha de dados no fabrico, *Iزارo APS* e *Iزارo MES*, onde o escalonador do Iزارo Grey é uma das ferramentas utilizadas.

Logo	Empresa	Cidade	País	Sector	Actividade	Descrição Actividade
	ACC.ELÁSTICOS LESOL, S.A.	San Sebastián	ESPAÑA	Industrial	Borracha, Cerâmica e Plástico	Fabricação de peças anti vibração.
	FAPOMED – IND. CONF. PRODUTOS MÉDICO CIRÚRGICOS, S.A.	Felgueiras	PORTUGAL	Industrial	Materiais Cirúrgicos	Produtos Médico Cirúrgicos
	LARRAÑAGA PLÁSTICOS, S.A.	Usúrbil	ESPAÑA	Industrial	Químico	Transformação de materiais plásticos. Peças industriais em materiais plásticos.
	CAF. CONSTRUCCIONES Y AUXILIAR DE FERROCARRILES	Beasain	ESPAÑA	Industrial	Ind. Auxiliar Equipamentos Transporte	Locomotoras de comboios, vagões e material rolante
	TETRA PAK TUBEX	Lisboa	PORTUGAL	Industrial	Auxiliar Ind. Alimentar	Produtos Auxiliares Indústria Alimentar
	Industrias Alga	Abadiño	ESPAÑA	Industrial	Ind. Auxiliar Equipamento Transporte	Fabricação de acessórios para o sector de equipamento transporte, válvulas, motores de arranque.
	Bollinghaus Steel	Vieira de Leiria	Portugal	Industrial	Produtos Laminados	
	M. C. Graça, Lda.	Carregado	PORTUGAL	Industrial	Auxiliar Ind. Automóvel	Acessórios p/ indústria automóvel
	Grupo Armando Álvarez ASPLA	Torrelavega	ESPAÑA	Industrial	Químico	Transformação de materiais plásticos. Fabricação e impressão de filme plástico p/ embalagens.

Figura 12: Algumas referências de empresas que utilizam a aplicação Iزارo APS.

Na sua página da internet ([www.softi9.com](http://www.softi9.com)), a **Soft i9** apresenta um conjunto de testemunhos que levaram a empresa **Tetra Pak Tubex Portugal** a optar por esta aplicação:

- “A opção por estas soluções, surgiu devido ao facto de a empresa sentir a falta de uma solução de planeamento integrada no sistema de gestão global do negócio;”
- “As ferramentas que asseguravam estas funcionalidades eram ferramentas básicas do Microsoft Office, o que resultava na falta de perspectiva de médio e longo prazo da carga e utilização das linhas de produção”;
- A escolha recaiu nas soluções da Softi9 “pela credibilidade do suporte comercial e pela facilidade e flexibilidade do interface com o planeador”.

No site, são indicadas algumas vantagens da solução da soft i9:

- “Os sistemas de controle integrados, desde o planeamento de produção e cálculo de necessidades de recursos, passando pelo registo de falhas dos equipamentos de processo e expandindo ao seguimento e análise estatístico das variáveis de controlo de processo ‘são fundamentais, neste cenário de evolução’.”

Outro dos exemplos de implementação disponíveis na página da internet da *Soft i9* é o caso da *Aspla*. A *Aspla* é uma empresa que se dedica ao fabrico de embalagens flexíveis baseadas em filme de plástico para os sectores da alimentação, agrícola, bebidas, indústrias de lacticínios, etc., dispondo de um vasto leque de produtos personalizados para cada cliente.

Para esta planificação, a *ASPLA* escolheu o IZARO APS “por se tratar de uma ferramenta de planificação gráfica que permite a planificação de todas as Ordens de Fabrico de uma forma rápida e simples, centrando-se no agrupamento das características do produto e proporcionando um prazo de entrega realista ao departamento comercial da empresa. Desta forma, uma vez enviadas todas as Ordens de Fabrico do Sistema de Gestão para o IZARO APS, estas são sequenciadas graças aos algoritmos próprios deste escalonador, originando uma solução de acordo com os objectivos da empresa” ([www.softi9.com](http://www.softi9.com)).

### 3.3. Contextualização do trabalho

Tendo em conta a natureza das soluções de escalonamento produzidas pelo sistema IZARO Grey, surgiu a ideia da aplicação desenvolvida neste trabalho. Com esta aplicação, e como referido atrás, pretende desenvolver-se um conjunto de contributos, a serem eventualmente integrados no IZARO Grey, melhorando a sua funcionalidade.

Estas melhorias visam testar, de forma sistemática, a abordagem heurística desenvolvida em torno do esquema sequenciamento/escalonamento e da adopção de uma estratégia multi-critério mais clara para os planeadores.

No capítulo seguinte apresenta-se esta abordagem (*Sequenciar/Escalonar*), frisando os vários passos que são seguidos na resolução dos problemas de *Job-Shop*. Apresenta-se também a ferramenta informática criada para dar suporte à abordagem *Sequenciar/Escalonar*.

São ainda apresentados alguns exemplos de resolução de problemas *Job-Shop* utilizando um pequeno Sistema de Apoio à Decisão (SAD) onde se integra a aplicação desenvolvida.



# Capítulo 4

## A abordagem “sequenciar/escalonar” para o problema de Job-Shop

Neste capítulo, iremos apresentar a abordagem adoptada neste trabalho para o problema do escalonamento de Job-Shop, fazendo algumas notas prévias sobre alguns métodos e técnicas existentes relevantes para a abordagem proposta (regras de prioridade, técnicas heurísticas). Foi desenvolvida uma aplicação informática que visa introduzir possíveis melhorias no sistema actualmente utilizado no Izaro-Grey. É descrita a abordagem utilizada no desenvolvimento dessa aplicação informática.

De seguida, são apresentados alguns exemplos do funcionamento dessa aplicação. Os exemplos são de diferentes graus de complexidade e têm em vista demonstrar a forma como a aplicação informática lida com os dados e os trata para resolver o problema numa perspectiva multi-critério. Com base num conjunto reduzido de indicadores, o planeador pode então escolher uma solução de escalonamento que satisfaça adequadamente as suas preferências.

### 4.1. Principais técnicas utilizadas

Nesta secção apresentam-se sucintamente algumas técnicas úteis neste contexto. Começa-se por fazer uma referência às regras de prioridade utilizadas para a definição da sequência de tarefas. As regras de prioridade apesar de serem particularmente indicadas para problemas de uma única máquina, são aqui utilizadas no problema de *Job-Shop* para a construção da solução inicial. A partir desta solução, são constituídas novas soluções com que se procura melhor satisfazer os objectivos em jogo.

#### 4.1.1. Regras de Prioridade (priority rules)

No contexto de uma única máquina as **regras de prioridade** são utilizadas para a criação da sequência segundo a qual um conjunto de tarefas deve ser realizado. Algumas destas regras serão utilizadas para gerar a sequência inicial do problema de **Job-Shop**.

Existe uma grande variedade de **regras de prioridade**, de entre as quais se destacamos seguintes por serem as mais utilizadas nos casos práticos:

- EDD (Earliest Due Date - Data de entrega mais cedo) - são primeiro sequenciadas as tarefas com data de entrega mais cedo;
- SPT (Shortest Processing Time - Tempo de Processamento mais Curto) - são sequenciadas primeiro as tarefas que demorem menos tempo a ser processadas;
- LPT (Longest Processing Time - Tempo de Processamento mais Longo) - são sequenciadas primeiro as tarefas que demorem mais tempo a ser processadas;
- FCFS (First Come First Served) - é sequenciada primeiro, a primeira tarefa a estar disponível;
- LWR (Least Work Remaining) - a tarefa cujo tempo de processamento restante for menor, será a primeira da sequência;
- Random - as tarefas são ordenadas de forma aleatória.

No caso de problemas de **Job-Shop**, após ser definida a sequência das tarefas através de uma destas regras, procede-se ao escalonamento dessa sequência de tarefas, que consiste em alocar as operações das tarefas aos recursos, colocando a operação no primeiro espaço disponível, ou seja, deve ser alocada ao primeiro intervalo de tempo que o recurso esteja disponível e que ocorra depois da operação anterior da mesma tarefa.

#### 4.1.2. Heurísticas de melhoramento

Como já se verificou anteriormente, os problemas de escalonamento de **Job-Shop** são problemas de optimização combinatoria de difícil resolução (**NP-Hard**). Em virtude dessa dificuldade de resolução serão utilizados métodos heurísticos baseados em procedimentos de **Pesquisa Local**.

A **Pesquisa Local** consiste basicamente na deslocação de uma solução para a outra na sua vizinhança de acordo com determinadas regras previamente definidas. Essa sequência de soluções obtidas ao longo da pesquisa define uma **trajectória** no espaço das soluções (Madureira,1995).

A **Pesquisa Local** tem como ponto de partida uma sequência (solução inicial) que pode, por exemplo ser obtida por uma das **regras de prioridade**. A partir dessa solução inicial, e

por cada iteração, procede-se a troca entre duas tarefas que estejam no espaço de vizinhança. Para a definição da vizinhança é utilizado um intervalo de vizinhança definido por  $\beta$ , ajustado no início do procedimento. No procedimento de *Pesquisa Local*, é aleatoriamente escolhida uma tarefa da solução inicial. Escolhida essa tarefa, procede-se à escolha aleatória de uma segunda tarefa, que se encontre dentro do intervalo de vizinhança da primeira tarefa seleccionada. Escolhidas estas duas tarefas, é feita a troca de posição das duas, sendo assim gerada uma nova solução que é avaliada. Na segunda iteração, o processo repete-se, mas desta vez a solução inicial será a solução que resultou da iteração anterior.

Uma extensão natural desta abordagem (e que poderá constituir um desenvolvimento futuro do trabalho) passava pelo recurso a *metaheurísticas*, ou seja, algoritmos de *Pesquisa Local* com estratégias mais complexas de pesquisa. Entre estes algoritmos encontram-se a pesquisa tabu, pesquisa “simulated annealing”, etc...

#### 4.1.3. Multi-objectivo

Os problemas práticos envolvem, quase invariavelmente, objectivos múltiplos o que exige uma abordagem diferente da tradicional e mais complexa.

No caso dos problemas de escalonamento, esta complexidade é aumentada pelo seu carácter combinatório.

Um problema de Optimização Combinatória Multi-objectivo pode ser formulado da seguinte forma (ver, por exemplo, Viana (2004)):

$$\min f(X) = \{f_1(X), \dots, f_m(X)\}$$

sujeito a:

$$X \in D$$

onde  $m$  é o número de funções objectivo,  $X$  representa o vector das variáveis de decisão,  $D$  é o espaço de soluções admissíveis  $f_k(X)$  indica os objectivos a minimizar.

Segundo Viana (1997), “na verdade, esta representação não é absolutamente correcta sob o ponto de vista ‘matemático’, já que normalmente não há uma solução que conduza, simultaneamente ao valor óptimo para todos os objectivos. É por esta razão que em contextos multi-objectivo, não se usa o termo optimização mas termos tais como não dominância...”

A imagem da solução  $X$  no espaço dos objectivos é dada pelo ponto  $z^X = f(X)$ .

Diz-se então que o ponto  $z^X$  domina  $z^Y$  se  $f_k(X) \leq f_k(Y), \forall k$  com  $f(X) \neq f(Y)$  e representa-se por  $(z^X \succ z^Y)$ .

Então a solução  $X$  domina a solução  $Y$ , se a imagem de  $X$  dominar a imagem de  $Y$ , ou seja, uma solução  $X$  domina a solução  $Y$  se for melhor (ou pelo menos de igual valor) no que se refere a todos os objectivos.

Assim sendo, a solução  $X$  é não dominada se não existir nenhuma solução admissível que domine a solução  $X$ , isto é, se não existir uma solução admissível que seja melhor em todos os objectivos do que a solução  $X$ . O conjunto de soluções não dominadas é conhecido por conjunto de **Pareto** ou conjunto de soluções eficientes.

Estes conceitos foram utilizados neste trabalho no processo de melhoramento das soluções (no sentido de se considerarem simultaneamente os vários objectivos em jogo). A ideia fundamental da abordagem adoptada consiste em identificar um conjunto de soluções não dominadas (correspondentes a compromissos entre objectivos), que sejam fornecidas ao planeador e que das quais este possa escolher uma.

## 4.2. Abordagem adoptada

A abordagem adoptada neste trabalho, de sequenciar/escalonar as tarefas, pode contribuir, de forma significativa, para a melhoria das soluções de escalonamento. Numa primeira fase os dados são guardados numa base de dados estruturada. Lidos os dados (Read Data) da base de dados, procede-se à ordenação (SORT) das tarefas utilizando uma regra de prioridade.

Após serem ordenadas as tarefas, procede-se ao escalonamento (scheduling) das mesmas, alocando as operações aos recursos correspondentes. O resultado desse escalonamento é apresentado num diagrama de Gantt. Para além desse diagrama, os dados são avaliados (Evaluate) segundo os objectivos definidos no início do processamento e são apresentados os valores relativos a esses objectivos, de modo ao planeador poder tomar decisões.

O planeador poderá ainda dar início ao processo de trocas (Swap), com o qual a aplicação, através de um processo sistemático, apresenta um conjunto de soluções não dominadas.

Estes pontos estão representados na Figura 13 e serão apresentados com maior detalhe nas secções seguintes.

Esta aplicação foi desenvolvida em Visual Basic (VB).

### 4.2.1. Leitura de Dados (*Read Data*)

Como é natural, foi necessário guardar todos os dados relativos às tarefas, operações e recursos associados ao problema de escalonamento em questão, tornando possível o uso desses dados a partir da aplicação em VB.

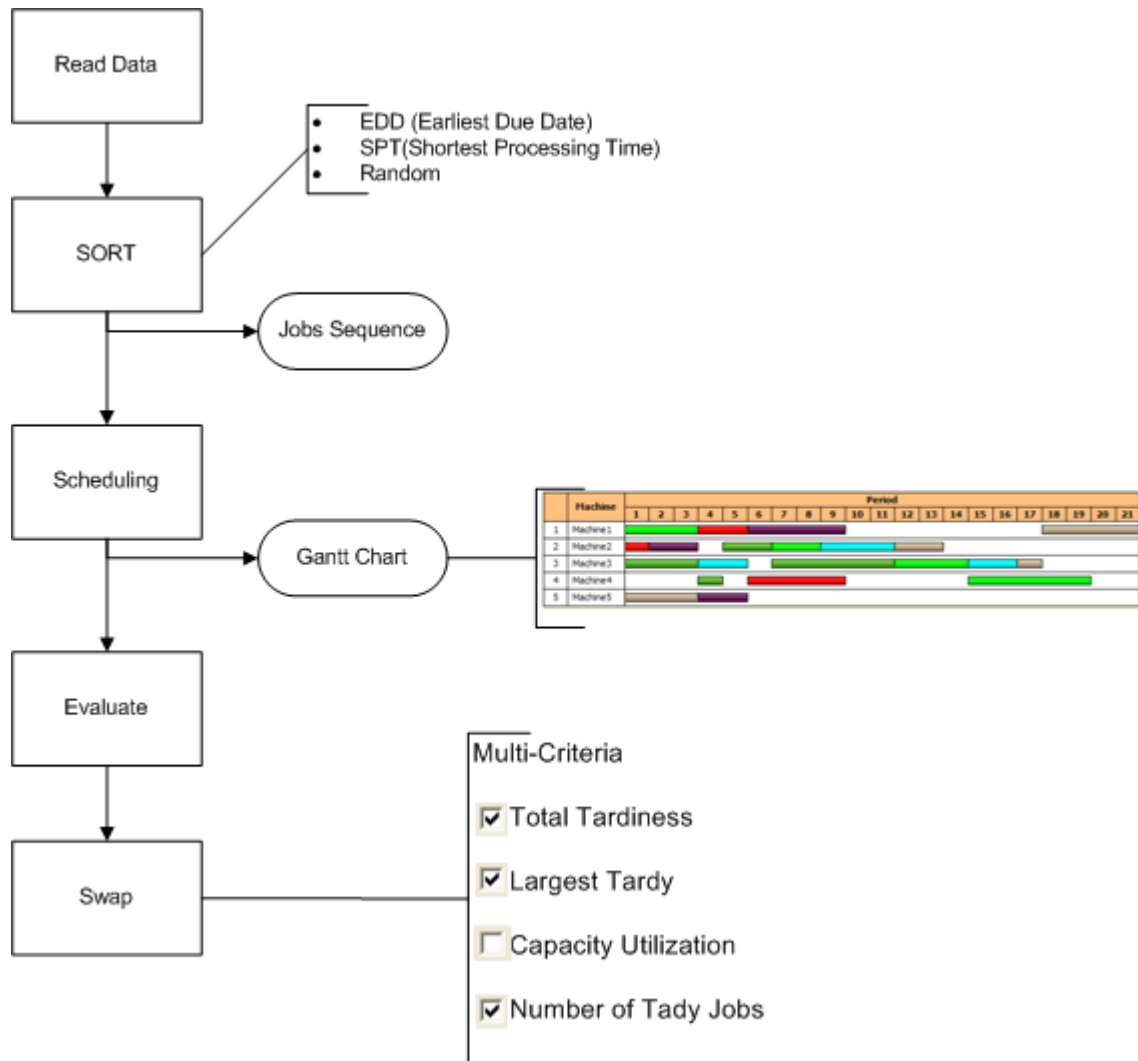


Figura 13: Estrutura da aplicação

No início do desenvolvimento os dados eram inseridos directamente no VB de modo a facilitar a aprendizagem da linguagem de programação. Posteriormente, passou-se a ler os dados guardados num ficheiro de texto (.txt), mas de modo a melhorar a sua organização, estes foram inseridos numa base de dados *Microsoft Access*. A sua interacção com o VB é feita de forma simples e correcta, pelo que, esta foi a solução adoptada como definitiva. No módulo Read Data final, é usado um “*procedure*” em que é feita uma pesquisa à base de dados, e os dados resultantes dessa pesquisa são guardados em variáveis do VB.

A base de dados está organizada da forma expressa na Figura 14.

Na tabela “*Job*” são guardados os dados relativos às datas de entrega (Due Date) de cada tarefa. É ainda guardada informação relativa à Release Date e ao prazo limite de entrega da tarefa (Deadline). Estes dois últimos campos não estão em uso pelo sistema actual, mas seriam algo a ser implementado em trabalho futuro. Quanto à tabela “*Operation*” temos os campos “*Job*”, que guarda o número da tarefa a que corresponde a operação indicada no campo “*operation*”; o tempo de processamento dessa tarefa, guardado no campo “*ProcTime*”

(*Processing Time*); e o campo “*Machine*” que contém a indicação da máquina onde é realizada a operação da tarefa em questão.

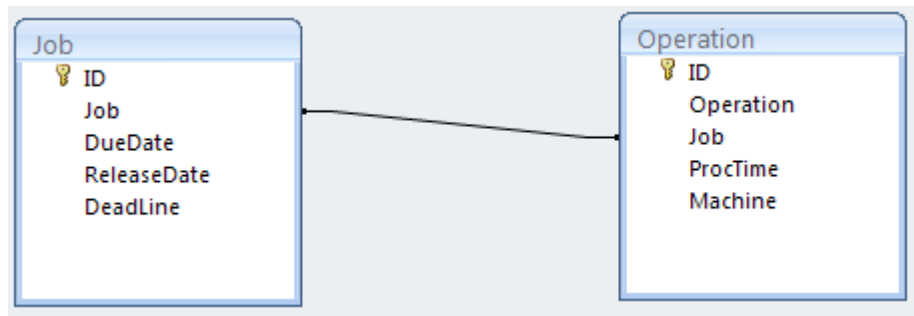


Figura 14: Esquema da relação entre as tabelas da Base de Dados usado

#### 4.2.2. Sequenciamento (*Sort*)

Após a leitura dos dados é necessário ordená-los de forma a gerar uma sequência de tarefas que é designada por *solução inicial*. A *solução inicial* é a sequência a partir da qual se tenta alcançar melhores resultados para determinados objectivos predefinidos.

Para gerar a sequência que dá origem à *solução inicial*, é dado ao planeador o poder de escolher uma de três **Regras de Prioridade** diferentes.

##### 4.2.2.1 EDD

Na **regra de prioridade EDD** (*Earliest Due Date*) as tarefas são ordenadas de acordo com as suas datas de entrega, ou seja, a primeira tarefa é aquela cuja data de entrega ocorre mais cedo. Com esta regra tenta-se à partida minimizar os atrasos na entrega dos produtos. Este método é vastamente utilizado pelas empresas, principalmente, em casos onde a data de entrega é um factor muito importante.

No algoritmo usado para gerar a sequência inicial segundo a **regra de prioridade EDD**, começa-se seguindo a numeração de cada tarefa. Verifica-se a data de entrega da primeira tarefa e atribui-se o primeiro lugar na sequência, de seguida, verifica-se a data de entrega da tarefa seguinte. Se a data de entrega for inferior à da tarefa numero um, o número da tarefa é copiado para a segunda posição, e a nova tarefa é colocada em primeiro lugar, caso contrário será guardada na segunda posição. O processo repete-se.

Quando a data de entrega é menor do que um grupo de tarefas, essas tarefas são deslocadas uma unidade para o final da sequência, por exemplo, se a data de entrega da quarta tarefa for inferior à data de entrega das tarefas 2 e 3, já distribuídas na sequência, a tarefa da posição 3 passa para a posição 4 da sequência, a posição 2 passa para a 3 e a quarta tarefa passa a ocupar a posição 2 da sequência. Este processo repete-se até todas as tarefas fazerem parte da sequência, que será então designada por sequência inicial.

#### 4.2.2.2 - SPT

Com o uso da **regra de prioridade SPT** (*Shortest Processing Time*), o processo é semelhante, apenas com a diferença das tarefas serem ordenadas segundo o seu tempo de processamento, isto é, são primeiro processadas as tarefas cujo tempo de processamento seja menor ficando para o fim da sequência as tarefas cujo tempo de processamento seja mais elevado.

#### 4.2.2.3 - Random

Na aproximação **Random** (aleatória), como o próprio nome indica, as tarefas são ordenadas de forma aleatória. Em termos de implementação, é gerado um número aleatório do intervalo de 1 até ao número de Tarefas (*Jobs*). De seguida esse número é comparado com os números aleatórios anteriormente gerados e ordenados, e caso o número já exista na sequência, é gerado novo número aleatório. Este processo é repetido até todas as tarefas pertencerem à sequência.

##### **Sequência de Tarefas**

A **Sequência de Tarefas** (*Jobs Sequence*) é resultado da ordenação inicial, obtida por uma das **regras de prioridade**. Os índices das tarefas são guardados por meio de um *array* chamado *sequence*. Em *sequence(1)* é guardado o índice da tarefa que é realizada em primeiro lugar, em *sequence(2)* é guardado o índice da tarefa realizada em segundo lugar e assim sucessivamente até todas as tarefas estarem guardadas, pela ordem determinada no sequenciamento (SORT).

#### 4.2.3. Escalonamento (*Scheduling*)

Após ser gerada a Sequência de Tarefas (*Jobs Sequence*), designada também por solução inicial, procede-se ao seu Escalonamento (*Scheduling*). O Escalonamento consiste em distribuir as tarefas pelos recursos disponíveis consoante o recurso que a operação da tarefa necessite.

Começa-se por escalonar a primeira tarefa da sequência. No primeiro período todos os recursos estão livres. Então, para a primeira operação o setup que existe é o chamado setup inicial. Uma vez definido esse setup, é alocada a tarefa ao recurso durante o tempo correspondente.

A segunda operação da primeira tarefa é alocada ao recurso correspondente logo que a primeira operação termine. Antes do início da segunda operação deve ser alocado o tempo de setup de modo a que o recurso esteja disponível para quando a primeira operação terminar, a segunda possa ter início.

As restantes operações da primeira tarefa são escalonadas da mesma forma. Neste momento, os setups são todos definidos como setups iniciais, mas irão resultar períodos livres antes de cada operação que irão ser aproveitados para, se existir tempo disponível, alocar as restantes operações e respectivos setups.

Para efeitos da determinação dos tempos de “setup”, define-se “característica” como sendo um atributo da tarefa, relevante para um dado recurso quando realizar determinada operação dessa tarefa. Por exemplo uma característica, numa máquina de pintura, será a cor a utilizar em determinada operação.

Terminada a primeira tarefa dá-se início ao escalonamento da segunda tarefa da sequência. O início de processamento de cada operação é definido como sendo igual ao fim de processamento da operação anterior da mesma tarefa. O passo seguinte é verificar se o recurso onde a operação é realizada, não está ocupado durante o tempo necessário para a operação, desde o início do setup até ao final do processamento. Os tempos de setup dependem da característica anterior e da seguinte, portanto ao escalonar devem ser tidos em conta os tempos de setup das operações das tarefas já escalonadas, pois colocar uma operação antes de outra, leva a que o tempo de setup dessa tarefa já anteriormente escalonada seja alterado.

Caso exista tempo suficiente para o setup, para o processamento da operação a escalonar e ainda para a alteração do tempo de setup da operação já escalonada, então a operação é alocada nesse intervalo de tempo, e o tempo de setup da operação anteriormente alocada é ajustado tendo em conta que o setup passa a ser da característica da nova tarefa alocada para a tarefa anteriormente alocada. Caso não exista tempo suficiente, o início de processamento irá ser alterado, passando a ser igual ao fim de processamento da operação já alocada, mais o tempo de setup entre a característica da operação já alocada e a da tarefa a alocar. O processo repete-se até que todas as operações e todas as tarefas estejam alocadas.

O resultado do escalonamento é expresso por meio de uma matriz de ocupação de recursos, onde se pode observar os períodos de tempo que as tarefas e os tempos de setup das operações ocupam ao longo do escalonamento. Esta matriz será caracterizada mais tarde nas secções relativas aos resultados e aos testes computacionais. Outro dos resultados importantes deste escalonamento é um diagrama de Gantt.

#### 4.2.4. Diagrama de Gantt

O diagrama de Gantt (Gantt Chart), é uma forma de representar graficamente a distribuição das operações das tarefas pelos diferentes recursos existentes.

Com o Diagrama de Gantt é mostrada, de forma clara, como irão ser distribuídas as operações das tarefas pelas máquinas respectivas ao longo do tempo. Num diagrama de Gantt, como o da Figura 15, do lado esquerdo temos os recursos disponíveis, sendo cada linha um recurso. Na parte superior temos os períodos de tempo, sendo que cada coluna representa um período de tempo que pode representar 1 hora, 1 dia, 1 semana, etc.

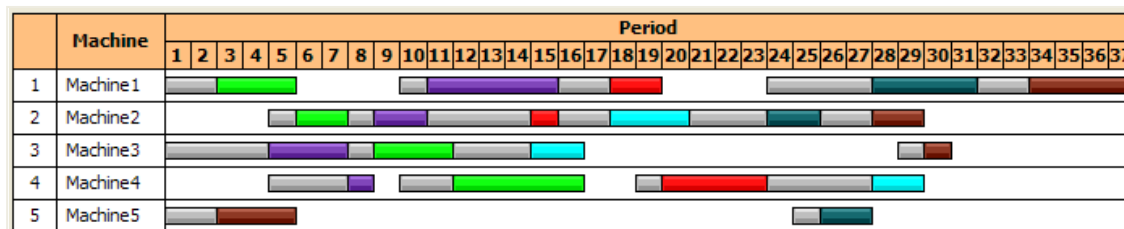


Figura 15: Exemplo de um Diagrama de Gantt

No diagrama, cada tarefa é representado por uma cor, e a alocação de uma operação dessa tarefa a uma máquina é representada por uma barra da cor correspondente. A barra prolonga-se durante os períodos de tempo nos quais a máquina estará ocupada a realizar essa operação dessa tarefa.

Neste trabalho, foi utilizada como base uma aplicação já existente, o “JanChart.ocx”. Esta aplicação era limitada, mas com algumas alterações foi possível tornar o diagrama de Gantt dinâmico, podendo este ajustar-se automaticamente ao número de períodos e de recursos utilizados, consoante os casos que fossem sendo resolvidos.

#### 4.2.5. Avaliação (*Evaluate*)

Após o escalonamento (Scheduling) é necessário avaliar (Evaluate) os resultados que se obtêm com a sequência inicial. Para este efeito, são utilizados os seguintes indicadores:

- Atraso Total (Total Tardiness);
- Atraso Máximo (Largest Tardy);
- Tempo de Setup (Setup Times);
- Tempos de Espera (Waiting Times);
- Número de Tarefas Atrasdas (Number of Tardy Jobs);
- Utilização dos Recursos (Capacity Utilization).

O módulo da avaliação é uma parte muito importante no escalonamento, pois é ele que nos permite calcular o valor das soluções para os objectivos que pretendemos otimizar. Estes valores permitem a comparação de soluções alternativas e estão na base dos algoritmos de melhoramento.

Os valores guardados para se poder avaliar o critério Atraso Total, são o resultado do somatório dos atrasos de todas as tarefas escalonadas. O atraso de cada tarefa consiste no número de períodos entre o período do prazo de entrega (Due Date) e o período onde a tarefa é terminada.

#### 4.2.6. Trocas (*Swap*)

Por fim, com vista a melhorar a solução de escalonamento produzida, o planeador pode seleccionar a função Trocas. Esta função consiste em efectuar repetidas trocas (*Swap*) entre pares de tarefas (*Jobs*) escolhidas aleatoriamente, de modo a gerar novas soluções.

O número de trocas que são efectuadas pela aplicação são predefinidas, podendo ser alteradas a qualquer momento. Para efeito dos testes realizados neste trabalho, este número foi de 100.

Nesta “pesquisa local”, primeiro é escolhida aleatoriamente uma das tarefas da sequência inicial, ou seja é gerado aleatoriamente um número entre um e o número total de tarefas. Após escolhida a primeira tarefa, procede-se à escolha da segunda tarefa, que deve estar contida dentro de um intervalo de vizinhança definido por  $\beta$ , que foi definido (para efeito de testes) como 2. Gera-se então um segundo número aleatório entre 0 e 1 que é multiplicado por  $\beta$ . De seguida gera-se um terceiro número aleatório entre 0 e 1, caso este terceiro número seja superior a 0,5, então o segundo número aleatório multiplicado por  $\beta$  será somado ao primeiro número aleatório. Caso seja menor, então será subtraído, desta operação resulta a segunda tarefa, que irá trocar de posição na sequência de ordenação com a outra tarefa anteriormente obtida. Daqui surge uma nova sequência que irá ser sujeita ao módulo de avaliação, e caso os valores obtidos com esta nova sequência, para os objectivos seleccionados inicialmente pelo planeador, sejam não dominados pelas soluções já anteriormente obtidas, então esta sequência é adicionada ao conjunto de soluções

Por outro lado, se os resultados obtidos por esta solução fizeram com que haja uma solução que seja dominada por esta, ou seja, que todos os valores, para os objectivos pretendidos, sejam piores (menores no caso de se pretender maximizar um objectivo, ou maiores no caso de se pretender minimizar um objectivo), então esta sequência toma o lugar dessa solução que é eliminada. Caso haja mais do que uma solução que seja dominada, todas essas soluções dominadas serão eliminadas e a nova sequência será apenas adicionada uma vez, de modo a ter soluções repetidas.

### 4.3. Exemplo

Para ilustrar a abordagem desenvolvida, em particular na utilização das regras no sequenciamento de tarefas e o seu escalonamento, irá ser utilizado um pequeno exemplo.

Consideram-se 3 tarefas, cada uma com 3 operações. Pretende-se escalonar essas operações em 4 máquinas diferentes. Nas tabelas 1 e 2, apresentam-se os dados para este exemplo:

Tabela 1: Tempos de processamento

	<u>Tempo de Processamento</u>			Data de entrega
	Operação 1	Operação 2	Operação 3	
Tarefa 1	3	2	1	10
Tarefa 2	2	3	2	14
Tarefa 3	1	2	3	12

Tabela 2: Máquinas onde são realizadas as operações

	<u>Máquina onde é processada</u>		
	Operação 1	Operação 2	Operação 3
Tarefa 1	1	2	4
Tarefa 2	3	2	3
Tarefa 3	2	1	4

Seguindo a abordagem adoptada, é necessário em primeiro lugar sequenciar as tarefas de modo a saber a ordem pela qual devem ser processadas. Para este exemplo vamos escolher a regra de prioridades EDD. Assim sendo, tendo em conta a Tabela 1, a primeira tarefa da sequência será a 1 (verde), de seguida a 3 (vermelho) e por fim a 2 (amarelo).

Ordenadas as tarefas vamos então proceder ao escalonamento. Como todas as tarefas estão disponíveis no instante  $t=0$ , não têm mais nenhuma restrição para além da sequência e do facto de não poderem ser interrompidas as operações.



Figura 16: Instante Inicial

Assim sendo, quando começamos a escalonar a tarefa verde todos os recursos estão disponíveis. Portanto a operação 1 da tarefa 1 (verde) é alocada na primeira posição e como o seu tempo de processamento é 2, ocupa os dois primeiros períodos.

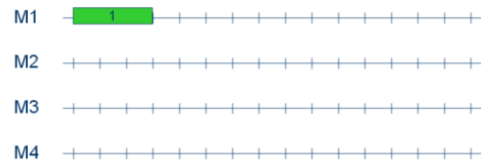


Figura 17: Primeira operação da primeira tarefa escalonada

Como os restantes recursos se encontram disponíveis, a operação seguinte começa logo no período a seguir a terminar a operação anterior e assim sucessivamente até todas as operações dessa tarefa estarem terminadas.

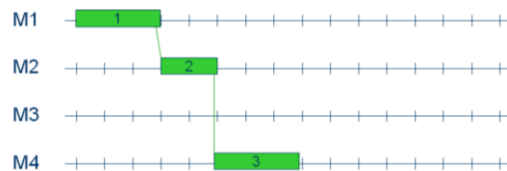


Figura 18: Resultado do escalonamento da primeira tarefa

Terminada a tarefa 1 (verde) passamos à tarefa 3 (vermelho). A primeira operação da tarefa vermelha é realizada na máquina 2 que já tem uma operação alocada à tarefa verde, mas como o tempo de processamento da operação 1 da tarefa 3 (vermelho) é inferior ao início de processamento da operação 2 da tarefa 1 (verde), então a operação 1 da tarefa 3 (vermelho) pode ser alocada no instante inicial e antes da operação 2 da tarefa 1 (verde).

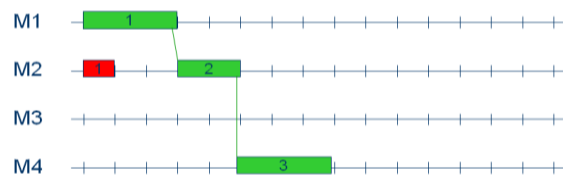


Figura 19: Primeira operação da tarefa 3 (vermelho)

De seguida a tarefa vermelha tem uma operação a realizar na máquina 1, mas como no instante em que termina a operação 1 da tarefa 3 (vermelho), a máquina está ocupada com a operação 1 da tarefa 1 (verde), então, a operação 2 da tarefa 3 (vermelho) terá que esperar que essa operação termine, como se pode ver na Figura 20.

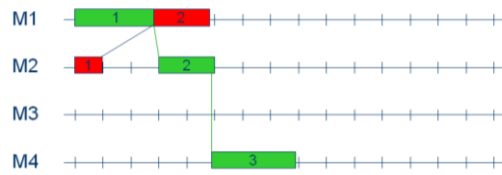


Figura 20: Espera da 2ª operação da tarefa 3 (vermelho)

Na operação 3 da tarefa 3 (vermelho), volta a acontecer que o recurso já se encontra ocupado no instante em que a operação 2 termina, obrigando a que a operação 3 tenha um tempo de espera.

No caso da tarefa 2 (amarelo), para a operação 1 não há nenhum problema, visto o

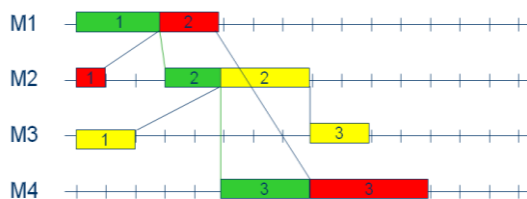


Figura 21: Escalonamento final

recurso estar disponível, mas quando esta operação termina, a operação 2 não pode começar imediatamente, apesar de o recurso estar disponível nesse período, porque se a operação 2 começasse nesse período iria colidir com a operação 2 da tarefa 1 (verde). Assim sendo, a operação 2 da tarefa amarela irá ter que esperar que a operação 2 da tarefa 1 (verde) termine. Depois a operação 3 da tarefa 2 (amarelo) não tem qualquer tipo de restrição, sendo então escalonada no período de tempo imediatamente a seguir ao instante em que termina a operação 2.

O processo de escalonamento adoptado no projecto é em tudo semelhante ao deste exemplo. Como se pode ver as operações são alocadas ao primeiro espaço disponível, evitando qualquer colisão que possa existir com outras tarefas já escalonadas e por consequência com maior prioridade.

#### 4.4. Aplicação desenvolvida

Nesta secção vamos apresentar as funcionalidades da aplicação resultante do trabalho desenvolvido. A interface da aplicação foi sofrendo diversas alterações ao longo do projecto, tendo em vista facilitar a utilização do sistema por parte do planeador. Pretende-se, em geral, disponibilizar diversas opções de escolha, mas sem que essas opções sejam demasiadas, de modo a que o SAD não venha a complicar o processo de decisão.

### a) *Seleccção das regras de prioridade*

Como tal começamos pela interface inicial, através da qual, como se pode ver na Figura 22, são dadas ao planeador diversas opções de escolha. Aqui ele deve proceder à escolha da heurística (algoritmo de sequenciamento) segundo a qual pretende que seja gerada a solução inicial. Por omissão, está seleccionada a regra de prioridade EDD.

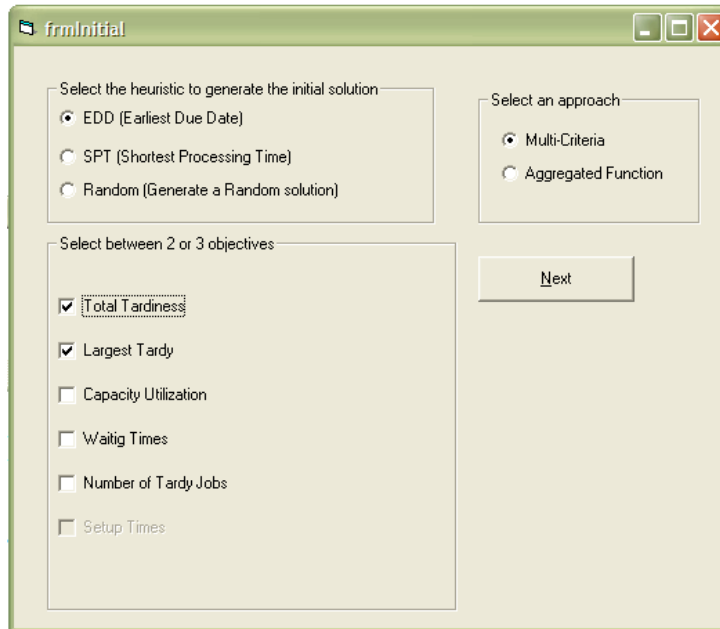


Figura 22: Vista da página inicial da aplicação desenvolvida

### b) *Seleccção dos objectivos*

Outra das opções disponibilizadas ao planeador é a opção de escolha dos objectivos/critérios que ele pretende melhorar. Aqui é pedido ao planeador para seleccionar entre 2 ou 3 objectivos, uma vez que a aplicação se destina a resolver problemas numa lógica multi-objectivo. Mas caso o planeador pretenda tratar apenas um objectivo, a aplicação irá na mesma resolver o problema, sendo que irá ser mostrada uma pequena janela (ver Figura 23) na qual se pergunta se o utilizador pretende continuar a resolver o problema com apenas um único objectivo, ou se apenas se tratou de um erro na escolha dos objectivos.

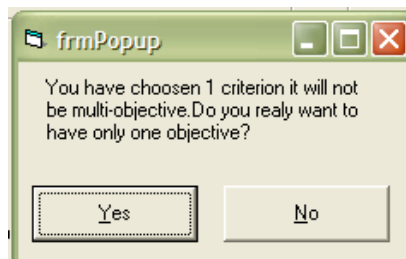


Figura 23: Janela de aviso no caso de ser seleccionado apenas um objectivo.

### c) *Seleccção da aproximação*

Para além da selecção dos objectivos, o planeador tem a opção de escolha da aproximação a seguir entre multi-critério ou função agregada. Multi-critério consiste em

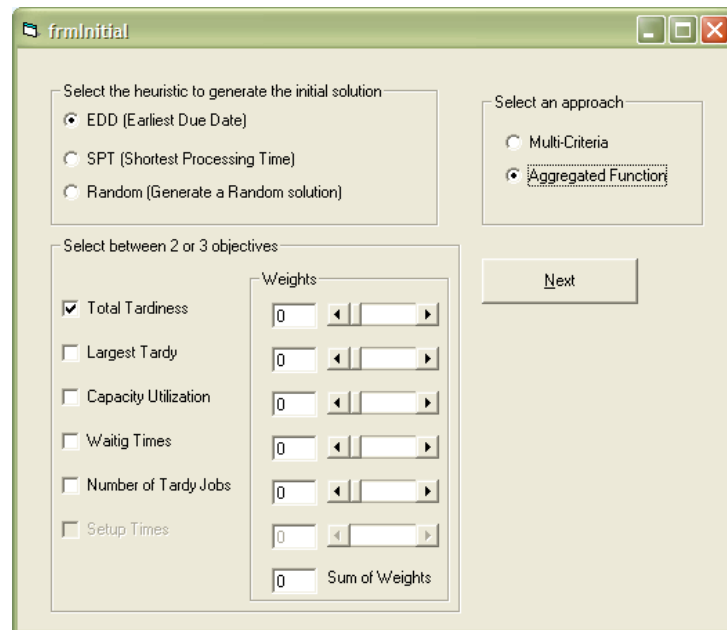


Figura 24: Página inicial onde aparece a atribuição de pesos aos objectivos

obter soluções de sequências que melhorem cada um dos objectivos seleccionados pelo planeador. No caso da função agregada, os valores dos objectivos seleccionados são normalizados, sendo atribuídos diferentes pesos a esses objectivos. Depois esses objectivos “pesados” são somados e daí resulta um valor que se pretende otimizar. Para tal surgem novas opções na janela, como se vê na Figura 24.

### d) *Atribuição dos pesos*

Nesta janela são ainda atribuídos os pesos relativos aos objectivos seleccionados. Estes pesos devem ser escolhidos deslocando a barra horizontal à direita de cada objectivo, ajustando-a para o valor pretendido do peso, sendo que o somatório dos pesos deverá ser igual a 1, caso contrário, a aplicação retorna uma mensagem de erro.

### e) *Escalonamento resultante*

Após a introdução destas opções, o planeador deverá carregar no botão *next*, surgindo uma nova janela onde é apresentado o diagrama de Gantt como resultado do escalonamento da sequência gerada pela heurística construtiva inicialmente definida pelo utilizador.

Juntamente com o diagrama de Gantt, são fornecidas diversas informações adicionais. Do lado esquerdo temos informação sobre o início e o fim da cada operação. No centro, em

cima, encontra-se uma tabela onde é possível verificar a ocupação dos recursos, de forma algo semelhante ao diagrama de Gantt, com a diferença de que em vez das barras a indicar as operações de cada tarefa, está o número da tarefa que em cada instante está a utilizar determinado recurso.

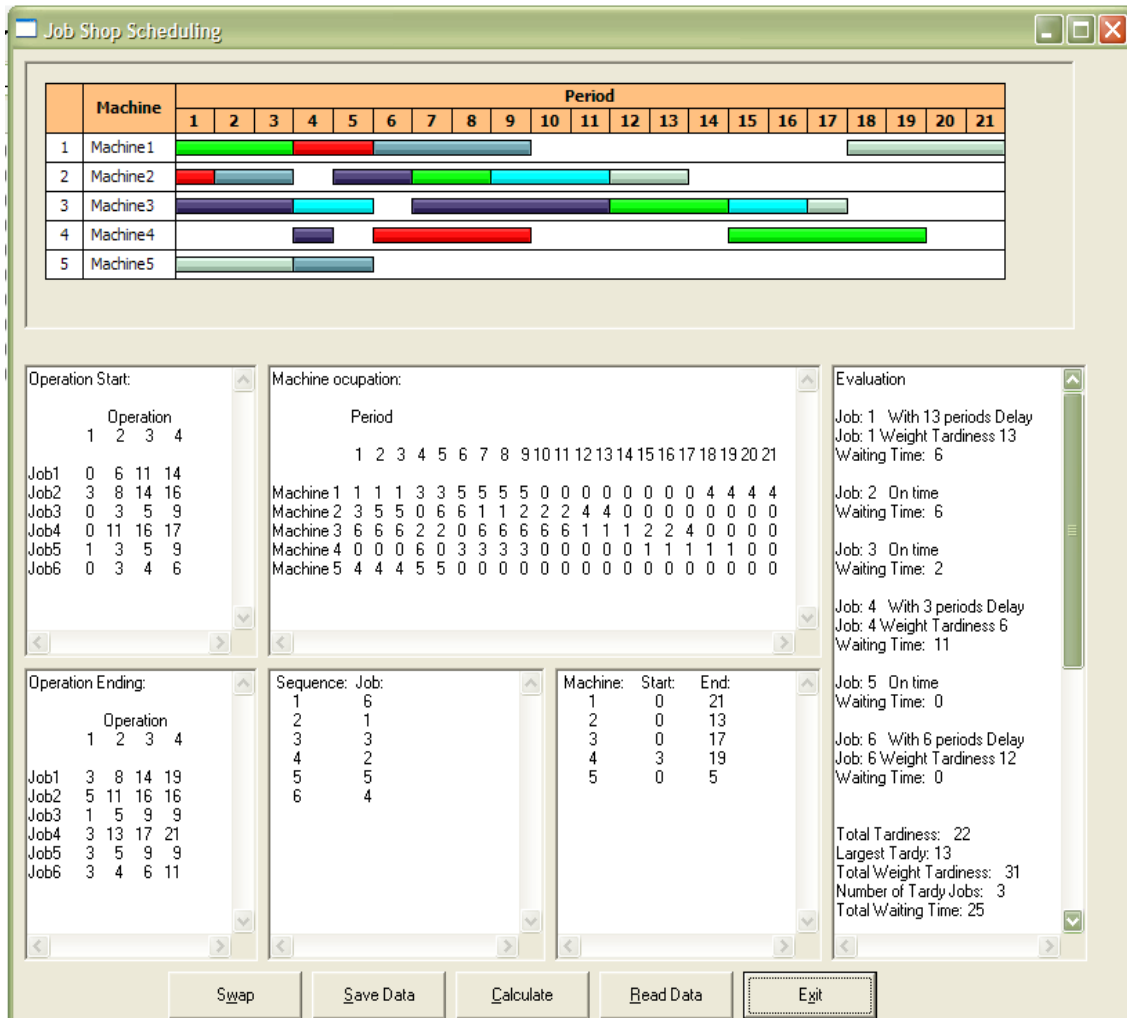


Figura 25: Página principal da aplicação

Ao centro à esquerda, temos a informação da sequência inicial construída pela heurística definida no início, pelo planeador. No centro à direita, está a informação relativa ao momento em que cada recurso começa e termina de ser utilizado.

Por fim, existe do lado direito, a informação relativa à avaliação de todos os dados resultantes deste escalonamento. Para as tarefas é possível ver o tempo de atraso relativamente ao prazo de entrega de cada tarefa, o tempo que cada tarefa esteve à espera para ser operada e ainda os valores de cada um dos objectivos possíveis, ou seja, do atraso total, do atraso máximo, do número de tarefas atrasadas, do tempo total de espera, da utilização dos recursos e do tempo total de setups. No fim desta lista estão os dados relativos aos recursos, com a informação sobre o tempo em que cada recurso esteve em utilização, o

tempo em que cada recurso esteve parado e informação sobre qual a percentagem de utilização desse recurso.

#### f) *Processo de Trocas*

Por selecção do planeador, quando este pressiona o botão “Swap”, surge uma nova janela onde o planeador dá início ao processo de trocas, sendo executado o procedimento anteriormente descrito no tópico 4.2.6. , com a realização das trocas entre duas tarefas

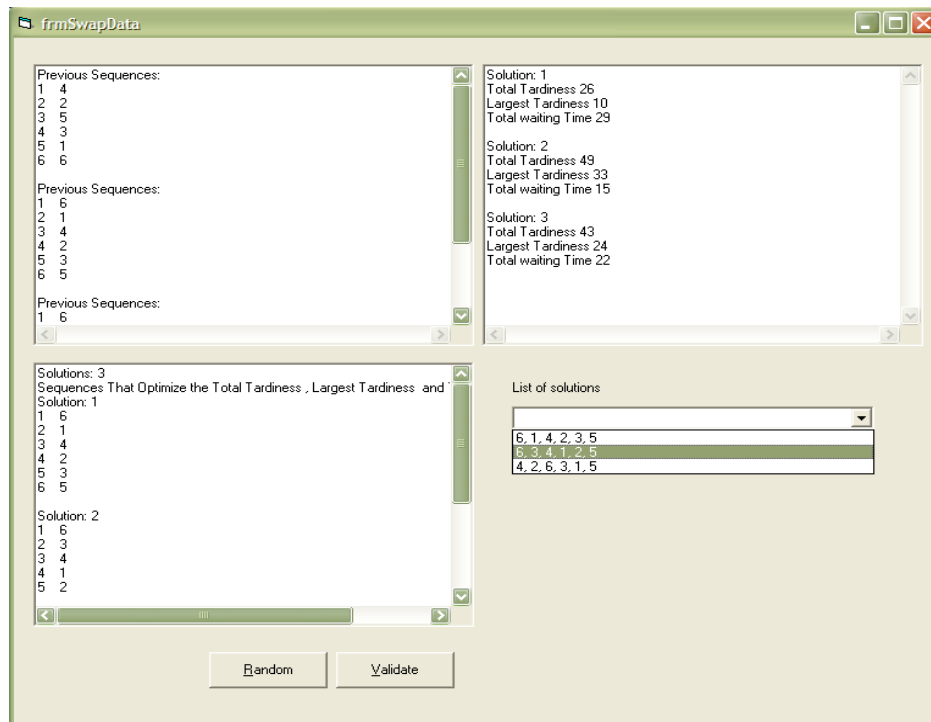


Figura 26: Vista geral da janela com um exemplo de um conjunto de soluções

aleatórias numa determinada vizinhança. Deste processo de pesquisa, resulta um conjunto de soluções que são apresentadas juntamente com os objectivos previamente seleccionados na janela inicial, como mostra a Figura 26. Nesta figura pode-se ver que apenas são mostrados, para cada solução, os 3 objectivos seleccionados, neste caso, o Atraso Total (*TotalTardiness*), o Atraso Máximo (*Largest tardy*) e o Tempo de Espera Total (*Total Waiting Time*) entre as operações de todas as tarefas de cada uma das soluções.

#### g) *Seleção de uma das soluções*

Agora o planeador tem a oportunidade de seleccionar, por meio de uma DropBox apresentada no lado direito da janela, qualquer uma das sequências do conjunto de soluções, e visualizar os valores dos restantes objectivos.

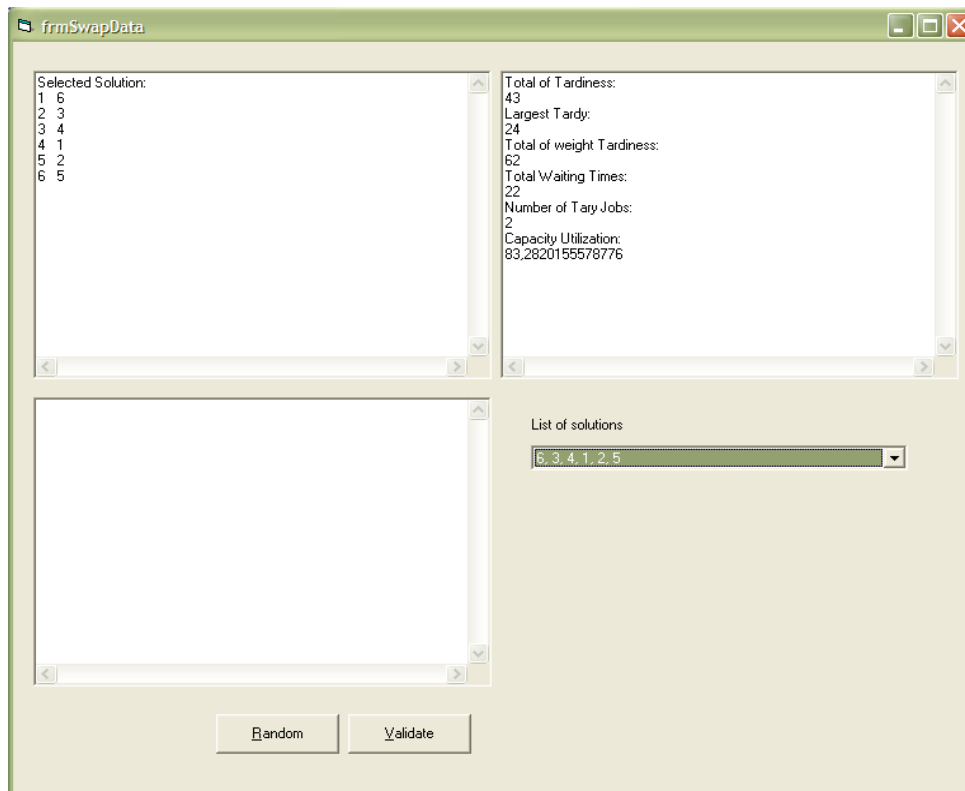


Figura 27: Vista detalhada de uma das seqüências do conjunto de soluções resultante

Nesta janela, o planeador pode alternar entre as diversas seqüências do conjunto de soluções e ir visualizando os valores dos objectivos. Ao validar uma dada seqüência, pressionando o botão “*validate*”, o planeador tem acesso novamente à janela principal, onde é possível visualizar o resultado do escalonamento da seqüência seleccionada de entre o conjunto de soluções.

### ***Exemplo ilustrativo***

A título de exemplo resolveu-se um problema de escalonamento muito simples, a seguir apresentado. Existem 3 tarefas a ser escalonadas, cada uma delas com 3 operações. Não são aqui considerados tempos de setup entre tarefas.

Tabela 3: Dados

	Tarefa 1		Tarefa 2		Tarefa 3	
	Máquina	Tempo de processamento	Máquina	Tempo de processamento	Máquina	Tempo de processamento
Operação 1	1	3	3	2	2	1
Operação 2	2	2	2	3	1	2
Operação 3	3	3	1	3	4	4

Na Tabela 3, está indicado o número da máquina / recurso onde cada operação de cada tarefa está alocada, e qual o respectivo tempo de processamento.

A Figura 28 mostra o escalonamento das Operações das Tarefas apresentadas na Tabela 3. A Tarefa 1 é representada pela cor Verde, a Tarefa 2 pela cor azul, e a Tarefa 3 pela cor vermelha. Neste pequeno exemplo não foram utilizadas para as heurísticas construtivas as datas de entrega de cada tarefa, pelo que, a sequência inicial, seguiu apenas a ordem da numeração das tarefas, ou seja, primeiro foi escalonada a tarefa 1, depois a tarefa 2 e por fim a tarefa 3.

	Machine	Period													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Machine1	Green			Red										
2	Machine2	Red			Green			Blue							
3	Machine3	Blue				Green									
4	Machine4									Blue		Red			

Figura 28: Resultado do escalonamento das tarefas apresentadas na Tabela 3

De seguida, consideram-se dois exemplos de dimensão e características diferentes, sendo o segundo destes exemplos mais próximo das aplicações encontradas na prática.

## 4.5. Exemplo 1

Considere-se uma situação envolvendo 6 tarefas, com até 4 operações, distribuídas por 5 máquinas. Neste exemplo, são utilizadas as regras de prioridade EDD, SPT e Random, para sequenciar as tarefas. Considera-se também que existem tempos de setup entre as várias tarefas.

### *Dados*

Tabela 4: Dados relativos ao primeiro exemplo com 6 tarefas

<i>Operação</i>	<i>Tarefa</i>	<i>Tempo</i>	<i>Máquina</i>	<i>Prazo de Entrega</i>
1	1	3	1	6
2	1	2	2	6
3	1	3	3	6
4	1	5	4	6
1	2	2	3	16
2	2	3	2	16
3	2	2	4	16
1	3	1	2	13
2	3	2	1	13
3	3	4	4	13
1	4	3	5	18
2	4	2	2	18
3	4	1	3	18
4	4	4	1	18
1	5	2	2	16
2	5	2	5	16
3	5	4	1	16
1	6	3	3	5
2	6	1	4	5
3	6	2	2	5
4	6	5	1	5

Na Tabela 4, estão representados os dados que associam cada operação de cada tarefa a uma máquina e a um tempo de processamento. Como se pode verificar, ao se acrescentar apenas mais quatro tarefas, a tabela cresceu bastante, o que leva a antever que, com valores de um caso real, em que se chegam a escalonar largas centenas, e muitas vezes alguns milhares de tarefas, as matrizes que contêm estas relações possam ser de grande dimensão.

### **Setups**

Neste exemplo, vamos ainda considerar uma outra matriz que contém os dados relativos aos tempos de setup em cada máquina para a preparação da transição entre as diferentes tarefas de acordo com as suas características.

Tabela 5: Extracto da matriz de setups para o exemplo 1

<b>Setup</b>				
<b>Recurso</b>	<b>Característica Inicial</b>	<b>Característica Seguinte</b>	<b>Tempo de Setup</b>	
1	1	1	1	0
1	1	1	2	1
1	1	1	3	4
1	1	1	4	2
1	1	1	5	1
1	1	1	6	3
1	1	1	7	2
1	2	1	1	3
1	2	2	2	0
1	2	2	3	1
1	2	2	4	4
1	2	2	5	2
1	2	2	6	1
1	3	3	7	2
1	3	3	1	5
1	3	3	2	5
1	3	3	3	0

Na Tabela 5 apenas estão alguns dados relativos a um dos recursos utilizados na realização das diversas operações, uma vez que a matriz completa tem 282 linhas Tendo em conta estas matrizes de valores foi executada a aplicação, seleccionando a Regra de Prioridade EDD e três objectivos: o Tempo Total dos Atrasos, o Tempo Total em espera entre operações e o número de tarefas atrasadas.

### Sequenciamento e escalonamento

Da sequenciação por meio da Regra de Prioridade EDD resulta que a ordem das tarefas na seqüência é a seguinte: 6,1, 3, 2, 5, 4. O resultado do escalonamento pode ser visualizado no diagrama de Gantt da Figura 29. Neste diagrama observamos que os setups são representados pela cor cinzenta.

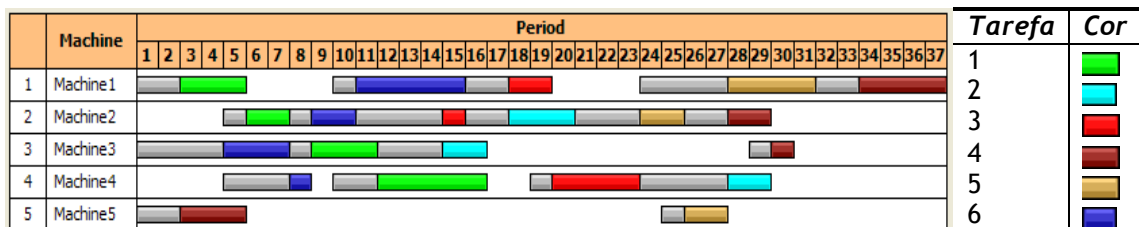


Figura 29: Diagrama de Gantt relativo ao exemplo 1

Foi definido que os setups deveriam ser programados para que quando uma operação de uma tarefa termina, o setup para a operação seguinte esteja terminado, de modo a que a operação seguinte possa começar logo que possível. É claro que isto só acontecerá caso o recurso onde a operação seguinte irá ser realizada esteja livre, senão o setup dessa operação deverá aguardar pelo momento em que seja disponibilizado o recurso para então alocar essa operação ao recurso.

Similarmente ao diagrama de Gantt temos a matriz de ocupação dos recursos, que nos permite observar os períodos de tempo em que cada recurso está alocado a uma determinada tarefa.

Tabela 6: Ocupação dos recursos no exemplo 1

Períodos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	
Recurso 1	1	1	1	1	1	0	0	0	0	6	6	6	6	6	6	3	3	3	3
Recurso 2	0	0	0	0	5	5	5	5	5	5	5	5	4	4	4	4	4	4	
Recurso 3	0	0	0	0	1	1	1	6	6	6	3	3	3	3	3	2	2	2	2
Recurso 4	6	6	6	6	6	6	6	1	1	1	1	2	2	2	2	2	0	0	0
Recurso 5	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	
Recurso 6	0	0	0	0	6	6	6	6	0	1	1	1	1	1	1	1	0	0	3
Recurso 7	3	3	3	3	2	2	2	2	2	2	0	0	0	0	0	0	0	0	
Recurso 8	4	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Recurso 9	0	0	0	0	0	5	5	5	0	0	0	0	0	0	0	0	0	0	

Na Tabela 6, são apresentados os dados relativos à ocupação dos recursos: as linhas de cor cinzenta mais escura representam os períodos de 1 a 19, e a cinzento mais claro, representam os períodos de 20 a 37, que é o período onde termina a última operação da última tarefa escalonada. Esta divisão foi necessária de modo a se poderem visualizar na totalidade os dados para este exemplo.

Nesta matriz os setups e as operações são assinalados com o número da tarefa correspondente. Ou seja, nos períodos em que determinado recurso esteja alocado, ou ao setup ou à operação de determinada tarefa, o número desta tarefa é colocado na posição relativa ao período correspondente.

### **Avaliação**

Quanto aos objectivos seleccionados pelo planeador, os resultados são apresentados na seguinte Tabela 7.

Tabela 7: Avaliação dos resultados

<i>Tarefa</i>	<i>Tempo de Atraso (em períodos)</i>	<i>Tempo de espera entre operações (em períodos)</i>
Tarefa: 1	10	1
Tarefa: 2	Sem atraso	8
Tarefa: 3	Sem atraso	2
Tarefa: 4	19	25
Tarefa: 5	Sem atraso	0
Tarefa: 6	10	0
Total	39	36
Tarefas em atraso	3	

Com base nestes valores, o planeador pode fazer uma análise acerca dos objectivos que definiu inicialmente e daqui poderá concluir que, por exemplo, um dos motivos pelo qual a tarefa 4 é entregue com atraso de 19 períodos poderá ser o facto de, entre operações, esta tarefa ter 25 períodos de tempo de espera.

Por exemplo, relativamente à tarefa 6, verifica-se que não existem tempos de espera entre as operações, mas no entanto essa tarefa é entregue com 10 períodos de atraso, e uma vez que esta tarefa foi a primeira a ser escalonada (resultado do sequenciamento inicial - EDD), isto por antever que existem problemas na definição dos prazos de entrega. Poderá assim existir um problema na interligação entre sectores da empresa nomeadamente entre a produção e as vendas, uma vez que o prazo de entrega não foi possível de ser cumprido.

Ainda relativamente aos dados sobre as tarefas e suas operações, existem ainda mais dois quadros que indicam os períodos de início e fim da cada operação de cada tarefa, (ver Tabela 8).

Nesta tabela são apenas tidos em conta os valores dos períodos em que uma operação de uma tarefa começa ou termina de ser realizada num recurso, não sendo tidos em conta os períodos de tempo em que está a ser realizado o setup do recurso.

Relativamente aos dados, por exemplo, na tarefa 6, verificamos que todos os períodos de início de cada operação coincidem com o fim da operação anterior, podendo então concluir-se que, como já tínhamos verificado na Tabela 7, não existem tempos de espera entre operações para a tarefa 6.

Tabela 8: Dados relativos ao início e fim das operações

	Operações							
	1	2	3	4	1	2	3	4
<b>Tarefa1</b>	2	5	8	11	5	7	11	16
<b>Tarefa2</b>	14	17	27	29	16	20	29	0
<b>Tarefa3</b>	14	17	19	23	15	19	23	0
<b>Tarefa4</b>	2	27	29	33	5	29	30	37
<b>Tarefa5</b>	23	25	27	31	25	27	31	0
<b>Tarefa6</b>	4	7	8	10	7	8	10	15
	<b>Início da Operação</b>				<b>Fim da Operação</b>			

### **Recursos**

Outra informação interessante que resulta também do escalonamento é acerca dos dados relativos aos recursos. Neles podemos analisar vários indicadores como por exemplo, o tempo durante o qual o recurso se encontra em processamento (Busy), o tempo em que se encontra parado (Empty) e ainda a relação entre o período de tempo em que o recurso esteve disponível e o tempo em que esteve em utilização (Usage), expressa em percentagem.

Estes dados mais uma vez podem ser importantes para o planeador, permitindo verificar se há recursos em subutilização, como por exemplo o recurso 5, que desde o período em que inicia a primeira operação e o período em que termina a última operação, apenas é utilizado em 8 períodos (o que dá uma relação inferior a 30% entre o tempo de utilização e o tempo em que está sem operar). Ao contrário do recurso 5, temos o recurso 2, que está sempre em utilização, apenas parando para efectuar os setups entre tarefas.

Tabela 9: Transcrição dos dados relativos à utilização dos recursos

<i>Recurso</i>	
1	29 períodos em processamento
	8 períodos em vazio
	78,37% de utilização
2	25 períodos em processamento
	0 períodos em vazio
	100% de utilização
3	18 períodos em processamento
	12 períodos em vazio
	60% de utilização
4	22 períodos em processamento
	3 períodos em vazio
	88% de utilização
5	8 períodos em processamento
	19 períodos em vazio
	29,62% de utilização

### **Melhoramentos**

Analisados estes dados, o planeador tem a possibilidade de utilizar o módulo de Trocas (Swap), com o qual se procura encontrar sequências que constituam melhores soluções relativamente aos 3 objectivos seleccionados no início.

Terminado o processo de trocas, é obtido o primeiro conjunto de soluções (ver Tabela 10).

Tabela 10: Primeiro conjunto de soluções

<i>Sequência</i>	<i>Objectivos</i>
1 6, 4, 1, 2, 3, 5	Atraso Total 26 Tempo Total de Espera 29 Número de Tarefas em Atraso 3
2 6, 4, 3, 1, 5, 2	Atraso Total 49 Tempo Total de Espera 27 Número de Tarefas em Atraso 3

Comparando os valores dos objectivos da solução inicial com os valores de cada uma das sequências do conjunto de soluções, podemos ver que a solução inicial é dominada pela sequência 1, uma que vez que tanto para o Tempo de Atraso Total, como para o Tempo Total

de Espera, obtemos piores resultados com a solução inicial do que com as sequência 1. Como o terceiro objectivo, o número de tarefas atrasadas, possui o mesmo valor para as duas, então a sequência 1 toma lugar no conjunto de soluções e a solução inicial deixará de constar nesse conjunto.

Relativamente à segunda sequência, verificamos que para o objectivo Tempo Total de Espera existe uma melhoria do valor relativamente à sequência 1, mas para o objectivo relativo ao Tempo de Atraso Total obtemos um valor superior, logo ambas as soluções são não dominadas, fazendo por isso as duas parte do conjunto de soluções para análise.

Neste momento o planeador poderá repetir o processo das trocas ou decidir escolher uma das sequências do conjunto de soluções e visualizar mais detalhes sobre o escalonamento dessa sequência. Caso o planeador pretenda pode então validar essa sequência e ter acesso a todos os dados anteriormente apresentados, ou seja, o diagrama de Gantt, a matriz de ocupação dos recursos, as matrizes de início e fim das tarefas, o valor dos objectivos, etc.

Na tentativa de proceder a melhorias nas soluções já encontradas, repetimos o processo de trocas. Surgiu então um novo conjunto de soluções não dominadas, e as soluções já existentes foram retiradas do conjunto, uma vez que eram dominadas pelo menos por uma das soluções do novo conjunto.

Tabela 11: Segundo conjunto de soluções

Sequência	Objectivos
1	1, 4, 2, 5, 6, 3
	Atraso Total 38
	Tempo Total de Espera 25
2	3, 4, 2, 1, 6, 5
	Atraso Total 47
	Tempo Total de Espera 13
3	1, 4, 6, 2, 5, 3
	Atraso Total 31
	Tempo Total de Espera 33
4	4, 2, 6, 1, 5, 3
	Atraso Total 43
	Tempo Total de Espera 22
5	6, 1, 2, 3, 4, 5
	Atraso Total 26
	Tempo Total de Espera 13
6	4, 1, 6, 2, 3, 5
	Atraso Total 37
	Tempo Total de Espera 28
	Número de Tarefas em Atraso 2

Podemos verificar através da Tabela 11 que existe apenas uma solução (sequência 5) em que o objectivo de minimizar o número de tarefas em atraso (Number of Tardy Jobs) se

mantém em 3, mas essa solução é não dominada uma vez que, apesar de ser pior que todas as outras soluções para esse objectivo, é bastante melhor nos outros objectivos. A solução 5 é a que tem menor valor para o objectivo do atraso total (Total Tardiness) e iguala a menor no objectivo do tempo total de espera (Total Waiting Time).

### Comparação

Na Figura 30 podemos fazer uma comparação entre todas as soluções que, para o objectivo do número de tarefas em atraso, apresentam um valor de 2 tarefas em atraso. Aqui podemos ver que, de entre elas, não existe uma solução cujo par de objectivos (Atraso Total e Tempo Total de Espera) domine as outras soluções - são as soluções não dominadas encontradas pela aplicação.

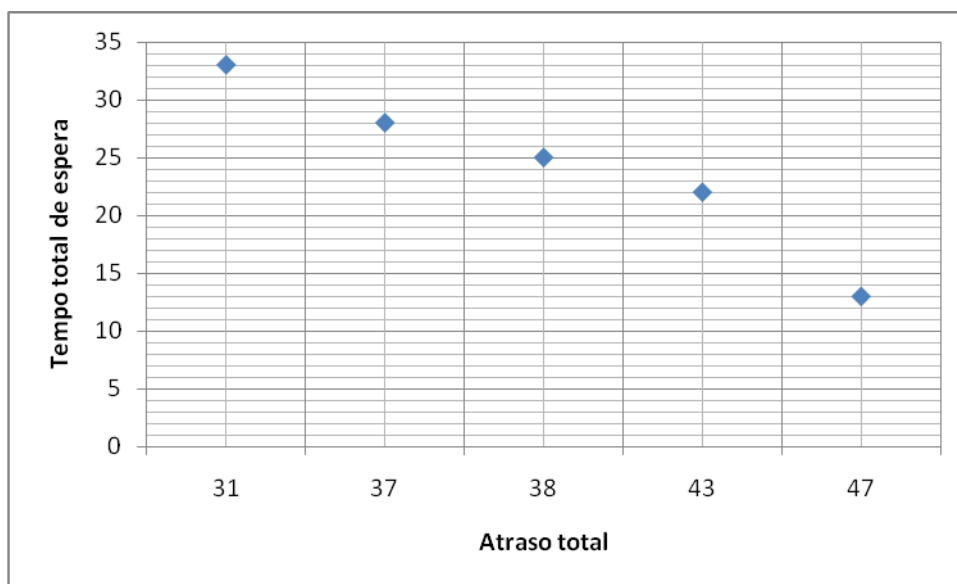


Figura 30: Soluções não dominadas (com 2 tarefas em atraso)

Nenhuma das soluções domina as outras, pois não há nenhuma que seja melhor em todos os objectivos. Com a apresentação deste conjunto de soluções, cabe ao planeador a escolha sequenciado escalonamento a utilizar. Ele pode, por exemplo, querer seleccionar uma sequência que corresponda a um compromisso natural entre os dois objectivos, como é o caso da solução (38, 25), que não sendo a melhor de todas em nenhum dos objectivos, se situa num ponto intermédio interessante.

Mas o planeador poderá eventualmente valorizar mais um dos objectivos e considerar que o outro não é assim tão importante, seleccionando, por exemplo, a solução (47,13). Esta solução é a melhor para a minimização do atraso total e a pior para o objectivo “minimizar o tempo total de espera”.

É possível então, escolher uma das sequências, validá-la, verificar o seu escalonamento e mesmo assim não estando satisfeito com esse escalonamento, o planeador, pode novamente

dar início ao processo de trocas e tentar procurar novos resultados, sem nunca correr o risco de perder qualquer das soluções não dominadas até agora encontradas.

## 4.6. Exemplo 2

Por fim a aplicação foi testada com uma instância de um problema construído a partir de um caso real. Os valores usados nesta instância são dados relativos a uma empresa do ramo dos plásticos. Assim sendo o primeiro passo foi a conversão dos dados, de XML para a base de dados Access. A quantidade de dados revelou ser um problema uma vez que, no caso real, estamos a falar de 777 tarefas, o que se traduz numa matriz das operações com mais de 2500 linhas e uma matriz de setups com mais de 17000 valores. Como tal optámos por reduzir a quantidade de dados a usar no teste. Reduziu-se a quantidade de tarefas para 50 e fez-se corresponder cada minuto a um período.

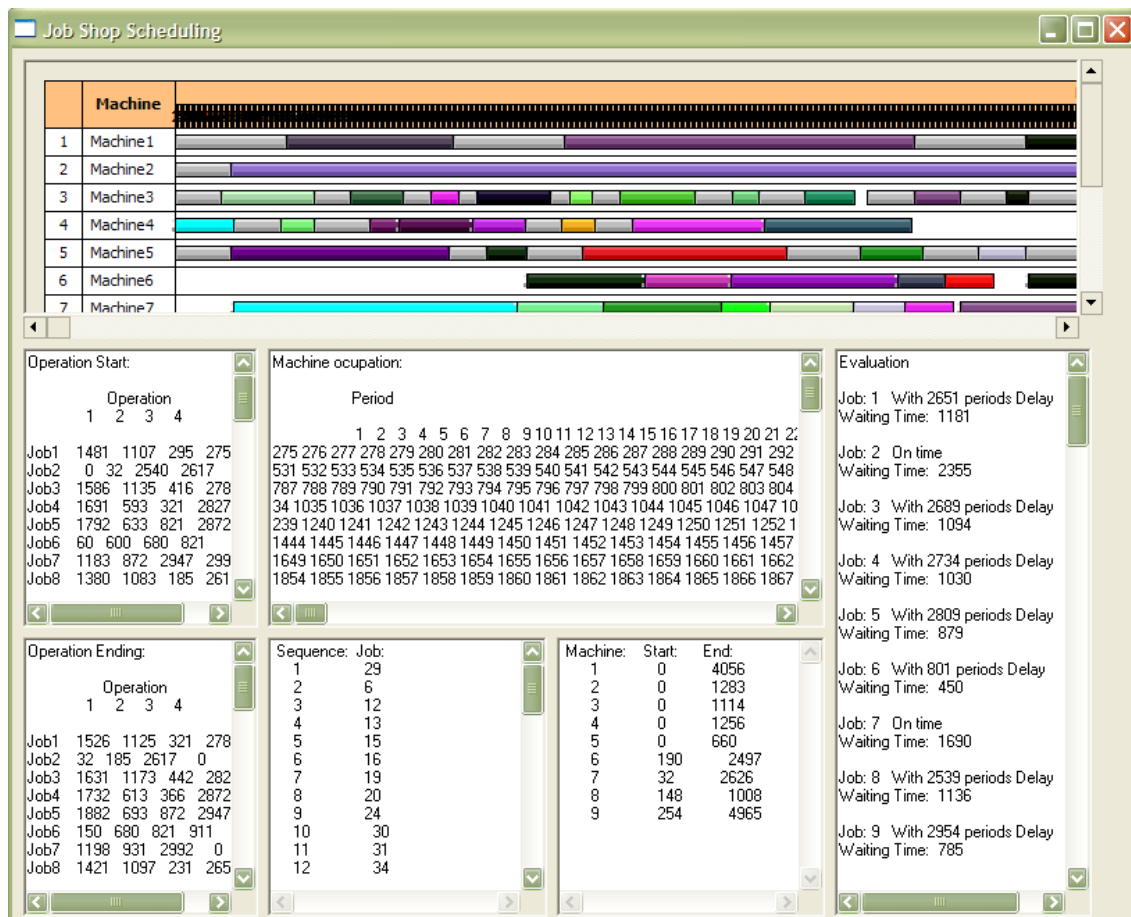


Figura 31: Resultados obtidos para uma instância retirada de um caso real

Assim sendo, e seleccionando como heurística construtiva a regra de prioridade EDD, obteve-se a seguinte sequência inicial: 29, 6, 12, 13, 15, 16, 19, 20, 24, 30, 31, 34, 37, 42, 49, 2, 8, 22, 41, 1, 3, 4, 5, 7, 9, 10, 11, 14, 17, 18, 21, 23, 25, 26, 27, 28, 32, 33, 35, 36, 38, 39, 40, 43, 44, 45, 46, 47, 48, 50.

Tendo-se seleccionado no início como objectivos a minimização do Total dos Atrasos e o Atraso Máximo, obtivemos o valor de 85350 minutos para o Atraso Total, o que corresponde a um somatório de todos os atrasos de 59 dias e 6 horas. O valor obtido para o Atraso Máximo foi de 4827 minutos, ou seja, 3 dias e 8 horas de atraso.

Num caso real como aquele em que foi baseado este exemplo, muitas das vezes não existe tempo de setup entre operações consecutivas, uma vez que dentro das tarefas realizadas (neste caso 50), existem muitas que possuem as mesmas características. Este facto é possível de verificar na Figura 31, onde se verifica que existe o escalonamento de operações seguidas sem setup, ou melhor, com tempo de setup nulo entre elas.

De novo, dá-se início ao processo de trocas no sentido de atingir melhorias para os objectivos seleccionados de início. Na Tabela 12, estão representados os valores dos objectivos do conjunto de soluções obtido através do processo de trocas.

Nesta tabela, não é transcrita a sequência completa das tarefas, sendo apenas indicado como referência o número da solução que representa essa sequência, uma vez que a quantidade de dados tornaria a apresentação demasiado confusa e o que realmente é importante apresentar, neste contexto, são os valores dos objectivos das soluções obtidas pelo processo de trocas.

Tabela 12: Resultado do processo de trocas para o segundo exemplo

<i>Solução</i>	<i>Objectivos</i>
1	Atraso Total 60805 Atraso Máximo 4375
2	Atraso Total 64704 Atraso Máximo 4327
3	Atraso Total 61681 Atraso Máximo 4358
4	Atraso Total 51686 Atraso Máximo 4385
5	Atraso Total 50998 Atraso Máximo 4608
6	Atraso Total 48892 Atraso Máximo 4614
7	Atraso Total 62396 Atraso Máximo 4346
8	Atraso Total 64416 Atraso Máximo 4345
9	Atraso Total 61925 Atraso Máximo 4349

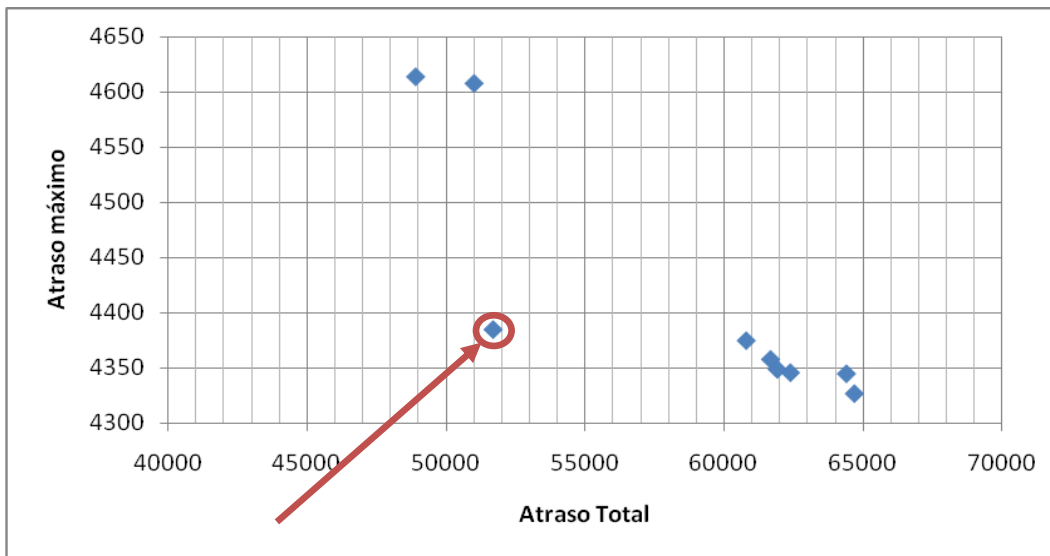


Figura 32: Gráfico das soluções obtidas

Na Figura 32 podemos verificar que quando os valores para um dos objectivos são bastante elevados, normalmente tomam valores mais baixos no outro objectivo. Esta representação permite no entanto evidenciar uma solução que, apesar de não ser nunca a melhor em qualquer dos objectivos, se destaca pelo facto de conseguir ter um valor relativamente baixo para ambos os objectivos.

A decisão quanto à solução a ser adoptada dependerá do que o planeador achar mais importante. Ele poderá querer optar pela solução mais equilibrada e seria a identificada a vermelho, ou poderá optar por qualquer uma das outras, dependendo do que o planeador ache que possa vir a ser mais ou menos interessante na situação em causa.

A aplicação cumpre assim o seu papel de fornecer os dados necessários para que o planeador possa tomar as melhores decisões.

# Capítulo 5

## Conclusões e desenvolvimentos futuros

O escalonamento das operações é um dos aspectos fundamentais na gestão da produção numa empresa industrial. Sem uma adequada alocação no tempo das tarefas (ordens de fabrico) às máquinas, podemos estar a desperdiçar enormes quantidades de tempo (recursos), com uma clara diminuição da produtividade e prejuízos elevados. Os resultados apresentados ilustram como se podem melhorar soluções neste domínio.

O trabalho apresentado nesta dissertação enquadra-se num projecto mais vasto de desenvolvimento de uma ferramenta de escalonamento (Izaro Grey) integrada em sistemas mais alargados de gestão usados em ambientes industriais.

No decorrer do desenvolvimento do Izaro Grey, foram sendo acrescentadas diversas funcionalidades ao escalonador, tornando-se difícil proceder a testes sobre os aspectos mais essenciais da abordagem adoptada, baseada em duas fases (primeiro, Sequenciar e depois, Escalonar as tarefas). Como tal, surgiu a necessidade de desenvolver um SAD (Sistema de Apoio à Decisão) que, utilizando a abordagem “Sequenciar/Escalonar”, não considerasse as restrições adicionais que existem no Izaro Grey, permitindo assim avaliar de uma forma dirigida as especificidades dessa abordagem.

Para além disso, o sistema criado tem a capacidade de, após ser gerada uma primeira solução, desencadear um processo de melhorias, do qual resulta um conjunto de soluções que se espera correspondam a bons compromissos entre múltiplos objectivos considerados.

Deste modo, um SAD como o desenvolvido neste trabalho, e eventualmente integrado num sistema de planeamento mais abrangente, como o Izaro-Grey, pode ser de grande importância para o escalonamento das operações. Esta ferramenta permite ao planeador o uso de múltiplos critérios e, visto que no mundo real o que se pretende é otimizar vários objectivos em simultâneo, podemos, por exemplo, querer minimizar o número de tarefas em

atraso, mas ao mesmo tempo maximizar a utilização dos recursos. Estes objectivos, muitas das vezes caminham em sentidos opostos e tornam as escolhas dos planeadores muito difíceis, justificando-se assim a utilização de um SAD como o desenvolvido neste projecto.

Ao abordar o problema, começando por sequenciar as tarefas segundo uma regra de prioridades, está-se desde logo a tentar otimizar os objectivos pretendidos, isto é, ao ordenar as tarefas numa determinada sequência, o ponto de partida para o processo de melhorias é, à partida, satisfatório.

Por outro lado a abordagem multi-critério parece ter grande potencial, permitindo apresentar ao planeador um conjunto de soluções que vai de encontro às necessidades por ele definidas no início do planeamento. Esse conjunto de soluções, apresenta apenas as soluções não dominadas, ou seja, aquelas que correspondem a compromissos entre os objectivos considerados.

Como complemento à utilização de um esquema de trocas como processo de melhorias, poder-se-ia implementar no futuro uma abordagem meta-heurística multi-objectivo. Novas abordagens deste tipo permitiriam uma análise comparativa dos algoritmos, em termos de eficiência e de eficácia, possibilitando o desenvolvimento de um método mais geral e robusto.

Outro ponto que justifica um desenvolvimento futuro é a optimização de código de modo a poder melhorar os tempos de execução, principalmente da parte gráfica do diagrama de Gantt.

## Referências

Baker, Kenneth; “Heuristic Procedures for Scheduling Job Families with Setups and Due Dates”, 1999.

Baker, Kenneth; Trietsch, Dan; “Principles of Sequencing and Scheduling”, Wiley, 2009

Blazewicz, J.; Ecker, K.; Pesch, E.; Schmidt, G.; Weglarz, J.; “Handbook on Scheduling, from theory to Applications”, Springer, 2007.

Brucker, Peter; “Scheduling Algorithms”, Springer, 2004.

Chase, Richard B.; Jacob, F. Roberts.; Aquilano, Nicholas J.; “Operations Management for Competitive Advantage”, Mc-Graw Hill, 2006.

Fernandes da Costa, Lino António Antunes Algoritmos; “Evolucionários em Optimização Uni e Multi-objectivo”, Braga, Universidade do Minho, 2003.

Koh, Shie-Gheun; Koo, Pyung-Hoi; Ha, Jae-won; Lee, Woon-Seek; “Scheduling parallel batch processing machines with arbitrary job sizes and incompatible job families”, 2004.

Krajewski, Lee J.; Ritzman, Larry P.; “Operations management - strategy and Analysis”, Addison-Wesley, 1996.

Kuehn, Wolfgang; Draschba, Clemens; “Simulation Based Job Shop production analyser”; 2004.

Lourenço, Helena Ramalinho Dias; “A Computational Study of the Job-Shop and the Flow-Shop Scheduling Problems”, Cornell University, 1993.

Madureira, Ana Maria Dias; “Aplicações de Meta-Heurísticas em Problemas de Sequenciamento”, Porto, 1995.

Madureira, Ana Maria; Pinho de Sousa, Jorge; “Aplicações de Meta-Heurísticas em Problemas de escalonamento de uma única máquina”, Investigação Operacional/Vol. 16, 1996, pp 115-133.

Madureira Pereira, Ana Maria Dias; “Aplicação de Meta-Heurísticas ao Problema de Escalonamento em Ambiente Dinâmico de Produção Discreta”, Braga, Universidade do Minho, 2003.

Oey, Kasin; Mason, Scott J.; “Scheduling batch processing machines in complex job shops”, 2001.

Ozlen, Melih; Azizoglu, Meral; “Multi-objective integer programming: A general approach for generating all non-dominated solutions”, 2000.

Pinedo, Michael L.; “Scheduling - Theory, Algorithms, and Systems”; Springer, 2008.

Pinho de Sousa, Jorge; Apontamentos da disciplina Gestão de Operações, Faculdade de Engenharia da Universidade do Porto, 2008.

Pinho de Sousa, Jorge; Guardão, Luis; “Inovar nas Ordens de fabrico”, TecnoMetal, 2008.

Ripon, Kazi Shah Nawaz; Tsang, Chi-Ho; Kwong, Sam; “Multi-Objective Evolutionary Job-Shop Scheduling Using Jumping Genes Genetic Algorithm, 2006.

T’kindt, Vincent; Billaut, Jean-Charles, “Multicriteria Scheduling - Theory, Models and Algorithms”, Springer, 2006.

Tra, Niem-Trung; “Comparison of Scheduling Algorithms for a Multi-Product Batch-Chemical Plant with a Generalized Serial Network”, 2000.

Viana, Ana; “Aplicação de Meta-heurísticas Multiobjectivo ao Problema de Sequenciamento de Actividades com Restrições de Recursos”, Porto, Faculdade de Engenharia da Universidade do Porto, 1997.

Viana, Ana; “Metaheuristics for the Unit Commitment Problem - The constraint Oriented Neighbourhoods Search Strategy”, Faculdade de Engenharia da Universidade do Porto, 2004.

## Sites

<http://www.softi9.pt>

<http://www.preactor.com/Home.aspx>

<http://www.stern.nyu.edu/om/software/lekin/index.htm>