



# **SISTEMAS DISTRIBUÍDOS DE PRODUÇÃO EM AMBIENTE DE PRODUÇÃO SIMULTÂNEA**

Tese submetida à Escola de Engenharia da Universidade do Minho para obtenção do grau de Doutor em Engenharia de Produção e Sistemas, sob orientação científica do Doutor Sílvio do Carmo Silva, Professor Associado da Universidade do Minho.

Rui Manuel Lima  
2003



## **AGRADECIMENTOS**

Ao Professor Sílvio do Carmo Silva pelo empenho e rigor colocados em momentos chave deste projecto.

A todos os que profissionalmente mostraram, directa e indirectamente, que acreditavam neste trabalho.

Ao Paulo e ao Rui pelo companheirismo e apoio efectivo.

Aos meus pais, um eterno agradecimento. Sinto profundamente que tudo o que atinjo tem algo deles.

Ao Jorge, Fernando e Nelson pela cumplicidade permanente.

À Susana pelas palavras certas no momento certo.

À minha família e aos meus amigos.

À Cristina pela energia que transmite e por acreditar, mesmo quando parece que mais ninguém acredita.



## RESUMO

Novos paradigmas de produção emergem na economia global, aparentemente baseados em três tendências principais. Estas tendências são: aumento da dinâmica de reconfiguração dos sistemas de produção, distribuição dos recursos do sistema de produção, e virtualidade dos sistemas, i.e. sistemas que são configurados em termos organizacionais, sem alteração da implantação física dos equipamentos. A primeira tendência deve-se ao aumento de requisitos do produto, nomeadamente ao nível da qualidade, variedade e entrega. A segunda tendência resulta de uma necessidade de melhoria do desempenho do sistema de produção, e a terceira, deve-se à impossibilidade de agrupamento físico de recursos de produção distribuídos globalmente e às possibilidades oferecidas pelos sistemas de comunicação existentes.

Com base nas referidas tendências surgem novos requisitos de projecto e operação de sistemas de produção. Podem-se seleccionar alguns destes requisitos, resumidos da seguinte forma: *aumento da reconfigurabilidade do sistema baseada em recursos de produção distribuídos e na entrega rápida dos produtos encomendados*. Com base nestes requisitos, apresenta-se uma definição e desenvolve-se um modelo de Sistema Distribuído de Produção (SDP), denominado MSDP. Neste modelo existe uma relação estreita entre a estrutura genérica do sistema distribuído de produção e a estrutura de transformações de estado do produto encomendado. Desenvolve-se ainda um modelo conceptual baseado em agentes de software do MSDP, e uma representação formal do modelo com base em Redes de Petri e em mecanismos de modelação apresentados por Lecompte, Deschamps and Bourrières (2000).

O modelo formal desenvolvido é aplicado a várias configurações de sistemas, relacionadas com a estrutura de algum produto encomendado. Esta aplicação permite verificar a possibilidade de utilização do modelo para representar diferentes SDPs e para estudar o comportamento de SDPs de acordo com a filosofia de Produção Simultânea. Esta filosofia tem como principal objectivo a redução do prazo de entrega de encomendas através da redução do tempo de percurso de fabrico de cada encomenda.

O estudo realizado neste trabalho permite concluir que o modelo pode ser utilizado para projecto e avaliação de sistemas distribuídos de produção dedicados a cada encomenda. Além disso, verificou-se que se obtêm ganhos importantes de tempo de percurso de fabrico e de qualidade de serviço de produção para pequenos graus de simultaneidade. Isto foi avaliado para algumas estruturas, por aplicação de estratégias de Produção Simultânea baseada na utilização de recursos de produção alternativos e na partição de encomendas. Como esperado, o melhor desempenho obtém-se para a partição máxima, i.e. partições em lotes de processamento de uma unidade. No entanto, na maioria dos casos testados, a maior parte do ganho verifica-se para pequenos graus de partição. Estes resultados foram obtidos num enquadramento de experimentação que não considerou nem custos de transporte e preparação nem de posse. Baseado nos resultados obtidos e no facto dos dois primeiros custos aumentarem com o aumento da partição e os custos de posse tenderem a diminuir com esse aumento, para tirar vantagem efectiva da Produção Simultânea, devem adoptar-se graus de partição moderados.



## **ABSTRACT**

Apparently, a new paradigm of production is emerging in the new and global economy based on three main tendencies. Such tendencies are: increasing dynamics of manufacturing systems reconfiguration, distribution of the manufacturing resources of a production system, and virtuality of systems, i.e. systems that are configured only on organizational and control terms, not on physical rearrangement of facilities. The first tendency is due to increasing product demand requirements, namely quality, variety and delivery. The second, resulting from a need for enhancement of production performance and, the third, is due to the unfeasibility of physically grouping and rearranging world wide distributed production facilities and due to capabilities offered by world wide communication and logistic systems.

Based on the referred tendencies new requirements arise for production systems design and operation. Some can be selected and synthesized as: *increased ability for system reconfiguration based on distributed resources and fast delivery of each product ordered*. Based on such requirements a definition of Distributed Production Systems (SDP) is put forward together with a model, the MSDP model, of such systems. Such model is closely related with the processing structure of an ordered product to be manufactured. Based on the MSDP model, a formal representation is developed using Petri Nets and modelling mechanisms put forward by Lecompte, Deschamps and Bourrières (2000), and an agent based conceptual model is presented.

The formal representation of the model is applied to a variety of systems configuration instances according with the product ordered. The application aim is twofold: first, testing the model suitability for representing and reconfiguring different SDP and, second, for studying the behaviour of SDP, under the Simultaneous Manufacturing philosophy. This philosophy aims at minimizing time delivery of orders through maximum reduction of production flow time of each production order.

The study carried out in this work lead to the conclusion that the model is suitable for representation, design and evaluation of distributed production systems dedicated to each product order. Further, a small degree of simultaneous manufacturing was proved enough to achieve important gains in order flow time reduction and quality of the manufacturing service. This was evaluated, for a few system configurations with strategies to simultaneous manufacturing based on simultaneous allocation of order manufacturing requirements to replicated machines, i.e. manufacturing resources, and also based on order splitting at a few different levels. As expected, the best values of performance were obtained under maximum order splitting, i.e. under split batches of size one. However, in most cases tested, the lion share of benefits from splitting was obtained for small degrees of it. These results were obtained under a framework for experiments that did not take into account neither handling and set-up costs nor holding costs. Based on the findings reported and onto the fact that the first two costs increase with order splitting and that holding costs tend to decrease with it, then, it was concluded that, to take effective advantageous of simultaneous manufacturing, moderate degrees of order splitting should be adopted.





## ÍNDICE

<b>Agradecimentos</b> .....	<b>iii</b>
<b>Resumo</b> .....	<b>v</b>
<b>Abstract</b> .....	<b>vii</b>
<b>Índice</b> .....	<b>ix</b>
<b>Lista de Figuras</b> .....	<b>xv</b>
<b>Lista de Abreviaturas e Acrónimos</b> .....	<b>xix</b>
<b>I - Introdução</b> .....	<b>1</b>
1. Enquadramento e Objectivos .....	3
1.1 Definição e Representação de Sistemas Distribuídos de Produção .....	4
1.2 Produção Simultânea.....	5
1.3 Projecto de Sistemas Distribuídos de Produção .....	6
2. Estrutura da Dissertação .....	7
<b>II - Sistemas Distribuídos de Produção</b> .....	<b>9</b>
1. Novos Paradigmas de Gestão da Produção.....	12
1.1 Fábrica Fractal.....	12
1.1.1. Similitude.....	12
1.1.2. Auto-Organização .....	13
1.1.3. Dinâmica .....	14
1.1.4. Orientação por Objectivos.....	14
1.2 Sistemas de Produção Holónicos.....	15
1.3 Sistemas de Produção Biónicos.....	16
2. Revisão Bibliográfica sobre Sistemas Distribuídos de Produção.....	17
2.1 Definição de Sistema Distribuído de Produção e dos seus Conceitos.....	19
2.2 Modelo e Projecto de Sistema Distribuído de Produção.....	19
2.3 Especificação Formal de Sistemas Distribuídos de Produção .....	26
2.4 Programação e Controlo da Produção em Sistemas Distribuídos de Produção.....	26
2.5 Sistemas CAPP e CAM em Sistemas Distribuídos de Produção .....	30
2.6 Tecnologia Aplicada nos Sistemas Distribuídos de Produção.....	31
3. Sistema Distribuído de Produção.....	31
3.1 Sistema.....	32
3.2 Sistema Distribuído Heterogéneo de Informação .....	33
3.3 Sistema de Produção .....	34
3.4 Definição de Sistema Distribuído de Produção .....	35

3.5	Características de Sistemas Distribuídos de Produção.....	37
3.5.1.	Distribuição.....	37
3.5.2.	Autonomia .....	38
3.5.3.	Reconfiguração.....	38
4.	Considerações Finais .....	39
<b>III</b>	<b>Modelo do Sistema Distribuído de Produção.....</b>	<b>41</b>
1.	Modelo do Sistema Distribuído de Produção.....	43
1.1	Enquadramento do Modelo .....	43
1.1.1.	Sistemas de Produção Orientados ao Produto.....	43
1.1.2.	Modelos de Sistemas de Produção Orientados aos Objectos .....	45
i.	Vantagens da modelação Orientada aos Objectos.....	45
1.2	Modelação do Sistema Distribuído de Produção.....	46
1.2.1.	Caracterização do Produto .....	48
1.2.2.	Caracterização das Células de Produção .....	51
1.2.3.	Coordenação e Controlo da Produção .....	52
i.	Processo de Configuração do Sistema Distribuído de Produção.....	53
1.2.4.	Informação necessária para a criação do sistema .....	54
2.	Representação Formal do Sistema Distribuído de Produção .....	55
2.1	Modelação de Produtos e Transformações.....	56
2.1.1.	Breve Referência às Redes de Petri .....	57
2.1.2.	Modelação do Produto e do Processo .....	58
i.	Exemplo.....	58
2.1.3.	Relação entre o Armazenamento e as Transformações.....	60
i.	Exemplo.....	62
2.1.4.	Processo de Agregação .....	62
i.	Configuração Genérica .....	65
ii.	Relação entre Quantidades dos Níveis de Agregação .....	66
iii.	Exemplo.....	70
2.1.5.	Restrições de Precedência.....	73
i.	Exemplo.....	74
2.2	Modelação de Recursos .....	75
2.2.1.	Restrições de Procura .....	75
2.2.2.	Restrições de Disponibilidade .....	75
2.2.3.	Afectação de Recursos.....	76
i.	Exemplo.....	78
3.	Exemplo de Aplicação do Modelo Formal MSDP .....	79
3.1	Modelo MSDP de um Sistema de Produção do tipo Linha Flexível de Fabrico.....	80
3.2	Modelação do Produto e do Processo .....	82
3.2.1.	Processo de Agregação .....	84

i.	Modelação dos Níveis de Agregação .....	84
ii.	Modelação da Transição entre Níveis de Agregação .....	86
3.3	Restrições de Precedência de Processos.....	90
3.4	Modelação de Recursos .....	91
3.5	Afectação de Recursos .....	93
4.	Considerações Finais .....	94
<b>IV</b>	<b>- Produção Simultânea .....</b>	<b>97</b>
1.	Definição.....	99
1.1	Breve Abordagem à Engenharia Concorrente / Simultânea.....	99
1.2	Produção Simultânea.....	100
1.3	Medida de Simultaneidade .....	101
1.3.1.	Grau de Simultaneidade do Sistema .....	101
1.3.2.	Grau de Simultaneidade de uma Transformação .....	102
1.4	Princípios de Simultaneidade.....	103
2.	Configuração Dinâmica do Sistema .....	105
2.1	Configuração Específica .....	106
2.1.1.	Exemplo.....	107
2.2	Possibilidades de Configuração do Sistema .....	110
2.2.1.	Modelo Local de Afectação de Recursos.....	111
2.2.2.	Modelo Global de Afectação de Recursos .....	113
3.	Formulação do Problema de Selecção de Recursos .....	115
3.1	Modelo Local de Afectação de Recursos.....	115
3.1.1.	Estratégia de Selecção de Recursos 1x1.....	115
i.	Menor Prazo de Entrega .....	116
ii.	Menor Custo .....	116
3.1.2.	Estratégia de Selecção de Recursos 1xq.....	116
i.	Menor Prazo de Entrega .....	117
ii.	Menor Custo .....	117
3.2	Modelo Global de Afectação de Recursos .....	117
3.2.1.	Restrições Genéricas.....	118
i.	Variáveis de Decisão .....	118
ii.	Limites Superiores .....	120
iii.	Número de recursos .....	120
iv.	Procura .....	121
v.	Disponibilidade .....	121
vi.	Tempos de recursos – relação entre execução e instante de conclusão.....	121
vii.	Tempos de transformações – relação com os recursos .....	121
viii.	Transformações precedentes.....	122
ix.	Transformações paralelas .....	122
3.2.2.	Restrições de Agenda.....	123
i.	Definição da Agenda .....	123
ii.	Execução sem interrupção num período livre da agenda.....	123

3.2.3.	Optimização da Selecção de Recursos .....	124
i.	Minimização do tempo total de execução .....	124
ii.	Minimização do número de recursos .....	124
iii.	Minimização do instante de conclusão do processamento .....	125
iv.	Formulações Mistas .....	128
4.	Considerações Finais .....	129
<b>V - Experimentação do Modelo MSDP em Ambiente de PS .....</b>		<b>131</b>
1.	Possibilidades de Configuração do Sistema .....	133
1.1	Plano de Experimentação .....	133
1.2	Caso de Teste .....	134
1.3	Modelo Local de Afectação de Recursos .....	137
1.3.1.	Descrição da Experimentação .....	137
1.3.2.	Resultados .....	139
1.4	Modelo Global de Afectação de Recursos .....	140
1.4.1.	Descrição da Experimentação .....	141
1.4.2.	Resultados .....	142
1.5	Considerações Finais sobre Possibilidades de Configuração do Sistema .....	145
2.	Aplicação do Modelo MSDP em Ambiente de PS a Várias Encomendas .....	146
2.1	Plano de Experimentação .....	147
2.2	Medidas de Desempenho .....	148
2.2.1.	Instante de conclusão do Conjunto das Encomendas .....	149
2.2.2.	Somatório dos Instantes de Conclusão das Encomendas .....	149
2.2.3.	Qualidade de Serviço de Produção (QMS) .....	150
2.2.4.	Utilização de Recursos .....	150
2.3	Descrição da Experimentação .....	151
2.4	Caso AA_FJS3x4 .....	153
2.4.1.	Caso AA_FJS3x4_01 .....	154
2.4.2.	Caso AA_FJS3x4_10 .....	155
2.5	Caso AA_FJS4x6 .....	159
2.5.1.	Caso AA_FJS4x6_1 .....	160
2.5.2.	Caso AA_FJS4x6_10 .....	161
2.6	Caso AA_STRUCT .....	165
2.7	Considerações Finais sobre o Processamento de Várias Encomendas .....	168
<b>VI - Modelo MSDP Baseado em Agentes de Software .....</b>		<b>171</b>
1.	Agentes .....	173
1.1	Enquadramento da Utilização de Agentes .....	174
1.1.1.	Sistema Aberto .....	175
1.1.2.	Distribuição .....	175
i.	Distribuição do Controlo .....	175

ii.	Distribuição da Informação.....	175
iii.	Distribuição de Materiais, Pessoas e Meios de Produção.....	176
1.2	Definições de Agentes.....	176
1.2.1.	Agentes de Software.....	176
i.	Perspectiva Genérica.....	177
ii.	Perspectiva Técnica.....	177
iii.	Perspectiva de Utilização.....	177
1.2.2.	Agentes Inteligentes.....	178
1.3	Sistemas Multiagente.....	181
1.3.1.	Arquitectura da Estrutura de Ligação.....	182
1.3.2.	Comunicação / Coordenação.....	182
1.4	Ferramentas de Modelação e Desenvolvimento.....	184
1.4.1.	Linguagens.....	185
1.4.2.	Ambientes.....	185
2.	Modelo Multiagente do MSDP.....	187
2.1	Requisitos do Sistema.....	188
2.2	Ferramentas Seleccionadas.....	189
2.2.1.	Especificação de Sistemas de Informação.....	189
2.2.2.	Ambiente de Suporte a Agentes de Software.....	190
2.2.3.	Exemplo de Aplicação das Ferramentas Seleccionadas.....	191
2.3	Descrição Genérica do Modelo.....	193
2.3.1.	Agente Cliente - Descrição Genérica.....	193
2.3.2.	Agente Encomenda - Descrição Genérica.....	194
2.3.3.	Agente Recurso - Descrição Genérica.....	194
2.3.4.	Coordenação dos Agentes.....	195
2.4	Especificação do Modelo.....	195
2.4.1.	Modelo Lógico Estático.....	195
2.4.2.	Comunicação / Coordenação.....	197
2.4.3.	Agente Cliente.....	197
2.4.4.	Agente Encomenda.....	198
2.4.5.	Agente Recurso.....	200
3.	Considerações Finais.....	202
<b>VII - Conclusão.....</b>		<b>205</b>
1.	Conclusões e Contribuições do Trabalho Desenvolvido.....	207
1.1	Definição de Sistema Distribuído de Produção.....	207
1.2	Modelo do Sistema Distribuído de Produção.....	208
1.3	Produção Simultânea.....	209
1.4	Estudo do Modelo MSDP em ambiente de PS.....	210
2.	Limitações do Trabalho Desenvolvido e Perspectivas de Trabalho Futuro.....	212

2.1	Definição de Sistema Distribuído de Produção .....	212
2.2	Modelo do Sistema Distribuído de Produção.....	212
2.3	Produção Simultânea .....	213
2.4	Sistema Multiagentes .....	214
<b>Referências Bibliográficas .....</b>		<b>215</b>
<b>Anexos .....</b>		<b>A</b>
1.	Experimentação .....	C
1.1	Xpress-MP 2003 - Código Mosel .....	C
1.2	Xpress-MP 2003 – Dados de Entrada.....	F
1.3	Experimentação AA .....	G
1.3.1.	Folha de Cálculo .....	G
1.3.2.	Programação em VBA.....	G
2.	Modelo de Agentes de Software .....	I
2.1	Exemplo de Aplicação das Ferramentas Seleccionadas .....	I

## LISTA DE FIGURAS

Figura 1: Recursos Distribuídos de Produção.....	5
Figura 2: Simultaneidade de utilização de recursos de produção por partição de um lote.....	6
Figura 3: Fractais Similares - adaptado de Warnecke (1993). ....	13
Figura 4: Sistema Holónico - retirado de Tharumarajah, Wells and Nemes (1996). ....	16
Figura 5: Comparação entre o sistema biológico e de produção .....	17
Figura 6: Representação esquemática de Sistema .....	32
Figura 7: Representação de Subsistemas. ....	33
Figura 8: Modelo do Sistema de Produção segundo Hitomi (1996). ....	34
Figura 9: Representação de Produtos e Células por agregação de elementos similares.....	48
Figura 10: Estruturas de produto alternativas. ....	49
Figura 11: Produto final – Caixa para fotografias.....	50
Figura 12: Estrutura do Produto com quantidades unitárias. ....	50
Figura 13: Representação dos diferentes níveis da estrutura do Sistema Distribuído de Produção.....	51
Figura 14: Exemplo de representação de Produtos e Células. ....	52
Figura 15: Encomenda e o processo local de formação de uma célula de produção. ....	53
Figura 16: Encomenda e o processo global de formação de uma célula de produção.....	53
Figura 17: Programação Generalizada - Lecompte, Deschamps and Bourrières (2000). ....	56
Figura 18: Exemplo de Rede de Petri. ....	57
Figura 19: Estrutura do Produto com transformações - representação vertical e horizontal.....	58
Figura 20: Rede de Petri representando a dinâmica do sistema de produção.....	59
Figura 21: Exemplos de sistemas com representação de processos de produção alternativos. ....	61
Figura 22: Exemplos de níveis com agregação em que se mantém o mesmo número de objectos. ....	63
Figura 23: Exemplos de matrizes de transição entre níveis de agregação. ....	65
Figura 24: Exemplos de Relacionamento entre Quantidades dos Níveis de Agregação. ....	67
Figura 25: Exemplo de agregação. ....	69
Figura 26: Rede de Petri referente ao nível de agregação $v=l=2$ . ....	71
Figura 27: Rede de Petri referente ao nível de agregação $v=l-1=1$ .....	71
Figura 28: Rede de Petri referente ao nível de agregação $v=0$ .....	72
Figura 29: Sistema de produção composto por diferentes estágios - adaptado de Azevedo (1999).....	80
Figura 30: Estrutura do Produto associada a quatro estágios sequenciais de produção. ....	81
Figura 31: Rede de Petri do sistema de produção .....	82
Figura 32: Rede de Petri referente ao nível de agregação $v=l=3$ . ....	85
Figura 33: Rede de Petri referente ao nível de agregação $v=l-1=2$ .....	85
Figura 34: Rede de Petri referente ao nível de agregação $v=l-2=1$ . ....	85
Figura 35: Configuração genérica: estrutura simples de um produto. ....	107
Figura 36: Configuração genérica: Rede de Petri do passo 1. ....	107

Figura 37: Configuração genérica: Rede de Petri do passo 2. ....	108
Figura 38: Gráfico de Gantt - Agenda dos recursos candidatos. ....	109
Figura 39: Gráfico de Gantt – Aplicação do princípio (i). ....	109
Figura 40: Gráfico de Gantt – Aplicação do princípio (ii.a). ....	109
Figura 41: Gráfico de Gantt – Aplicação do princípio (ii.b). ....	110
Figura 42: Gráfico de Gantt – Aplicação do princípio (iii). ....	110
Figura 43: Curva de variação da função $1/x$ . ....	112
Figura 44: Modelo local - Prazo de Entrega vs Grau de Simultaneidade. ....	112
Figura 45: Modelo global - Prazo de Entrega vs Grau de Simultaneidade. ....	114
Figura 46: Modelo global - Custos de Preparação e de Posse vs Grau de Simultaneidade. ....	114
Figura 47: Estruturas genéricas de base. ....	134
Figura 48: Diagrama de Fluxo do processo de experimentação para modelo local. ....	138
Figura 49: Curvas de Variação dos Instantes de Conclusão das Transformações. ....	140
Figura 50: Diagrama de Fluxo do processo de experimentação para o modelo global. ....	142
Figura 51: Curva de Variação do Instante de Conclusão. ....	144
Figura 52: Curvas de Variação dos Custos e do Instante de Conclusão. ....	145
Figura 53: Diagrama de Fluxo do processo de experimentação. ....	152
Figura 54: Representação idêntica para todos os produtos do caso AA_FJS3x4. ....	153
Figura 55: AA_FJS3x4_01 – Gráfico de Gantt. ....	154
Figura 56: AA_FJS3x4_10 - Gráfico de Gantt - P3-2-1-4. ....	157
Figura 57: AA_FJS3x4_10_5 - Gráfico de Gantt - P3-2-1-4. ....	157
Figura 58: AA_FJS3x4_10_1 - Gráfico de Gantt - P3-2-1-4. ....	158
Figura 59: Representação idêntica para todos os produtos do caso AA_FJS4x6. ....	159
Figura 60: AA_FJS4x6_1 - Gráfico de Gantt - P3-6-2-4-5-1. ....	160
Figura 61: AA_FJS4x6_10 - Gráfico de Gantt - P3-6-2-4-5-1. ....	163
Figura 62: AA_FJS4x6_10_1 - Gráfico de Gantt - P3-6-2-4-5-1. ....	163
Figura 63: AA_FJS4x6_10_5 - Gráfico de Gantt - P3-6-2-4-5-1. ....	164
Figura 64: AA_STRUCTURE – Redes de Petri dos produtos P1 a P4. ....	166
Figura 65: AA_STRUCTURE - Gráfico de Gantt – P4-3-2-1-5. ....	167
Figura 66: Ilustração de Agente num determinado ambiente. ....	177
Figura 67: Ilustração do teste de inteligência de Alan Turing. ....	178
Figura 68: Agente Inteligente - Russell and Norvig (1995). ....	179
Figura 69: Engenharia de Software - Ciclo de Vida de Clássico. ....	188
Figura 70: JavaSpaces como espaço de partilha de objectos. ....	191
Figura 71: JavaSpaces e Jini - retirado de Halter (2002). ....	191
Figura 72: Exemplo de Utilização de JavaSpaces – Relações do Objecto “Customer”. ....	192
Figura 73: Exemplo de Utilização de JavaSpaces – Objectos do tipo “Entry”. ....	192
Figura 74: Modelo Estrutural do Sistema na Rede. ....	194
Figura 75: Modelo Lógico Estático da Informação de Agentes Recurso. ....	196



Figura 76: Modelo Lógico da Actividade de Lançamento de Encomenda do Agente Cliente. ....	198
Figura 77: Modelo Lógico de Actividade de Gestão Local de Encomenda de Agentes Encomenda. ....	199
Figura 78: Modelo Lógico de Actividade de “Criação de Oferta” de Agentes Recurso. ....	201
Figura 79: Modelo Lógico de Actividade de “Aceitação de Pedido” de Trabalho. ....	202



## **LISTA DE ABREVIATURAS E ACRÓNIMOS**

CAD:	“Computer Aided Design”
CAPP:	“Computer Aided Process Planning”
CASE:	“Computer Aided Software Engineering”
CIM:	“Computer Integrated Manufacturing”
CORBA:	“Common Object Request Broker Architecture”
FCM:	“Focused Cellular Manufacturing”
FFS:	“Flexible Flow Shop”
GS:	Grau de Simultaneidade
ISO:	“International Organization for Standardization”
JIT:	“Just-In-Time”
LT:	“Lead Time”
OKP:	“One-of-a-Kind Production”
OO:	Orientado aos Objectos / “Object Oriented”
OPIM:	“One Product Integrated Manufacturing”
OSI:	“Open Systems Interconnection”
PE:	Prazo de Entrega
PN:	“Petri Net” – Rede de Petri
PS:	Produção Simultânea
QMS:	“Quality of Manufacturing Service”
SDP:	Sistema Distribuído de Produção
SPOP:	Sistema de Produção Orientado ao Produto
STEP:	“STandard for The Exchange of Product model data”
UML:	“Unified Modelling Language”
UT:	Unidades de Tempo
VBA:	“Visual Basic for Applications”
WIP:	“Work In Process”



## **I - INTRODUÇÃO**

A motivação e os objectivos definidos para especificação, projecto e estudo de sistemas distribuídos de produção conduzem todo o trabalho apresentado, sob o título de Sistemas Distribuídos de Produção em Ambiente de Produção Simultânea.



Neste trabalho de investigação pretende-se, em termos gerais, dar uma contribuição para o estudo de sistemas distribuídos de produção, para o estudo da filosofia de Produção Simultânea e para o estudo da sua aplicação no processo de geração de configurações de sistemas distribuídos de produção. Assim sendo, apresenta-se neste capítulo o enquadramento e objectivos gerais do trabalho, um conjunto de objectivos específicos e a estrutura da dissertação.

## **1. ENQUADRAMENTO E OBJECTIVOS**

Face aos novos paradigmas da economia mundial surgem novos paradigmas de produção. De acordo com estes paradigmas, os sistemas de produção deverão ser capazes de satisfazer especificações cada vez mais personalizadas de cada cliente, no menor tempo possível. Uma solução possível é recorrer à utilização de recursos distribuídos que, de forma articulada e conjugada oferecem os seus serviços como elementos de um serviço alargado, traduzido na produção e conclusão da encomenda, eventualmente incluindo a sua distribuição ao cliente. Como se depreende da natureza de prestação de serviços associada aos recursos, cada um deles é autónomo, sujeitando-se apenas à prestação do serviço solicitado na data acordada.

Nesta perspectiva, para entregar uma encomenda, é necessário executar um processo de gestão que inclui a selecção de recursos de produção, reunidos para oferecerem os seus serviços, que de acordo com apropriados mecanismos de coordenação possibilitam a obtenção do produto encomendado na quantidade e data desejada.

Estamos no fim de contas a falar de um processo dinâmico, adaptado a cada encomenda, que poderíamos chamar de projecto do sistema de produção para essa encomenda. Encontramos, portanto, características tais como a distribuição de recursos e sistemática configuração dos sistemas, seguidos de nova configuração, i.e. reconfiguração, para os adaptar às necessidades de transformação de cada encomenda. Estas características, nomeadamente dinamismo de configuração dos sistemas de produção, baseado num leque de recursos ou serviços disponíveis, e sua orientação à encomenda, i.e. ao produto, foram eleitas como essenciais no conceito de Sistema Distribuído de Produção adoptado nesta dissertação.

Sob o ponto de vista dos sistemas distribuídos de produção, a questão está em prestar um serviço de qualidade aos clientes que, garantida que seja a qualidade do produto, se pode traduzir, essencialmente, na entrega mais rápida possível dos produtos encomendados. Por este facto, o seu papel principal orienta-se a garantir a produção de cada produto, sendo necessário um processo de selecção de recursos autónomos, que conjugada e articuladamente formam um sistema dedicado ou orientado ao produto, ou mais precisamente à encomenda de um produto. Isto traduz-se, em última análise, no que se designa genericamente por Sistema de Produção Orientado ao Produto (SPOP), descrito em Silva, S. C. and Alves (2002), e em particular por OPIM – One Product Integrated Manufacturing, introduzido por Putnik and Silva (1995).

A necessidade de rápida entrega do produto sugere que, os diferentes serviços e objectos, i.e. produtos, constituintes do produto final, poderão vantajosamente ser produzidos em paralelo, ou simultaneamente, por diferentes recursos. Este paralelismo ou simultaneidade, levada ao extremo, poderá permitir a minimização do tempo de entrega do produto. Esta abordagem à simultaneidade de processamento orientada ao produto foi

proposta por Silva, S. C. and Putnik (1995) com o nome de Produção Simultânea – “Simultaneous Manufacturing”.

Estas três vertentes, nomeadamente a da selecção de recursos distribuídos, a produção orientada ao produto, aqui instanciada na produção orientada a cada encomenda de um produto, e a produção simultânea, formam conjuntamente uma estratégia para desenvolver instrumentos ou modelos úteis de planeamento produtivo, i.e. selecção de recursos e sua utilização no tempo. O desenvolvimento destes modelos e o seu estudo orientam o trabalho aqui desenvolvido. Desta forma, pode-se dizer que o objectivo deste trabalho se insere no projecto e operação de sistemas distribuídos de produção numa óptica de Produção Simultânea.

Pelo exposto, foi decidido eleger como objectivo genérico deste trabalho de investigação o *projecto e avaliação do desempenho de sistemas distribuídos de produção em ambiente de Produção Simultânea*. Neste objectivo estão contidos os seguintes conceitos estruturantes fundamentais: Sistemas Distribuídos de Produção (SDP) e Produção Simultânea (PS). Ambos os conceitos devem ser definidos e especificados formalmente, de forma a proceder-se a uma análise e desenvolvimento baseados no seu inter-relacionamento.

A principal motivação da aplicação da Produção Simultânea assenta no pressuposto de que contribui para a redução do prazo de entrega de encomendas, por redução do tempo em curso ou de percurso do seu fabrico, contribuindo portanto, indirectamente, para o aumento da satisfação dos clientes.

Há portanto uma necessidade de desenvolver e analisar princípios operacionais inerentes à PS, formas da sua implementação prática e verificação do comportamento do prazo de entrega de uma encomenda face à sua produção simultânea, i.e. a simultaneidade de utilização de todos os recursos possíveis na sua execução. Embora a Produção Simultânea tenha como premissa a satisfação de uma encomenda de cada vez, a análise desenvolve-se por implementação sucessiva da PS a encomendas que sucessivamente têm de ser produzidas em SDPs.

## **1.1 Definição e Representação de Sistemas Distribuídos de Produção**

O objectivo da dissertação, como foi referido, tem subjacente a abordagem a dois conceitos principais e ao seu relacionamento. Tais conceitos são o de SDP e de PS. Para projecto e avaliação de SDPs é necessário criar modelos de acordo com a sua definição.

Dos programas de investigação em curso nos últimos dez anos, surgiram diversos paradigmas relacionados com sistemas distribuídos de produção, referidos por Tharumarajah, Wells and Nemes (1996), como por exemplo os Sistemas de Produção Biónicos<sup>1</sup>, Holónicos<sup>2</sup> e Fractais. Estes sistemas partilham conceitos similares, de modelação e de utilização de meios de produção autónomos, i.e. com a capacidade de auto-organização. Tal capacidade de auto-organização depende da estratégia de gestão, isto é, dos objectivos, restrições e critérios de funcionamento do sistema.

Quaisquer sistemas são, por definição, baseados em elementos distribuídos inter-relacionados. Assim sendo, qualquer sistema de produção também se baseia em algum tipo de distribuição estrutural dos elementos de produção que o compõem. Na Figura 1 representam-se dois exemplos hipotéticos de sistemas

---

<sup>1</sup> “Bionic”.

<sup>2</sup> “Holonic”.



de produção com utilização distribuída de meios de produção, i.e. recursos de produção, em que um dos recursos é partilhado pelos dois sistemas. Não parece então suficiente, utilizar a característica de distribuição de elementos do sistema para definição de um sistema distribuído de produção. Estes sistemas, são baseados em elementos heterogéneos e surgem como resposta a um aumento da complexidade de requisitos do sistema, nomeadamente no que se refere à capacidade de reconfiguração e autonomia dos elementos de produção.

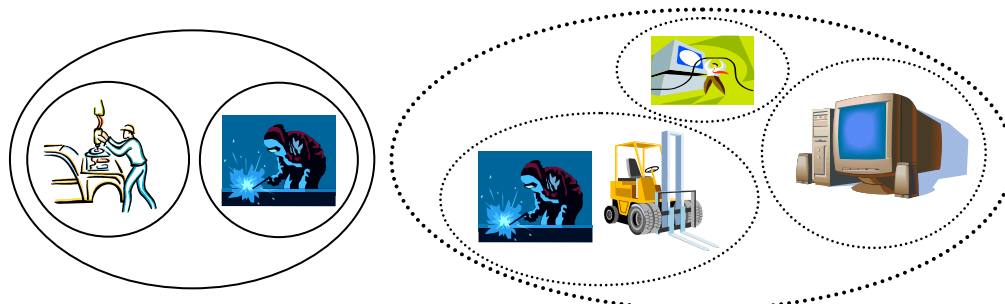


Figura 1: Recursos Distribuídos de Produção.

Neste trabalho, apresenta-se uma *definição* de SDP, que serve de base para a *concepção de um modelo de criação* de SDPs que satisfaçam os requisitos de produção para uma encomenda de um produto de cada vez. Esta abordagem de SDPs orientados à encomenda, como foi referido, está directamente relacionada com a filosofia de PS. O modelo referido apresenta conceitos de projecto directamente ligados a uma representação estrutural do produto associada ao seu processo produtivo e, ainda, à agregação de recursos, que são formalmente especificados através de Redes de Petri.

## 1.2 Produção Simultânea

A filosofia de Produção Simultânea, apresentada por Silva, S. C. and Putnik (1995), aplicada aos SDPs, *requer o estudo de mecanismos e a concepção de modelos* que permitam analisar as suas implicações sobre o comportamento dos sistemas de produção.

O objectivo motivador da aplicação da Produção Simultânea é o de redução do prazo de entrega da encomenda, como consequência da previsível redução do seu prazo de produção. De facto, tal filosofia advoga o princípio genérico da *utilização dedicada e simultânea de todos os recursos possíveis para produção de uma encomenda de cada vez*. A aplicação deste princípio pressupõe o escalonamento dos recursos para uma encomenda, explorando a sua utilização dedicada simultânea como se nenhuma outra encomenda existisse no horizonte próximo.

A utilização simultânea de recursos, na produção da mesma encomenda, pode conseguir-se com, ou sem, a sua partição - “batch splitting” - e com, ou sem, sobreposição de processamento de lotes - “batch overlapping”. A partição duma encomenda em várias encomendas, permite o processamento simultâneo destas nos recursos de produção do sistema. A sobreposição do processamento de uma encomenda, por outro lado, não divide a encomenda, mas baseia-se na possibilidade de enviar parte dela para o recurso seguinte, onde a encomenda é processada, antes de terminar o processamento de toda a quantidade da encomenda no recurso anterior. Desta forma, a execução de duas transformações diferentes consecutivas, sobre o mesmo lote de produção, realiza-se simultaneamente durante um determinado período de tempo.

Não havendo partição da encomenda, nem sobreposição de produção, pode ainda considerar-se a produção simultânea da encomenda associada à utilização simultânea dos recursos apenas para esta encomenda, como se nenhuma outra existisse. O processamento simultâneo de uma encomenda pode intensificar-se à custa da acção conjunta do aumento do número de recursos de produção e da partição e sobreposição de lotes.

A Figura 2 ilustra a possibilidade de partição de um lote de três unidades em dois lotes, um de uma unidade e outro de duas unidades, que são processados simultaneamente em dois recursos de produção. Neste exemplo, os dois recursos executam o mesmo processo de soldadura.

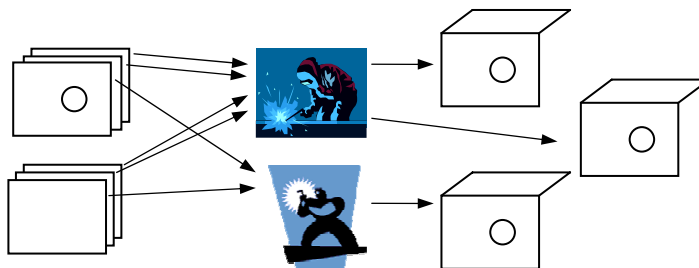


Figura 2: Simultaneidade de utilização de recursos de produção por partição de um lote.

No contexto do estudo de Sistemas Distribuídos de Produção em ambiente de Produção Simultânea é necessário compreender o comportamento do prazo de entrega de uma encomenda em função da simultaneidade de utilização dos recursos. Tal compreensão em conjunto com a noção de SDP e com o modelo formal de SDP desenvolvidos neste trabalho, possibilitam a criação de uma configuração apropriada de cada sistema de produção para a satisfação de cada uma das encomendas a produzir.

### **1.3 Projecto de Sistemas Distribuídos de Produção**

Um dos objectivos parciais subjacentes ao objectivo genérico apresentado é o de projecto de SDPs. Pretende-se, no âmbito deste objectivo, *desenvolver um modelo que permita construir configurações* de SDPs baseadas na aplicação dos princípios de Produção Simultânea.

O modelo e respectiva especificação formal do SDP, relacionado com a Produção Simultânea, permite enquadrar o desenvolvimento destes sistemas, com definição das condições e critérios para a sua configuração e reconfiguração. Não se pretende proceder à implantação física dos recursos, mas antes à sua configuração em células virtuais (Mclean and Brown (1987)) a partir de meios de produção, ou recursos, existentes. Também não se pretende efectuar o estudo de implantações de meios de produção, nem analisar formas de tomada de decisão sobre os melhores processos e/ou meios de produção a aplicar, mas encontrar formas de decidir sobre a selecção de meios de produção existentes para a produção de cada encomenda.

A análise do modelo de SDPs desenvolvido neste trabalho, debruça-se sobre a capacidade de representação e de geração de configurações de sistemas com distribuição de recursos. Assim sendo, pretende-se mostrar a capacidade de representação de Sistemas Distribuídos de Produção através da aplicação do modelo em vários casos. Pretende-se, igualmente, efectuar um estudo da forma de afectação de recursos para SDPs em ambiente de PS e estudar a variação do prazo de entrega relativamente à simultaneidade de produção. A aplicação da PS em sistemas de produção, para execução consecutiva de

várias encomendas, tem implicações sobre prazos de entrega e utilização de recursos que devem ser clarificados.

## **2. ESTRUTURA DA DISSERTAÇÃO**

A presente dissertação divide-se em capítulos relacionados com os objectivos estabelecidos, de acordo com uma estrutura que conduz o trabalho, através da fundamentação, definição e modelação de sistemas distribuídos de produção, definição de Produção Simultânea e especificação do relacionamento com os SDPs, experimentação do modelo desenvolvido, apresentação de um projecto do modelo de SDPs baseado em agentes de software e finalmente, conclusões do trabalho. Todos os capítulos contêm uma introdução e são concluídos com secções de considerações finais.

Nas primeiras páginas da dissertação apresenta-se informação de identificação, caracterização e estruturação deste documento, com título, agradecimentos, resumo, índices e lista de acrónimos.

O primeiro capítulo corresponde à introdução do trabalho com apresentação do enquadramento e do objectivo genérico e respectivos objectivos parciais do trabalho, seguido pela explanação da estrutura da dissertação.

O capítulo II - Sistemas Distribuídos de Produção - é dedicado ao enquadramento dos conceitos relacionados com sistemas distribuídos de produção, baseado numa revisão bibliográfica, seguida de um trabalho de definição de Sistema Distribuído de Produção. Esta definição baseia-se em conceitos de sistemas, sistemas distribuídos de informação e em objectivos de estabelecimento de uma definição que permita distinguir os SDPs de outros sistemas de produção.

No capítulo III - Modelo do Sistema Distribuído de Produção - é apresentado um modelo para o sistema distribuído de produção, de acordo com a definição do capítulo anterior. Esta definição é baseada em conceitos de elementos de produção similares, que são utilizados recursivamente, seguindo linhas de investigação dos sistemas de produção Biónico, Holónico e Fractal. Além disso, o sistema é orientado ao produto, na medida em que um sistema só existe para a satisfação de uma encomenda de um produto. O modelo apresentado é formalmente especificado pelo recurso a Redes de Petri, com um conjunto de mecanismos específicos para a modelação de Sistemas Distribuídos de Produção. Finalmente apresenta-se um exemplo de modelação de SDPs baseado numa Linha de Fabrico Flexível, verificando a possibilidade de representação de um sistema de produção conhecido através do modelo formal multinível desenvolvido.

O capítulo IV - Produção Simultânea - agrega os conceitos definidores desta abordagem de gestão da produção. Utiliza-se a definição desta abordagem para estabelecer as suas características e princípios de simultaneidade de utilização de recursos. A utilização de um destes princípios de simultaneidade permite estabelecer a relação com o modelo de Sistemas Distribuídos de Produção, especificando formas de geração de configurações de sistemas, isto é, de selecção de recursos de produção, de uma forma local ou global.

O capítulo V - Experimentação do Modelo MSDP - representa o trabalho de experimentação do modelo em diferentes situações e o de análise dos resultados obtidos nos testes. Este trabalho de experimentação divide-se em duas secções, com objectivos diferentes. O primeiro objectivo é o de estudar o comportamento do prazo de entrega de encomendas relativamente à simultaneidade de utilização de recursos, para uma

estrutura genérica. Finalmente, pretende-se estudar os efeitos da aplicação da Produção Simultânea, de acordo com os modelos de afectação de recursos desenvolvidos no capítulo anterior. Esta secção incide sobre a análise da satisfação de várias encomendas sucessivas em diferentes sistemas de produção seleccionados, considerando negligenciáveis as operações de transporte e preparação da produção para selecção de recursos.

No capítulo VI - Modelo MSDP Baseado em Agentes de Software - apresenta-se um modelo de um sistema baseado em Agentes de Software que permita implementar o modelo de Sistemas Distribuídos de Produção desenvolvido.

O capítulo VII - Conclusão - apresenta as contribuições e conclusões do trabalho e propostas de desenvolvimento futuro.

Os últimos capítulos são os de apresentação das referências bibliográficas, que fundamentaram e apoiaram este trabalho, aos quais se seguem anexos complementares à dissertação apresentada.

## **II - SISTEMAS DISTRIBUÍDOS DE PRODUÇÃO**

Face aos novos desafios que se colocam às empresas, com a evolução do mercado em direcção à alteração permanente dos requisitos dos clientes, é necessário encontrar sistemas de produção que reajam rapidamente, alterando a sua própria configuração. Desta realidade surgiram nos últimos anos paradigmas de organização de empresas e da produção que se podem agregar numa classificação: sistemas distribuídos de produção. Estes sistemas carecem, no entanto, de definições globalmente aceites, modelos de concepção, linguagens de especificação formal, e formas de gestão. Neste capítulo apresenta-se uma definição de Sistema Distribuído de Produção e identificam-se características importantes.



A associação entre empresas tende a requerer a interação entre os seus sistemas de produção. As formas mais comuns de relacionamento entre estes sistemas são as de subcontratação e gestão da cadeia de fornecimento, que não são caracterizadas por terem mecanismos explícitos para lidar com a rápida alteração das ligações estabelecidas. Conceitos mais recentes de associação entre empresas baseiam-se em relações mais flexíveis e dinâmicas com exploração intensiva de tecnologia de informação e comunicação. Algumas destas associações incluem:

- Empresa Estendida<sup>3</sup> - empresas que se associam ao longo da cadeia produtiva de uma forma mais ou menos fixa para produzir determinado tipo de produtos.
- Empresa Virtual<sup>4</sup> - empresas que se associam durante um espaço de tempo usualmente reduzido para explorar uma oportunidade de negócio e cuja ligação tem uma elevada dependência da tecnologia da informação.
- OPIM<sup>5</sup> - empresas que vão utilizar recursos disponíveis (possivelmente noutras empresas), distribuídos globalmente, para criar um sistema eficiente de produção dedicado a um produto e que se dissolverá após a conclusão da tarefa.

Tradicionalmente, é usual, serem os sistemas de produção concebidos e operados localmente dentro de uma instalação fabril, onde se transformam matérias primas em produtos finais. Esta é, na essência, a função base de qualquer sistema de produção: usar e transformar materiais de entrada, físicos ou não, em produtos finais, tangíveis ou intangíveis. Hoje é possível, em tempo útil e com um custo adequado, recorrer a redes de comunicação locais ou globais, para transformar matérias primas e produzir produtos, tirando partido da natureza distribuída da localização dos recursos.

Os sistemas distribuídos de produção podem ser equacionados, pelo menos, segundo dois aspectos: o da distribuição espacial dos recursos e o da distribuição da tomada de decisão. A constatação da existência de diferentes visões sobre sistemas distribuídos de produção, faz com que se torne pertinente apresentar uma definição, analisar a sua aplicabilidade, formas de concepção e formas de os gerir.

Há vários trabalhos de investigação a decorrer, no âmbito da exploração da distribuição dos recursos de produção, que se baseiam nas seguintes premissas e requisitos das empresas do futuro:

- guiadas pelo mercado;
- integração dos fornecedores no ciclo de vida do produto;
- adaptabilidade à mudança, capazes de produzir pequenas quantidades com elevada qualidade a custos reduzidos;
- compostas por pequenas unidades autónomas distribuídas globalmente, cooperando para um mesmo objectivo, apoiadas nas novas tecnologias da informação.

O programa de investigação e desenvolvimento IMS<sup>6</sup> sobre sistemas de produção inteligentes é um exemplo do esforço realizado internacionalmente no estudo de novos paradigmas apoiados nos conceitos

---

<sup>3</sup> "Extended Enterprise".

<sup>4</sup> "Virtual Enterprise".

<sup>5</sup> "One Product Integrated Manufacturing".

<sup>6</sup> "Intelligent Manufacturing Systems" - <http://www.ims.org/>.

anteriores, como sejam<sup>7</sup>: Produção Ágil, Sistemas de Produção Distribuídos e Autónomos, Sistemas de Produção Biónicos e Empresas Fractais.

## **1. NOVOS PARADIGMAS DE GESTÃO DA PRODUÇÃO**

Nesta secção apresentam-se, de forma sumária, alguns dos novos paradigmas de gestão da produção, e as suas características, realçando as suas semelhanças e diferenças.

### **1.1 Fábrica Fractal**

O conceito de Fábrica Fractal foi introduzido por Warnecke (1993) que se inspirou no trabalho do matemático Mandelbrot, criador da teoria da geometria fractal. Esta teoria baseia-se na possibilidade de gerar iterativamente estruturas complexas similares, a partir de regras simples. Do mesmo modo, o sistema de produção da Fábrica Fractal é composto por uma organização de unidades similares (fractal) que providenciam serviços de produção (Warnecke (1993); Tharumarajah, Wells and Nemes (1996) ;Kádar, Monostori and Szelke (1998)).

A definição de fractal segundo Warnecke (1993):

Um fractal é uma entidade da empresa agindo independentemente cujos objectivos e desempenho podem ser descritos de uma forma precisa.

Os fractais apresentam as seguintes características: similitude<sup>8</sup>, auto-organização<sup>9</sup>, orientação por objectivos<sup>10</sup> e dinâmica.

#### **1.1.1. Similitude**

Os fractais são entidades similares, quanto à sua estrutura e quanto aos seus objectivos, que executam serviços. Não são entidades idênticas, porque para o mesmo problema, mesmos objectivos e mesmas entradas podem ter estruturas diferentes, conforme ilustração da Figura 3. Nomeadamente, podem recorrer a recursos de produção diferentes, com diferentes processos produtivos. Os fractais podem, de uma forma independente, associar-se com fractais externos à fábrica, criando dessa forma uma rede de fractais, que vistos como recursos, podem constituir uma rede de empresas. Por sua vez, os objectivos globais têm que ser transformados em objectivos específicos de cada fractal individual.

Uma possibilidade aceite na fábrica fractal é a da existência de entidades centralizadoras, caso se considere necessário ou mais vantajoso. Um exemplo é o do recurso a uma unidade central de planeamento de recursos ou de apoio ao planeamento, ou ainda a uma entidade de concentração de conhecimento especializado.

---

<sup>7</sup> “Agile Manufacturing; Autonomous & Distributed Manfg Systems; Bionic Manfg Systems; Fractal Company”.

<sup>8</sup> “Self-similar”.

<sup>9</sup> “Self-organization”.

<sup>10</sup> “Goal orientation”.



Todos os mecanismos de apoio da empresa devem estar disponíveis para todos os fractais. Isto significa que todas as entidades devem ter acesso completo à informação da empresa, que não deve ser monopolizada.

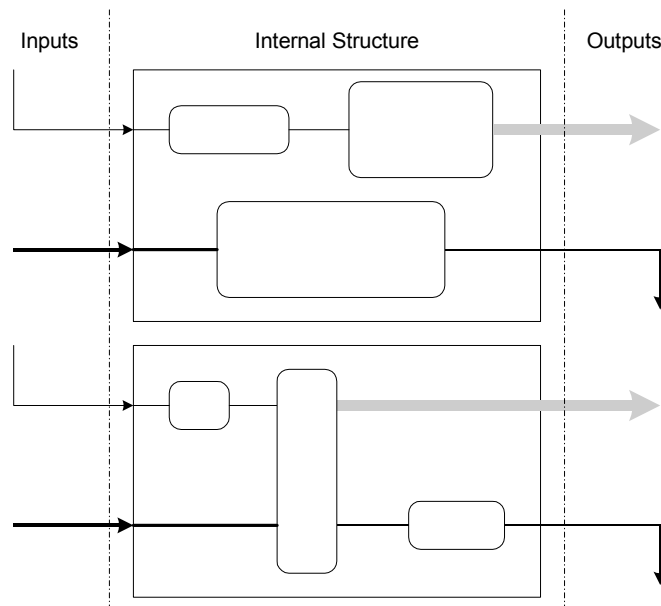


Figura 3: Fractais Similares - adaptado de Warnecke (1993).

### **1.1.2. Auto-Organização**

Os fractais organizam os procedimentos de uma forma óptima, por aplicação de métodos adequados, determinam e formulam os seus objectivos num processo dinâmico e tomam decisões sobre as ligações adequadas para a sua estrutura. Os fractais reestruturam, regeneram e dissolvem-se a si próprios. Diferentes fractais recorrem a diferentes métodos para projectarem o seu sistema e efectuarem o planeamento e gestão da produção em função, por exemplo, dos recursos disponíveis ou da procura e quantidade a produzir. Os recursos disponíveis podem ter diferentes características, nomeadamente relativas a habilidade, desempenho e grau de flexibilidade. Podemos assim ter, numa mesma empresa, fractais organizados de diferentes formas, nomeadamente em linha de produção, células de Grupo de Trabalho, ou células JIT<sup>11</sup>, em função das decisões tomadas pelos próprios fractais.

O objectivo numa fábrica fractal é o de existirem formas de organização, em que elementos vitais, os fractais, se estruturam a eles próprios e em conjunto sirvam o sistema global. Os fractais estão ligados em rede, por eficientes sistemas de informação e de comunicação, sendo controlados por eles próprios a natureza e extensão do acesso aos dados. Nesta fábrica é fundamental a utilização eficiente e intensiva de modelo e sistemas de informação, sendo os seguintes realçados por Warnecke (1993):

- Paradigmas de modelação, como os de orientação aos objectos e agentes de software, que permitam o suporte sistemático à modelação por fractais.
- Sistemas CIM abertos e transparentes.
- Utilização de sistemas periciais<sup>12</sup>.

<sup>11</sup> "Just-In-Time".

<sup>12</sup> "Expert systems".

- Recurso à simulação como ferramenta de avaliação antes da execução de determinadas operações.
- Sistemas de programação da produção, de execução e de controlo, baseados em conhecimento.

### **1.1.3. Dinâmica**

Os sistemas de produção são, por tradição, baseados numa visão estática da estrutura do sistema, i. e., os sistemas depois de projectados, mantêm-se inalterados por longos períodos, e as mudanças importantes dão-se em ciclos muito alargados. Mesmo os sistemas cuja estrutura se baseia na ligação entre unidades básicas, sejam elas segmentos de produção, ilhas de produção, ou células de produção, são fundamentalmente estáticos. Os fractais são dinâmicos e têm uma vitalidade inerente, que os leva a alterar a sua estrutura e as ligações existentes com outros fractais.

A vitalidade significa que é necessário, permanentemente, descobrir e tirar vantagem dos factores de sucesso. Warnecke (1993) enumera alguns dos factores internos de sucesso que influenciam a vitalidade do sistema, como sejam: custos; potencial de produção; investigação e desenvolvimento; eficiência da gestão; potencial de compras e distribuição; potencial financeiro; logística; localização; funcionários. Por sua vez, os factores de sucesso externos que influenciam a vitalidade são enumerados como sendo os seguintes: mercados de escoamento<sup>13</sup>; mercados de fornecimento<sup>14</sup>; concorrência; legislação.

O desempenho do fractal é sujeito a constante análise e avaliação, para que em qualquer momento e em função das conclusões, se proceda a alterações da sua estrutura e das ligações existentes, em benefício do sistema global. A avaliação pode efectuar-se com diferentes critérios, como sejam: o produto; procedimentos e materiais (podem evoluir); necessidade de informação e comunicação (a quebra de uma ligação pode levar à alteração da estrutura); movimentação de materiais; funcionários; necessidades de experiência e aprendizagem.

### **1.1.4. Orientação por Objectivos**

Os objectivos específicos de cada fractal devem ser gerados e adaptados através da coordenação entre os fractais participantes no processo, em função das necessidades. Este processo deve ser suportado por um mecanismo de herança que assegure a consistência do *sistema de objectivos*. Trata-se de um processo iterativo, não totalmente autónomo, existindo na realidade uma hierarquia na definição de objectivos. Esta hierarquia não é, por outro lado, totalmente rígida, permitindo a influência de objectivos “locais”, aparentemente mais individuais. Nos níveis mais elevados de hierarquia estabelecem-se objectivos da empresa, procedendo-se assim a um enquadramento de objectivos individuais, que não poderão contrariar os objectivos globais. Neste processo é necessária a existência de mecanismos de controlo de objectivos conflituosos e duplicados. Existe, no entanto, um ponto importante na fábrica virtual, que é o da influência dos níveis mais baixos da hierarquia sobre a formação dos objectivos globais. Esta influência pode ser explícita, sendo em muitos casos implícita, resultando do facto do sentido individual dos fractais influenciarem o sentido global da empresa. A empresa deve reconhecer e aproveitar explicitamente a

---

<sup>13</sup> “Outlet markets”.

<sup>14</sup> “Supply markets”.

potencialidade individual dos seus elementos, sendo por outro lado, implicitamente, conduzida num determinado sentido por esses mesmos elementos.

## **1.2 Sistemas de Produção Holónicos**

Os sistemas de produção holónicos<sup>15</sup>, descritos por Tharumarajah, Wells and Nemes (1996), são baseados em unidades de produção que estão contidas umas dentro de outras, criando subsistemas até à composição final do sistema de produção. Cada uma destas unidades de produção denomina-se Holon, tendo características simultaneamente do todo e de parte do todo. Representa o todo porque pode conter outros Holons, que por sua vez conterão outros holons com o intuito de criar um subsistema do sistema produtivo. Representa uma parte do todo porque pode estar contido noutros Holons e assim sucessivamente, até compor um sistema produtivo. Este conceito foi desenvolvido no âmbito de um projecto do programa internacional IMS, já referido anteriormente. Embora o trabalho de Koestler tenha servido de inspiração aos sistemas holónicos, neste projecto procedeu-se à definição de alguns conceitos:

- Holon: Um módulo de construção, autónomo e cooperante, de um sistema de produção para transformação, transporte, armazenamento e/ou validação de objectos, de informação ou físicos. É constituído por uma parte de processamento de informação e em muitos casos de processamento físico. Um Holon pode fazer parte de outro Holon.
- Autonomia: a capacidade de uma entidade criar e controlar a execução dos seus próprios planos e/ou estratégias.
- Cooperação: Um processo pelo qual um conjunto de entidades desenvolve planos aceites por todas as entidades, e os executa.
- Holarquia: Um sistema de holons que coopera para atingir um determinado objectivo. A Holarquia define as regras básicas para cooperação entre holons, limitando dessa forma a sua autonomia.
- Sistema de Produção Holónico: é a holarquia que integra todas as actividades de produção (nomeadamente a encomenda, projecto, planeamento, produção e marketing) que permitem implementar a empresa ágil.
- Atributos holónicos: atributos que fazem de uma entidade um holon, sendo o conjunto mínimo destes atributos a autonomia e a cooperação.

Um holon pode ser definido pelas suas funções ou tarefas, e funciona numa holarquia dentro de determinados limites de autonomia, definidos pelas regras impostas. As alterações no ambiente envolvente resultam na reacção dos holons que comunicam as suas mudanças de estado ao centro controlador, que por sua vez ajusta o seu curso de acção.

Um holon é ele próprio uma holarquia de holons, que são em certa medida controlados pelas regras que ele impõe. Por sua vez o holon está subordinado ao holon do qual faz parte. A Figura 4 representa a holarquia e as relações de “todo” – “parte de”.

---

<sup>15</sup> Termo adaptado do inglês ‘holonic’, que por sua vez representa a característica respeitante ao ‘holon’ que resulta da contracção da palavra grega ‘holo’, que representa o todo, com o sufixo ‘on’, que sugere uma parte (como em ‘proton’ ou ‘neutron’). O Holon representa simultaneamente o todo e uma parte do todo (Tharumarajah, Wells and Nemes (1996)).

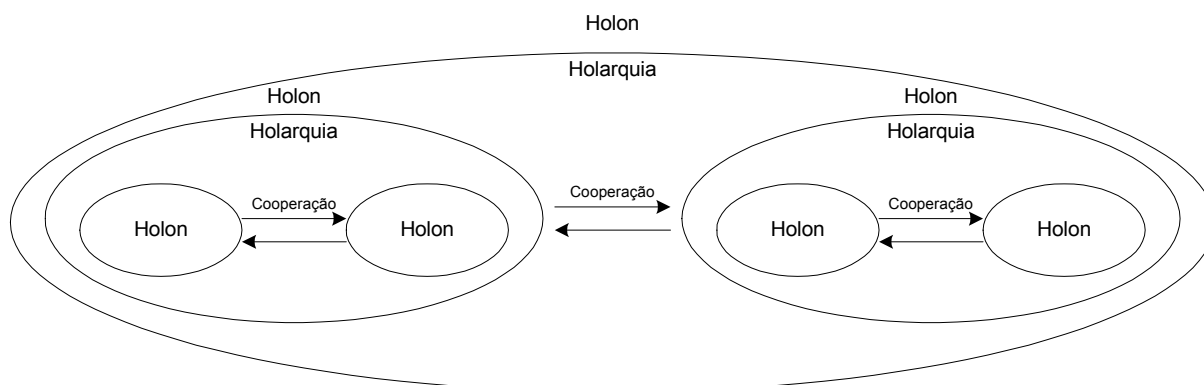


Figura 4: Sistema Holónico - retirado de Tharumarajah, Wells and Nemes (1996).

### 1.3 Sistemas de Produção Biónicos

Okino apresenta um novo conceito de sistema de produção inspirado na organização biológica dos seres vivos, denominada Produção Biónica (Tharumarajah, Wells and Nemes (1996), Kádar, Monostori and Szelke (1998)). Os seres vivos são compostos por entidades que se organizam e funcionam de uma forma autónoma dentro de uma estrutura hierárquica. As células agrupam-se em tecidos que por sua vez criam órgãos e estes os organismos, seres vivos, espécies, ...

A célula é a mais pequena entidade dos organismos vivos capaz de vida independente, podendo, isolada ou interagindo com outras células, executar todas as funções da vida. A célula é delimitada por uma membrana pela qual entram e saem os elementos químicos necessários ao seu funcionamento e controlo, estabelecendo a ligação entre o seu ambiente interno e externo e composta por um núcleo e por compartimentos especializados (organelo). No sistema de produção, a unidade de produção, equivalente à célula do sistema biológico recebe e fornece objectos de produção (materiais, produtos, e informação) do ambiente envolvente.

Nos sistemas biológicos existem fundamentalmente 3 tipos de interacção química, nomeadamente:

- Interacção entre as células e a matriz celular através de moléculas receptoras existentes na membrana.
- Interacção por junção entre células que depende fundamentalmente do tecido em que as células se encontram e de moléculas específicas que transportam informação relativa ao tipo de ligação a efectuar.
- Comunicação química e eléctrica entre células, possibilitada por moléculas sinalizadoras segregadas e que se deslocam pelo espaço entre células, ligando-se a proteínas receptoras existentes noutras células, podendo desencadear alterações de comportamento.

No funcionamento de sistemas distribuídos de produção a comunicação será efectuada pelo fluxo de materiais e de informação, regulada por unidades coordenadoras que permitirão a integração das actividades das unidades autónomas de produção. Estas unidades coordenadoras terão igualmente a função de interligar células entre diferentes níveis da hierarquia (Figura 5).

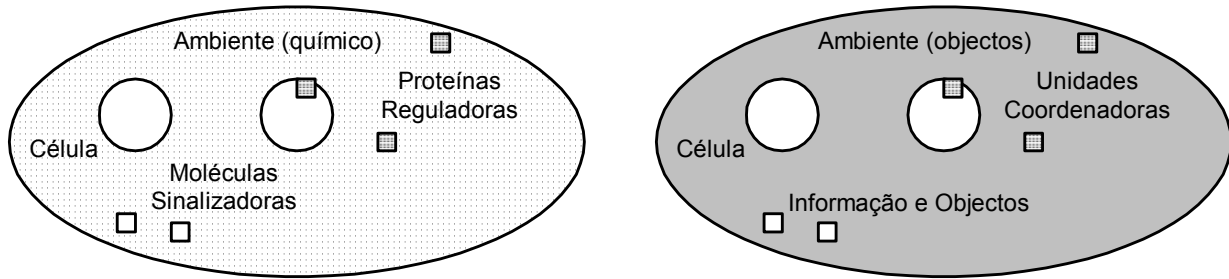


Figura 5: Comparação entre o sistema biológico e de produção

O processo de concepção de formas de vida baseia-se na transmissão de informação do tipo ADN entre as células. As células por acção de catalisadores específicos crescem através da absorção de moléculas e posterior divisão em duas células que contêm ambas a mesma informação do ADN da célula original. A diferenciação entre o funcionamento e estrutura das células dos organismos faz-se através do reconhecimento que as células iniciais têm da posição em que se encontram na montagem do organismo, realçando determinados cromossomas do ADN. Esta informação é passada para as células criadas a partir daquele ponto do crescimento. A causa deste mecanismo não é conhecida, existindo apenas algumas teorias que o tentam explicar. Num sistema de produção biónico supõe-se que também se podem criar unidades de produção por transmissão da informação necessária ao desenvolvimento da funcionalidade requerida.

O sistema Biónico de Produção é composto por um elemento básico denominado *modelon*, que por sua vez é composto por modelons de níveis hierárquicos inferiores que compõem uma hierarquia de modelons filhos, numa estrutura de modelons dentro de modelons. Além de ser composto por outros modelons, é também composto por operadores que realizam operações de coordenação e por um ambiente comum para armazenamento e troca de informação entre modelons (Figura 5).

Num exemplo de aplicação apresentado por Tharumarajah, Wells and Nemes (1996), um modelon de nível superior pode criar uma especificação para o projecto de um produto e em seguida enviar informação do tipo ADN para uma base de modelons, de forma a reunir modelons num nível inferior, para o projecto de partes do produto. Pela propagação deste processo é possível a criação de um sistema que projectará o produto. Num outro exemplo, um modelon consegue gerir a cooperação entre os recursos (outros modelons) para a produção duma peça, a partir da informação sobre a forma de produção da mesma e sobre os recursos necessários.

## **2. REVISÃO BIBLIOGRÁFICA SOBRE SISTEMAS DISTRIBUÍDOS DE PRODUÇÃO**

Os novos paradigmas de sistemas de produção definem um enquadramento genérico para o sistema de produção e para o seu funcionamento. É possível consultar publicações de revistas e conferências científicas com a descrição de trabalho realizado com o objectivo de enquadrar e definir de forma concisa estes conceitos, para apresentação de modelos e formas de projecto destes sistemas, especificação formal dos mesmos, análise de diferentes formas de gestão ou de programação da produção, com aplicação a casos de estudo e recorrendo a diferentes tecnologias. Estes temas são utilizados para classificar as referências bibliográficas relacionadas com Sistemas Distribuídos de Produção analisadas nesta secção, apresentando-se na Tabela 1 uma lista e a respectiva classificação das mesmas.

Tabela 1: Lista e classificação de referências bibliográficas relacionadas com sistemas distribuídos de produção.

	Referências	Definição	Modelo e projecto	Especificação Formal	Sistema de Informação	Programação e Controlo Prod.	CAPP CAM	Casos de aplicação	Tecnologia aplicada	Número de Citações
1.	Agrawal and Graves (1999)							X	X	0
2.	Archimède, Charbonnaud and Mercier (2003)					X				
3.	Barata <i>et al.</i> (2001)						X		X	
4.	Brun and Portioli (1999)					X				0
5.	Butala and Sluga (2002)		X						X	
6.	Camarinha-Matos <i>et al.</i> (1998)		X						X	11
7.	Crowe and Stahlman (1995)	X				X				2
8.	Davidrajah and Deng (2000)		X		X					0
9.	Fujii, Kaihara and Morita (2000)		X						X	0
10.	Gjerdrum, Shah and Papageorgiou (2001)					X			X	0
11.	Gunasekaran (1998)	X								15
12.	Gupta <i>et al.</i> (1998)						X		X	7
13.	Kádar, Monostori and Szelke (1998)		X						X	3
14.	Lecompte, Deschamps and Bourrières (2000)			X						0
15.	Maturana, Shen and Norrie (1999)		X						X	7
16.	Ming <i>et al.</i> (1998)		X				X		X	2
17.	Miyashita (1998)		X						X	3
18.	Monostori, Kádar and Hornyák (1998)		X				X			2
19.	Shen, Maturana and Norrie (2000b)		X						X	1
20.	Sluga and Butala (2001)		X							
21.	Sluga, Butala and Bervar (1998)		X							3
22.	Sousa and Ramos (1998)		X			X				7
23.	Sousa and Ramos (1999)					X				6
24.	Sun, Zhang and Nee (2001)						X		X	0
25.	Tharumarajah, Wells and Nemes (1996)		X							13
26.	Trentesaux, D., Tahon and Ladet (1998)					X			X	3
27.	Tsukada and Shin (1998)					X				5
28.	Van Brussel <i>et al.</i> (1998)		X							19
29.	Wiendahl and Ahrens (1997)					X				5
30.	Wu, N., Mao and Qian (1999)		X							2
31.	Zaremba <i>et al.</i> (1997)			X						1
32.	Zhao, J., Cheung and Young (1999)		X		X				X	0
33.	Zhao, Xiaobo, Wang and Luo (2000a)			X						1
34.	Zhao, Xiaobo, Wang and Luo (2000b)			X						2
35.	Zhao, Xiaobo, Wang and Luo (2001a)			X						1
36.	Zhao, Xiaobo, Wang and Luo (2001b)			X						0

Nesta dissertação foi realizada pesquisa bibliográfica de uma forma contínua tentando justificar, enquadrar e suportar o trabalho realizado. Esta pesquisa pode dividir-se em pesquisa de referência, como a que foi apresentada na secção anterior e em quase todos os capítulos, e pesquisa de suporte, apresentada com o intuito de enquadrar e justificar a utilização e desenvolvimento de vários conceitos.

Além deste tipo de pesquisa, realizou-se aquela que se pode denominar de pesquisa de enquadramento do trabalho sobre sistemas distribuídos de produção, realizado pela comunidade científica e que se apresenta nesta secção. Esta resulta de dois tipos de acção. Uma acção de pesquisa permanente sobre trabalhos que se podem enquadrar nos sistemas distribuídos de produção e que resulta da consulta de algumas revistas científicas, do cruzamento de informação e da orientação científica desta dissertação. Também se realizou um outro tipo de acção de pesquisa, sistemática e generalizada, durante um determinado período de tempo. O resultado desta acção de pesquisa, por ser demasiado alargado, foi limitado através de alguns critérios. Esta pesquisa resultou da utilização do serviço RAM (“Recent Advances in Manufacturing”) no site EEVL (<http://www.eevl.ac.uk/ram/index.php>) com as palavras “distributed manufacturing” ou “distributed production”. Nesta pesquisa são seleccionados todos os artigos que tenham, em qualquer posição do texto, a

palavra “distributed” e qualquer uma das outras: “manufacturing” ou “production”. Depois de reunido o maior número possível de artigos resultantes desta pesquisa, limitou-se o seu resultado, considerando apenas os artigos com um número de citações igual ou superior a três. Com zero, uma e duas citações obtém-se um número demasiado elevado de referências – na ordem das 160 –, muitas delas com citações dos próprios autores noutros artigos, tendo-se por essa razão optado por três ou mais citações como critério de selecção. Nesta secção são apresentados trabalhos, cuja selecção resulta, dos vários tipos de pesquisa referidos.

## **2.1 Definição de Sistema Distribuído de Produção e dos seus Conceitos**

Um sistema de produção não hierárquico<sup>16</sup> é definido por Crowe and Stahlman (1995) como sendo constituído por nós de processamento que recorrem a uma comunicação ponto a ponto, conforme necessário. Neste sistema, cada nó tem completo conhecimento das suas habilidades e é capaz de comunicar com outros nós para o substituírem ou complementarem na execução de tarefas. O controlo está distribuído pelos nós autónomos (também denominados de agentes ou células). As vantagens dos sistemas de produção não hierárquicos serão: complexidade reduzida do sistema global; redução de custos e atrasos provocados por tarefas de supervisão; manutenção e modificação do sistema mais simples, devido ao carácter modular e configurável da estrutura; tolerância a falhas; responsabilização do trabalhador. As desvantagens destes sistemas serão: ausência de soluções analíticas globais; hardware e software que forneçam a funcionalidade total ainda não estão disponíveis; tendência para a duplicação de dados; dificuldade em rastrear o trabalho de produção; possibilidade de bloqueio; dados nem sempre disponíveis no local certo.

As empresas ágeis são capazes, segundo Gunasekaran (1998), de rapidamente se reorganizarem e mesmo reconfigurarem, de forma a responder a uma oportunidade de mercado imediata, que poderá ser temporária. Para cumprir com este objectivo, deverão integrar-se num sistema de produção distribuído, composto por uma rede de serviços de produção que poderão ser utilizados por qualquer empresa como extensão das suas habilidades internas. *Este sistema de produção distribuído fisicamente será assim composto por agentes heterogéneos e estruturas de controlo que operam autonomamente durante o planeamento e execução das tarefas de produção.* A este tipo de sistema juntar-se-ão outros elementos que possibilitarão a implementação de uma empresa ágil: ferramentas e métricas para formação de empresas virtuais; ferramentas e métricas para formação rápida da associação de empresas; Engenharia Concorrente; sistemas de informação integrada de produtos, produção e empresas; ferramentas para rápida construção de protótipos; comércio electrónico.

## **2.2 Modelo e Projecto de Sistema Distribuído de Produção**

A arquitectura de Empresa Virtual (VE – “Virtual Enterprise”) apresentada por Camarinha-Matos *et al.* (1998) baseia-se num conceito de redes de empresas que cooperam como nós de uma rede de fornecedores, clientes e funções especializadas, criando uma única entidade de negócio. Uma das características distingue esta nova forma de acordo de cooperação entre empresas é o da utilização de tecnologias de informação para apoiar a criação de uma rede de informação. A aliança pode ser de curta ou de longa duração, com empresas

---

<sup>16</sup> “Heterarchy”.

que entram e abandonam a rede de uma forma dinâmica e outras que se mantêm ao longo do tempo. Este conceito admite a possibilidade de alianças baseadas numa empresa dominante e alianças baseadas em parceiros de igual dominância, sendo uma das variantes baseada na associação de algumas empresas em federações que fornecem infra-estruturas de coordenação e gestão. A arquitectura da Empresa Virtual deve permitir que a empresa mantenha a sua independência e ao mesmo tempo coopere de uma forma integrada. Assim sendo, cada empresa será encapsulada num nó estrutural que abrange o seu sistema de informação de planeamento e controlo de produção num módulo interno e que comunica a partir de um módulo de cooperação. Este módulo terá as funções de comunicação de informação comercial por EDI (“Electronic Data Interchange”), informação de produtos por STEP, informação do sistema de qualidade, informação de segurança, informação pública de suporte à rede de empresas, informação de interface de mapeamento, infra-estrutura de rede. As funcionalidades da Empresa Virtual são divididas em infra-estrutura de rede, serviços básicos, cooperação intermédia e cooperação avançada. A arquitectura das Empresas Virtuais é, inicialmente, baseada em dois tipos de nós fundamentais o do Coordenador da Rede e o do Nó da Empresa, que foi referido atrás. O coordenador da rede fornece as seguintes funcionalidades: regista novas empresas; fornece assistência às novas empresas para criarem o módulo de cooperação; mantém informação sobre o directório da rede; gera notícias sobre a configuração da rede; pode, possivelmente, servir de testemunha para negociações entre empresas. O Nó da Empresa ainda tem funcionalidades que permitem gerir a comunicação externa com outros nós e com o coordenador. O sistema de informação da Empresa Virtual é baseado num sistema federado / distribuído de gestão de informação orientado aos objectos denominado PEER, que suporta a troca de informação entre nós cooperantes autónomos e distribuídos. Neste sistema a informação e o controlo estão distribuídos ao longo da rede, sendo a inter-relação entre dois nós estabelecida através de Esquemas de Informação (“Schemas”). Todos os nós são representados por diversos esquemas: um esquema integrado com toda a informação que o nó conhece e só acessível internamente; um esquema local com parte da informação do esquema integrado; vários esquemas de exportação baseados no esquema local e que pode ser disponibilizado para diferentes nós externos; vários esquemas de importação a partir dos quais se obtém informação externa e que vai sendo utilizado para aumento do esquema integrado. A utilização destes esquemas permite manter a confidencialidade dos dados internos e evitar a replicação de informação. O funcionamento do sistema PEER apoia-se num Dicionário de Comunidade com informação sobre a sociedade de nós que pode ser consultada em qualquer momento por todos os nós.

O sistema de Produção Virtual modelado por Davidrajuh and Deng (2000), baseia-se numa empresa anfitriã que reúne um conjunto de outras empresas para construir o sistema. Cada uma das empresas publica a informação sobre os seus produtos recorrendo a XML (“eXtensible Markup Language”), dando-se a interacção por meio de um sistema de multiagentes com agentes móveis.

O modelo de Fábrica Virtual Distribuída (DFV – “Distributed Virtual Factory”) apresentado por Fujii, Kaihara and Morita (2000), é baseado na ligação por uma rede de computadores, de subsistemas referentes a diferentes áreas da fábrica. Cada uma destas áreas tem um sistema de simulação que é integrado no sistema global. Este modelo preconiza a integração de ilhas de automação e/ou simulação no sistema da Fábrica Virtual Distribuída. Um dos subsistemas corresponde a um sistema de gestão da fábrica, que recolhe



informação distribuída e toma decisões em conformidade com as estratégias a adoptar. Este sistema utiliza períodos de tempo (“Time Bucket”) como mecanismo para a sincronização dos subsistemas de simulação distribuídos, em que cada subsistema efectua a sua simulação de uma forma independente e concorrente, de acordo com o seu relógio de simulação. No final de um período cessa a execução e espera-se que todos os subsistemas cessem a sua própria execução, estabelecendo-se então a comunicação entre todos os subsistemas da fábrica. Em seguida, apresentam um modelo de simulação para determinação de custos de produção baseado em sete subsistemas: gestão da fábrica; fábrica de cooperação; processador a; processador b; armazenamento de matérias primas e de produtos semi-acabados; linha de montagem. Esta fábrica virtual processa 5 produtos com 50 tipos de materiais segundo um fluxo de processamento predefinido, podendo funcionar num ambiente tradicional do tipo “push” ou “pull”. Com base neste modelo e num modelo formal de atribuição de custos de actividades aos produtos, é executada uma simulação que permite efectuar uma análise de custos detalhada, e dessa forma fazer uma oferta para o sistema de empresas virtuais, por determinação do custo de produção na fábrica virtual.

Kádar, Monostori and Szelke (1998) apresentam um resumo de conceitos de sistemas de produção avançados, como, Sistemas de Produção Biónicos, Holónicos e Fractais. Baseado nestes conceitos apresentam um sistema orientado aos objectos de simulação de um sistema de produção holónico. Este sistema é constituído por dois agentes fundamentais: Agentes Máquina representando os recursos de produção e também referido em Monostori, Kádar and Hornyák (1998); Agente Encomenda para coordenação dos trabalhos. O agente máquina é composto por um módulo de registo na comunidade, um de comunicação e um de supervisão da máquina com as possíveis acções do agente descritas numa base de conhecimento. Este agente armazena informação sobre as aptidões e a agenda do recurso numa base de dados interna. O agente Encomenda coordena a produção verificando a possibilidade de aceitação das encomendas e atribuição de tarefas às máquinas. O processo de atribuição e coordenação de trabalhos efectua-se pela aplicação de um protocolo de comunicação baseado na Rede de Contrato<sup>17</sup> de Smith. Os produtos são representados em ficheiros com os planos de processo alternativos. O agente Encomenda tem um módulo de registo, um módulo de comunicação, um de processamento das encomendas e outro de gestão dos trabalhos a atribuir às máquinas. A verificação da possibilidade global de fabrico permite iniciar um processo de requisição de ofertas por parte das máquinas, com posterior selecção de máquinas para execução dos trabalhos. Neste sistema apenas o módulo de registo funciona de uma forma centralizada.

Maturana, Shen and Norrie (1999) apresentam a arquitectura do sistema de produção multiagente MetaMorph e uma aplicação protótipo. O sistema de produção é formado por subsistemas heterogéneos cooperantes que estabelecem uma coligação alargada virtual e dinâmica, isto é, lida com os aspectos de coligação estratégica entre parceiros da Empresa Virtual. Neste sistema pretende-se aplicar Engenharia Concorrente ao sistema de projecto do produto e do processo e ao sistema de gestão da produção para diminuir o tempo de lançamento do produto no mercado. A arquitectura do sistema MetaMorph baseia-se numa federação de agentes coordenados por agentes de mediação, que utilizam técnicas de mediação de mercados e de apoio à união directa entre agentes para estabelecerem a comunicação entre elementos do

---

<sup>17</sup> “Contract Net”.

sistema. Os agentes de mediação fornecem mecanismos de apoio à comunicação entre qualquer tipo de agente do sistema, e os agentes Recurso representam dispositivos e operações de produção. Os Agentes Recurso são cooperativos e autónomos e os Agentes Mediadores além desta características, ainda apresentam características de aprendizagem. Os agentes Recurso formam coligações para trabalharem em cooperação de forma a resolverem problemas colocados por tarefas que lhes são requisitadas por agentes Mediadores. Se os agentes Recurso não conseguirem resolver o problema na sua coligação, procuram outras coligações que o possam fazer. Os agentes Mediadores são compostos por diversas funcionalidades que lhes permitem executar actividades de diversos domínios: empresa; especificação e projecto do produto; organização virtual; planeamento e programação da produção; execução; comunicação através do protocolo KQML; aprendizagem. Com base nesta arquitectura estes agentes podem ser especializados num dos domínios aumentando a sua utilidade numa determinada área. Cada empresa tem que ter um agente mediador de alto nível que funcione como integrador do sistema, fornecendo uma visão global. Este agente reconhece todos os agentes mediadores de níveis inferiores e todos os recursos. As tarefas de alto nível são decompostas por agentes mediadores estáticos, i.e. Agentes Mediadores Activos que negociam subtarefas com clones de agentes Recurso. Estes clones representam os Agentes Recurso no local de negociação, evitando aumento de perda de tempo por efeitos de comunicação. Os Agentes Mediadores denominados Gestores de Agentes de Dados fornecem funcionalidade para pesquisa de outras coligações que permitam resolver problemas que não estão totalmente resolvidos. Neste trabalho ainda se descreve, sucintamente, o processo de aprendizagem dos agentes mediadores que se utiliza para atribuição de tarefas a recursos se forem reconhecidos padrões passados.

Ming *et al.* (1998) apresentam um modelo de representação de um centro de maquinagem desenvolvido com objectos distribuídos CORBA, de forma a facilitar a possibilidade de actuação desses objectos em diferentes plataformas e a sua ligação a outros centros de maquinagem. Os objectos desenvolvidos representam diversas funções do centro, nomeadamente: análise de características do produto, planeamento de processo, processamento de imagem, maquinagem, monitorização, controlo de maquinagem, e agente de cooperação com outros centros de maquinagem.

Miyashita (1998) apresenta o sistema CAMPS multiagente de resolução de problemas de planeamento e programação da produção. O sistema distribuído de produção é composto por um agente gestor que recebe encomendas de clientes e as divide em trabalhos que envia para os agentes planeadores. O agente planeador envia os trabalhos para os agentes programadores de produção e tem a responsabilidade de integrar as soluções obtidas e manter as restrições globais. O agente programador representa um recurso e constrói uma sequência para as operações atribuídas pelo planeador. A resolução de conflitos entre programas de produção desenvolvidos pelos diversos agentes programadores pode ser efectuado por três processos aplicados sucessivamente até que se obtenham soluções realizáveis: alteração das restrições intra-agente; alteração das restrições inter-agente; alteração do problema. Os dois primeiros processos de resolução de conflitos dependem apenas dos agentes programadores, e resultam da inclusão de restrições adicionais, como restrições de precedência entre operações, que permitam obter programas de produção realizáveis. Em último caso o agente planeador altera o problema de forma a obter soluções. As acções que este agente pode aplicar

são as seguintes: aumentar o prazo de entrega; aumentar a capacidade do recurso; cancelar trabalhos; subcontratar trabalhos a outros agentes planeadores; subcontratar operações a outros agentes programadores.

Shen, Maturana and Norrie (2000b) descrevem a arquitectura do sistema inteligente MetaMorph II de projecto e produção baseado em agentes de software. Este sistema resultou da evolução dos sistemas multiagente MetaMorph descrito atrás e do sistema DIDE. Este sistema foi desenvolvido, originalmente, para projecto de Engenharia, podendo a sua arquitectura ser utilizada para desenvolver sistemas distribuídos de produção. Esta arquitectura baseia-se em agentes cognitivos de software que permitem integrar ferramentas de Engenharia e especialistas humanos num ambiente aberto. Ao contrário do sistema MetaMorph, o sistema DIDE não utiliza uma arquitectura baseada em agentes mediadores. Os agentes organizam-se em função da organização da própria empresa e podem participar na resolução dos problemas se compreenderem as mensagens enviadas. O sistema MetaMorph II agrupa características dos dois sistemas referidos: sistema multiagente e sistema baseado em agentes mediadores. A arquitectura deste sistema baseia-se num agente mediador da empresa que facilita a integração de subsistemas funcionais representados por agentes mediadores. Os subsistemas representam empresas parceiras, funções de marketing, projecto, planeamento e programação, execução e fornecimento de materiais. Este sistema assegura a integração entre subsistemas de projecto, como o sistema DIDE, representados por mediadores de projecto e subsistemas de planeamento e programação de produção representados por agentes recurso, através da garantia que estes fornecem de aptidão e capacidade de fabrico. Os recursos de produção são agrupados por tipo, como ferramentas, máquinas, transportadores, trabalhadores. Estes agrupamentos são coordenados por agentes mediadores. No entanto, é possível estabelecer comunicação directamente com agentes sem passar pelo mediador. Outras empresas associam-se ao sistema registando os seus mediadores de empresa como qualquer outros mediador de subsistema. Os especialistas humanos podem ser representados no sistema através de agentes tal como no sistema DIDE.

O modelo de sistema holónico apresentado por Sousa and Ramos (1998), baseia-se em agentes inteligentes que representam os recursos de produção, as tarefas de produção e um gestor de tarefas. O utilizador coloca uma encomenda, com uma determinada data de entrega, ao *Gestor de Tarefas*, que cria agentes *Tarefa* para lidar com os pedidos de encomenda e gere as alterações provocadas por alterações do ambiente (falhas de máquinas, alterações na encomenda...). O agente *Tarefa*, negocia directamente com os agentes *Recurso* de modo a executar a tarefa pretendida. A informação sobre o plano de produção, de modo a conhecer os recursos com os quais deve negociar, é fornecida por um outro sistema.

Segundo Sluga and Butala (2001) algumas das mais esperadas características de novos sistemas de produção são: arquitectura aberta multinível; capacidades de comunicação avançadas; tomadas de decisão descentralizadas; capacidade de auto-organização; redefinição dos sistemas de trabalho em função de autonomia, adaptabilidade evolucionária, reconfiguração, cooperação, interactividade, orientação à tarefa, comunicação, coordenação, e aprendizagem. Este trabalho está centrado na forma de estruturação do sistema, baseada em mecanismos de mercado, para o cumprimento de algum objectivo e na forma como o sistema pode adaptar a sua estrutura em função de alterações internas ou externas.

Sluga, Butala and Bervar (1998) apresentam o modelo de um sistema de produção baseado em agentes. Em Butala and Sluga (2002) aprofundam este modelo e apresentam a forma de coordenação dos agentes. Este modelo baseia-se no conceito de sistemas de produção elementares (EWS – “Elementary Work Systems”) com comportamento autónomo, capacidade e aptidão para desempenhar uma determinada operação de produção. Este EWS é composto por elementos de hardware necessários para identificação, controlo, optimização e implementação do processo de trabalho, e por um operador humano autónomo para tomadas de decisão e para projecto. A auto-organização destes sistemas, baseia-se numa arquitectura implementada com a tecnologia multiagente e controlada por mecanismos de regulação de mercado. Introduzem a noção de agente de sistema de trabalho virtual (VWS – “Virtual Work System”) para representar um EWS no sistema. Este é um elemento de software com as seguintes funcionalidades básicas: sensor, avaliador, accionador e mecanismos de inferência. A coordenação do sistema resulta da colocação de divisão de tarefas por parte de agentes mediadores, que avaliam e seleccionam as melhores ofertas dos VWS para essas tarefas. Posteriormente os agentes mediadores atribuem tarefas aos VWS através de contratos estabelecidos, isto é, através de trocas de mensagens de atribuição e confirmação entre os agentes.

Tharumarajah, Wells and Nemes (1996) é um artigo de referência de descrição dos conceitos de sistemas de produção Biónicos, Holónicos e Fractais, que apresenta descrições detalhadas destes conceitos e alguns exemplos de aplicação. Não se vão repetir, nesta secção, estes conceitos que foram largamente descritos na primeira secção deste capítulo que contém inúmeras referências a este artigo.

O modelo de sistema de produção Holónico PROSA apresentado por Van Brussel *et al.* (1998), que é o artigo mais citado dos que são referenciados nesta secção, é baseado em três holons básicos mais um adicional. Os holons básicos são o holon Produto, Encomenda e Recurso, sendo o holon adicional o holon “Staff”. Este modelo tem princípios de representação Orientados aos Objectos, nomeadamente com representação de holons como especialização de classes mais genéricas e por utilização de mecanismos de herança. O modelo é apresentado por diagramas de classes em linguagem UML, que representam o aspecto estático do modelo, com uma parte das propriedades e métodos de algumas classes. Também apresentam as relações entre os holons básicos. Um sistema holónico de produção, isto é, uma holarquia, é constituída por pelo menos um holon Produto, pelo menos um holon Recurso e por zero ou mais holons Encomenda. O holon Produto tem funções de fornecimento de informação sobre o produto, de gestão dessa mesma informação, e de fornecimento de funções de projecto e planeamento de processo. O holon Recurso tem funções de gestão do recurso de produção, como fábrica, oficina, máquinas, meios auxiliares, ferramentas e matérias primas, podendo ser uma parte em software e outra em hardware, podendo aceitar ou recusar tarefas de produção, controlar processos de produção, planear manutenção e gerir informação sobre sub-recursos. Este recurso pode ser especializado em outros holons recurso como equipamento, “workstation”, “shop” e “fábrica”, que por sua vez também podem ser especializados em robots, máquinas NC e células de montagem entre outros. Este artigo apresenta uma visão similar à que se apresenta em Lima and Silva (1998), sobre a possibilidade dos recursos pertencerem a diferentes sistemas de produção (holarquias) simultaneamente mas substancialmente diferente sobre a noção de recursos autónomos e não autónomos, inexistente no modelo PROSA. O holon Encomenda só pode estar associado a um produto final, e encarrega-se da atribuição de

tarefas aos holons Recurso. Não é formalmente abordada a questão da autonomia de recursos e a ligação com recursos não autónomos, nem a forma de geração de configurações do sistema em função de novas encomendas, nem se estende a modelação holónica aos produtos e encomendas. O holon “Staff” adiciona funcionalidades de especialização, sistemas periciais, sistemas proprietários ou algoritmos centralizados ao modelo PROSA.

O modelo desenvolvido nesta dissertação foi inicialmente apresentado numa versão parcial em Lima and Silva (1998), baseando-se numa visão fractal do sistema de produção e também em três classes de objectos fundamentais Produto, Recurso e Encomenda. As principais diferenças do modelo de Van Brussel *et al.* (1998) relativamente ao que é apresentado nesta dissertação está na clarificação do conceito de autonomia dos recursos, na generalização do conceito de produto a todos os elementos da estrutura do produto final, na modelação de todos os três elementos do sistema de produção de forma similar baseada na agregação de elementos, na apresentação de um mecanismo de configuração e reconfiguração baseado na encomenda e na representação formal desta agregação que permite gerar configurações de sistemas de produção.

O modelo de Empresa Virtual Ágil (AVE – “Agile Virtual Enterprise”) referido por Wu, N., Mao and Qian (1999) é composto por um sistema de produção com recursos distribuídos, que podem pertencer a diferentes empresas, com a possibilidade de reconfiguração dinâmica de acordo com o produto encomendado. Quando uma empresa tem uma encomenda para um produto novo e quer configurar uma AVE, inicia um processo de recolha de propostas. Esta empresa torna-se na empresa principal que vai reunir à sua volta outras empresas, que fizeram ofertas para fornecer algum dos serviços requeridos. A Empresa Virtual Ágil é uma empresa orientada ao produto, no sentido em que é criada para responder à necessidade de fornecimento de algum produto. No entanto, neste artigo, a decomposição das tarefas, logo dos serviços requeridos, é efectuada em função dos processos produtivos. Para a selecção de empresas que executem os serviços requeridos pela Empresa Virtual Ágil recorre-se a uma formulação inteira do problema e transformam-no numa formulação por grafos para o qual apresentam um algoritmo de resolução. A formulação é efectuada considerando as ofertas de cada possível fornecedor e os custos de transporte entre empresas da AVE. Não é apresentada uma forma automática de configurar o sistema, que depende, no exemplo apresentado, dos processos definidos à partida e para os quais se procura parceiros. A própria empresa principal pode, no entanto, apresentar propostas para os serviços requeridos. Outra das características do modelo é o de distinguir custos de transporte e de produção.

Zhao, J., Cheung and Young (1999) apresentam um modelo de sistema de informação para empresas virtuais orientado aos objectos, com recurso à representação UML. O modelo do sistema de produção é centrado num objecto representativo de instalações de produção denominado “Facility”, que corresponde à generalização de diferentes instalações, como sejam Empresas, Fábricas, “Shop”, Células e Postos de Trabalho. A instalação é ainda composta por diferentes objectos:

- Recursos representados pelo objecto “Resource” com a respectiva descrição e que pode, por sua vez, ser composto por diferentes tipos de recursos (tipos de máquinas) e por matérias primas que pode processar;

- Processos representados pelo objecto “Process” com a respectiva descrição e composto por diferentes tipos de processos;
- Estratégias representadas pelo objecto “Strategy” com a respectiva descrição da estratégia de utilização do recurso.

## **2.3 Especificação Formal de Sistemas Distribuídos de Produção**

Lecompte, Deschamps and Bourrières (2000) apresentam um modelo algébrico, apoiado em Redes de Petri, para representação de recursos agregados e respectivas tarefas agregadas, para formalização do problema de atribuição de recursos em diferentes níveis de agregação numa Empresa Virtual. Este modelo algébrico permite ligar formalmente os recursos elementares de uma empresa aos recursos agregados, nível a nível, até ao nível mais abstracto associado à tarefa mais genérica.

Zaremba *et al.* (1997) apresentam uma formalização algébrica para controlo de sistemas distribuídos de veículos automáticos (AGV - “Automatic Guided Vehicle”). O sistema AGVS (“AGV System”) corresponde a um sistema distribuído de processos autónomos e repetitivos (fluxos dos veículos), que partilham um conjunto comum de secções de caminho.

O sistema de produção reconfigurável (RMS – “Reconfigurable Manufacturing System”) é definido por Zhao, Xiaobo, Wang and Luo (2000a) como um sistema que pode adoptar diferentes configurações, e em que cada uma dessas configurações é utilizada para produzir uma família de produtos similares. Quando uma família de produtos é seleccionada, através de critérios de selecção de famílias de produtos, recorre-se a uma das hipóteses de configuração para produção de encomendas daquela família. Cada vez que é seleccionada uma família de produtos diferente da anterior para ser produzida, dá-se uma alteração de configuração com um determinado custo para o sistema. A avaliação do desempenho do sistema segundo diferentes critérios permite que o sistema evolua. Por exemplo, a alteração das condições de mercado poderão aconselhar a revisão do limite de encomendas aceites por família de produtos. Este sistema é representado formalmente por um modelo estocástico, que permite, a partir dos custos de reconfiguração e do lucro obtido na produção, formalizar:

- o problema de configuração óptima do sistema, para o qual são apresentados dois algoritmos de resolução em Zhao, Xiaobo, Wang and Luo (2000b);
- a política de selecção óptima da família de produtos a produzir, para a qual são apresentados dois procedimentos de resolução em Zhao, Xiaobo, Wang and Luo (2001a).

A medida do desempenho depende do número de encomendas aceites em relação ao número de encomendas efectuadas, sendo apresentadas duas forma de resolução em Zhao, Xiaobo, Wang and Luo (2001b).

## **2.4 Programação e Controlo da Produção em Sistemas Distribuídos de Produção**

Archimède, Charbonnaud and Mercier (2003) apresentam um modelo para avaliação de programas de produção estabelecidos entre locais de produção distribuídos, denominado MSR-PAC. Este modelo permite estudar a sensibilidade dos planos de produção em presença de perturbações. Este modelo é composto por módulos de programação de produção, comando e monitorização do sistema de fabrico. Um modelo

multiagente baseado num protocolo de coordenação do tipo Redes de Contrato permite obter programas de produção para um conjunto de recursos distribuídos. Este programa toma em consideração os tempos de transporte entre locais de produção e tempos de processamento. O modelo MSR-PAC utiliza esta solução de programação para actuar sobre o sistema de fabrico, enviando ordens de produção. A informação de monitorização é utilizada para construir novos programas. O MSR-PAC verifica a existência de perturbações sobre o programa de produção e avalia o seu efeito, decidindo-se pela necessidade de construir novos programas de uma forma parcial ou global.

Brun and Portioli (1999) apresentam um sistema baseado em multiagentes para a programação da produção em sistemas de produção que envolvem montagem, exigindo a coordenação da produção de componentes, por oposição a sistemas que apenas abordam o problema da ordenação de operações sequenciais. Em cada montagem existe um quadro<sup>18</sup> em que cada agente filho, correspondente a componentes ou submontagens, publica e actualiza permanentemente a sua data de entrega esperada. Este quadro é visível para todos os agentes relacionados com aquela montagem. Assim sendo, a oferta de cada agente dependerá também das datas esperadas dos outros agentes, não sendo necessário acelerar o seu trabalho se houver outros atrasados. Neste modelo existe um conjunto de agentes relacionados com a estrutura do produto e com os recursos associados ao processamento das várias partes dessa estrutura.

No sistema de produção *quase* não hierárquico de Crowe and Stahlman (1995), pretende-se reduzir as desvantagens dos sistemas não hierárquicos, colocando um coordenador de células que toma decisões sobre a atribuição de trabalhos às células, considerando critérios globais. Estes critérios estarão relacionados fundamentalmente com o balanceamento da carga das células, o impedimento de bloqueios e a disponibilização da informação correcta.

Na área da logística de cadeias de fornecimento, Gjerdrum, Shah and Papageorgiou (2001) apresentam um modelo do sistema baseado em sete agentes fundamentais, que representam o cliente, as fábricas, os armazéns de distribuição e agentes de coordenação. Neste modelo, representativo da cadeia de fornecimento, a selecção das diferentes ligações é efectuada por critérios de menor custo para encomendas com datas de entrega definidas pelo cliente.

Monostori, Kádár and Hornyák (1998) apresentam um sistema de programação da produção apoiado num modelo multiagente de um sistema de produção holónico. Este modelo multiagente é detalhado em Kádár, Monostori and Szelke (1998). Cada Agente Encomenda selecciona um plano de processo para um dos produtos da encomenda, considerando-o um trabalho - job - a programar, e anuncia as tarefas desse trabalho. Os agentes Máquina têm um tempo limite para efectuarem ofertas. Estes agentes fazem ofertas se tiverem a aptidão de processamento daquela tarefa, reservando o tempo de execução daquela tarefa no tempo livre da sua agenda. Pode haver várias reservas no mesmo período de tempo livre, mas apenas uma delas se pode tornar fixa, se o recurso for seleccionado. A programação de um trabalho só termina com a selecção da oferta para a última tarefa desse trabalho. O processo de programação continua com os trabalhos correspondentes aos planos de processo dos produtos seguintes da encomenda.

---

<sup>18</sup> "Blackboard".

No modelo de sistema holónico apresentado por Sousa and Ramos (1998), os agentes Recurso, depois de receberem pedidos para execução de uma determinada tarefa, analisam a sua agenda e seleccionam janelas de tempo em que a podem executar. Em seguida, enviam esta informação para os agentes que executam tarefas posteriores, até à última tarefa. A informação sobre as janelas de tempo seleccionadas por cada agente Recurso para a execução de uma tarefa, são novamente enviadas, em sentido contrário, da última tarefa para os agentes predecessores. Finalmente, todos os agentes Recurso respondem ao agente *Tarefa*, que por sua vez, toma a decisão de atribuir as tarefas aos agentes. A informação de possibilidade de aceitação da encomenda é então enviada ao gestor de tarefas. Para evitar conflitos de negociação entre tarefas e recursos, o gestor de tarefas só permite o início da negociação de tarefas que não utilizem recursos já em negociação. Este processo está descrito em detalhe em Sousa and Ramos (1999).

Sousa and Ramos (1999) apresentam um protocolo de negociação, baseado no protocolo de Rede de Contrato de Smith, para programação dinâmica da produção. Esta programação baseia-se na premissa do cumprimento de datas de entrega das tarefas. O Holon Tarefa recebe uma tarefa do módulo de gestão de tarefas e obtém planos de produção a partir do holon de planeamento de processo, passando a conhecer a sequência de operações a executar. Em seguida envia pedidos aos Holons Recurso, indicando a tarefa, uma janela inicial de execução e informação sobre as operações anteriores e seguintes. Os agentes Recurso vão passar por duas fases antes realizarem uma oferta. Uma primeira fase de influência dos agentes Recurso seguintes com envio de janelas livres onde se poderão realizar essas operações e uma segunda fase inversa com envio de janelas livres para as operações anteriores. Depois de receberem as janelas dos recursos das operações anteriores e posteriores vão construir uma ofertas que enviam ao agente Tarefa. Este agente selecciona tarefas em função da aplicação de heurísticas de escalonamento. Se não for possível obter uma solução de coordenação entre os holons Recurso, o holon Tarefa tenta alterar os recursos para as operações. Se for possível obter uma solução, notificam-se os holon Recurso e envia-se a solução para o Gestor de Tarefas. Para evitar conflitos entre recursos aptos para a mesma operação, o gestor de Tarefas só envia tarefas para negociação se não utilizarem os mesmos recursos. Para esse efeito mantém uma lista de recursos em negociação. Um Holon Tarefa pode libertar recursos se os considerar menos aptos para uma determinada operação, possibilitando assim, o início de outras negociações. Quando o holon Recurso detecta uma falha da qual não consegue recuperar, envia avisos para os holons Tarefa com o qual tem compromissos. Este holon reinicia processos de negociação para substituir aquele recurso. Na presença de encomendas prioritárias com datas de entrega exigentes, o holon recurso pode ter que atrasar outras operações. Neste caso, o holon Tarefa pode ter que negociar com outros recursos para lhe para que sejam atribuídas janelas livres, necessárias para o cumprimento daquela encomenda.

O Sistema Distribuído de Produção descrito por Trentesaux, D., Tahon and Ladet (1998) é composto por agentes que controlam um ou mais recursos de produção. Estes agentes cooperam entre si para determinar soluções de programação de produção e comunicam verticalmente com utilizadores do sistema que lhes atribuem tarefas. Os agentes determinam soluções de programação de produção para tarefas atribuídas globalmente pelos utilizadores, após verificação da existência de suficiente capacidade disponível. O Sistema Híbrido é estruturalmente idêntico ao Distribuído, mas operacionalmente diferente, na medida em que



determina a programação da produção apenas para o recurso estrangulamento<sup>19</sup>, e a partir desse ajusta o programa dos outros recursos. O recurso estrangulamento é determinado para cada conjunto de tarefas a programar, como aquele que restringe o débito do sistema, isto é, aquele que limita o número de unidades produzidas por unidade de tempo. Neste trabalho, os autores comparam o desempenho de três arquitecturas de controlo de actividade produtiva a partir das medidas de média do atraso das tarefas e desvio padrão do atraso. As arquitecturas comparadas são a Arquitectura centralizada e as duas descritas, Distribuída e Híbrida. Na arquitectura centralizada obtém-se uma programação mais eficiente, aplicação industrial mais viável, mas uma menor reactividade e flexibilidade relativamente a acontecimentos imponderáveis. Na arquitectura Distribuída obtém-se soluções de programação menos eficientes e implementações industriais menos viáveis por não haver garantia de obtenção de uma programação óptima nem utilizar dados de previsão, dificultando a sincronização de trabalhos. Por outro lado, permite obter uma maior reactividade e flexibilidade relativamente a acontecimentos imponderáveis. A solução Híbrida fornece soluções intermédias de programação e viabilidade industrial e mantém boas características de reactividade e flexibilidade herdadas do sistema distribuído.

Tsukada and Shin (1998) uma abordagem distribuída ao problema de partilha de ferramentas entre células de sistemas de produção flexíveis. Este sistema é composto por controladores de células com dois tipos de agentes: Agente de Gestão de Tarefas e Agente de Gestão de Ferramentas. O Agente de Gestão de Tarefas é responsável por programar as tarefas e requisitar a utilização de ferramentas adequadas ao Agente de Gestão de Ferramentas. Se não for possível cumprir com o programa, rejeita-se a tarefa, requisita-se a reestruturação completa do programa de utilização das ferramentas ou negocia-se com outros Agentes de Gestão de Tarefas a utilização dos seus períodos de afectação da ferramenta pretendida. Perante encomendas de última hora, uma célula tenta assegurar as tarefas que lhe são atribuídas por três métodos locais: sem alterar o programa; alterando apenas as tarefas não prioritárias; removendo tarefas não prioritárias e alterando as prioritárias. Se não for possível executar a tarefa sem ferramentas adicionais pode pedi-las emprestadas a outras células, se dessa forma resolver o problema de programação segundo um dos métodos referidos.

O sistema de produção “auto-organizado”<sup>20</sup> preconizado por Wiendahl and Ahrens (1997), será composto por unidades funcionais autónomas interligadas, de forma a poder ser estabelecida a troca de serviços necessária ao cumprimento de objectivos globais. Estas unidades funcionais correspondem a agentes inteligentes, que em vários casos de simulação efectuada pelos autores, foram agentes humanos. No sistema de produção apresentado existem dois agentes fundamentais: Agentes de Encomenda, que actuam em representação do cliente; Agentes Máquina, que actuam em representação da empresa. O sistema opera segundo o conceito de regras de mercado. Os agentes do mesmo tipo competem pelos serviços oferecidos pelo outro tipo de agente, nomeadamente: os Agentes Encomenda competem pela utilização dos recursos das empresas, e os Agentes Máquina competem pela melhor oferta para utilização dos seus recursos. Existe ainda um conflito entre os objectivos dos agentes de tipo diferente, em que os Agentes Encomenda tentam pagar o

---

<sup>19</sup> “Bottleneck”.

<sup>20</sup> “Self-organized”.

mínimo pelos recursos dos Agentes Máquina, que tentam obter o máximo pela utilização desses recursos. O custo da transação, segundo os autores, penaliza este tipo de sistema em relação ao sistema centralizado, desde que não exista auxílio de formas automáticas de comunicação e de processamento da informação.

## **2.5 Sistemas CAPP e CAM em Sistemas Distribuídos de Produção**

Uma das partes do sistema distribuído de produção corresponde ao planeamento de processo auxiliado por computador. Alguns autores debruçam-se sobre este problema, nomeadamente sobre a forma de gerar automaticamente um plano de processo a partir do projecto do produto. Sun, Zhang and Nee (2001), deram mais uma contribuição neste campo, construindo um sistema multiagentes para a resolução deste problema. Este sistema baseia-se numa arquitectura federada, apoiada num agente coordenador (“Facilitator”) da actividade de todo o sistema. Recebe as mensagens dos outros agentes e reenvia-as para o agente pretendido. Os outros agentes correspondem às diversas funções do sistema: projecto, estimação de custos, planeamento de processo, avaliação da possibilidade de produção, descrição da habilidade de produção, identificação de características (“features”) do produto e avaliação da possibilidade de montagem.

Barata *et al.* (2001) apresentam um resumo de tecnologias e normas que podem ser utilizadas na implementação de sistemas integrados de fabrico, que servem de enquadramento à implementação de um sistema baseado em agentes para controlo de um sistema flexível de fabrico denominado NovaFlex. Uma das características deste sistema é o da existência de um processo de *agentificação* (“*agentification*”) dos equipamentos físicos do sistema. Estes são integrados no sistema através de um encapsulamento (“wrapper”) que permite o seu controlo através do sistema multiagente. O sistema multiagente têm dois tipos de organização baseadas no agrupamento de agentes. No primeiro o agrupamento funciona num sistema de gestão de agentes que fornece, numa organização estabelecida a longo prazo, funcionalidades relacionadas com os equipamentos de produção. No segundo tipo de organização os agentes funcionam em consórcio enquanto o objectivo que levou ao seu agrupamento existir.

Gupta *et al.* (1998) apresentam um sistema de planeamento automático de processo para operações de quinagem (“*bending operations*”) de chapa. Este sistema baseia-se numa arquitectura distribuída que permite gerar modelos abertos para gerar planos de processo, com encapsulamento de conhecimento especializado em módulos de planeamento especializado. O sistema é constituído por um módulo de planeamento de operações que comunica com módulos especializados: planeamento de preparação e de ferramentas; planeamento de posições de fixação para quinagem; planeamento de movimento. O planeador central envia questões aos planeadores especializados e em função das respostas desenvolve o plano de processo parcialmente optimizado. O módulo de planeamento recebe o projecto CAD, entrega um plano de processo que envia para o gestor de planos que por sua vez envia para a estação de quinagem. Este plano pode ser actualizado em função dos problemas que possam surgir na execução e reutilizado para as restantes unidades do lote. O planeador central cria um espaço de estados com início numa folha de metal e conclusão na peça pretendida, isto é, na folha dobrada na forma requerida. Este módulo vai gerando estados intermédios com operações adicionais de quinagem alternativas que vai seleccionando em função da avaliação das respostas dos planeadores especializados. A partir do desenho CAD o sistema determina

automaticamente: a sequência de operações, as ferramentas e arras de robot necessárias, a disposição de ferramentas, as posições de fixação para quinagem, e os movimentos do robot para fabrico da peça.

Agrawal and Graves (1999) descrevem um sistema de determinação de custos para a produção de placas de circuito impresso, num ambiente distribuído de produção. Este sistema determina custos e tempos de produção, analisando as possibilidades de produção e ligando os sistemas de projecto aos sistemas de fabrico, que podem estar distribuídos por diferentes empresas.

## **2.6 Tecnologia Aplicada nos Sistemas Distribuídos de Produção**

A principal tecnologia utilizada na representação de sistemas de produção distribuídos tem sido a dos sistemas multiagente, nomeadamente por:

- Butala and Sluga (2002) – implementam um sistema multiagente em Eclipse, linguagem baseada em PROLOG.
- Gjerdrum, Shah and Papageorgiou (2001) – JATLite (<http://java.stanford.edu/>);
- Kádar, Monostori and Szelke (1998) – Pacote Simple++ com linguagem Simtalk.
- Maturana, Shen and Norrie (1999) – Sistema multiagente desenvolvido com VisualWorks 2.5 em linguagem Smalltalk e protocolo KQML para comunicação entre agentes.
- Ming *et al.* (1998) – CORBA based intelligent agents;
- Miyashita (1998) – Sistema multiagente desenvolvido em linguagem JAVA e protocolo KQML para comunicação entre agentes.
- Shen, Maturana and Norrie (2000b) – Sistema multiagente desenvolvido com VisualWorks 2.5 em linguagem Smalltalk, interfaces desenvolvidas 3D studio e Visual C++ e protocolo KQML para comunicação entre agentes.
- Sousa and Ramos (1998) – sistema multiagente sem referência à tecnologia utilizada.
- Sun, Zhang and Nee (2001) – JATLite (<http://java.stanford.edu/>);

Entre os autores que recorreram a tecnologia orientada aos objectos temos:

- Agrawal and Graves (1999) – CORBA e STEP.
- Barata *et al.* (2001) – C++ Builder, DCOM.
- Camarinha-Matos *et al.* (1998) – Sistema de Gestão de Informação PEER.
- Fujii, Kaihara and Morita (2000) - Linguagem C, Microsoft Access 97; Smpl, SLAM II, Fortran.
- Ming *et al.* (1998) – CORBA based intelligent agents.
- Trentesaux, D., Tahon and Ladet (1998) – Linguagem C++.
- Zhao, J., Cheung and Young (1999) - Object Store (OODBMS), Visual C++.

## **3. SISTEMA DISTRIBUÍDO DE PRODUÇÃO**

Não existe uma definição de Sistema Distribuído de Produção unanimemente aceite pela comunidade dos sistemas de produção. As definições utilizadas, em grande parte dos trabalhos, apoiam-se, explicita ou implicitamente, em conceitos relacionados com sistemas de produção Biónicos, Holónicos e Fractais, e na agilidade de utilização dos meios de produção. A definição aqui apresentada sobre Sistemas Distribuídos de Produção baseia-se em conceitos de Sistemas, Sistemas de Produção e Sistemas Distribuídos de Informação. Considera-se ainda que estes sistemas se apoiam em diferentes tipos de recursos, logo, são sistemas

heterogéneos. A referência a estes conceitos constitui a base de construção da definição do Sistema Distribuído de Produção.

### 3.1 Sistema

Tomando em consideração as definições de sistema, apresentadas na Tabela 2, é necessário, para os sistemas distribuídos de produção, definir os seus objectivos e componentes. Os objectivos do sistema distribuído de produção são os objectivos de um sistema de produção capaz de responder a mercados cada vez mais exigentes, onde se podem identificar algumas das seguintes características: menor tempo de vida dos produtos; menores prazos de entrega; maior participação do cliente em decisões de projecto do produto. De uma forma genérica, os componentes de um sistema distribuído de produção são elementos de produção interligados capazes de processar materiais de entrada e entregar produtos.

Tabela 2: Definições de Sistema

Autor	Definição de Sistema
Athey (1982)	<i>Conjunto de componentes trabalhando em conjunto para o objectivo da totalidade do sistema.</i>
Wu, B. (1994)	<i>Colecção de componentes inter-relacionados de uma forma organizada, que trabalham em conjunto para atingir um fim lógico.</i>

Esta definição será útil na construção de um modelo representativo e gerador de sistemas distribuídos de produção. Um modelo de um sistema representa uma parte desse sistema, relevante para os objectivos definidos. Tomando em consideração as definições de sistema, apresentadas na Tabela 2 e a ilustração de um modelo da Figura 6, é necessário, para um modelo de sistema distribuído de produção, que para além da definição do sistema, se defina o ambiente e o relacionamento com o ambiente. Resumindo, é necessário definir objectivos, fronteiras, entradas e saídas, subsistemas e respectivo inter-relacionamento.



Figura 6: Representação esquemática de Sistema

A análise dos subsistemas (Figura 7) de um sistema pode ser realizada, segundo Gale and Eldred (1996), através de três abordagens fundamentais: decomposição funcional – quais os subsistemas que desempenham determinadas funções; decomposição estrutural - quais os elementos constituintes do sistema; decomposição espacial – tomando a distância física como critério. Nesta análise, é no entanto necessário considerar a característica da *totalidade*<sup>21</sup> de qualquer sistema. É esta característica que conduz o desempenho dos vários subsistemas em direcção ao seu objectivo, acrescentando propriedades que não existiriam doutra forma no sistema, isto é, propriedades emergentes.

<sup>21</sup> “Wholeness”.

O ambiente do sistema fica definido pela delimitação das fronteiras do mesmo. Estas fronteiras podem ser definidas, ainda segundo Gale and Eldred (1996), com base em diversos critérios: pelos objectivos que se pretendem atingir, pelos limites físicos, pela necessidade de otimizar resultados ou em contrapartida de suboptimizar, e / ou pela finalidade da investigação em curso.

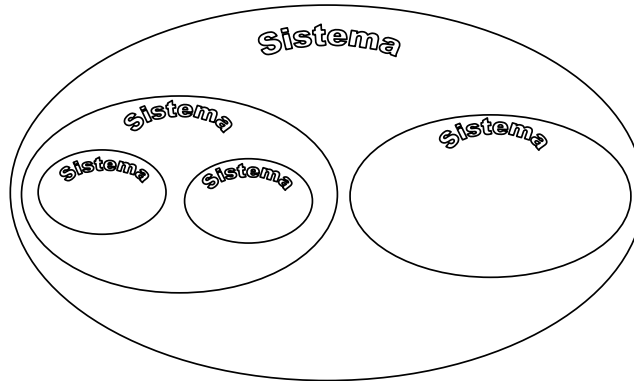


Figura 7: Representação de Subsistemas.

### 3.2 Sistema Distribuído Heterogéneo de Informação

Constata-se que os Sistemas Distribuídos Heterogéneos de Informação, com base nas definições apresentadas na Tabela 3, são sistemas compostos por vários elementos de processamento de informação, possivelmente diferentes, a comunicarem por meios diferentes para atingir um determinado objectivo. As duas primeiras definições atribuem este significado às seguintes denominações: Sistema de Computadores Distribuídos Heterogéneos; Rede de Computadores.

Coulouris, Dollimore and Kindberg (2001) baseiam a definição de sistemas distribuídos em elementos de processamento a funcionarem em redes de computadores, assumindo assim, a existência de meios e protocolos de comunicação que interligam computadores em rede. O facto de os elementos de processamento comunicarem através de mensagens, leva à caracterização do sistema de acordo com as seguintes propriedades:

- Concorrência – cada elemento executa as suas tarefas e concorre pela utilização de recursos partilhados que são outros elementos do sistema.
- Sem relógio global – como a comunicação só se realiza por mensagens não é possível assegurar a existência de uma referência temporal precisa para todos os elementos.
- Falhas independentes – cada componente do sistema pode falhar de uma forma independente, deixando os outros a funcionar.

Tabela 3: Definições de Sistemas Distribuídos Heterogéneos de Informação

Autor	Definição de Sistema Distribuído Heterogéneo de Informação
Shuey, Spooner and Frieder (1997)	<i>Sistema de Computadores Distribuídos Heterogéneos é composto por diferentes computadores executando processos diferentes de diversas aplicações, que comunicam através de ambientes de transmissão de informação distintos.</i>
Wu, J. (1999)	<i>Rede de Computadores é um sistema composto por elementos de processamento de informação e sistemas de comunicação distintos.</i>
Coulouris, Dollimore and Kindberg (2001)	<i>Sistema Distribuído é um sistema em que diferentes componentes de hardware ou de software, localizados em redes de computadores, comunicam e coordenam as suas acções através do envio de mensagens.</i>

O sistema distribuído de informação não heterogéneo é composto por elementos de processamento a comunicarem entre si. Neste caso, tanto os elementos de processamento como os meios de comunicação podem ser iguais, formando sistemas de processamento paralelo. No âmbito dos sistemas de informação, Wu, J. (1999) define o Sistema Distribuído de Computadores como uma composição de elementos de processamento que funcionam de uma forma cooperante em processos da mesma aplicação.

### 3.3 Sistema de Produção

As definições de Sistema de Produção, apresentadas na Tabela 4, caracterizam os objectivos e componentes destes sistemas. O objectivo do sistema de produção pode resumir-se ao objectivo de obtenção de produtos. Os produtos são caracterizados como sendo alguma “coisa”, algum objecto tangível ou não, com maior utilidade para o sistema do que os materiais de entrada. São objectos que podem ser vendidos ou utilizados por outros sistemas, ou subsistemas. Os componentes do Sistema de Produção são elementos ou recursos de produção, capazes de executar os processos de produção que adicionam utilidade aos objectos de entrada do sistema.

Tabela 4: Definições de Sistema de Produção

Autor	Definição de Sistema de Produção
Wu, B. (1994)	<i>Sistema que utiliza um conjunto de processos de produção para converter matérias primas em formas mais úteis e eventualmente em produtos finais.</i>
Hitomi (1996)	<i>Produção é o processo de produzir bens económicos, bens tangíveis e serviços intangíveis, a partir de recursos de produção, criando, assim, utilidade por adição de valor.</i>

Um sistema de produção, conforme ilustração apresentada na Figura 8, é composto por alguns meios de produção, que sob o efeito de trabalho produtivo vão processar os objectos de produção, de forma a entregar algum(ns) produto(s). Este sistema produtivo pode associar-se com outros sistemas produtivos, formando um sistema mais alargado com a habilidade de criar novos produtos. Esta associação tem que se apoiar em meios de comunicação que se podem basear no deslocamento de pessoas, ou num misto entre pessoas e meios automáticos como tapetes rolantes e AGVs, ou ainda em redes de sistemas de informação.

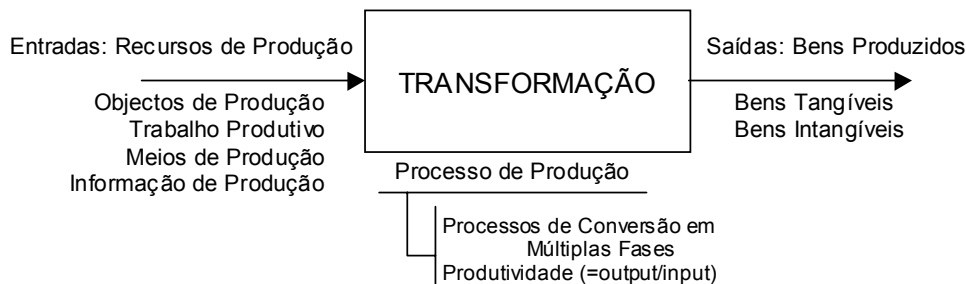


Figura 8: Modelo do Sistema de Produção segundo Hitomi (1996).

Podem adoptar-se perspectivas do sistema de produção dependentes dos componentes e dos objectos do sistema. Estas podem ser analisadas do ponto de vista da movimentação física, e neste caso os objectos podem movimentar-se entre recursos transformadores, ou não se movimentar esperando que o recurso se desloque para o processar. Isto significa que qualquer um destes elementos, os objectos ou os recursos, ou ambos, se podem deslocar fisicamente num determinado sistema de produção.

Adoptando-se uma perspectiva estática da estrutura dos componentes do sistema de produção, o ponto de vista do elemento de processamento que executa um processo transformador, é o de produção local. Seja qual for o local de transformação, ela dá-se nesse local utilizando os recursos aí existentes, para transformar os materiais à sua entrada em materiais transformados à sua saída. Para gerir um elemento de processamento, temos que conhecer, fundamentalmente, as suas aptidões, a sua capacidade padrão e a capacidade disponível. A aptidão do elemento de processamento ou recurso de produção permite determinar a possibilidade de utilização do recurso para executar determinado processo de transformação. A capacidade padrão é utilizada para determinar tempos de execução. A capacidade disponível, dependente das encomendas aceites, permite atribuir a execução de trabalhos em períodos de tempo livres.

Adoptando-se a mesma perspectiva estática da estrutura dos componentes do sistema de produção, o ponto de vista dos objectos de produção (matérias primas, produtos semiacabados, informação), é o de produção distribuída. As matérias-primas são produzidas ou recolhidas num determinado local, iniciando aí uma viagem que, de transformação em transformação, as vai levar ao seu estado de comercialização, isto é, de produto final.

A análise detalhada das definições de Sistemas em geral, e de Sistemas de Produção em particular, mostra que todos os sistemas são de alguma forma distribuídos. São compostos por diferentes elementos que comunicam entre si para atingir um determinado objectivo. Se comunicam, têm algum tipo (grau) de distribuição entre os elementos que é necessário ligar.

Por um lado a crescente consciencialização e exigência dos clientes com a conseqüente transformação do mercado, e por outro a evolução tecnológica, conduz à exploração de novos paradigmas de Gestão da Produção.

### **3.4 Definição de Sistema Distribuído de Produção**

As definições de Sistema Distribuído de Produção, apresentadas na Tabela 5, caracterizam os objectivos e componentes destes sistemas segundo diversos autores. Estas definições caracterizam estes sistemas em função da relação entre Sistema de Produção e distribuição. Autores como Leitão and Restivo (2000), Shen, Maturana and Norrie (2000b), Barata *et al.* (2001), Sluga and Butala (2001), Tharumarajah (2001), baseiam os seus trabalhos, implícita ou explicitamente, em conceitos de sistemas distribuídos de produção similares ao da definição de Gunasekaran (1998), inspirados nos paradigmas de sistemas de produção Biónicos, Holónicos e Fractais. Estes conceitos exploram fortemente características de autonomia dos elementos de produção. A capacidade de configuração e reconfiguração autónoma da relação estrutural entre elementos do sistema está também presente nestes trabalhos, tal como nos sistemas de produção ágeis (Sanchez and Nagi (2001); Gunasekaran, Mcgaughey and Wolstencroft (2001)), ou de empresas virtuais (Camarinha-Matos, Afsarmanesh and Osório (2001); Carvalho, Putnik and Cunha (2003)).

A segunda definição, baseia-se numa noção de desempenho do sistema de produção, que se pode considerar distribuído quando a distância não influencia o sistema. Esta definição limita o sistema de produção distribuído a sistemas que exploram a transferência de objectos com custo negligenciável. Isto pode acontecer quando se explora a transferência de objectos não tangíveis, através de redes de sistemas de informação com custos de comunicação muito inferiores aos custos de transformação física. Pressupõe ainda

a exploração da utilização de sistemas abertos, que reduzem os custos de integração, fundamentalmente, aos custos de implementação do sistema, respeitando as normas relevantes.

Tabela 5: Definições de Sistema Distribuído de Produção

Autor	Definição de Sistema Distribuído de Produção
Gunasekaran (1998)	<i>O sistema de produção distribuído fisicamente é composto por agentes heterogéneos e estruturas de controlo que operam autonomamente durante o planeamento e execução das tarefas de produção.</i>
Putnik et al. (1998)	<i>Sistema distribuído de produção é um sistema de manufactura cuja funcionalidade e desempenho é independente da distância física entre elementos do sistema.</i>

No caso mais geral de um sistema de produção, os elementos de processamento são distintos e trabalham, em diversos ambientes, simultaneamente em diferentes processos de diferentes produtos, comunicando entre si de diferentes formas. Se neste sistema houver preocupação com o desempenho, então a optimização da utilização dos recursos, ao nível do projecto e gestão do sistema, será fundamental.

O termo Distribuído, associado ao Sistema da Produção, emerge da necessidade de encontrar novas formas de organização e gestão, que resolvam problemas associados à crescente necessidade de adaptabilidade às alterações. A seguinte definição resulta de uma parte do trabalho inicial desta dissertação, publicada numa primeira versão em Lima, Silva and Martins (1999b) e numa versão melhorada em Lima, Silva and Martins (1999a). Este trabalho sofreu um desenvolvimento adicional apresentado neste capítulo, nomeadamente, no aprofundamento de conceitos e no enquadramento de outras definições de Sistemas Distribuídos de Produção, de Sistemas e Sistemas Distribuídos de Informação.

Sistema Distribuído de Produção é composto por uma rede de elementos autónomos de processamento, com a possibilidade de reconfiguração dinâmica do sistema produtivo.

Esta definição coloca de parte os sistemas de produção convencionais, por não permitirem uma reconfiguração dinâmica. Por outro lado, requer formas de gestão que permitam a reconfiguração do sistema, integrando no modelo várias partes do sistema, nomeadamente elementos de processamento, entradas e saídas do sistema e a comunicação entre elementos de processamento. Existe ainda um conjunto de desafios que se levantam na modelação, projecto e operação destes sistemas:

- Heterogeneidade – todos os elementos de produção do sistema podem ser diferentes, podendo estar interligados das mais variadas formas. É necessário modelar estas diversas formas e a sua integração.
- Abertura – os sistemas devem ser abertos, isto é, deve conhecer-se a sua interface, de modo que exista a possibilidade de qualquer outra entidade associar o seu próprio elemento ao sistema. Esta característica permite aumentar a utilidade do sistema e reduzir o tempo de integração.
- Segurança – protecção de informação, identificação dos elementos de produção e confiança na informação transmitida, são alguns dos principais problemas de segurança destes sistemas.
- Gestão de avarias – é necessário estar ciente da possibilidade de avaria independente de qualquer um dos elementos, projectando o sistema para lidar com os diferentes tipos de avarias. Será necessário ultrapassar a avaria, por detecção, repetição do trabalho ou redundância, mas também por estabelecimento de mecanismos que aumentem a confiança na utilização do sistema. Por



exemplo, assegurando que todos os elementos têm um baixo índice de avarias e pela existência de contrapartidas para os elementos prejudicados pela avaria.

- Concorrência – todos os recursos de produção têm que estar preparados para lidar com a concorrência de requisição dos seus trabalhos, assegurando respostas e posterior processamento fiáveis.
- Transparência – a utilização dos recursos deve ser transparente para os utilizadores, para os quais, do ponto de vista operacional, não deve interessar quem executa nem a forma de execução do trabalho.

Nem todos os aspectos referidos são abordados nesta dissertação, mas a heterogeneidade, abertura do sistema e transparência, são motivações do desenvolvimento de um modelo baseado nesta definição.

### **3.5 Características de Sistemas Distribuídos de Produção**

A definição de Sistemas Distribuídos de Produção levanta a necessidade de esclarecer alguns dos termos utilizados e de apresentar as características que distinguem estes sistemas dos sistemas de produção convencionais: distribuição, autonomia e reconfiguração dinâmica.

#### **3.5.1. Distribuição**

A distribuição, embora seja inerente a qualquer sistema, é em muitos casos negligenciada para efeitos de simplificação de análise do sistema. Nestes sistemas, a distribuição do sistema é fulcral, logo, é necessário avaliar o seu efeito, aplicando modelos que incluam a possibilidade de representação da ligação entre os subsistemas. Num sistema distribuído de produção o modelo deve permitir representar os subsistemas de transferência de materiais.

A distribuição física, ou lógica, indica algum tipo de distanciamento entre os elementos do sistema e a necessidade de comunicação entre os mesmos. Daqui se depreende que a fronteira de definição dos subsistemas vai definir o *grau de distribuição*. Os sistemas são distribuídos por natureza, distinguindo-se pelo grau de distribuição, que depende das fronteiras definidas para a modelação do sistema, que pode ser a oficina de fabrico, a fábrica, a empresa, o grupo de empresas, o País ou o Mundo. Os elementos distribuídos podem ser máquinas numa determinada instalação fabril, diferentes empresas em vários países, ou máquinas de diferentes empresas em locais completamente distintos. O grau de distribuição pode resultar da análise de diversos factores, como distância, operações de comunicação, ou operações de integração entre elementos do sistema.

A associação de elementos de processamento num sistema distribuído, implica a exploração de hardware e software que permita a comunicação entre os mesmos. Por outro lado, é necessário que os diferentes subsistemas se compreendam, isto é, o significado e contexto da informação tem que ser compreendido por todos os elementos. O desenvolvimento destes sistemas requer a existência de um modelo conceptual que seja compreendido e aceite por todos os elementos, de forma que haja uma utilidade real na criação de Sistemas Distribuídos de Produção globais.

### **3.5.2. Autonomia**

Autonomia é um conceito geral cuja definição, segundo o Dicionário de Português da Porto Editora (<http://www.infopedia.pt/>), diz respeito à *possibilidade que uma entidade tem de estabelecer as suas próprias normas*. Este conceito de definição e enquadramento complexo, é alvo de estudo da filosofia, do direito e da política. Segundo Kant, será *a liberdade da vontade racional que só obedece à lei por ela mesma legislada*. Estas definições gerais podem contribuir para a clarificação deste conceito, no âmbito dos sistemas distribuídos de produção.

Um elemento de um sistema distribuído de produção está incluído num sistema de produção, nalgum sistema empresarial, relacionando-se com outros sistemas de produção da mesma empresa ou de outras, pelo que a sua autonomia nunca será total. Este elemento tem que se reger pelos objectivos do sistema em que está integrado. Haverá, desta forma uma noção de integração dos seus objectivos nos do sistema, existindo sempre, no caso dos sistemas empresariais, objectivos impostos, directa ou indirectamente pela entidade que coordena ou desencadeia a formação do sistema distribuído. Além disso, terá que se reger por leis de relacionamento entre entidades económicas do sistema legislativo em que se enquadra.

Guiando-nos pelos conceitos aqui apresentados, a autonomia é sempre restringida pelos objectivos do sistema. Pode então, relativamente a esta característica referir-se o conceito de *grau de autonomia*. Este é uma medida da capacidade do elemento do sistema estabelecer as suas próprias normas de funcionamento. A distribuição dos elementos permite que exista distribuição do controlo, que está, por outro lado, intrinsecamente ligada à autonomia dos elementos, ou antes, ao grau de autonomia. Quanto maior for a capacidade de um elemento do sistema tomar decisões e controlar o efeito dessas decisões, maior será a autonomia desse elemento.

No âmbito dos Sistemas Distribuídos de Produção, todos os subsistemas têm que atingir os seus objectivos, de forma a contribuir para os objectivos dos sistemas em que se inserem. No entanto, e concordando com a noção apresentada por Warnecke (1993), para a definição de objectivos e sub-objectivos do sistema de fractais da Fábrica Fractal, os próprios elementos do sistema podem influenciar os objectivos gerais através dos seus resultados e das suas intenções. Esta noção pode ser clarificada se se considerar a possibilidade de um destes elementos ser uma pessoa, com objectivos próprios que influenciem os objectivos gerais do sistema.

No Sistema Distribuído de Produção preconiza-se a capacidade de execução autónoma de processos de transformação do sistema de produção, sem interferência relativamente à forma de execução por parte de qualquer outro elemento ou sistema. Este tipo de autonomia deixa alguns caminhos livres, como seja a possível participação de um elemento de processamento em diferentes sistemas, desde que contribua para os objectivos de todos eles. A forma como esta autonomia será implementada é, actualmente, fruto de investigação no âmbito dos sistemas de produção, sistemas de informação e inteligência artificial.

### **3.5.3. Reconfiguração**

Para que o sistema esteja apto a responder às alterações do ambiente, é necessário que exista a possibilidade da sua organização se alterar de uma forma dinâmica. Os Sistemas Distribuídos de Produção devem ter a capacidade de se reconfigurarem, alterando a associação entre elementos do sistema, como

resposta a alterações do ambiente ou dos objectivos. A dinâmica da reconfiguração é um conceito definido, neste contexto, como a capacidade de alteração, o que, no entanto, não será suficiente se não estiver associada ao horizonte temporal desta alteração. É preciso notar que todos os sistemas se podem adaptar e dessa forma reconfigurar. Neste âmbito, no caso mais geral (ideal), esta reacção deverá realizar-se instantaneamente (zero segundos). O conceito de reconfiguração do sistema está intrinsecamente relacionado com a noção de Produção Ágil (Gunasekaran, Mcgaughey and Wolstencroft (2001); Vernadat (1999); Rabelo and Camarinha-Matos (1996)). Esta é a característica tecnologicamente mais exigente da definição de um Sistema Distribuído de Produção.

O *grau de reconfiguração* do sistema pode ser uma forma de medir esta característica dos Sistemas Distribuídos de Produção, e depende da capacidade da organização do sistema se alterar, relacionada com as razões de alteração e com o tempo dessa alteração.

#### **4. CONSIDERAÇÕES FINAIS**

O estudo de paradigmas de organização da produção, como os dos Sistemas de Produção Biónicos, Holónicos e Fractais, em conjunto com a análise de inúmeros trabalhos desenvolvidos no âmbito da distribuição de sistemas de produção, permitiu construir o enquadramento para a definição de Sistema Distribuído de Produção.

Apresenta-se, assim, uma definição de Sistema Distribuído de Produção, apoiada nos conceitos de Sistemas, Sistemas Distribuídos de Informação e Sistemas de Produção, no contexto dos novos paradigmas de organização da produção. Nesta definição, os conceitos de autonomia, distribuição e reconfiguração dinâmica são características fundamentais do sistema. No entanto, é necessário considerar a essência não absoluta destes conceitos, que são definidos de uma forma relativa.

A definição de Sistema Distribuído de Produção, e a das suas características, não nos indica a forma do sistema nem de criação de um modelo do sistema que permita efectuar a sua representação, projecto e coordenação. É necessário criar um modelo de enquadramento que permita este desenvolvimento. Num mercado em constante e rápida mutação, em que os clientes podem ter um papel activo nas fases do ciclo de vida do produto, é necessário definir novos modelos que permitam uma maior flexibilidade (agilidade) e adaptação dinâmica do sistema produtivo às alterações existentes. Acredita-se que uma das formas de construir estes sistemas se baseia em modelos de unidades de produção autónomas similares que prestam serviços a outras unidades de produção.



### **III - MODELO DO SISTEMA DISTRIBUÍDO DE PRODUÇÃO**

A definição de Sistema Distribuído de Produção é baseada na descrição de um modelo genérico, seguido da especificação formal do mesmo. O modelo apresentado, cumprindo com os objectivos do trabalho, é um modelo de um sistema de produção orientado à satisfação de uma encomenda de cada vez. É um sistema de satisfação não apenas de uma encomenda, mas de um único produto de uma encomenda de cada vez. A representação inequívoca de um sistema deve basear-se em linguagens formais. Neste capítulo apresenta-se uma representação formal do modelo do Sistema Distribuído de Produção baseada em Redes de Petri.



Para efectuar o estudo de um sistema de produção nas suas vertentes de projecto, operação e controlo é necessário utilizar a definição do sistema, um modelo e uma representação desse modelo. A definição do Sistema Distribuído de Produção apresentada no capítulo anterior serve de base à concepção de um modelo para projecto do sistema. Este modelo fornece mecanismos de concepção e representação de um sistema distribuído de produção baseado na encomenda de um produto. A representação formal deste modelo permite defini-lo de uma forma precisa e desenvolver métodos de optimização para selecção de recursos para os vários elementos de produção do sistema.

## **1. MODELO DO SISTEMA DISTRIBUÍDO DE PRODUÇÃO**

Um modelo de um Sistema Distribuído de Produção, assumindo a autonomia dos recursos de produção, deve dar resposta ao problema de configuração e reconfiguração do sistema a partir do conhecimento dos recursos distribuídos disponíveis. Sendo o conceito de Produção Simultânea um conceito chave a aplicar ao Sistema Distribuído de Produção, o modelo desenvolvido adopta algumas das suas premissas. A Produção Simultânea assume o escalonamento completo de uma ordem de produção de cada vez utilizando todos os recursos disponíveis. Neste modelo assume-se que a encomenda de um produto, vista como uma ordem de produção de um produto, é o factor que desencadeia o processo de configuração do sistema. Por configuração do sistema entende-se a representação e selecção de meios de produção distribuídos que permitam executar os processos de produção necessários para satisfazer a encomenda. O processo de configuração do sistema deve ser dinâmico, dependente do estado dos meios de produção, podendo ser aplicado na reconfiguração do sistema motivada por alteração de objectivos ou como forma de dar resposta a falhas do sistema.

### **1.1 Enquadramento do Modelo**

O projecto de um sistema de produção e a sua operação são processos iterativos e contínuos, de tal forma interligados, que devem ser considerados e modelados conjuntamente. Desta forma, os processos e entidades de modelação devem responder a essa necessidade de interactividade e interligação. O modelo desenvolvido e utilizado neste trabalho, considera estes requisitos, ligando o processo de projecto do sistema a cada encomenda de um produto. Este modelo é baseado em Sistemas de Produção Orientados ao Produto, mais especificamente, orientados a um produto de uma encomenda. A teoria Orientada aos Objectos serve de base de modelação porque apresenta, por um lado, características úteis para a modelação de sistemas com as características descritas e por outro, garantias de uma modelação completa, com precisão e com simplicidade.

#### **1.1.1. Sistemas de Produção Orientados ao Produto**

Segundo Silva, S. C. and Alves (1997) e Alves, Lima and Silva (2003), um Sistema de Produção Orientado ao Produto (SPOP) é composto por um conjunto de recursos ou células de produção que simultaneamente produzem um produto ou um leque de produtos similares, incluindo todas as fases de fabricação e de montagem. Neste âmbito, um produto pode ser composto por vários componentes, correspondendo a uma estrutura de produto com vários níveis. A estrutura de um SPOP pode corresponder a uma linha de produção ou de montagem, ou em outros casos corresponder a um conjunto coordenado de

recursos interligados. Esta abordagem à modelação de sistemas de produção inclui, em grande parte, o conceito apresentado por Putnik and Silva (1995) da Empresa Virtual do tipo OPIM – “One Product Integrated Manufacturing”.

A concentração do sistema empresarial numa determinada direcção foi uma noção introduzida por Skinner (1974), com a Fábrica Focada<sup>22</sup>. Com esta abordagem pretende-se concentrar a fábrica nos processos fundamentais para o sucesso do negócio. Este conceito nasceu da necessidade de responder ao problema da proliferação do número de produtos da empresa, com ciclos de vida mais curtos, produzidos em volume inferior e com mais requisitos de alteração por parte do consumidor. Assume-se, neste conceito, que a fábrica se torna mais competitiva se se dedicar ou focar na execução de determinada tarefa, processo ou produto, aumentando desta forma as competências da empresa na produção. Este aumento de competência advém do aumento da repetição, experiência e homogeneidade de tarefas e da redução da diversidade das mesmas.

O SPOP explora a noção de concentração em produtos similares, tentando explorar as vantagens da focagem do sistema de produção, nomeadamente, em processos de produção similares e na redução do tipo de produtos e materiais a gerir.

A concorrência de nível global impele, segundo Tu (1997), as empresas para a produção rápida de produtos personalizados ou mesmo totalmente individualizados, recorrendo em alguns casos, a sistemas de produção criados para cada produto e utilizados uma única vez, isto é sistemas OKP<sup>23</sup>. Exemplos de sistemas OKP são os sistemas de construção de navios ou de construção de estruturas em aço. Uma forma de resposta aos requisitos referidos por Tu (1997) reside na adopção de estruturas organizacionais virtuais e ágeis. Os sistemas OKP de produção de produtos individualizados têm, em geral, dificuldades adicionais de adaptação a estas estruturas organizacionais por terem investimentos elevados em maquinaria pesada, estruturas físicas e força de trabalho considerável. As características de uma Empresa Virtual OKP como um novo paradigma de produção são: elevada personalização; “uma” abordagem com sucesso ao produto; utilização racional das tecnologias e recursos; planeamento e controlo de produção adaptável; influência contínua do cliente sobre o processo de produção; desenvolvimento e produção do produto baseado na evolução de protótipos; controlo distribuído e autonomia inter-organizacional; estrutura de Empresa Virtual e produção global.

Al-Mubarak, Khumawala and Canel (2003) definem células focadas de produção como uma implantação de recursos em função de cada produto final, formando assim células de máquinas para fabricar e montar cada um desses produtos. A produção celular focada (FCM<sup>24</sup>) distingue-se da produção celular, segundo estes autores, por não tentar tirar vantagem da semelhança de processos no agrupamento de recursos de produção. No estudo desenvolvido conclui-se que a aplicação da FCM reduz a média dos instantes de conclusão dos produtos finais e reduz também a média do inventário de trabalho em processamento (WIP<sup>25</sup>). A produção celular reduz os tempos de preparação, pelo que produz melhores resultados para lotes de reduzidas dimensões ou quando o tempo de preparação é elevado.

---

<sup>22</sup> “Focused Factory”.

<sup>23</sup> “One-of-a-Kind Production”.

<sup>24</sup> “Focused Cellular Manufacturing”.

<sup>25</sup> “Work In Process”.



### **1.1.2. Modelos de Sistemas de Produção Orientados aos Objectos**

É possível encontrar inúmeros trabalhos de investigação sobre modelação de sistemas de produção baseados no conceito de Orientação aos Objectos. Esta modelação está, principalmente, relacionada com a modelação de sistemas de informação a utilizar no apoio às funções de produção. No entanto, algumas das características desta teoria podem servir de enquadramento à modelação de sistemas de produção, pelo que se apresenta uma pequena amostra de alguns dos trabalhos de modelação de sistemas de informação orientados aos objectos.

Yoo (1997) apresenta uma base de dados de apoio a um sistema CIM<sup>26</sup>. Trata-se duma base de dados CAD baseada em classes de objectos com a descrição estrutural e tecnológica das peças produzidas.

Hoche, Broomhead and Grieve (1996) descrevem um modelo genérico para desenvolvimento de software orientado aos objectos, para controlo de células de produção. Este modelo baseia-se numa classe de objecto para controlo da célula e em classes de objectos que representam elementos reais das células, como operadores, robots e máquinas. O objecto de controlo tem a função de implementar o método de controlo e coordenar a comunicação entre objectos. Um trabalho de implementação de um sistema de controlo baseado neste modelo é descrito em Hoche *et al.* (1996).

Aguiar, Murgatroyd and Edwards (1996) apresentam uma ferramenta computacional de apoio à Engenharia de Software (CASE) baseada em objectos para modelação de recursos de produção, denominada SEW-OSA. Esta ferramenta permite modelar os recursos como objectos, acompanhando várias fases do ciclo de vida do software, nomeadamente, ao nível da análise de requisitos, projecto e implementação.

Adiga (1997) apresenta uma metodologia para a construção de software orientado aos objectos para sistemas de produção. As linguagens de modelação mais utilizadas nesta metodologia são os diagramas de Entidade – Relacionamento (ER) e Diagramas de Fluxo de Mensagens (MFD). De acordo com esta metodologia o sistema de informação para a produção deve ter três módulos fundamentais, módulo de estado do sistema de produção, módulo de decisão e módulo de interfaces.

O trabalho de modelação de empresas de Mertins, Sussenguth and Jochem (1992) baseia-se no recurso a classes de objectos fundamentais: produto, encomenda e recurso. O modelo lógico estático da informação baseia-se nestes objectos, em objectos especializados descendentes e no relacionamento entre eles. O modelo funcional baseia-se num modelo de actividade genérico e num conjunto de estruturas de encadeamento das funções. Nesta modelação preconiza-se ainda a utilização de vistas da informação adequadas a cada utilizador.

#### ***i. Vantagens da modelação Orientada aos Objectos***

A teoria OO tem sido fundamentalmente utilizada na concepção e desenvolvimento de software, tendo sido largamente descritas as suas vantagens por autores da comunidade de OO.

- Facilidade de Modelação

Na teoria OO os objectos são as principais entidades do sistema, sendo desenvolvido o modelo do sistema a partir dos objectos existentes (Adiga (1997); Meyer (1988)), fisicamente ou não. Esta

---

<sup>26</sup> "Computer Integrated Manufacturing".

característica facilita o desenvolvimento de modelos de sistemas de produção (Adiga (1997) ). Nos objectos é encapsulada a informação e as respectivas funções de processamento.

- Reutilização

A reutilização de software pode ser efectuada, segundo Yourdon (1994), em diversos níveis, nomeadamente: especificação, projecto, código fonte, informação, e código compilado. A reutilização ao nível da especificação e do projecto é efectuada fundamentalmente pelo mecanismo de herança. Por este mecanismo é possível conceber novas classes a partir de classes já existentes, especializando as suas funcionalidade em novas classes que herdaram as características das classes ascendentes. Outro mecanismo corresponde à utilização de outros objectos como parte da classe em construção. Adiga (1997) referiu exemplos de algumas livrarias de classes aplicadas a sistemas de produção, que assim exploram a possibilidade de reutilização de classes de objectos.

- Teste e Evolução do Sistema

Os sistemas OO podem crescer a partir de pequenos elementos até elementos mais complexos completamente funcionais, permitindo a modelação por extensão de elementos concebidos anteriormente (Adiga (1997); Carmichael (1994)). A abordagem OO é adequada a um desenvolvimento evolucionário do sistema, alargando as capacidades do sistema, com alterações mínimas até que todas as funções estejam implementadas. Além disso, um modelo OO do sistema pode ser usado para controlar um sistema físico associado, ou para simular o seu comportamento aumentando o conhecimento do sistema e agindo sobre o mesmo em função dessa simulação.

- Ocultação da Informação

A definição da informação visível para o exterior do objecto é conhecida como princípio de ocultação da informação<sup>27</sup> (Meyer (1988)). Em cada objecto define-se a informação e as funções que devem ou não ser visíveis para o exterior, reforçando a independência dos objectos relativamente à forma como implementam as funcionalidades que oferecem. A interface com o utilizador do objecto não deve ser alterada, embora a implementação possa ser. Este princípio também permite criar uma camada de segurança, protegendo dados que não devem ser visíveis para o exterior.

## **1.2 Modelação do Sistema Distribuído de Produção**

O modelo do Sistema Distribuído de Produção deve representar elementos de produção autónomos interligados entre si com a capacidade de reconfiguração da estrutura do sistema. Isto significa que o modelo representa elementos com a capacidade autónoma de realizar o trabalho a que se propõe, comunicando com outros elementos de forma a com eles se estruturar para atingir o objectivo do sistema. Nesta secção descreve-se, de uma forma genérica, o modelo de Sistemas Distribuídos de Produção desenvolvido neste trabalho, que será designado por MSDP.

Como o Sistema Distribuído de Produção pode ser composto por recursos de produção globalmente distribuídos, pretende-se que o modelo utilizado permita seleccionar recursos que assegurem a satisfação dos requisitos de uma encomenda. Em qualquer local, um utilizador, tendo acesso a uma rede de comunicação,

---

<sup>27</sup> "Information Hiding".

deve poder colocar uma encomenda de um produto, e obter uma resposta de entrega desse produto por um determinado recurso. Para que um recurso possa entregar um determinado produto tem que ter a aptidão e a capacidade requeridos, isto é, tem que executar o processo de produção correcto.

O objectivo do sistema é o de satisfazer as encomendas dos clientes com aplicação dos princípios de Produção Simultânea. A Produção Simultânea apoia-se na premissa de satisfação de uma ordem de produção de cada vez. Por satisfação de uma ordem de produção de cada vez entende-se o seu escalonamento completo, isto é, selecção e atribuição de recursos às tarefas de produção requisitadas pela ordem de produção.

Assume-se que cada Sistema Distribuído de Produção é criado para a satisfação dos requisitos de produção de uma encomenda de um único produto.

Dessa forma, o modelo representa um Sistema de Produção Orientado ao Produto, mais especificamente, um sistema de produção orientado à produção do único produto de uma única encomenda. O cliente encomenda um produto que um sistema se responsabiliza por entregar. Na visão do cliente o processo de produção executado por um recurso corresponde a uma operação de entrega de um produto. Esta visão de recurso de produção apoiada em propriedades de ocultação de informação, similares à da teoria OO, mantém a simplicidade do modelo e a autonomia dos elementos do sistema. Qualquer recurso é visto como um elemento capaz de executar um processo de produção, e em qualquer altura se pode alterar o processo físico de transformação, sem afectar a forma como é visto e utilizado no modelo do sistema. Resulta desta análise, a conclusão de que, em geral, qualquer produto sofrerá uma operação de transporte após uma transformação física. No entanto, para generalização do modelo, qualquer operação, seja ela de transformação física ou de transporte, terá a mesma representação. Utiliza-se uma noção de transformação lógica do estado do produto, isto é, um produto que sofre uma operação de transporte muda de estado passando a ser um produto transportado.

Considera-se, neste modelo, a configuração e reconfiguração dinâmica do sistema. Como este sistema é por natureza virtual, a reconfiguração baseia-se na alteração dinâmica da ligação existente entre recursos durante o período necessário para a satisfação da encomenda. Não se considera a alteração da sua implantação física. Este SPOP é concebido dinamicamente para satisfação de uma encomenda, tirando partido dos recursos disponíveis onde quer que se encontrem, podendo assim, utilizar um conjunto de meios distribuídos globalmente.

Considera-se que todos os elementos da estrutura de um produto, tais como matérias primas, componentes, semi-acabados e produtos finais, são produtos. Qualquer um destes produtos pode, em determinada fase de um sistema de produção, ser encomendado por clientes e fornecido por recursos de produção. O recurso de produção com capacidade de entregar um determinado produto de uma determinada encomenda é uma célula de produção autónoma, isto é, um subsistema de todo o sistema de produção. O conjunto de todas as células compõe um sistema de produção para um determinado produto final. A modelação do sistema de produção através da teoria OO é usualmente considerada como natural e intuitiva. Esta modelação não é uma modelação pura OO, mas inspirada nos mesmos conceitos, visualizando-se a composição do sistema baseada em dois tipos de objectos, os produtos e as células.

Neste modelo, cada produto de uma encomenda é entregue por um Sistema Distribuído de Produção, isto é, por uma célula autónoma de produção. Esta forma de modelar o sistema repete-se para todos os elementos da estrutura do produto, reflectindo uma visão fractal do sistema, com células dentro de células cada uma delas com a responsabilidade de entregar um produto da estrutura do produto final. A Figura 9 ilustra esta visão de produtos dentro de produtos e células dentro de células.

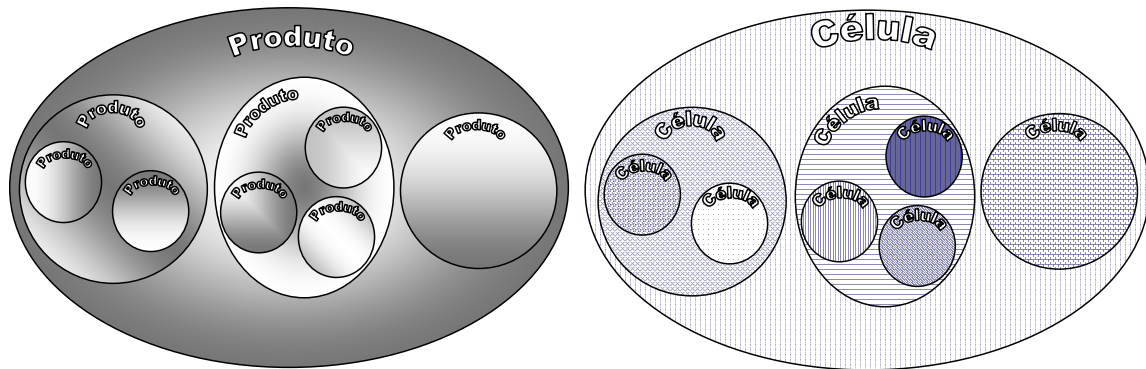


Figura 9: Representação de Produtos e Células por agregação de elementos similares.

Uma célula corresponde a um elemento de produção, que processa autonomamente o produto. Isto significa que as transformações necessárias e as funções de controlo são completamente executadas pela célula. Um conjunto de células pode cooperar para aumentar as aptidões ou capacidades requeridas por um produto, formando dessa forma uma nova célula. As células serão assim elementos de outras células, e o conjunto de todas as células de um Sistema Distribuído de Produção formam a sua configuração genérica, por não representar os recursos de produção. Este conceito de célula é similar ao conceito apresentado por Silva, S. C. and Putnik (1995), embora, neste caso, se restrinja a células orientadas ao produto.

Resumindo, o sistema distribuído de produção é visto como uma rede de células de produção que executam um serviço originando a entrega de um produto. Produtos são todos os elementos da estrutura do produto final. O produto é considerado o resultado do serviço fornecido por uma célula de produção, isto é da execução de uma transformação de estado, que resulta na entrega desse produto. Cada uma destas células pode ser constituída por outras células que entregam outros produtos necessários para a sua própria produção.

### **1.2.1. Caracterização do Produto**

O produto é caracterizado pela sua estrutura, cuja especificação condiciona, em primeira análise, a estrutura do próprio sistema. A estrutura do sistema é uma réplica organizacional da estrutura do produto, havendo tantas células autónomas de produção como transições entre elementos da estrutura do produto. Se os recursos de produção tiverem um grau de autonomia suficiente, podem propor a entrega de um produto, alterando a estrutura de referência. Embora esta hipótese exista, não é explorada na modelação utilizada.

A normalização da especificação de produtos tem sido abordada em diversos trabalhos, principalmente ao nível da descrição geométrica do produto com especificações como a norma IGES – “Initial Graphics Exchange Specification” da organização NIST – “National Institute of Standards and Technology” do governo

Norte Americano e DXF – “Drawing interchange (X) Format” introduzido pela companhia AutoDesk no programa CAD “AutoCAD”. A norma STEP da ISO, descrita em Product Data Technology Advisory (1997), é considerada a mais promissora forma de representação de produtos, que se baseia em atributos como: geometria, forma, composição ou estrutura, materiais, tolerância e propriedades. O processo de produção e a quantidade encomendada são fundamentais para identificação e selecção dos recursos. Segundo Tharumarajah, Wells and Nemes (1996), Okino no âmbito da Produção Biónica aborda este problema por identificação de máquinas, ferramentas e outros meios de auxiliares necessários para a execução do processo de produção.

Coloca-se neste modelo o problema da especificação do produto, do seu relacionamento com o processo produtivo necessário para a sua entrega e do relacionamento deste com os recursos de produção. Neste modelo pretende-se seleccionar recursos para a execução das transformações necessárias para cada produto final, pelo que se assume que o relacionamento entre produto, processo de produção e recurso tem uma solução conhecida, sendo informação de entrada do modelo.

O produto é um elemento da estrutura do produto final, que resulta da *mudança de estado* de um ou mais produtos. Esta mudança de estado resulta da execução de uma transformação por parte de uma célula autónoma de produção.

A estrutura de um produto está, neste trabalho, associada ao seu processo de produção, em que os níveis existentes representam processos executados sobre os elementos da estrutura. A mudança dos processos pode estar associada à mudança da estrutura do produto e à consequente mudança de estrutura do modelo do Sistema Distribuído de Produção. Esta associação entre estruturas do produto e do sistema permite manter a unicidade e simplicidade de representação, mas por outro lado pode levar à multiplicação de estruturas para o mesmo fim. O mesmo produto realizado por processos alternativos é representado por estruturas diferentes. Considere-se uma chapa de metal que sofre duas operações que podem ser executadas em qualquer ordem, uma resultante do processo de furar e outra de estampagem. Conforme ilustrado pela Figura 10 e de acordo com os conceitos de estrutura do produto apresentados, este produto pode ser representado por duas estruturas alternativas.

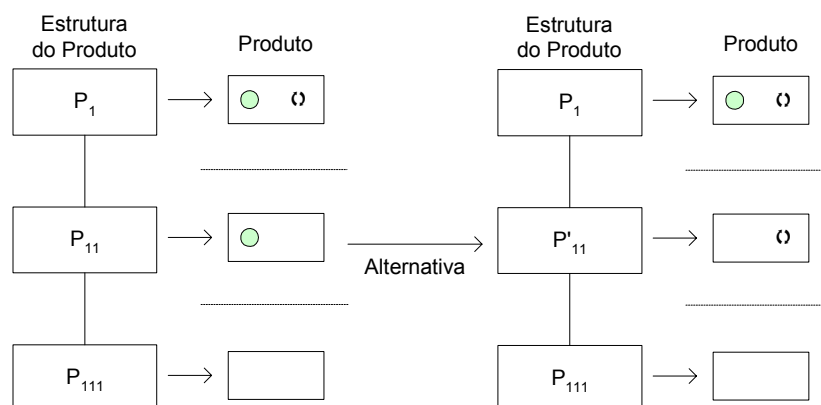


Figura 10: Estruturas de produto alternativas.

Na Figura 11 apresenta-se um produto simples que serve para ilustrar alguns dos conceitos deste processo de modelação. Este produto é uma caixa em chapa de metal para armazenar fotografias.

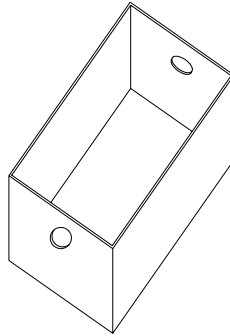


Figura 11: Produto final – Caixa para fotografias.

Este produto pode ser representado pela estrutura apresentada na Figura 12, com os seguintes elementos constituintes:

- O elemento  $P_1$  corresponde ao produto final, isto é, à caixa para fotografias.
- O elemento  $P_{11}$  corresponde à base da caixa.
- O elemento  $P_{12}$  corresponde ao conjunto completo das paredes da caixa.
- O elemento  $P_{121}$  corresponde a um canto de duas paredes soldadas, em que uma parede tem um furo circular que serve de pega da caixa.
- O elemento  $P_{122}$  corresponde ao outro canto de duas paredes soldadas, em que uma parede tem um furo oval que serve de pega da caixa.

Os processos de produção simplificados que permitem obter cada um dos elementos desta estrutura são processos de soldadura de chapa. O elemento  $P_1$  resulta do processo de soldar a base com o conjunto das paredes da caixa. O elemento  $P_{12}$  resulta do processo de soldar os dois cantos das paredes da caixa.

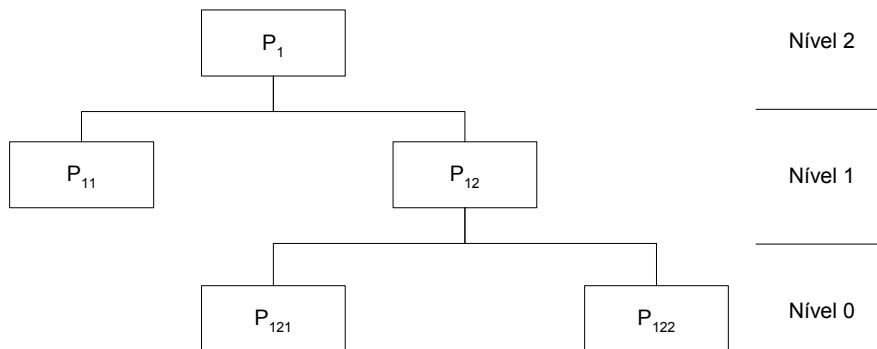


Figura 12: Estrutura do Produto com quantidades unitárias.

Entre cada um dos níveis da estrutura do produto executa-se, neste exemplo, apenas uma transformação de mudança de estado dos objectos de produção. Considerando-se que os elementos  $P_{12}$ ,  $P_{121}$  e  $P_{122}$  resultam de uma aquisição de componentes e que se pretende modelar e controlar o sistema incluindo esse ponto de aquisição, é possível estender esta estrutura de forma a ter mais três transformações, cada uma delas de obtenção destes elementos. Neste caso existirão cinco transformações de estado. É preciso notar que o resultado de qualquer serviço, isto é, acção de transformação, é um objecto de produção num determinado estado.

### 1.2.2. Caracterização das Células de Produção

Neste modelo um produto de uma encomenda é produzido numa célula dedicada. Isto significa que esta é projectada com o único objectivo de satisfazer aquela encomenda daquele produto. A célula resulta da selecção de recursos com aptidão e capacidade disponível que satisfaçam os requisitos da encomenda. Tal não restringe a utilização destes recursos a uma única célula. Se tiverem aptidão e capacidade disponível os recursos podem ser candidatos a serem utilizados noutras células.

Cada célula é composta por recursos de produção seleccionados a partir de um conjunto de recursos candidatos. Isto significa que na formação do sistema é necessário, para a formação de cada célula, resolver o problema de selecção de recursos para formação das células face à estrutura do produto, conforme ilustrado pela Figura 13.

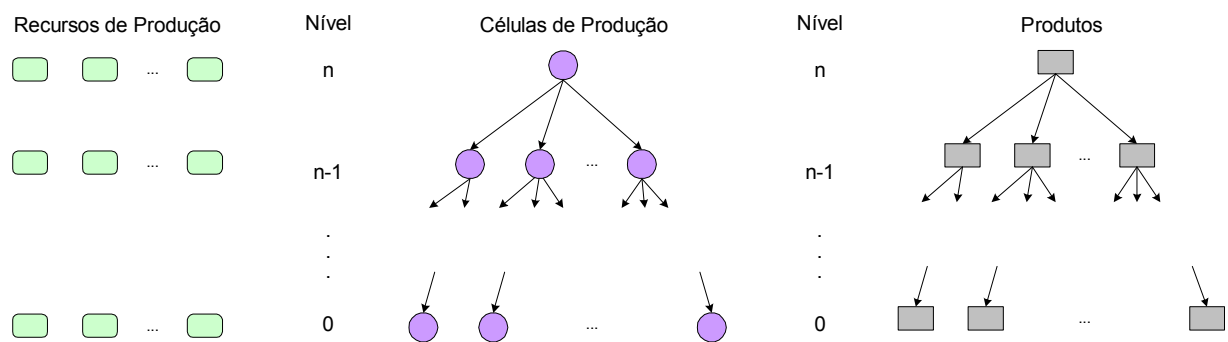


Figura 13: Representação dos diferentes níveis da estrutura do Sistema Distribuído de Produção

Neste modelo a árvore da estrutura do produto é similar à árvore de estrutura das células, em que a cada célula corresponde uma única tarefa de transformação: a de produzir o produto resultante do processamento na célula. Isto significa, que para cada encomenda de um produto é criado um sistema de produção por selecção de recursos para cada célula.

A formação do sistema pode basear-se na selecção de recursos para uma única célula que entrega o produto final, considerando-se assim a total agregação das transformações e produtos do sistema. Por outro lado pode basear-se na selecção de recursos para as células responsáveis pela execução das transformações dos níveis  $n$  e  $n-1$ , havendo agregação dos elementos do sistema nos níveis inferiores. Estas células são responsáveis pela produção dos elementos dos níveis  $n$  e  $n-1$  da estrutura do produto. Este processo de formação das células pode continuar até ao processo de selecção correspondente a todos os elementos da estrutura. Neste último caso não existe agregação de nenhuma célula nem de nenhum elemento da estrutura. Este processo de agregação permite considerar o processo de formação do Sistema Distribuído de Produção em qualquer nível em função da informação disponível, e das fronteiras definidas para o sistema.

Neste modelo uma célula autónoma é composta por um ou mais recursos autónomos seleccionados para uma transformação. Um recurso autónomo é um meio de produção que tem capacidade para executar totalmente o processamento requerido para a entrega de um produto. O recurso tem capacidade para organizar, planear e controlar o trabalho que leva à entrega ou conclusão do produto. Este recurso pode ser composto por recursos não autónomos no âmbito daquela transformação. Recursos não autónomos podem

ser uma máquina e um operador que apenas em conjunto poderão criar um recurso autónomo. Outro exemplo seria: uma máquina de corte, um conjunto de ferramentas e um operador. Os recursos não autónomos podem, em conjunto, compor um recurso com capacidade suficiente para processar um determinado produto, formando assim um recurso autónomo.

Uma encomenda do produto da Figura 11 terá associado um Sistema Distribuído de Produção como aquele que é ilustrado pela Figura 14. Cada produto da estrutura do produto da Figura 12 será entregue por uma célula correspondente. Cada uma destas células será composta por um ou mais recursos autónomos com a capacidade de execução da respectiva transformação, e que utiliza como objectos de produção de entrada os produtos descendentes da estrutura do produto.

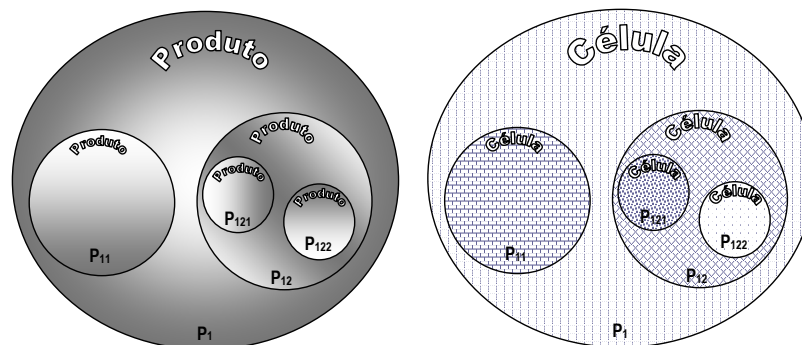


Figura 14: Exemplo de representação de Produtos e Células.

### **1.2.3. Coordenação e Controlo da Produção**

Este modelo de Sistemas Distribuídos de Produção é baseado na autonomia das células de produção e o processo de coordenação e controlo da produção resulta do processo de formação do sistema de produção, isto é, do processo de configuração e possível reconfiguração. Neste processo de formação seleccionam-se recursos autónomos para a formação de cada célula que executa cada uma das transformações de mudança de estado dos produtos, pertencentes à estrutura do produto final, criando-se assim, uma configuração específica para o Sistema Distribuído de Produção.

O elemento chave neste processo de coordenação e controlo da produção é a encomenda, que é a entidade que desencadeia os processos de planeamento e controlo de produção. A encomenda tem um sentido alargado, podendo estar associada a cada transformação do estado dos produtos. Assim sendo, a encomenda é aplicada num sentido genérico para selecção de recursos para cada célula. Os requisitos de produção são definidos, directa ou indirectamente, com base na encomenda, e incluem a identificação das quantidades e do produto requerido. Do processo de criação do Sistema Distribuído de Produção resulta o estabelecimento de prazos de entrega, que dependem da disponibilidade dos recursos.

A célula de produção será criada em função de critérios de decisão baseados em factores como prazo de entrega e custo. O processo de criação da célula, ou de criação da configuração do Sistema Distribuído de Produção, deve realizar-se, no modelo MSDP, com base nos referidos critérios de decisão, de acordo com a filosofia de Produção Simultânea.



**i. Processo de Configuração do Sistema Distribuído de Produção**

No processo de configuração do Sistema Distribuído de Produção do MSDP são consideradas duas formas alternativas de criação da configuração do sistema: Processo Local e Processo Global. No processo local, ilustrado pela Figura 15, cada célula é criada, isto é, são seleccionados recursos, com base numa encomenda gerada para cada transformação. No processo global cada célula é criada com base numa única encomenda, a partir da informação sobre todos os recursos que podem executar as transformações de mudança de estado de todos os produtos da estrutura do produto final. Em ambos os casos a selecção de recursos baseia-se nos princípios de Produção Simultânea, explorando a possibilidade de atribuição de transformações a vários recursos alternativos e em simultâneo, com o objectivo de reduzir o prazo de entrega. A redução do prazo de entrega resulta da redução do tempo de percurso de fabrico, isto é, do tempo que medeia entre o início e a conclusão do processamento de todas as transformações para produção do produto.

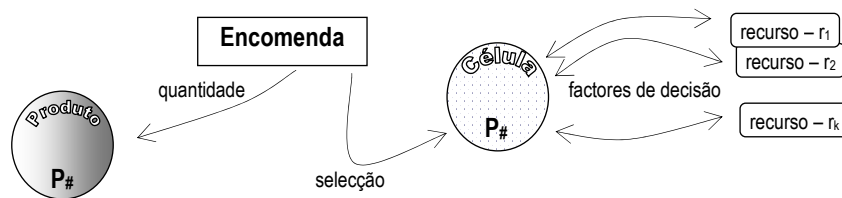


Figura 15: Encomenda e o processo local de formação de uma célula de produção.

A repetição do processo local ilustrado pela Figura 15 para as encomendas de todos os produtos da estrutura permite criar o Sistema Distribuído de Produção. Outra forma de criação deste sistema resulta da aplicação do processo global, ilustrado pela Figura 16, de criação das células em função dos recursos disponíveis para uma encomenda de um produto. Por aplicação de critérios de decisão preestabelecidos, seleccionam-se, de forma integrada, os recursos para todas as transformações.

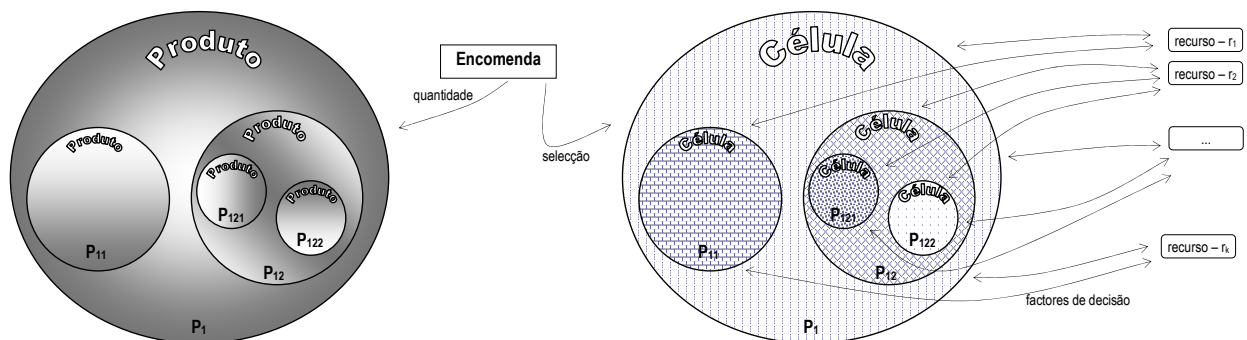


Figura 16: Encomenda e o processo global de formação de uma célula de produção.

A criação do Sistema Distribuído de Produção para a satisfação de uma encomenda de uma unidade da caixa de fotografias da Figura 11, com estrutura de produto  $P_1$  representada na Figura 12, depende do produto, dos recursos disponíveis, do processo de configuração global ou local e dos critérios de decisão. Neste exemplo, pretende-se criar a configuração do sistema, modelando o processo de obtenção dos elementos  $P_{12}$ ,  $P_{121}$  e  $P_{122}$ , conforme descrição apresentada no último parágrafo da secção 1.2.1 deste capítulo. Dessa forma será necessário considerar 5 transformações de estado dos produtos respeitantes aos 5

elementos da estrutura. Admite-se para tal, que por cada componente, existem 4 recursos autónomos para execução de cada uma das transformações em causa, com tempo completamente livre. Cada um dos recursos consegue executar algumas transformações com tempos unitários, i.e. tempos de entrega de uma unidade de produto indicados, na Tabela 6.

Tabela 6: Tempos unitários de produção propostos por cada recurso autónomo para cada componente.

Tempos	r <sub>1</sub>	r <sub>2</sub>	r <sub>3</sub>	r <sub>4</sub>	r <sub>5</sub>	r <sub>6</sub>	r <sub>7</sub>	r <sub>8</sub>	r <sub>9</sub>	r <sub>10</sub>	r <sub>11</sub>	r <sub>12</sub>	r <sub>13</sub>	r <sub>14</sub>	r <sub>15</sub>	r <sub>16</sub>
P <sub>1</sub>	3	2	3	4												
P <sub>11</sub>	2				3	4	3									
P <sub>12</sub>		1						3	2	4						
P <sub>121</sub>		3									3	3	3			
P <sub>122</sub>			2											2	3	5

Neste exemplo é possível seleccionar, de uma forma visual simples, um conjunto de recursos de produção que conduziriam ao menor prazo de entrega para uma encomenda de uma unidade de P<sub>1</sub>. Uma solução possível é apresentada na Tabela 7, correspondente a 6 UT de prazo de entrega após início do processamento. O processamento de P<sub>12</sub> pode iniciar-se no instante 3 após conclusão de P<sub>121</sub> e P<sub>122</sub>. O processamento de P<sub>1</sub> pode iniciar-se após conclusão de P<sub>11</sub> e P<sub>12</sub>, que se pode dar no instante 4 (3 UT + 1 UT).

Tabela 7: Recursos de produção seleccionados.

Tempos	r <sub>1</sub>	r <sub>2</sub>	r <sub>3</sub>	r <sub>4</sub>	r <sub>5</sub>	r <sub>6</sub>	r <sub>7</sub>	r <sub>8</sub>	r <sub>9</sub>	r <sub>10</sub>	r <sub>11</sub>	r <sub>12</sub>	r <sub>13</sub>	r <sub>14</sub>	r <sub>15</sub>	r <sub>16</sub>
P <sub>1</sub>	3	2	3	4												
P <sub>11</sub>	2				3	4	3									
P <sub>12</sub>		1						3	2	4						
P <sub>121</sub>		3									3	3	3			
P <sub>122</sub>			2											2	3	5

#### **1.2.4. Informação necessária para a criação do sistema**

A encomenda de um produto desencadeia a criação de um sistema distribuído de produção por selecção de recursos de produção que formam células. A cada produto corresponde uma célula, que é composta pela selecção de recursos de produção adequados para aquela encomenda. Assim sendo, além da estrutura do produto e da quantidade encomendada é necessário conhecer para cada recurso de produção:

- Aptidão: indicação dos produtos que o meio de produção pode executar.
- Tempo Unitário: o tempo de execução de uma transformação para produção de uma unidade de um determinado produto, isto é, o tempo que medeia entre o início e a conclusão da execução dessa transformação .
- Disponibilidade: indicação da agenda ou da oferta do recurso num determinado horizonte.
- Custo: custo associado à satisfação de cada encomenda.

## **2. REPRESENTAÇÃO FORMAL DO SISTEMA DISTRIBUÍDO DE PRODUÇÃO**

O modelo de Sistemas Distribuídos de Produção apresentado nesta dissertação baseia-se numa representação de sistemas de produção embebidos em sistemas de produção com uma correspondência directa com a estrutura de produto. Um sistema agregado tem menos produtos e transformações, tornando possível lidar com uma menor quantidade de dados, reduzindo a dimensão dos problemas de cálculo de necessidade de materiais, necessidade de capacidade e de programação de produção.

A agregação de dados é utilizada com o objectivo de redução da dimensão de problemas, em diversos sistemas, nomeadamente nos sistemas de planeamento e controlo de produção do tipo MRP. Nos sistemas MRP determinam-se necessidades de materiais para os períodos de planeamento, com base em dados sobre prazos de entrega (LT – “Lead Time”) dos materiais, que são impostos pelos utilizadores do sistema. Estes dados resultam de uma agregação de dados sobre a capacidade dos recursos de produção, que não são definidos em função da carga dinâmica desses mesmos recursos, mas impostos em função da experiência e de uma utilização predefinida para o sistema. Como o LT não é definido em função da carga dinâmica do sistema, o instante de conclusão de uma encomenda pode ser totalmente diferente do prazo de entrega estabelecido.

A ligação entre níveis de agregação do sistema pode permitir determinar as necessidades de materiais e capacidade requerida do sistema, de uma forma mais precisa em qualquer nível de agregação. Pode-se, inclusive, resolver problemas de programação de produção em qualquer nível de agregação, afectando recursos a transformações de produção.

O modelo de Sistemas Distribuídos de Produção desenvolvido requer a utilização de métodos de afectação de recursos locais ou globais, para aplicação dos processos, referidos atrás, de configuração das células de produção. O desenvolvimento de métodos de afectação precisos deve apoiar-se numa representação formal do sistema. Assim sendo, utiliza-se, neste trabalho, uma representação formal, baseada em Redes de Petri e na formulação algébrica apresentada por Lecompte, Deschamps and Bourrières (2000), que os autores designam por Programação Generalizada<sup>28</sup>. Chega-se assim a um modelo formal de um sistema que possibilita a exploração de vantagens, tais como as de unicidade da representação, possibilidade de manuseamento preciso das entidades do sistema, relacionamento entre níveis de agregação e desenvolvimento de modelos de afectação de recursos.

Nesta secção, apresenta-se a representação formal do modelo de Sistemas Distribuídos de Produção, isto é, representação formal do MSDP. Esta secção é dividida em duas secções fundamentais de modelação de produtos e transformações e de modelação dos recursos. A modelação de produtos e transformações também contém a modelação da agregação de níveis e de restrições de precedência. A modelação de recursos também contém a modelação de restrições e do processo de afectação de recursos.

---

<sup>28</sup> “Generalized Scheduling”.

## 2.1 Modelação de Produtos e Transformações

No trabalho sobre “Programação Generalizada para Empresas Virtuais”, Lecompte, Deschamps and Bourrières (2000) apresentam uma representação formal algébrica de sistemas de produção, baseada na visão da Fábrica Fractal de Warnecke (1993). No conceito de Programação Generalizada a visão dos sistemas de produção é uma visão baseada nos processos de produção. Tais processos são representados por Redes de Petri. Lecompte, Deschamps and Bourrières (2000) apresentam a formalização do sistema de produção e da ligação entre vários níveis de agregação do sistema. Em cada nível de agregação há pelo menos uma transformação de produção, i.e. tarefa, correspondente a um processo de produção, que agrega várias transformações do nível de agregação imediatamente inferior. Como se pode ver na Figura 17, níveis de agregação superior são compostos por menos tarefas. Em cada nível de agregação pode existir um processo de afectação de meios de produção a cada tarefa, isto é, pode resolver-se um problema de programação de produção.

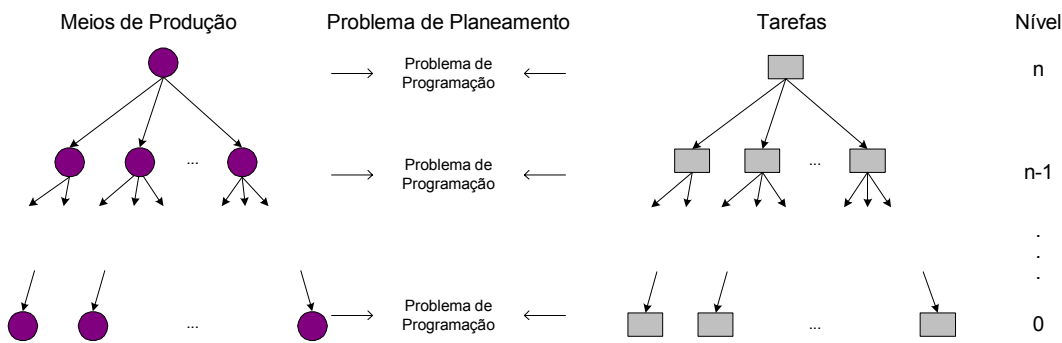


Figura 17: Programação Generalizada - Lecompte, Deschamps and Bourrières (2000).

A formulação algébrica apresentada por Lecompte, Deschamps and Bourrières (2000) permite lidar com a modelação de sistemas embebidos noutros, e assim aplicar-se na representação do modelo de Sistemas Distribuídos de Produção desenvolvido neste capítulo. Na representação da Programação Generalizada não é apresentado um modelo do sistema de produção, ou a forma de construir estes modelos, não havendo assim nenhum mecanismo para efectuar a ligação entre níveis de agregação.

Neste capítulo, apresenta-se a formulação algébrica da Programação Generalizada, a extensão efectuada neste trabalho a essa formulação e a aplicação da formulação resultante ao modelo de Sistemas Distribuídos de Produção. Esta representação formal do Sistema Distribuído de Produção desenvolvido neste trabalho é, também, designada por modelo formal do MSDP. O modelo MSDP inclui um mecanismo de transição entre níveis, que relaciona diferentes níveis de agregação de transformações através da estrutura do produto. Este mecanismo é utilizado para geração dinâmica da configuração do sistema de produção a partir de informação sobre a encomenda, sobre os recursos disponíveis e sobre a estrutura do produto que contém informação sobre as transformações a executar. Este relacionamento entre a estrutura do produto e as transformações a executar no sistema de produção, e a ligação com o processo de construção do modelo do Sistema Distribuído de Produção são aspectos que definem e distinguem o modelo desenvolvido nesta dissertação.

### 2.1.1. Breve Referência às Redes de Petri

As Redes de Petri são utilizadas neste trabalho e na Programação Generalizada por Lecompte, Deschamps and Bourrières (2000) para representação e modelação de sistemas de produção. Por essa razão, torna-se necessário efectuar uma introdução a este conceito. Deve notar-se, que a notação utilizada nesta secção, serve apenas para a introdução das Redes de Petri e não é utilizada em mais nenhuma parte desta dissertação.

Segundo Proth and Xie (1996) a Rede de Petri (PN) é um grafo direccionado com dois tipos de nós denominados locais e transições<sup>29</sup>. Os arcos direccionados ligam alguns locais a algumas transições, com um peso associado. Um arco direccionado nunca liga um local a outro nem uma transição a outra. Na representação gráfica da Rede de Petri um local é representado por um círculo e uma transição por um rectângulo ou barra. Cada local pode ter marcas<sup>30</sup> que permitem modelar o comportamento dinâmico do sistema. A marcação<sup>31</sup> do sistema é um vector com um número de elementos igual ao número de locais, sendo cada elemento inteiro não negativo, igual ao número de marcas nesse local.

Formalmente, uma Rede de Petri, PN, é um 5-tuplo, isto é, um conjunto de 5 elementos diferentes:

$$PN = (P, T, A, W, M_0).$$

Sendo:

- $P = \{p_1, p_2, \dots, p_m\}$  - Conjunto Finito de Locais.
- $T = \{t_1, t_2, \dots, t_n\}$  - Conjunto Finito de Transições, sendo  $P \cap T = \emptyset$ .
- $A \subseteq (P \times T) \cup (T \times P)$  - Conjunto Finito de Arcos direccionados que ligam locais a transições e transições a locais.
- $W : A \rightarrow \{1, 2, \dots\}$  - Função Peso associado aos Arcos direccionados que indica o número de marcas necessárias para que as transições se executem e o número de marcas resultantes dessa transições.
- $M_0 = [m_1, m_2, \dots, m_m]$  - Vector Marcação Inicial que indica o número inicial de marcas em cada local.

No exemplo da Figura 18 a Rede de Petri tem 4 locais, referenciados por  $p_1$  a  $p_4$ , e 5 transições, referenciadas por  $t_1$  a  $t_5$ . O vector de marcação é  $M_0 = [3, 5, 1, 2]$ . Quando não se indica o peso dos arcos, assume-se que esse peso é unitário, que é o caso deste exemplo.

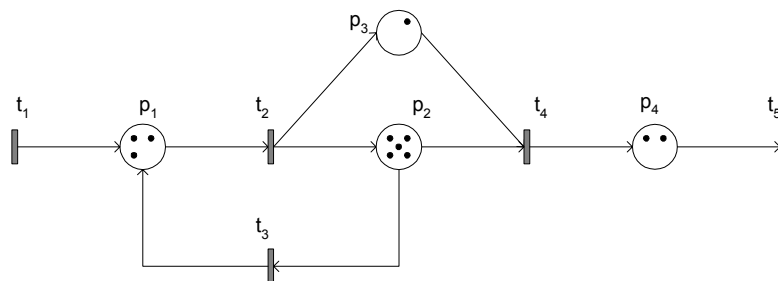


Figura 18: Exemplo de Rede de Petri.

<sup>29</sup> "Places and Transitions".

<sup>30</sup> "Token".

<sup>31</sup> "Marking".

### 2.1.2. Modelação do Produto e do Processo

A aplicação da Programação Generalizada recorre a uma representação genérica de sistemas de produção, baseada em Redes de Petri, cuja aplicação ao modelo do Sistema Distribuído de Produção desenvolvido neste trabalho, requer a definição dos formalismos utilizados. Assim, os locais da Rede de Petri representam os objectos de produção e as transições representam as transformações referidas no modelo MSDP, que são representados por:

- $O$  ( $\dim O = m$ ) – conjunto dos  $m$  objectos de produção  $o_i$ ,  $i=1, \dots, m$ , i.e. produtos do sistema;
- $T$  ( $\dim T = n$ ) – conjunto das  $n$  transições  $t_j$ ,  $j=1, \dots, n$ , i.e. transformações executáveis no sistema.

A partir da Rede de Petri do sistema podem obter-se as matrizes individuais de consumo (**Pre**) e fornecimento (**Post**) de objectos de produção pelas transformações, e a matriz que integra os consumos e fornecimentos (**C**), todas representadas na Equação 1. O símbolo  $\mathbb{N}$  representa vectores ou matrizes em que todos os elementos são inteiros não negativos.

$$\underline{\text{Pre}} : \mathbf{O} \times \mathbf{T} \rightarrow \mathbb{N}; \underline{\text{Post}} : \mathbf{O} \times \mathbf{T} \rightarrow \mathbb{N}; \mathbf{C} = \text{Post} - \text{Pre} \quad \text{Equação 1}$$

Cada elemento  $C_{ij}$  da matriz **C** corresponde à quantidade do objecto  $o_i$  consumido (valor negativo) ou produzido (valor positivo) pela transformação  $t_j$ . Estas quantidades consumidas ou produzidas são designadas, nesta dissertação, por quantidades de utilização.

#### i. Exemplo

Para clarificação do processo de modelação utiliza-se recursivamente o exemplo da caixa de fotografias (Figura 11, página 50), cuja estrutura de produto, baseada na Figura 12, está representada na Figura 19 à esquerda. Esta representação resulta da adição de três elementos,  $P_{111}$ ,  $P_{1211}$  e  $P_{1221}$ , para modelação do seu processo de aquisição, conforme descrição do último parágrafo da secção 1.2.1 deste capítulo. Além da adição dos produtos, acrescenta-se um conjunto de pequenos rectângulos que representam as transformações que fornecem o produto colocado imediatamente acima. A introdução da representação das transformações na estrutura do produto permite ilustrar o conceito, introduzido na secção III - 1.2.1, de mudança de estado dos produtos como resultado da execução de uma transformação por parte de uma célula autónoma.

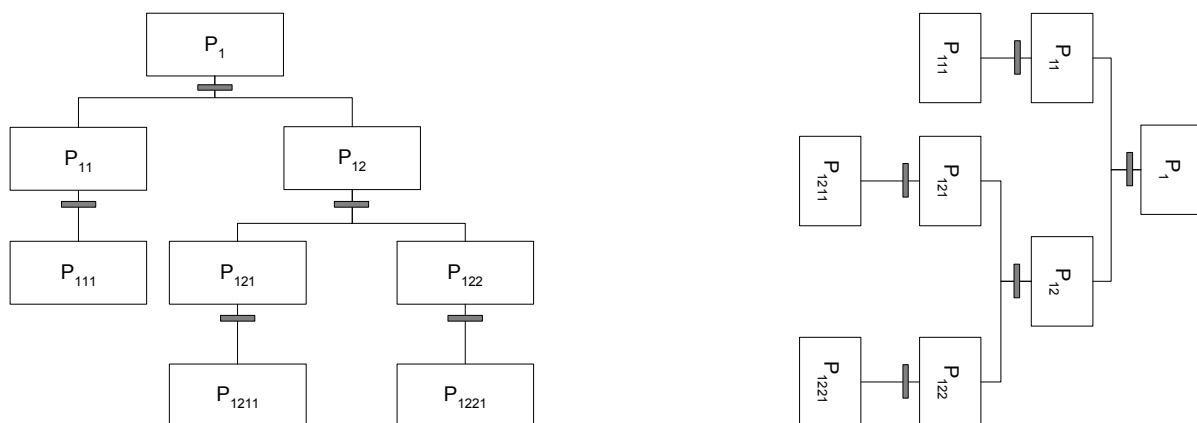


Figura 19: Estrutura do Produto com transformações - representação vertical e horizontal.

A rotação da estrutura do produto, representada na Figura 19 à direita, aproxima-a de uma Rede Petri. Os objectos correspondem aos produtos e as transformações correspondem a células de produção do modelo de Sistemas Distribuídos de Produção desenvolvido neste capítulo. A representação formal de Programação Generalizada permite representar este exemplo através da Rede de Petri da Figura 20. Efectua-se assim, nesta dissertação, a ligação entre um modelo de SPOP com uma representação de sistemas de produção com uma visão fundamentalmente baseada nos processos de produção.

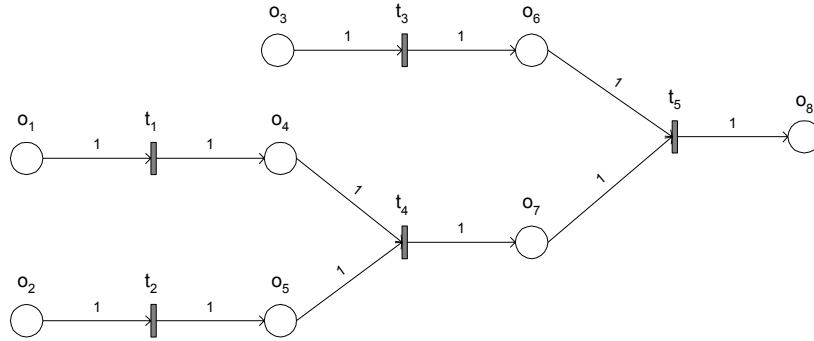


Figura 20: Rede de Petri representando a dinâmica do sistema de produção.

Esta Rede de Petri é composta por 8 objectos (locais) e 5 transformações (transições):

- $O = \{o_1, \dots, o_8\}, m = 8.$
- $T = \{t_1, \dots, t_5\}, n = 5.$

Os objectos  $o_4$  a  $o_8$  correspondem respectivamente aos produtos  $P_{121}, P_{122}, P_{11}, P_{12},$  e  $P_1$ . As células de produção de cada um destes produtos referente às transformações de entrega destes produtos são representadas pelas transformações  $t_1$  a  $t_5$ . Os objectos  $o_1$  a  $o_3$  são exactamente iguais aos objectos  $o_4$  a  $o_6$ , isto é, aos produtos  $P_{1211}, P_{1221}$  e  $P_{111}$ , sendo introduzidos para permitir a modelação das células de entrega desses componentes, pela execução das transformações  $t_1$  a  $t_3$ . Esta rede de Petri representa o nível mais elementar das transformações a executar para produzir um produto a partir dos seus componentes.

As matrizes da Equação 1 de representação dos objectos consumidos e fornecidos são:

$$\mathbf{Pre} = \begin{matrix} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \\ o_6 \\ o_7 \\ o_8 \end{matrix} \begin{bmatrix} t_1 & t_2 & t_3 & t_4 & t_5 \\ 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \\ & & & & & 1 \\ & & & & & & 1 \\ & & & & & & & 1 \end{bmatrix}; \mathbf{Post} = \begin{matrix} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \\ o_6 \\ o_7 \\ o_8 \end{matrix} \begin{bmatrix} t_1 & t_2 & t_3 & t_4 & t_5 \\ & & & & \\ & & & & \\ 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}; \mathbf{C} = \begin{matrix} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \\ o_6 \\ o_7 \\ o_8 \end{matrix} \begin{bmatrix} t_1 & t_2 & t_3 & t_4 & t_5 \\ -1 & & & & \\ & -1 & & & \\ & & -1 & & \\ 1 & & & -1 & \\ & 1 & & -1 & \\ & & 1 & & -1 \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix}.$$

Interpretando a matriz  $\mathbf{C}$  pode verificar-se, por exemplo, que a transformação  $t_1$  consome um objecto  $o_1$  e fornece um objecto  $o_4$ , e que a transformação  $t_4$  consome um objecto  $o_4$  e um objecto  $o_5$  e fornece um objecto  $o_7$ . Neste exemplo, todas as quantidades de consumo e produção, i.e. quantidades de utilização, de objectos são unitárias.

### 2.1.3. Relação entre o Armazenamento e as Transformações

Lecompte, Deschamps and Bourrières (2000), relacionam o armazenamento de objectos com as transformações do sistema através de vectores de quantidades armazenadas de objectos e de quantidade de trabalho executada nas transformações. A quantidade armazenada<sup>32</sup>  $S_i$  de cada objecto de produção, i.e. produto da estrutura de produto final, é representada no vector  $\mathbf{S}$  do nível de armazenamento, i.e. das quantidades armazenadas de todos os objectos representados no sistema. A realização de uma transformação do sistema, consome quantidades de determinados objectos e fornece quantidades de determinados objectos especificadas pelos pesos dos arcos da rede, isto é, pelas quantidades de utilização. O número de vezes que se executa essa transformação influencia a quantidade armazenada de cada objecto envolvido. Este número de vezes de execução de uma determinada transformação corresponde à *quantidade de trabalho*  $W_j$  executada nessa transformação, sendo o conjunto destas quantidades agrupada no vector  $\mathbf{W}$  de quantidade de trabalho do sistema. Assim:

- $\mathbf{S} = [S_1, \dots, S_i, \dots, S_m]^T$  – nível de armazenamento de todos os objectos da empresa.
- $\mathbf{W} = [W_1, \dots, W_j, \dots, W_n]^T$  – quantidade de trabalho de cada uma das  $n$  transformações do sistema.

Se o nível de armazenamento do sistema num determinado momento for representado por  $\mathbf{S}$  e se se pretender alterar esse nível de armazenamento para  $\mathbf{S}'$ , então é necessário executar uma quantidade de trabalho que altere o armazenamento da forma pretendida. Supondo que apenas a execução da transformação  $t_j$  fornece o objecto  $o_i$ , e que da execução da transformação  $t_j$ , uma única vez, resultam duas unidades do objecto  $o_i$ , então  $C_{ij}$  é igual a 2. Assim sendo, a diferença entre  $S_i$  e  $S_i'$  é dada pelo produto entre o número de vezes que se executam transformações  $W_j$ , que forneçam objectos  $o_i$ , pela quantidade que fornecem de cada uma das vezes. Esta relação é representada pela Equação 2, de igualdade entre a diferença no nível de armazenamento e o produto da quantidade dos objectos consumidos e fornecidos por uma execução de cada transformação pelo número de vezes que se executam as transformações.

$$\delta\mathbf{S} = \mathbf{S}' - \mathbf{S} = \mathbf{C}\mathbf{W}, \text{ com } \mathbf{S}, \mathbf{S}', \mathbf{W} \in \mathbb{N} \quad \text{Equação 2}$$

Conhecendo-se a quantidade a produzir ou consumir de todos os produtos obtém-se um sistema de equações com variável  $\mathbf{W}$ . A dimensão deste sistema é definida por  $m$  equações, correspondente à dimensão do vector  $\delta\mathbf{S}$ , e por  $n$  incógnitas, correspondente à dimensão do vector  $\mathbf{W}$ . Neste sistema de equações podem verificar-se os seguintes casos:

- i.  $m > n$ . Sistema com mais objectos do que transformações, isto é, com mais equações do que incógnitas, para o qual pode não haver solução possível. Este é o caso que se aplica ao modelo MSDP apresentado, que se baseia na estrutura do produto, configurando operações de fabrico de componentes e sua montagem. A determinação de soluções possíveis depende do estabelecimento de objectivos atingíveis.

<sup>32</sup> "Stock".



- ii.  $m = n$ . O número de objectos é igual ao número de transformações, isto é, o número de equações é igual ao de incógnitas.
- iii.  $m < n$ . Sistema com mais transformações do que objectos, isto é, com mais incógnitas do que equações.

O caso ii. descreve sistemas de produção com um número igual de transformações e de objectos, em que a possibilidade de solução do sistema depende da rede de transformações e dos objectivos estabelecidos. No caso iii. de sistemas de produção com mais transformações do que objectos, existe uma multiplicidade de percursos alternativos de produção, que exigem a selecção de uma solução através do estabelecimento de critérios de escolha do processo produtivo. Estes dois casos, verificam-se com a existência de processos alternativos para obter o mesmo produto, conforme ilustração da Figura 21. O modelo MSDP trata esta questão (secção III - 1.2.1, Figura 10), da possibilidade de existência de processos alternativos, com a replicação de estruturas de produto alternativas, que se reflectem em estruturas alternativas de sistemas de produção. Isto significa que nenhum destes casos, ii. e iii., se aplica ao modelo MSDP.



Figura 21: Exemplos de sistemas com representação de processos de produção alternativos.

No caso i., o caso que se verifica no modelo MSDP, com mais objectos do que transformações, se forem definidos objectivos irrealistas para a variação de armazenamento dos objectos não haverá solução para a resolução do sistema de equações representado. Assim, supondo que o nosso objectivo, para o sistema da Figura 20, é definido pelo seguinte vector  $\delta\mathbf{S}$  para os objectos  $o_1$  a  $o_8$ , então não haverá uma solução possível para a quantidade de trabalho necessária para atingir esse objectivo:

$$\delta\mathbf{S} = [-1, -30, -5, 2, 3, 2, 5, 5]^T .$$

Este vector define o aumento e diminuição do armazenamento pretendido. Para se aumentar o armazenamento de  $o_8$  em cinco unidades é necessário ter uma quantidade de trabalho em  $t_5$  de cinco unidades que vai consumir cinco unidades de  $o_6$ . Para se aumentar o armazenamento de  $o_6$  em duas unidades (objectivo), mais as cinco consumidas por  $t_5$ , é necessário ter uma quantidade de trabalho em  $t_3$  de sete unidades. Isto faz com que seja necessário consumir sete unidades de  $o_3$  sendo este valor diferente do objectivo estabelecido de cinco unidades consumidas. Nos casos ii. e iii., também se podem estabelecer objectivos que não permitem obter soluções para o problema. Nestes casos, a definição de um objectivo de aumento do nível de armazenamento de  $o_1$ , não permite determinar uma quantidade de trabalho, que satisfaça esse objectivo, para qualquer uma das transformações,.

**i. Exemplo**

Considera-se, para o exemplo da Figura 20, que se pretende determinar a quantidade de trabalho necessária para produzir uma determinada quantidade de produto final. Se se pretender produzir uma unidade do produto  $P_1$  ( $o_8$ ) e manter o nível de armazenamento dos produtos fabricados no sistema, obtém-se o seguinte vector referente à diferença das quantidades em armazenamento:

$$\delta S = S' - S = [\delta S_1, \delta S_2, \delta S_3, 0, 0, 0, 0, 1]^T.$$

Assim sendo, por aplicação da Equação 2, determina-se a quantidade de trabalho que é necessário executar em cada transformação:

$$\begin{aligned} \delta S &= CW \\ [\delta S_1, \delta S_2, \delta S_3, 0, 0, 0, 0, 1]^T &= C \times [W_1, W_2, W_3, W_4, W_5]^T \\ \begin{bmatrix} \delta S_1 \\ \delta S_2 \\ \delta S_3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} &= \begin{bmatrix} -1 & & & & \\ & -1 & & & \\ & & -1 & & \\ 1 & & & -1 & \\ & 1 & & -1 & \\ & & 1 & & -1 \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \end{bmatrix} \\ \begin{bmatrix} \delta S_1 \\ \delta S_2 \\ \delta S_3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} &= \begin{bmatrix} -W_1 \\ -W_2 \\ -W_3 \\ W_1 - W_4 \\ W_2 - W_4 \\ W_3 - W_5 \\ W_4 - W_5 \\ W_5 \end{bmatrix} \Leftrightarrow \begin{cases} -W_1 = \delta S_1 \\ -W_2 = \delta S_2 \\ -W_3 = \delta S_3 \\ W_1 - W_4 = 0 \\ W_2 - W_4 = 0 \\ W_3 - W_5 = 0 \\ W_4 - W_5 = 0 \\ W_5 = 1 \end{cases} \Leftrightarrow \begin{cases} -W_1 = \delta S_1 \\ -W_2 = \delta S_2 \\ -W_3 = \delta S_3 \\ W_1 = W_4 \\ W_2 = W_4 \\ W_3 = W_5 \\ W_4 = W_5 \\ W_5 = 1 \end{cases} \Leftrightarrow \begin{cases} \delta S_1 = -1 \\ \delta S_2 = -1 \\ \delta S_3 = -1 \\ W_1 = 1 \\ W_2 = 1 \\ W_3 = 1 \\ W_4 = 1 \\ W_5 = 1 \end{cases} \end{aligned}$$

A interpretação destes resultados indica que é necessário executar uma vez cada uma das transformações para se aumentar em uma unidade a quantidade armazenada do objecto  $o_8$ , correspondente ao produto  $P_1$ , e que é necessário consumir uma unidade dos objectos  $o_1$ ,  $o_2$  e  $o_3$ .

**2.1.4. Processo de Agregação**

No modelo MSDP várias transformações podem ser representadas por uma transformação de mais alto nível que as agrega. A utilização de agregação de níveis pode permitir a redução da dimensão do problema, já que, em níveis agregados existirão menos transformações, logo menos problemas de selecção de recursos para cada transformação. A aplicação deste conceito em diversos níveis é coincidente com a visão da Fábrica Fractal e com o modelo de Sistemas Distribuídos de Produção MSDP. Uma formulação genérica para o processo de agregação foi apresentada por Lecompte, Deschamps and Bourrières (2000), podendo ser ligeiramente adaptado ao modelo MSDP. Deve notar-se, que a noção de nível de agregação, no modelo de Programação Generalizada, não está associada à estrutura do produto, porque a representação dos sistemas de produção também não está. Assim, de uma forma geral, um nível de agregação superior, tem pelo menos uma transformação que agrega, pelo menos duas transformações do nível inferior.

No modelo de Sistemas Distribuídos de Produção o número de transformações de um determinado nível será sempre superior ao de um nível de agregação superior, coincidindo com a formulação apresentada na

Programação Generalizada. Isto porque a agregação de transformações significa precisamente que um conjunto de transformações será visto no nível superior como uma única transformação.

A notação utilizada, nomeadamente, os conjuntos  $T$  e  $O$  e as matrizes  $C$ ,  $W$  e  $\delta S$ , passam a ser indexadas ao nível  $v$  de agregação considerado. No modelo MSDP o número de transformações em dois níveis consecutivos é definido pela cardinalidade (**card**) do conjunto  $T$  representada na Equação 3.

A relação entre o número de objectos em níveis de agregação consecutivos é definido por Lecompte, Deschamps and Bourrières (2000), para a Programação Generalizada, pela Equação 4. Segundo esta definição o número de objectos, cardinalidade de  $O$ , de um determinado nível será sempre superior ao do nível de agregação imediatamente acima ( $v+1$ ).

	$\text{card } T^{v+1} < \text{card } T^v$	Equação 3
	$\text{card } O^{v+1} < \text{card } O^v$	Equação 4

A definição da relação entre o número de objectos em níveis consecutivos de agregação não é, no entanto, suficientemente genérica, na medida que não considera a possibilidade de agregação de várias transformações em paralelo que entreguem vários objectos ou o mesmo objecto, e que podem conduzir à existência do mesmo número de objectos entre níveis de agregação.

Na Figura 22 apresentam-se casos de exemplo com níveis de agregação em que se mantém o mesmo número de objectos. No Caso 1 da Figura 22 apresenta-se um exemplo, de duas transformações  $t_1$  e  $t_2$ , sob um fundo escuro, que entregam dois objectos em paralelo e que podem ser agregadas numa única transformação  $t_1$  do nível com agregação. No Caso 2 da Figura 22 apresenta-se um exemplo, de duas transformações alternativas  $t_1$  e  $t_2$ , sob um fundo escuro, que podem entregar o mesmo objecto e que podem ser agregadas numa única transformação  $t_1$  do nível com agregação. Esta agregação permite efectuar a gestão dessas transformações como uma única transformação, reduzindo a dimensão do problema de determinação da quantidade de trabalho das transformações.

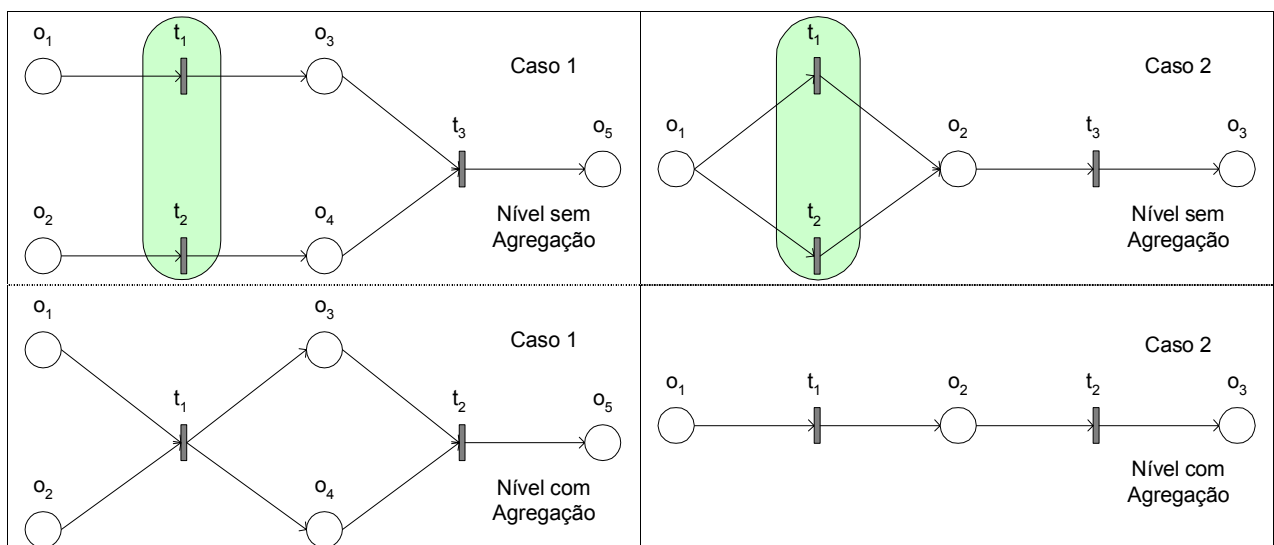


Figura 22: Exemplos de níveis com agregação em que se mantém o mesmo número de objectos.

Considerando uma formulação mais geral para a Programação Generalizada, o número de objectos de um determinado nível,  $v+1$ , será sempre não superior ao do nível de agregação  $v$  imediatamente abaixo. Assim sendo, a Equação 4 é substituída pela Equação 5.

---

$$\text{card } O^{v+1} \leq \text{card } O^v \quad \text{Equação 5}$$

No modelo de Sistemas Distribuídos de Produção desenvolvido nesta dissertação, um produto da estrutura do produto é obtido pela execução de uma transformação por uma célula autónoma, não se explorando a possibilidade de duas transformações entregarem o mesmo produto, o que leva à rejeição do Caso 2 da Figura 22. As alternativas de processo de produção podem ser exploradas apenas pela definição de novas estruturas de produto, conforme descrição apresentada na secção 1.2.1 deste capítulo. Neste modelo, também não é possível representar uma transformação divergente, em que uma célula autónoma entrega mais do que um produto, o que leva à rejeição do Caso 1 da Figura 22.

Como a estrutura do sistema está directamente ligada à estrutura do produto, de acordo com o modelo MSDP, o número de objectos de um determinado nível será sempre inferior ao do nível de agregação imediatamente abaixo, conforme definição apresentada na Equação 6. No nível mais alto de agregação só se consideram os objectos de entrada – matérias primas – e o produto de saída do sistema. No nível mais baixo consideram-se todos os objectos da empresa (matérias primas, componentes e produtos finais). Nos níveis intermédios alguns dos produtos estão embebidos nas transformações agregadas. Não é possível ter, num nível superior de agregação, mais objectos do que no nível inferior.

---

$$\text{card } O^{v+1} < \text{card } O^v \quad \text{Equação 6}$$

No modelo do Sistema Distribuído de Produção uma transformação que produz um produto, num nível superior, pode agregar transformações que produzem outros elementos da estrutura do produto do nível imediatamente abaixo.

A relação entre objectos em níveis consecutivos é especificada por Lecompte, Deschamps and Bourrières (2000) através da matriz binária  $\mathbf{M}_v^{v+1}$ , cujos elementos indicam e relacionam objectos existentes nos dois níveis. Esta matriz faz o relacionamento entre objectos de níveis diferentes, colocando em conformidade a nomenclatura desses objectos. As linhas da matriz representam os objectos do nível de agregação superior e as colunas, os objectos do nível inferior. Um elemento desta matriz com valor um representa um objecto que existe em ambos os níveis de agregação, e em cada linha ou coluna da matriz só pode existir um elemento com valor um. Isto significa que um produto não pode ser representado por dois produtos em outro nível. Esta matriz tem colunas completamente preenchidas por zeros, relativas aos objectos embebidos na agregação.

A relação entre transformações em níveis consecutivos é especificada pela matriz binária  $\mathbf{N}_v^{v+1}$ , cujos elementos relacionam transformações existentes nos dois níveis. As linhas da matriz representam transformações do nível de agregação inferior e as colunas do nível superior. Um elemento desta matriz com valor um relaciona uma transformação de agregação com uma de menor agregação. Uma linha desta coluna, correspondente a uma transformação agregada, pode ter vários elementos com valor um, respeitantes às transformações do nível inferior que estão agregadas no nível superior. Em todas as linhas da matriz só pode

existir um elemento com valor um. Isto significa que uma transformação só pode estar relacionada com uma única transformação agregada. Esta matriz não tem nenhuma coluna completamente preenchida por zeros, porque todas as transformações de um nível superior de agregação têm relação com pelo menos uma das transformações do nível inferior.

Um exemplo de matrizes  $\mathbf{M}_v^{v+1}$  e  $\mathbf{N}_v^{v+1}$  relativas ao sistema representado na Figura 20 é apresentado na Figura 23. A obtenção destas matrizes é mostrada mais à frente nesta secção.

$$M_0^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, N_0^1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figura 23: Exemplos de matrizes de transição entre níveis de agregação.

As quantidades de objecto consumido e fornecido por cada transformação em níveis consecutivos de agregação são relacionadas pelas duas matrizes referidas. Estas quantidades dependem das quantidades consideradas na Rede de Petri, e em particular para o modelo MSDP dependem da estrutura do produto. A definição destes relacionamentos, apresentada por Lecompte, Deschamps and Bourrières (2000) é formulada na Equação 7.

$$\exists \mathbf{M}_v^{v+1}, \mathbf{N}_v^{v+1} : \begin{cases} \mathbf{C}^{v+1} = \mathbf{M}_v^{v+1} \times \mathbf{C}^v \times \mathbf{N}_v^{v+1} \\ \mathbf{W}^v = \mathbf{N}_v^{v+1} \times \mathbf{W}^{v+1} \\ \delta \mathbf{S}^{v+1} = \mathbf{M}_v^{v+1} \times \delta \mathbf{S}^v \end{cases} \quad \text{Equação 7}$$

Lecompte, Deschamps and Bourrières (2000) apresentam também a formulação de generalização do relacionamento entre quaisquer níveis  $v$  e  $v+h$ , representada pela Equação 8.

$$\forall v, h \in \mathbb{N} : \begin{cases} \mathbf{C}^{v+h} = \mathbf{M}_v^{v+h} \times \mathbf{C}^v \times \mathbf{N}_v^{v+h} \\ \mathbf{W}^v = \mathbf{N}_v^{v+h} \times \mathbf{W}^{v+h} \\ \delta \mathbf{S}^{v+h} = \mathbf{M}_v^{v+h} \times \delta \mathbf{S}^v \end{cases} \quad \text{Equação 8}$$

Sendo  $\mathbf{L}$  e  $\mathbf{R}$  os produtos de matrizes à esquerda e direita respectivamente, podemos representar as matrizes de transição entre níveis da seguinte forma:

$$\begin{aligned} \bullet \quad \mathbf{M}_v^{v+h} &= \prod_{k=v}^{v+h-1} \mathbf{M}_k^{k+1} = \mathbf{M}_{v+h-1}^{v+h} \times \dots \times \mathbf{M}_v^{v+1}; \\ \bullet \quad \mathbf{N}_v^{v+h} &= \prod_{k=v}^{v+h-1} \mathbf{N}_k^{k+1} = \mathbf{N}_v^{v+1} \times \dots \times \mathbf{N}_{v+h-1}^{v+h}. \end{aligned}$$

### ***i. Configuração Genérica***

Projectar o sistema distribuído de produção que satisfaz uma encomenda, implica, entre outras acções, definir as transformações de mudança de estado dos elementos da estrutura do produto encomendado e,

aplicando a visão fractal do sistema, representar as transformações agregadas. Estas duas acções, associadas à representação apresentada com redes de Petri, permitem construir uma configuração genérica do sistema distribuído de produção. Esta configuração denomina-se genérica porque permite construir as Redes de Petri que representam os processos de transformação directamente associados à estrutura do produto, não incluindo a selecção nem afectação de recursos físicos que formam as células autónomas que executam as transformações, isto é, não incluindo a criação de uma configuração específica para o Sistema Distribuído de Produção. Em seguida apresenta-se o algoritmo, desenvolvido neste trabalho, que permite construir essa configuração genérica do Sistema Distribuído de Produção no modelo MSDP.

---

**ALGORITMO DE CONFIGURAÇÃO GENÉRICA**

1. Construir Rede de Petri referente à estrutura do produto. Nível mais baixo de agregação do Sistema Distribuído de Produção.
  2. Construir Rede de Petri referente ao nível de maior agregação. Este nível terá apenas uma transformação com saída referente ao produto final e entradas referentes a todos os produtos não transformados no sistema, isto é, produtos adquiridos.
  3. *ATRIBUIR* Rede de Petri do Sistema = Rede de Petri obtida em 2.
  4. *ENQUANTO* Rede de Petri do Sistema  $\neq$  Rede de Petri obtida em 1.
    - 4.1. *PARA TODAS* as Transformações Agregadas
      - 4.1.1. Expandir a Transformação
      - 4.1.2. Construir Rede de Petri
      - 4.1.3. *ATRIBUIR* Rede de Petri do Sistema = Rede de Petri obtida em 4.1.2
- 

Expandir uma transformação que produza um produto  $P_{\#}$  corresponde a desagregar essa transformação, nas transformações existentes na estrutura do produto que entregam os componentes directamente utilizados para processamento e entrega do produto. Neste processo geram-se matrizes **M** e **N** de transição entre níveis, cujos elementos têm valores um, ou zero, conforme exista, ou não, relacionamento entre os objectos (**M**) e entre as transformações (**N**) nos níveis de agregação. Este processo é exemplificado mais à frente, na secção III - 2.1.4.iii, e ilustrado na desagregação entre o nível representado pela Figura 26 e o nível representado pela Figura 27.

Conforme definição da matriz **M**, apresentada anteriormente, as colunas da matriz, referentes aos objectos embebidos na agregação do nível imediatamente acima, são totalmente preenchidas por zeros. Neste caso estes objectos não existem na Rede de Petri do nível com agregação. Os objectos existentes nos dois níveis são relacionados nesta matriz pelos elementos com valor um. Os elementos da matriz **N** que relacionam transformações não agregadas em níveis de agregação consecutivos têm valor um. As colunas da matriz **N**, referentes a transformações que agregam várias transformações do nível abaixo, têm elementos com valor um em todas as linhas referentes a estas transformações. Estas matrizes são exemplificadas mais à frente, na secção III - 2.1.4.iii, página 72.

## **ii. Relação entre Quantidades dos Níveis de Agregação**

No nível de menor agregação do modelo do Sistema Distribuído de Produção desenvolvido nesta dissertação, as quantidades de utilização, isto é, de consumo e fabrico, dos objectos pelas transformações são iguais às quantidades representadas na estrutura do produto. Estas quantidades estão representadas na Rede de Petri pelos pesos dos arcos. Quando uma transformação é executada uma vez pode produzir uma

quantidade diferente, i.e. maior ou menor, do que a quantidade consumida pela execução, também uma vez, da transformação seguinte. No caso da quantidade produzida ser maior, isto implica que um excesso de produção se vai adicionar às existências desse objecto, isto é, à quantidade armazenada desse objecto. No caso de ser menor, a quantidade em falta terá de ser obtida da quantidade armazenada existente. Como consequência haverá uma diminuição das existências desse objecto.

A execução da mesma quantidade de trabalho para obtenção de um determinado produto, em dois níveis diferentes, tem que utilizar a mesma quantidade de matérias primas. A primeira equação, do sistema representado na Equação 7 não permite, em determinados casos, calcular a mesma quantidade de matérias primas em dois níveis diferentes, como se verá em seguida.

No nível de agregação imediatamente acima ao de menor agregação, existe pelo menos uma transformação que agrega duas ou mais transformações (Equação 3) e existe pelo menos um objecto embebido (Equação 6). O facto de existir um objecto embebido significa que ele não é considerado para efeitos de cálculo nesse nível e por essa razão, não existem dados sobre o armazenamento desse objecto, que são agregados nesse nível. Neste processo de agregação, o consumo de matérias primas para produção deste objecto embebido tem que se reflectir no nível de agregação superior.

A agregação de dados de objectos embebidos, que são consumidos e fabricados em quantidades diferentes pelas transformações agregadas, coloca problemas de cálculo das quantidades de utilização de objectos no nível agregado. Nestas condições, as quantidades de utilização (pesos dos arcos das Redes de Petri) do nível agregado não são iguais às do nível não agregado, e no modelo do Sistema Distribuído de Produção não são iguais às da estrutura do produto. Não se encontram referências a este problema nem por Lecompte, Deschamps and Bourrières (2000), nem noutros trabalhos acerca deste modelo de formulação algébrica: Zolghadri, Lecompte and Bourrières (1998), Bourrières (1998) e Lecompte and Bourrières (1999).

Para melhor compreensão da forma de cálculo das quantidades de utilização de objectos, consideram-se os casos simples de agregação indicadas na Figura 24. Estes casos são, genericamente, os que se podem verificar na agregação de níveis resultantes da aplicação do algoritmo de Configuração Genérica, do modelo do Sistema Distribuído de Produção desenvolvido nesta dissertação.

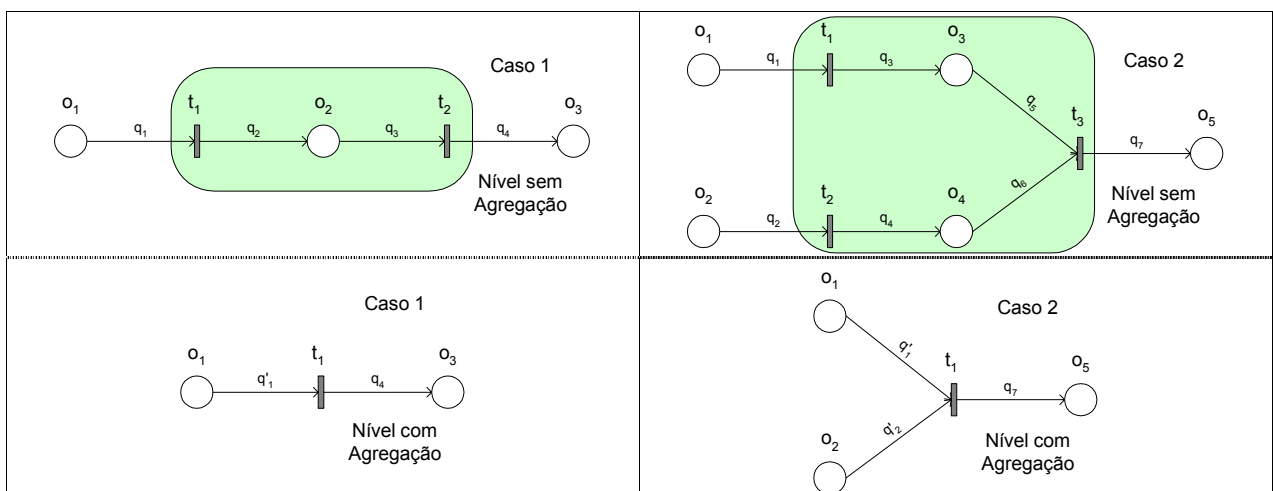


Figura 24: Exemplos de Relacionamento entre Quantidades dos Níveis de Agregação.

Para facilitar a leitura das Redes de Petri da Figura 24, colocam-se as transformações e objectos agregados, sob um fundo escuro, e mantém-se a nomenclatura dos objectos  $o_i$  e das quantidades utilizadas  $q_Q$  entre níveis. O Caso 2 mais genérico consideraria um número indeterminado de transformações em paralelo. Sem desprezar este facto, utiliza-se este caso para simplificar a definição do relacionamento entre quantidades de níveis de agregação sucessivos.

Considere-se, por exemplo, o Caso 1 com as seguintes quantidades de utilização:  $q_Q=1$ , com  $Q=1,2,4$  e  $q_3=2$ . A requisição de uma unidade de  $o_3$ , requer uma execução de  $t_2$ , e a execução de  $t_1$ , duas vezes, se não houver nenhuma quantidade armazenada de  $o_2$ . Dessa forma, o consumo do objecto  $o_1$  é igual a duas unidades. A mesma ordem de produção de  $o_3$ , aplicada no nível com agregação, tem que resultar na determinação da mesma necessidade de consumo de duas unidades de  $o_1$ . Assim sendo, quando existe uma agregação similar à do Caso 1 a determinação da quantidade de objecto  $o_1$  consumido resulta de:

$$q'_1 = q_1 \cdot (q_3/q_2)$$

Considere-se, por exemplo, o Caso 2 com nenhuma quantidade armazenada de  $o_3$  e  $o_4$ , e com as seguintes quantidades de utilização:  $q_Q=1$ , com  $Q=1,2,3,4,7$ ,  $q_5=2$  e  $q_6=3$ . A ordem de produção de uma unidade de  $o_5$  requer uma execução de  $t_3$ , que consome duas unidades de  $o_3$  e três unidades de  $o_4$ . Para se obter estas quantidades de  $o_3$  e de  $o_4$  é necessário executar  $t_1$  duas vezes e  $t_2$  três vezes. Dessa forma, o consumo do objecto  $o_1$  é igual a duas unidades e de  $o_2$  é igual a três unidades. A mesma ordem de produção de  $o_5$ , aplicada no nível com agregação, tem que resultar na determinação da mesma necessidade de consumo para  $o_1$  e  $o_2$ . Assim sendo, quando existe uma agregação similar à do Caso 2, a determinação das quantidades consumidas dos objectos  $o_1$  e  $o_2$ , resulta da aplicação de equações similares às duas seguintes:

$$\begin{aligned} q'_1 &= q_1 \cdot (q_5/q_3) \\ q'_2 &= q_2 \cdot (q_6/q_4) \end{aligned}$$

Considerando a situação geral de agregação, ilustrada pela Figura 25, qualquer objecto  $o_i$ , embebido numa agregação que é fabricado por uma transformação  $t_{jj}$  na quantidade  $q_i$  e consumido pela transformação  $t_j$  na quantidade  $q_i^*$ , é possível definir um factor de conversão  $fc_{jj,j,i}$  associado ao objecto  $o_i$ , entre as transformações  $t_{jj}$  e  $t_j$  de acordo com a Equação 9. Este factor de conversão permite relacionar as quantidades de entrada da transformação ( $jj$ ) de produção do objecto embebido ( $i$ ) com as de saída das transformações ( $j$ ) que consomem este objecto.

$$fc_{jj,j,i} = q_i^*/q_i \quad \text{Equação 9}$$

Aplicando esta equação ao Caso 1 anterior, confirma-se o factor de conversão obtido:

$$fc_{jj,j,i} = q_i^*/q_i \Leftrightarrow fc_{1,2,2} = q_{o_2}^*/q_{o_2} \Leftrightarrow fc_{1,2,2} = q_3/q_2$$

Aplicando esta equação ao Caso 2 anterior, confirmam-se os dois factores de conversão obtidos, relacionados com as transformações  $t_1$  e  $t_2$ , relativas à  $t_3$ :

$$\begin{aligned} fc_{jj,j,i} &= q_i^*/q_i \Leftrightarrow fc_{1,3,3} = q_{o_3}^*/q_{o_3} \Leftrightarrow fc_{1,3,3} = q_5/q_3 \\ fc_{jj,j,i} &= q_i^*/q_i \Leftrightarrow fc_{2,3,4} = q_{o_4}^*/q_{o_4} \Leftrightarrow fc_{2,3,4} = q_6/q_4 \end{aligned}$$



Considere-se, de acordo com a Figura 25, o objecto  $o_{ii}$  consumido por uma transformação  $t_{jj}$  de um nível sem agregação nas quantidades  $q_{ii}$ . Esta transformação  $t_{jj}$  produz um objecto  $o_i$  que é embebido pela agregação de níveis. Nesta situação, a determinação das quantidades consumidas  $q'_{ii}$  do objecto  $o_{ii}$  num nível de agregação imediatamente acima, resulta da aplicação da Equação 10. A aplicação desta equação aos Casos 1 e 2 anteriores confirmam os resultados obtidos.

$$q'_{ii} = q_{ii} \cdot fc_{jj,j,i} \quad \text{Equação 10}$$

No nível de menor agregação, a Rede de Petri representa as quantidades de utilização de todas as transformações, considerando que estas se executam uma vez. Dependendo da rede e dos pesos dos arcos, isto é, das quantidades de utilização, pode não haver uma solução inteira para o consumo de um determinado objecto da rede, resultante da execução, uma vez, da transformação de produção do produto final. Isto significa que, para se obter quantidades inteiras, é necessário executar transformações em quantidades de trabalho adequadas e que se considera a possibilidade de aumento ou diminuição das existências de alguns objectos. Este aspecto, reflecte-se na conversão de quantidades de utilização de objectos, que dessa forma, pode resultar em valores não inteiros (ver Equação 9) para essas quantidades em determinados níveis. Estes valores não inteiros podem ser manuseados pelo sistema algébrico, mas não têm interpretação física consistente, porque reflectem a execução de transformações em quantidades de trabalho não inteiras. Por este facto podem reduzir-se todas as quantidades, utilizadas pela transformação agregada, para números inteiros por arredondamento superior. Este arredondamento conduz à determinação de quantidades de utilização, no nível superior, de acordo com a execução das transformações, do nível inferior, em quantidades de trabalho em unidades inteiras.

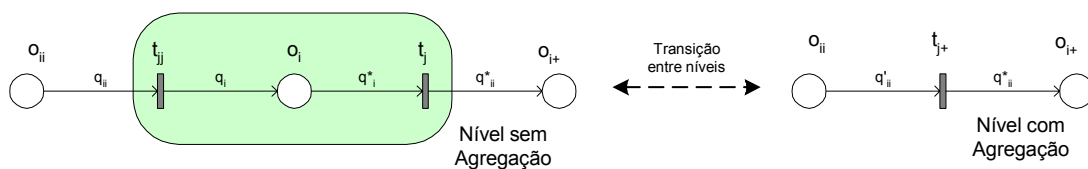


Figura 25: Exemplo de agregação.

Verificando-se a situação de agregação de um objecto que é consumido e fabricado em quantidades diferentes, descrita nesta secção, a primeira equação do sistema de equações definido por Lecompte, Deschamps and Bourrières (2000) e apresentado na Equação 7, não é válida. Nesta situação, esta equação não é válida porque atribui quantidades de utilização de objectos sempre iguais a outras quantidades existentes no nível de agregação inferior. Como se verificou, a determinação das quantidades utilizadas pelas transformações em níveis diferentes de agregação depende, nesta situação, do factor de conversão. A transformação agregada, em que existe um objecto embebido nas condições descritas, tem quantidades de consumo diferentes das quantidades de consumo da transformação que produz aquele objecto. É assim possível construir uma matriz  $\mathbf{N}^p$ , que relaciona as quantidades de consumo da transformação (linhas da matriz) de produção do objecto embebido, com as quantidades de consumo da transformação agregada (coluna da matriz). A coluna da matriz relativa à transformação agregada tem valores iguais aos factores de conversão  $fc_{jj,j,i}$  nas linhas relativas às transformações  $t_{jj}$  de produção do objecto embebido do nível sem agregação.

Esta matriz  $\mathbf{N}'$  pode resultar do produto de todos os elementos da matriz pelo valor um (1), excepto aqueles que são iguais aos  $fc_{jj,i}$ . Considerando, por exemplo,  $jj=2$  para o único factor de conversão de uma transformação que é agregada na transformação três do nível superior, a matriz  $\mathbf{Fc}_v^{v+1}$  é igual a:

$$\mathbf{Fc}_v^{v+1} =_{jj=2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & fc_{jj,i} & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Introduzindo a operação “o”, de multiplicação dos elementos de duas matrizes com as mesmas dimensões, a matriz  $\mathbf{N}'$  é definida pela aplicação da seguinte equação, de multiplicação de todos os elementos das matrizes  $\mathbf{Fc}$  e  $\mathbf{N}$ :

$$\mathbf{N}'^{v+1} = \mathbf{Fc}_v^{v+1} \circ \mathbf{N}_v^{v+1} \Leftrightarrow N_{v \ j1,j2}^{v+1} = Fc_{v \ j1,j2}^{v+1} \cdot N_{v \ j1,j2}^{v+1} \quad \text{Equação 11}$$

O sistema de equações da Equação 7, transforma-se no sistema apresentado na Equação 12.

$$\exists \mathbf{M}_v^{v+1}, \mathbf{N}_v^{v+1} : \begin{cases} \mathbf{C}^{v+1} = \mathbf{M}_v^{v+1} \times \mathbf{C}^v \times (\mathbf{Fc}_v^{v+1} \circ \mathbf{N}_v^{v+1}) \\ \mathbf{W}^v = \mathbf{N}_v^{v+1} \times \mathbf{W}^{v+1} \\ \delta \mathbf{S}^{v+1} = \mathbf{M}_v^{v+1} \times \delta \mathbf{S}^v \end{cases} \Leftrightarrow \begin{cases} \mathbf{C}^{v+1} = \mathbf{M}_v^{v+1} \times \mathbf{C}^v \times \mathbf{N}_v^{v+1} \\ \mathbf{W}^v = \mathbf{N}_v^{v+1} \times \mathbf{W}^{v+1} \\ \delta \mathbf{S}^{v+1} = \mathbf{M}_v^{v+1} \times \delta \mathbf{S}^v \end{cases} \quad \text{Equação 12}$$

Estas equações permitem relacionar os níveis de agregação, isto é, relacionar os objectos, as transformações e as quantidades de objecto utilizadas pelas transformações de dois níveis de agregação consecutivos.

### **iii. Exemplo**

O significado dos níveis de agregação e do relacionamento entre objectos e entre transformações pode ser melhor apreendido com um exemplo. Assim sendo, com base no exemplo utilizado neste capítulo, pode verificar-se que a Rede de Petri da Figura 20 corresponde ao nível mais baixo de agregação, denominado Nível 0. Apresentam-se em seguida os modelos dos vários níveis de agregação, entre o Nível 0 e o Nível Superior  $l$ , e o respectivo relacionamento. Neste exemplo não há necessidade de conversão de quantidades de utilização das transformações, conforme especificação apresentada na secção anterior, porque todas as quantidades da estrutura do produto são idênticas. A exemplificação da aplicação desta especificação é efectuada num exemplo apresentado na parte final deste capítulo.

#### **iii.i Níveis de Agregação**

As transformações de um determinado nível agregam transformações de níveis inferiores, podendo nesses casos haver objectos embebidos, que só se tornam visíveis nos níveis de agregação inferiores. Para simplificação da compreensão das matrizes  $\mathbf{C}^v$  que indicam, para o nível  $v$  de agregação, o consumo e produção de objectos pelas transformações, utilizam-se índices fora dos parêntesis rectos.

(a) Nível superior:  $v=l=2$

O passo 1 do Algoritmo de Configuração Genérica foi realizado na secção III - 2.1.2.i. O passo 2 deste mesmo algoritmo resulta na criação de um nível superior de agregação total, com todas as transformações representadas por uma única transformação de agregação (Figura 26). As entradas desta transformação são todos os objectos de entrada do sistema, que correspondem aos objectos obtidos externamente ao sistema de produção.

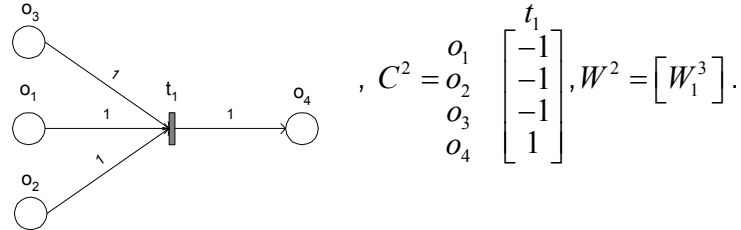


Figura 26: Rede de Petri referente ao nível de agregação  $v=l=2$ .

A transformação  $t_1$  da Figura 26 agrega todas transformações do sistema de produção, i.e. as transformações  $t_1$  a  $t_5$  da Figura 28. Os objectos  $o_1$ ,  $o_2$ ,  $o_3$  e  $o_4$  da Figura 26 coincidem, respectivamente, com os objectos  $o_1$ ,  $o_2$ ,  $o_3$  e  $o_8$  da Figura 28 e com os produtos  $P_{1211}$ ,  $P_{1221}$ ,  $P_{111}$  e  $P_1$  da Figura 19. Como esta Rede de Petri não coincide com a do passo 1 do Algoritmo de Configuração Genérica (secção III - 2.1.4.i) é necessário executar o passo 4.1.

(b) Nível 1:  $v=l-1=1$

Pela execução do passo 4.1, a única transformação agregada da Rede de Petri da Figura 26 tem que ser expandida. Esta transformação entrega o objecto correspondente ao produto  $P_1$ , que resulta do processamento imediato dos produtos  $P_{11}$  e  $P_{12}$ , pelo que aquela transformação se desagrega nas duas transformações que entregam estes dois produtos. Este processo de desagregação torna-se visível na Rede de Petri da Figura 27, estando realçado pela aplicação de um fundo mais escuro dos objectos e transformações agregadas no nível superior.

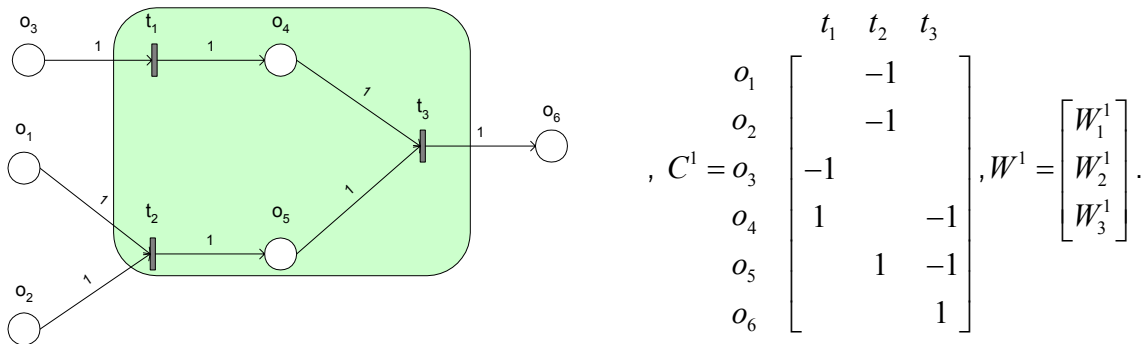


Figura 27: Rede de Petri referente ao nível de agregação  $v=l-1=1$ .

A transformação  $t_1$  da Figura 27 corresponde à transformação  $t_3$  do sistema de produção (Figura 28),  $t_2$  agrega as transformações  $t_1$ ,  $t_2$  e  $t_4$ , e  $t_3$  corresponde à transformação  $t_5$ . Os objectos  $o_1$ ,  $o_2$ ,  $o_3$ ,  $o_4$ ,  $o_5$  e  $o_6$  da Figura 27 coincidem, respectivamente, com os objectos  $o_1$ ,  $o_2$ ,  $o_3$ ,  $o_6$ ,  $o_7$  e  $o_8$  da Figura 28. Como esta rede

não coincide com a rede do Nível 0, é necessário proceder à desagregação das transformações agregadas, que neste caso corresponde apenas a  $t_2$ .

(c) Nível inferior:  $v=0$

Pela execução do passo 4.1, a única transformação agregada da Rede de Petri da Figura 27 tem que ser expandida. Esta transformação entrega o objecto  $o_5$ , correspondente ao produto  $P_{12}$ , que resulta do processamento imediato dos produtos  $P_{121}$  e  $P_{122}$ , pelo que aquela transformação se desagrega nas duas transformações que entregam estes dois produtos. Este processo de desagregação torna-se visível na Rede de Petri da Figura 28, estando realçado pela aplicação de um fundo mais escuro em volta dos objectos e transformações agregadas no Nível Superior.

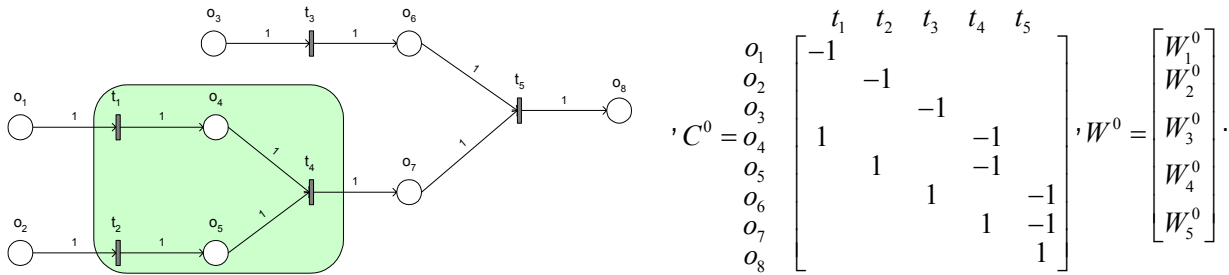


Figura 28: Rede de Petri referente ao nível de agregação  $v=0$ .

Neste nível não existe agregação, nem de transformações, nem de objectos, correspondendo ao nível elementar do sistema de produção modelado em secções anteriores.

iii.ii Modelação da Transição entre Níveis de Agregação

Pretende-se, para o exemplo em estudo, determinar as matrizes de transição entre níveis  $\mathbf{M}_v^{v+1}$  e  $\mathbf{N}_v^{v+1}$ , com  $v=0..l-1$ , a partir da Equação 7. É de realçar que estas matrizes se deduzem a partir das redes de Petri dos dois níveis em análise, verificando o mapeamento entre objectos e entre transformações nesses níveis.

(a) Transição entre o Nível Superior  $v+1=l=2$  e o Nível  $v=l-1=1$

A matriz  $\mathbf{M}_1^2$ , deduzida a partir das redes de Petri dos níveis 1 e 2, mapeia os objectos desses níveis, permitindo relacionar a quantidade de objectos em cada um dos níveis. Este relacionamento resulta da terceira equação do sistema de equações apresentado na Equação 7:

$$\delta \mathbf{S}^{v+1} = \mathbf{M}_v^{v+1} \times \delta \mathbf{S}^v \Leftrightarrow \delta \mathbf{S}^2 = \mathbf{M}_1^2 \times \delta \mathbf{S}^1 \Leftrightarrow [\delta S_1^2, \dots, \delta S_4^2]^T = \mathbf{M}_1^2 \times [\delta S_1^1, \dots, \delta S_6^1]^T \Leftrightarrow$$

$$\begin{bmatrix} \delta S_1^2 \\ \delta S_2^2 \\ \delta S_3^2 \\ \delta S_4^2 \end{bmatrix} = \begin{bmatrix} M_{1,1,1}^2 & \dots & M_{1,1,6}^2 \\ & & \\ & & \\ M_{1,4,1}^2 & \dots & M_{1,4,6}^2 \end{bmatrix} \times \begin{bmatrix} \delta S_1^1 \\ \dots \\ \delta S_6^1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} \delta S_1^2 \\ \delta S_2^2 \\ \delta S_3^2 \\ \delta S_4^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \delta S_1^1 \\ \dots \\ \delta S_6^1 \end{bmatrix}$$

A matriz  $\mathbf{N}_1^2$  é deduzida a partir das redes de Petri dos níveis 1 e 2, permitindo relacionar as transformações desses níveis, isto é, relacionar as quantidades de objecto consumido e produzido nas transformações dos dois níveis. Esta matriz tem valores unitários porque todas as quantidades  $q_Q$ , da Rede de

Petri do Nível 1, são iguais a um. O relacionamento entre quantidades de objectos resulta da primeira equação do sistema apresentado na Equação 7:

$$\mathbf{C}^{v+1} = \mathbf{M}_v^{v+1} \times \mathbf{C}^v \times \mathbf{N}_v^{v+1} \Leftrightarrow \mathbf{C}^2 = \mathbf{M}_1^2 \times \mathbf{C}^1 \times \mathbf{N}_1^2 \Leftrightarrow$$

$$\begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & -1 & 0 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} N_{1,1}^2 \\ N_{1,2}^2 \\ N_{1,2}^2 \end{bmatrix} \Leftrightarrow \begin{bmatrix} N_{1,1}^2 \\ N_{1,2}^2 \\ N_{1,2}^2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Esta matriz confirma o mapeamento das três transformações do Nível 1 para a única transformação do Nível 2, podendo verificar-se na resolução da segunda equação do sistema de equações referido que é necessário executar a mesma quantidade de trabalho, tanto nas três transformações do Nível 1 como na única transformação do Nível 2.

$$\mathbf{W}^v = \mathbf{N}_v^{v+1} \times \mathbf{W}^{v+1} \Leftrightarrow \mathbf{W}^1 = \mathbf{N}_1^2 \times \mathbf{W}^2 \Leftrightarrow \begin{bmatrix} W_{1,1}^1 \\ W_{1,2}^1 \\ W_{1,3}^1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \times [W_{1,1}^2] \Leftrightarrow \begin{bmatrix} W_{1,1}^1 \\ W_{1,2}^1 \\ W_{1,3}^1 \end{bmatrix} = \begin{bmatrix} W_{1,1}^2 \\ W_{1,2}^2 \\ W_{1,1}^2 \end{bmatrix}$$

(b) Transição entre o Nível v+1=1 e o Nível Inferior v=0

Seguindo o mesmo raciocínio deduzem-se as seguintes matrizes:

$$M_0^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, N_0^1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Nestas matrizes é possível verificar que a transformação  $t_2$  do Nível 1 agrega as transformações  $t_1$ ,  $t_2$  e  $t_4$  do Nível 0. Isto significa que no Nível 1 se pode agregar informação respeitante à célula que tem a responsabilidade de entrega do produto  $P_{12}$ . Esta agregação pode simplificar o processo de selecção de recursos envolvidos naquela célula.

**2.1.5. Restrições de Precedência**

Um dos aspectos a modelar em problemas de programação é o das precedências entre processos. Deste ponto de vista é necessário formular a dependência entre os tempos de entrega e os tempos de início das diversas transformações. Lecompte, Deschamps and Bourrières (2000) modelaram a precedência entre transformações através da formulação apresentada na Equação 14, deduzida a partir da matriz  $\mathbf{P}$  representada na Equação 13. Se nesta matriz algum elemento  $P_{ij}$  for diferente de zero então isto significa que a transformação  $j$  precede a transformação  $i$ .

$$\mathbf{P} = \mathbf{Pre}^T \times \mathbf{Post} \quad \text{Equação 13}$$

Com o intuito de calcular os instantes de início e conclusão de cada transformação, definem-se as seguintes matrizes:

- $\mathbf{T}^+ = [T^+_{1,\dots}, T^+_{j,\dots}, T^+_{n}]^T$  - vector dos tempos de conclusão de cada transformação.
- $\mathbf{T}^- = [T^-_{1,\dots}, T^-_{j,\dots}, T^-_{n}]^T$  - vector dos tempos de início de cada transformação.
- $\mathbf{P}^* : P^*_{ij} = \text{Min}\{1, P_{ij}\}$  - matriz binária, em que os elementos com valor um indicam que a transformação  $j$  precede a transformação  $i$ .

A determinação do instante de início de cada transformação em função das precedências estabelecidas é formulada pela operação definida na Equação 14, em que se define o instante de início de uma transformação como sendo igual ao instante de conclusão de todas as transformação precedentes.

$$\mathbf{T}^- = \mathbf{P}^* \otimes \mathbf{T}^+ \Leftrightarrow T^-_i = \text{Max}_j(P^*_{ij} \cdot T^+_j) \quad \text{com } i, j \in \{1, \dots, n\} \quad \text{Equação 14}$$

Assume-se nesta formulação que, para se iniciar o processamento de uma transformação, é necessário concluir a execução de todas as transformações precedentes. Esta formulação não considera a possibilidade de transferência, para a transformação seguinte, de objectos que resultem de uma parte do processamento já terminado, não permitindo modelar a sobreposição de processamento de lotes, i.e. “batch overlapping”.

### i. Exemplo

Aplicando esta forma de modelação ao Nível Inferior do exemplo em estudo obtemos a matriz  $\mathbf{P}$ , correspondente à rede de Petri da Figura 28. A interpretação desta matriz permite concluir que as transformações  $t_1$  e  $t_2$  precedem  $t_4$  e as transformações  $t_3$  e  $t_4$  precedem  $t_5$ . Neste exemplo  $\mathbf{P}$  é equivalente a  $\mathbf{P}^*$  porque todos os pesos dos arcos são unitários. Nas seguintes matrizes  $\mathbf{P}$  e  $\mathbf{P}^*$ , os elementos  $t_j$  são utilizados para auxiliarem a compreensão da matriz.

$$\mathbf{P} = \text{Pre}^T \cdot \text{Post} = \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{matrix} \begin{matrix} t_1 & t_2 & t_3 & t_4 & t_5 \\ \left[ \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{array} \right] & = & \mathbf{P}^* \end{matrix}$$

O vector dos tempos de início  $\mathbf{T}^-$  de cada transformação resulta da aplicação da Equação 14. Esta equação permite determinar o instante de início de todas as transformações em função das precedências de processamento. Neste caso o instante de início do processamento das três primeiras transformações não depende de nenhuma outra pelo que coincide com o instante zero a partir do instante que se define como de início de todo o processamento. Só se pode iniciar a execução da transformação  $t_4$  após o término das transformações  $t_1$  e  $t_2$ . A execução da transformação  $t_5$  só se pode iniciar depois das transformações  $t_3$  e  $t_4$ .

$$\mathbf{T}^- = \mathbf{P}^* \otimes \mathbf{T}^+ \Leftrightarrow T^-_i = \text{Max}_j(P^*_{ij} \cdot T^+_j) \Leftrightarrow \mathbf{T}^- = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \text{Max}(T^+_1, T^+_2) \\ \text{Max}(T^+_3, T^+_4) \end{bmatrix}$$

## 2.2 Modelação de Recursos

No processo de modelação de um sistema produtivo, é necessário incluir os recursos produtivos e as respectivas capacidades produtivas. Lecompte, Deschamps and Bourrières (2000) utilizam as seguintes variáveis como base da formulação das restrições de utilização dos recursos:

- $w_{jk}$  - quantidade de trabalho, não negativa, do tipo  $j$  executado no recurso  $k$ .
- $\alpha_{jk}$  - quantidade de trabalho do tipo  $j$  realizável pelo recurso  $k$  no horizonte de tempo  $[t, t+H]$ .

A variável  $w_{jk}$  representa o número de vezes que o recurso  $k$  executa a transformação  $j$ , sendo o total da quantidade de trabalho executada na transformação, dado pelo somatório destas variáveis em  $j$ , e o total de quantidade de trabalho executada pelo recurso, dado pelo somatório destas variáveis em  $k$ . A variável  $\alpha_{jk}$  representa a capacidade livre do recurso  $k$  para a execução da transformação  $j$ , i.e. número de vezes que pode executar aquela transformação no horizonte  $H$ .

### 2.2.1. Restrições de Procura

Restrições de procura correspondem às restrições relacionadas com a satisfação das quantidades encomendadas, sendo referidas por Lecompte, Deschamps and Bourrières (2000) como Restrições de Cobertura<sup>33</sup>. A quantidade de trabalho do tipo  $j$  executada nos recursos tem que ser igual à quantidade total de trabalho requerido pela transformação  $j$ .

$$\sum_{k=1}^r w_{jk} = W_j, j = 1, \dots, n \Leftrightarrow \begin{cases} w_{11} + \dots + w_{1k} + \dots + w_{1r} = W_1 \\ \dots \\ w_{n1} + \dots + w_{nk} + \dots + w_{nr} = W_n \end{cases} \quad \text{Equação 15}$$

### 2.2.2. Restrições de Disponibilidade

A quantidade de trabalho do tipo  $j$  realizável pelo recurso  $k$  no horizonte de tempo  $[t, t+H]$ ,  $\alpha_{jk}$  representa a capacidade disponível, medida em quantidade de trabalho, do recurso  $k$  relativa ao trabalho do tipo  $j$ , isto é, relativa à transformação  $j$ .

A quantidade total de trabalho  $w_{jk}$  realizado por um recurso, num horizonte  $H$ , não pode ser superior à sua capacidade total no mesmo horizonte. Assim sendo, a soma dos quocientes entre o trabalho realizado em cada transformação e a capacidade disponível não pode ser, em qualquer recurso, superior a uma unidade.

$$\sum_{j=1}^n \frac{w_{jk}}{\alpha_{jk}} \leq 1, k = 1, \dots, r \Leftrightarrow \begin{cases} w_{11}/\alpha_{11} + \dots + w_{j1}/\alpha_{j1} + \dots + w_{n1}/\alpha_{n1} \leq 1 \\ \dots \\ w_{1r}/\alpha_{1r} + \dots + w_{jr}/\alpha_{jr} + \dots + w_{nr}/\alpha_{nr} \leq 1 \end{cases} \quad \text{Equação 16}$$

Para se relacionar capacidade disponível, medida em quantidade de trabalho, do recurso com o seu tempo livre, introduzem-se, no modelo do Sistema Distribuído de Produção desenvolvido nesta dissertação, os seguintes parâmetros não negativos:

<sup>33</sup> "Covering".

- $c_{jk}$  – tempo de processamento unitário, i.e. tempo necessário para que o recurso  $k$  execute uma unidade de quantidade de trabalho do tipo  $j$ , ou por outras palavras, tempo para o recurso  $k$  executar uma vez a transformação  $j$ .
- $Tf_k$  – tempo livre do recurso  $k$  para processamento de qualquer transformação num horizonte de tempo  $H$ .

A relação entre o parâmetro  $\alpha_{jk}$  e  $c_{jk}$  é dado pela Equação 17, que expressa a capacidade disponível do recurso  $k$  para executar a transformação  $j$ , correspondente ao número de vezes que o recurso pode executar aquela transformação no seu tempo livre. Esta formulação não considera possíveis restrições impostas pela partição do tempo livre do recurso.

$$\alpha_{jk} = \text{int}(Tf_k / c_{jk}) \quad \text{Equação 17}$$

A restrição de disponibilidade pode, então, ser representada pela Equação 18, equivalente à Equação 16:

$$\sum_{j=1}^n c_{jk} \cdot w_{jk} \leq Tf_k, k = 1, \dots, r \quad \text{Equação 18}$$

Neste modelo MSDP define-se a convenção formulada na Equação 19:

$$Tf_k = 0 \Rightarrow \forall j \in \{1, \dots, n\} : (\alpha_{jk} = 0) \wedge (w_{jk} = 0) \wedge (w_{jk} / \alpha_{jk} = 0) \text{ com } k \in \{1, \dots, r\} \quad \text{Equação 19}$$

Segundo esta definição, se um recurso não tiver nenhum tempo livre, então o número de vezes  $\alpha_{jk}$  que pode executar qualquer transformação é igual a zero, a quantidade de trabalho  $w_{jk}$  que lhe é atribuída é nula, e para manter a validade da equação Equação 18, o quociente entre a quantidade de trabalho e a capacidade disponível também é igual a zero.

### 2.2.3. Afectação de Recursos

Introduzindo os vectores  $\Theta$  e  $\Upsilon$  para representar, respectivamente, todas as variáveis de quantidades de trabalho a atribuir a cada recurso relativas a cada transformação, e o vector da capacidade que cada recurso dispõe após afectação de recursos, torna-se possível definir o seguinte sistema de equações, em função das restrições de procura (Equação 15) e disponibilidade (Equação 16), apresentado em Lecompte, Deschamps and Bourrières (2000):

$$\begin{array}{l} \text{Restrições de Procura} \\ \text{Restrições de Disponibilidade} \end{array} \quad \begin{cases} A \cdot \Theta = W \\ \gamma \cdot \Theta \leq 1 \end{cases} \quad \xrightarrow{\text{transformado em}} \quad \begin{cases} A \cdot \Theta = W \\ \gamma \cdot \Theta + \Upsilon = 1 \end{cases}$$

- $\Theta = [w_{11}, \dots, w_{1k}, \dots, w_{1r}, \dots, w_{j1}, \dots, w_{jk}, \dots, w_{jr}, \dots, w_{n1}, \dots, w_{nk}, \dots, w_{nr}]^T$
- $\Upsilon = [\Upsilon_1, \dots, \Upsilon_k, \dots, \Upsilon_r]^T, \Upsilon_k \in [0, 1]$

Este sistema de equações permite determinar a quantidade de trabalho de cada recurso relativa a cada transformação, i.e. vector  $\Theta$ , e a capacidade que restará de cada recurso, i.e. vector  $\Upsilon$ , podendo representar-se na forma alternativa da Equação 20.



$$\hat{\mathbf{A}} \times \mathbf{Z} = \hat{\mathbf{W}}, \text{ com: } \hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \boldsymbol{\gamma} & \mathbf{1} \end{bmatrix}, \hat{\mathbf{W}} = \begin{bmatrix} \mathbf{W} \\ \mathbf{1} \end{bmatrix}, \mathbf{Z} = \begin{bmatrix} \boldsymbol{\Theta} \\ \Upsilon \end{bmatrix} \quad \text{Equação 20}$$

A dimensão vertical de  $\hat{\mathbf{A}}$  é igual à soma das dimensões verticais de  $\mathbf{A}$  e  $\boldsymbol{\gamma}$  e a dimensão horizontal é igual à soma das dimensões verticais de  $\boldsymbol{\Theta}$  e  $\Upsilon$ :

- $\dim_{vertical}(\hat{\mathbf{A}}) = \dim_{vertical}(\mathbf{A}) + \dim_{vertical}(\boldsymbol{\gamma}) \Leftrightarrow \dim_{vertical}(\hat{\mathbf{A}}) = n + r$ .
- $\dim_{horizontal}(\hat{\mathbf{A}}) = \dim_{vertical}(\boldsymbol{\Theta}) + \dim_{vertical}(\Upsilon) \Leftrightarrow \dim_{horizontal}(\hat{\mathbf{A}}) = nr + r$ .

Para clarificação do sistema de equações referido, apresentam-se as respectivas matrizes. O vector de variáveis é:

$$\mathbf{Z} = \begin{bmatrix} \boldsymbol{\Theta} \\ \Upsilon \end{bmatrix} = [w_{11} \quad \dots \quad w_{1r} \quad \dots \quad w_{j1} \quad \dots \quad w_{jr} \quad \dots \quad w_{n1} \quad \dots \quad w_{nr} \quad \Upsilon_1 \quad \dots \quad \Upsilon_k \quad \dots \quad \Upsilon_r]^T;$$

O vector de quantidade de trabalho requerida por cada transformação é:

$$\hat{\mathbf{W}} = \begin{bmatrix} \mathbf{W} \\ \mathbf{1} \end{bmatrix} = [W_1 \quad \dots \quad W_n \quad 1_1 \quad \dots \quad 1_r]^T.$$

As matrizes de parâmetros do sistema de equações tomam a seguinte forma:

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \boldsymbol{\gamma} & \mathbf{1} \end{bmatrix} = \begin{array}{c} \begin{array}{cccccccc|cccc} w_{11} & \dots & w_{1r} & \dots & w_{j1} & \dots & w_{jr} & \dots & w_{n1} & \dots & w_{nr} & \dots & \Upsilon_1 & \dots & \Upsilon_k & \dots & \Upsilon_r \\ \hline 1 & \dots & 1 & 0 & & & & & & & 0 & & 0 & & & & & \\ \dots & & & & & & & & & & & & & & & & & \\ W_j & 0 & \dots & 0 & 1 & \dots & 1 & 0 & \dots & 0 & & & & \dots & 0 & & & \\ \dots & & & & & & & & & & & & & & & & & \\ W_n & 0 & & & & & & & 0 & 1 & \dots & 1 & & & & \dots & & 0 \\ \hline 1_1 & \frac{1}{\alpha_{11}} & & & \frac{1}{\alpha_{j1}} & & & & \frac{1}{\alpha_{n1}} & & & & 1 & & & & & \\ \dots & & & & & & & & & & & & & & & & & \\ 1_r & & \frac{1}{\alpha_{1r}} & & & & \frac{1}{\alpha_{jr}} & & & & \frac{1}{\alpha_{nr}} & & & & \dots & 1 & & \\ & & & & & & & & & & & & & & & & \dots & 1 \end{array} \\ \begin{array}{c} n \\ \dots \\ r \end{array} \end{array}$$

Esta equação pode ser analisada em função das dimensões da matriz  $\hat{\mathbf{A}}$ :

- $n + r > nr + r$  - Corresponde ao caso em que existiriam mais equações do que variáveis, mas que nunca se verifica por ser  $r \geq 1$ .
- $n + r = nr + r$  - Neste caso existem tantas equações como variáveis, o que se verifica se  $r=1$ , e nesse caso está-se perante um caso trivial de afectação de recursos, já que haverá apenas um recurso.
- $n + r < nr + r$  - Caso usual em que existem mais variáveis do que equações.

No âmbito da modelação do Sistema Distribuído de Produção desenvolvido nesta dissertação introduziram-se, na subsecção anterior, dois parâmetros que, em conjunto com a Equação 18 permitem definir da capacidade disponível após afectação de recursos  $Y_k$ , em função do tempo livre do recurso. Esta definição, apresentada na Equação 21, é construída em função do tempo livre inicial  $Tf_k$  e do tempo resultante após a afectação de recursos  $Tr_k$ .

$$Y_k = Tr_k / Tf_k \quad \text{Equação 21}$$

Convencionam-se, através da Equação 22, que se o recurso  $k$  não tiver tempo livre então a capacidade resultante e o tempo resultante desse recurso são iguais a zero.

$$Tf_k = 0 \Rightarrow [(Y_k = 0) \wedge (Tr_k = 0)] \quad \text{com } k \in \{1, \dots, r\} \quad \text{Equação 22}$$

Neste caso, a formulação apresentada na Equação 20 transforma-se na Equação 23, em que  $\chi$  representa uma matriz constituída por elementos nulos e por tempos de processamento unitário.

$$\hat{\mathbf{A}}' \times \mathbf{Z}' = \hat{\mathbf{W}}', \text{ com: } \hat{\mathbf{A}}' = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \chi & \mathbf{1} \end{bmatrix}, \hat{\mathbf{W}}' = \begin{bmatrix} \mathbf{W} \\ \mathbf{Tf} \end{bmatrix}, \mathbf{Z}' = \begin{bmatrix} \mathbf{\Theta} \\ \mathbf{Tr} \end{bmatrix} \quad \text{Equação 23}$$

- $\mathbf{Tf} = [Tf_1, \dots, Tf_k, \dots, Tf_r]^T$
- $\mathbf{Tr} = [Tr_1, \dots, Tr_k, \dots, Tr_r]^T, Tr_k \in [0, Tf_k]$

### i. Exemplo

Na secção III - 1.2.3, Tabela 6, apresentam-se os dados referentes aos recursos do exemplo seguido neste capítulo, composto por 5 transformações. Nesses dados podem verificar-se os tempos de processamento unitário  $c_{jk}$  e o número de recursos igual a 16. A modelação do exemplo em estudo (Figura 28) resulta nas seguintes matrizes, de acordo com a Equação 23, com dimensões  $n=5$ ,  $r=16$  e  $nr=80$ .

Para este exemplo o vector de variáveis é:

$$\begin{aligned} \mathbf{Z}' = \begin{bmatrix} \mathbf{\Theta} \\ \mathbf{Tr} \end{bmatrix} &= [w_{1,1} \quad \dots \quad w_{1,r} \quad \dots \quad w_{j,1} \quad \dots \quad w_{j,r} \quad \dots \quad w_{n,1} \quad \dots \quad w_{n,r} \quad Tr_1 \quad \dots \quad Tr_k \quad \dots \quad Tr_r]^T \\ &= [w_{1,1} \quad \dots \quad w_{1,16} \quad \dots \quad w_{j,1} \quad \dots \quad w_{j,16} \quad \dots \quad w_{5,1} \quad \dots \quad w_{5,16} \quad Tr_1 \quad \dots \quad Tr_k \quad \dots \quad Tr_{16}]^T \end{aligned}$$

Para este exemplo o vector de quantidade de trabalho requerida por cada transformação é:

$$\begin{aligned} \hat{\mathbf{W}}' = \begin{bmatrix} \mathbf{W} \\ \mathbf{T} \end{bmatrix} &= [W_1 \quad \dots \quad W_n \quad Tf_1 \quad \dots \quad Tf_r]^T \\ &= [W_1 \quad \dots \quad W_5 \quad Tf_1 \quad \dots \quad Tf_{16}]^T \end{aligned}$$

A seguinte matriz representa os parâmetros do sistema de equações da Equação 23, que permite determinar as quantidades de trabalho a atribuir a cada recurso para executar cada transformação de forma a atingir os objectivos estabelecidos de variação das existências de cada objecto do sistema.

$$\hat{\mathbf{A}}^r = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \boldsymbol{\chi} & \mathbf{1} \end{bmatrix} = \begin{array}{c} \mathbf{W}_1 \\ \dots \\ \mathbf{W}_j \\ \dots \\ \mathbf{W}_5 \\ \mathbf{Tf}_1 \\ \dots \\ \mathbf{Tf}_{16} \end{array} \begin{array}{cccccccc|cccc} w_{1,1} & \dots & w_{1,16} & \dots & w_{j,1} & \dots & w_{j,16} & \dots & w_{5,1} & \dots & w_{5,16} & Tr_1 & \dots & Tr_k & \dots & Tr_{16} \\ \hline 1 & \dots & 1 & 0 & & & & & & & & 0 & 0 & & & & \\ \dots & & & & & & & & & & & & & & & & \\ \mathbf{W}_j & 0 & \dots & 0 & 1 & \dots & 1 & 0 & & & & & & \dots & 0 & & \\ \dots & & & & & & & & & & & & & & & & \\ \mathbf{W}_5 & 0 & & & & & & & \dots & 0 & 1 & \dots & 1 & & & & 0 \\ \mathbf{Tf}_1 & c_{1,1} & & & c_{j,1} & & & & c_{5,1} & & & & 1 & & & & \\ \dots & & \dots & & & \dots & & & & & & & & \dots & 1 & & \\ \mathbf{Tf}_{16} & & c_{1,16} & & & c_{j,16} & & & & c_{5,16} & & & & & & \dots & 1 \end{array} \begin{array}{l} n=5 \\ \\ \\ \\ r=16 \end{array}$$

$nr = 80$   $r = 16$

A configuração genérica de cada nível de agregação, permite determinar as quantidades de trabalho a executar em cada transformação, i.e. matriz  $\mathbf{W}$ , para satisfazer uma determinada quantidade de objecto encomendado. A quantidade de trabalho requerida por cada transformação, em conjunto com os dados sobre os tempos de processamento unitário  $c_{jk}$  e os tempos livres dos recursos, permitem construir o sistema de equações representado pela Equação 23. Este sistema de equações permite determinar uma configuração específica para a configuração genérica referida, isto é, permite afectar quantidades de trabalho a cada recurso para cada transformação. Uma solução deste sistema de equações, satisfaz a restrição de quantidade de trabalho requerida por cada transformação sem exceder o tempo livre de cada recurso. Esta solução não consideraria a agenda do tempo livre do recurso, isto é, a partição desse tempo livre, nem as precedências de processamento entre transformações.

**3. EXEMPLO DE APLICAÇÃO DO MODELO FORMAL MSDP**

A representação formal do modelo de Sistemas Distribuídos de Produção (MSDP) permite construir representações de sistemas de produção específicos. A utilização desta representação permite verificar a sua viabilidade como instrumento de projecto e representação de Sistemas Distribuído de Produção e como instrumento de apoio ao planeamento e controlo de produção nesses mesmos sistemas. Embora se tenha efectuado uma exemplificação dos conceitos de modelação e representação formal neste capítulo, considera-se necessário aplicar estes mesmos conceitos, de uma forma integrada, a um exemplo baseado em conceitos industriais.

Desta forma, apresenta-se um exemplo de representação de um sistema de produção baseado no modelo formal de sistemas distribuídos de produção (secção III - 2). Com base neste exemplo, também se utiliza o modelo formal para construção de uma configuração genérica (secção III - 2.1.4) e posterior configuração específica. Esta configuração específica corresponde a uma instância do sistema de produção resultante da resolução do sistema de equações representado pela Equação 23.

A resolução deste sistema de equações requer a utilização de algum critério. Neste exemplo, utiliza-se o critério de minimização do tempo total de utilização de todos os recursos, isto é, o somatório de todos os tempos de utilização de todos os recursos em todas as transformações. É preciso notar que podem existir vários recursos diferentes para a execução da mesma transformação e que neste caso, o critério favorecerá o

recurso mais rápido. Este é um critério simples, utilizado para introdução do problema de afectação de quantidades de trabalho aos recursos, em função da quantidade de trabalho requerida pelas transformações.

### 3.1 Modelo MSDP de um Sistema de Produção do tipo Linha Flexível de Fabrico

O exemplo aqui apresentado, designado por caso AZ\_FFS, é baseado na estrutura de um sistema produtivo apresentado por Azevedo (1999), cujo trabalho se debruçou sobre a Indústria de Circuitos Integrados. O sistema produtivo é apresentado como sendo do tipo Linha Flexível de Fabrico (FFS – “Flexible Flow Shop”). Um tal sistema é, genericamente composto por diversos estágios produtivos, sendo pelo menos um deles constituído por duas ou mais máquinas paralelas, isto é, máquinas capazes de executarem as mesmas transformações. No caso dito não flexível, há apenas uma máquina em cada estágio. Neste tipo de organização uma transformação faz parte de um processo de fabrico sequencial directo ou progressivo. Num FFS existe a possibilidade de selecção de unidades de processamento, em cada um dos estágios, dentre as que existirem para processar o mesmo tipo de operação.

Pretende-se verificar a capacidade de representação do modelo formal MSDP aplicado a este caso. A utilização deste representação formal, permite determinar a quantidade de trabalho requerida por cada transformação, para satisfazer a produção de uma encomenda de um produto. A agregação de níveis proposta neste modelo MSDP, permite associar as quantidades de materiais entre níveis, e determinar as suas necessidades a partir das quantidades de produto requeridas. Finalmente, a partir das quantidades de trabalho requeridas por cada produto, e sem exceder a quantidade de trabalho disponível por cada recurso, pretende-se efectuar a afectação de recursos às transformações. Esta afectação de recursos corresponde a uma instância do SDP.

O caso em estudo, representado na Figura 29, é composto por 4 estágios com várias unidades de processamento em cada estágio, num total de 11 recursos. Na Figura 29 estão exemplificados os fluxos de 3 ordens de produção diferentes, representando diferentes soluções para a programação da sua execução. Há um conjunto de dados adicionais, que vão ser gerados e apresentados ao longo desta secção de exemplificação, nomeadamente: uma estrutura de produto genérica adaptada ao sistema de produção da Figura 29; a quantidade de produto final encomendada; tempos de processamento unitário de cada recurso em cada transformação; capacidade livre de cada recurso.

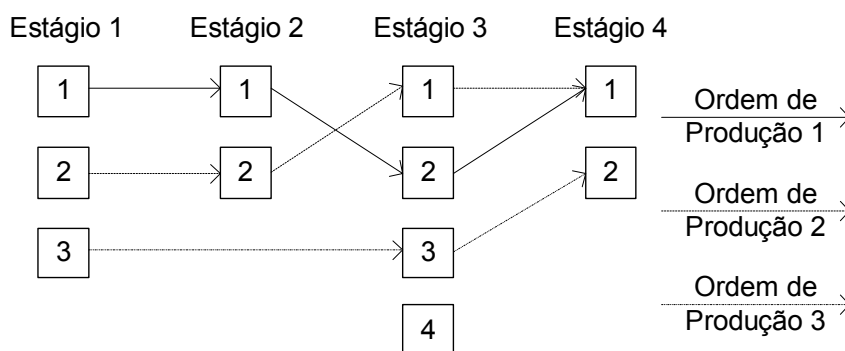


Figura 29: Sistema de produção composto por diferentes estágios - adaptado de Azevedo (1999).

Para poder tratar o caso em estudo com o Modelo do Sistema Distribuído de Produção apresentado neste capítulo, é necessário representar uma estrutura genérica dos produtos fabricados na FFS. De acordo com o modelo, essa estrutura tem um relacionamento estreito com o processo produtivo, podendo ser representada de forma genérica, neste caso, por 5 níveis. Os elementos de cada nível entram na constituição de apenas um elemento do nível superior de forma a manter uma produção do tipo FFS, com uma sequência de estágios. A Figura 30 representa a estrutura do produto a estudar e os respectivos estágios de processamento entre cada nível da estrutura.

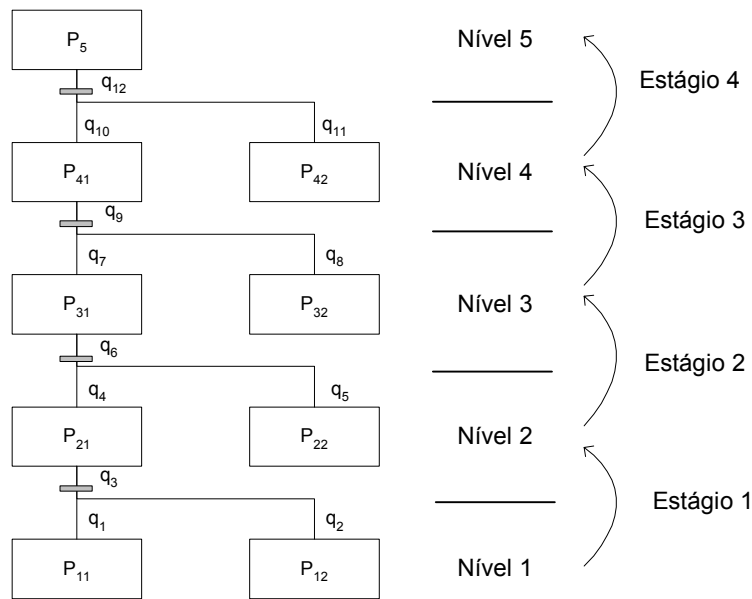


Figura 30: Estrutura do Produto associada a quatro estágios sequenciais de produção.

Na Figura 30, as quantidades envolvidas na produção do produto final  $P_5$ ,  $q_Q$ ,  $Q=1..12$ , representam as quantidades necessárias e geradas pelos processos de transformação de estado dos produtos componentes do produto final. As quantidades representadas na estrutura de produto correspondem às quantidades necessárias dos elementos descendentes para a produção de uma unidade de um elemento ascendente. Embora esta representação possa ser utilizada no modelo desenvolvido nesta dissertação, adopta-se uma notação genérica do ponto de vista da produção, em que se representam as quantidades necessárias e geradas pelos processos de transformação de estado dos produtos da estrutura do produto final. Como no caso em estudo não se encontra descrita a estrutura do produto, as quantidades de utilização apresentadas na Tabela 8, foram geradas de forma arbitrária. É possível verificar por observação conjunta da Figura 30 e da Tabela 8, que por exemplo, a transformação que produz duas unidades ( $q_6$ ) do produto  $P_{31}$  utiliza uma unidade ( $q_4$ ) de  $P_{21}$  e duas unidades ( $q_5$ ) de  $P_{22}$ .

Tabela 8: Quantidades  $q_Q$  consumidas e geradas por todas as transformações do sistema

$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$	$q_7$	$q_8$	$q_9$	$q_{10}$	$q_{11}$	$q_{12}$
2	3	1	1	2	2	3	4	1	2	2	1

### 3.2 Modelação do Produto e do Processo

A Rede de Petri da Figura 31 representa o sistema de produção do produto  $P_5$ , sendo constituída por 9 objectos, referentes ao produto final e aos respectivos componentes, e por 4 transformações, referentes aos 4 estágios de produção. Nesta Rede de Petri também se representam os elementos da estrutura do produto, associados aos objectos da rede.

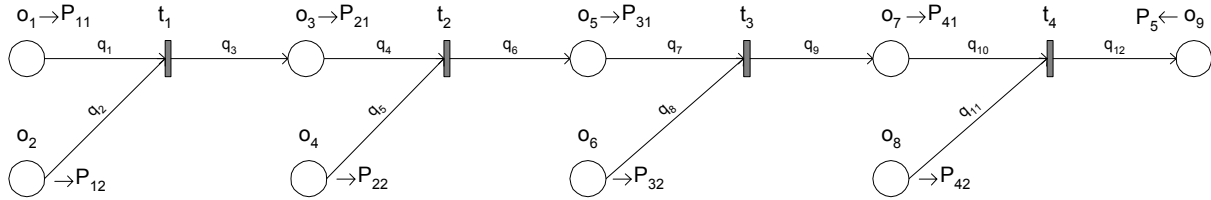


Figura 31: Rede de Petri do sistema de produção

Aplicando o modelo desenvolvido na secção 2 deste capítulo ao sistema de produção representado pela Rede de Petri da Figura 31, pode determinar-se a quantidade de trabalho a executar em todas as transformações em função da quantidade de produtos finais a produzir.

O caso em estudo é composto por 9 objectos e 4 transformações:

- $O = \{o_1, \dots, o_9\}, m = 9.$
- $T = \{t_1, \dots, t_4\}, n = 4.$

Nesta secção de exemplificação, utilizam-se, em várias matrizes, índices auxiliares, colocados fora dos parêntesis rectos. As matrizes da Rede de Petri da Figura 31 são:

$$\text{Pre} = \begin{matrix} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \\ o_6 \\ o_7 \\ o_8 \\ o_9 \end{matrix} \begin{bmatrix} & t_1 & t_2 & t_3 & t_4 \\ q_1 & & & & \\ q_2 & & & & \\ & q_4 & & & \\ & q_5 & & & \\ & & q_7 & & \\ & & q_8 & & \\ & & & q_{10} & \\ & & & q_{11} & \\ & & & & q_{12} \end{bmatrix}; \quad \text{Post} = \begin{matrix} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \\ o_6 \\ o_7 \\ o_8 \\ o_9 \end{matrix} \begin{bmatrix} & t_1 & t_2 & t_3 & t_4 \\ q_3 & & & & \\ & q_6 & & & \\ & & q_9 & & \\ & & & q_{12} & \\ & & & & \end{bmatrix}; \quad \text{C} = \text{Post} - \text{Pre} = \begin{matrix} o_1 \\ o_2 \\ o_3 \\ o_4 \\ o_5 \\ o_6 \\ o_7 \\ o_8 \\ o_9 \end{matrix} \begin{bmatrix} & t_1 & t_2 & t_3 & t_4 \\ -q_1 & & & & \\ -q_2 & & & & \\ q_3 & & -q_4 & & \\ & -q_5 & & & \\ & q_6 & & -q_7 & \\ & & -q_8 & & \\ & & q_9 & & -q_{10} \\ & & & -q_{11} & \\ & & & & q_{12} \end{bmatrix}.$$

Se se pretender produzir  $q$  unidades do produto  $P_5$  ( $o_9$ ) e tivermos como objectivo obter uma variação nula da quantidade de armazenamento de todos os produtos fabricados no sistema, i.e.  $o_3$ ,  $o_5$ , e  $o_7$ , então o vector referente à diferença das quantidades em armazenamento será:

$$\delta \mathbf{S} = \mathbf{S}' - \mathbf{S} = [\delta s_1, \delta s_2, 0, \delta s_4, 0, \delta s_6, 0, \delta s_8, q]^T.$$

Se considerarmos, para exemplificação, uma encomenda de 100 unidades do produto  $P_5$  ( $o_9$ ), então  $q=100$  no vector anterior.

A quantidade de trabalho que é necessário executar em cada transformação, para obter as quantidades finais dos produtos, resulta da Equação 2 e da resolução do respectivo sistema de equações:

$$\mathbf{C} \times \mathbf{W} = \delta \mathbf{S} \Leftrightarrow \mathbf{C} \times [W_1, W_2, W_3, W_4]^T = [\delta s_1, \dots, \delta s_9]^T \Leftrightarrow$$

$$\left\{ \begin{array}{l} -q_1 \cdot W_1 = \delta s_1 \\ -q_2 \cdot W_1 = \delta s_2 \\ q_3 \cdot W_1 - q_4 \cdot W_2 = \delta s_3 \\ -q_5 \cdot W_2 = \delta s_4 \\ q_6 \cdot W_2 - q_7 \cdot W_3 = \delta s_5 \\ -q_8 \cdot W_3 = \delta s_6 \\ q_9 \cdot W_3 - q_{10} \cdot W_4 = \delta s_7 \\ -q_{11} \cdot W_4 = \delta s_8 \\ q_{12} \cdot W_4 = \delta s_9 \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} -2 \cdot W_1 = \delta s_1 \\ -3 \cdot W_1 = \delta s_2 \\ W_1 - W_2 = 0 \\ -2 \cdot W_2 = \delta s_4 \\ 2 \cdot W_2 - 3 \cdot W_3 = 0 \\ -4 \cdot W_3 = \delta s_6 \\ W_3 - 2 \cdot W_4 = 0 \\ -2 \cdot W_4 = \delta s_8 \\ W_4 = 100 \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} \delta s_1 = -600 \\ \delta s_2 = -900 \\ W_1 = 300 \\ \delta s_4 = -600 \\ W_2 = 300 \\ \delta s_6 = -800 \\ W_3 = 200 \\ \delta s_8 = -200 \\ W_4 = 100 \end{array} \right.$$

Os objectos cuja diferença das quantidades em armazenamento é inferior a zero correspondem às matérias primas daquele produto. O facto do seu valor ser negativo significa que existe necessidade de consumo daquelas quantidades que, se consideram existentes em armazém. O valor de  $W_j$ , com  $j=1, \dots, 4$ , indica o número de vezes que é necessário executar a transformação  $j$ , para obter 100 unidades de  $o_8$ . A Tabela 9 representa, nas duas primeiras colunas, a quantidade de trabalho (capacidade) requerida para cada transformação  $W_j$ , determinada acima.

Tabela 9: Capacidade disponível por recurso e requerida por transformação

Período $[t, t + H]$			
Transformação	Capacidade Requerida ( $W_j$ )	Recurso	Capacidade Disponível ( $\alpha_{jk}$ )
1	300	1	210
		2	100
		3	35
2	300	4	240
		5	150
3	200	6	55
		7	100
		8	120
		9	10
4	100	10	40
		11	90

O caso em estudo é composto por 11 recursos, cada um com determinada capacidade disponível para a execução de determinada transformação. A exemplificação deste caso requer a utilização de dados sobre a capacidade disponível dos recursos, para posterior selecção de recursos para cada transformação. No modelo desenvolvido neste trabalho, a capacidade disponível para executar uma determinada transformação corresponde ao número de vezes que o recurso pode executar essa transformação no seu tempo não ocupado. Para obtenção de soluções possíveis geraram-se dados de capacidade disponível de cada recurso, impondo duas restrições: o somatório das capacidades disponíveis dos recursos que podem executar uma transformação tem que ser superior à capacidade requerida por essa transformação; um único recurso não pode ter capacidade disponível suficiente para executar a transformação, de forma que seja necessário seleccionar pelo menos dois recursos. Estes dados, apresentados nas últimas duas colunas da Tabela 9, são válidos para um determinado horizonte temporal definido por  $H$ .

Além destes dados, referentes à capacidade livre dos recursos, e dos dados apresentados na secção anterior (III - 3.1) é necessário conhecer o tempo de processamento unitário  $c_{jk}$  de cada recurso  $k$  para cada

execução da transformação  $j$ . Não se pretende estudar a influência dos tempos de processamento unitário na aplicação do modelo MSDP, pelo que estes valores podem ser gerados aleatoriamente. Pretende-se apenas, introduzir o problema de afectação de recursos e verificar a possibilidade de resolver este problema com o modelo formal do MSDP.

A definição dos tempos de processamento unitário de cada recurso em cada transformação de uma forma controlada, permitirá influenciar o processo de afectação de recursos para execução das transformações, e dessa forma permitirá verificar, mais facilmente, o resultado desse processo. Estes tempos, neste exemplo, são definidos de forma que os recursos com mais capacidade livre sejam os mais rápidos a executar a transformação. Com este objectivo, definem-se como sendo iguais ao inverso da capacidade livre, da seguinte forma:

$$\forall j \in \{1, \dots, n\}, k \in \{1, \dots, r\} : c_{jk} = 1/\alpha_{jk} .$$

Esta definição de dados deverá favorecer, com o critério de minimização do tempo total de utilização, a afectação de trabalho aos recursos mais rápidos e que neste caso, são os que têm mais capacidade livre.

### **3.2.1. Processo de Agregação**

O processo de agregação permite construir várias configurações genéricas de sistemas de produção em vários níveis de agregação. Em cada nível de agregação existem transformações que agregam outras, estabelecendo-se o relacionamento entre as quantidades de trabalho requeridas em cada nível. Além disso há objectos que são embebidos ao longo do processo de agregação, reduzindo-se, no nível de maior agregação, a matérias primas e ao produto final. Neste nível de maior agregação é assim, possível determinar de uma forma directa as necessidades de matérias primas em função da quantidade de produto final requerida.

O processo de agregação resulta da aplicação do algoritmo de Configuração Genérica, secção 2.1.4.i, de desagregação sucessiva, a partir do Nível Superior, de maior agregação, até ao Nível Inferior, de agregação nula. Este nível de agregação do sistema é representado pela Rede de Petri da Figura 31, correspondente a toda a estrutura do produto.

#### **i. Modelação dos Níveis de Agregação**

As transformações de um determinado nível agregam transformações de outros níveis, podendo nesses casos haver objectos embebidos nessa transformação, que só aparecem nos níveis de agregação inferiores.

##### **i.i Nível Superior $v=1=3$**

No Nível Superior a agregação será total, com todas as transformações representadas por uma única transformação de agregação (Figura 32). As entradas desta transformação serão todos os objectos de entrada do sistema, que poderão ser consideradas as matérias primas do sistema de produção.

A transformação  $t_1$  da Figura 32 agrega todas transformações do sistema de produção, isto é, as transformações  $t_1$  a  $t_4$  da Figura 31. Os objectos  $o_1$  a  $o_6$  da Figura 32 coincidem, respectivamente, com os objectos  $o_1, o_2, o_4, o_6, o_8$  e  $o_9$  da Figura 31.



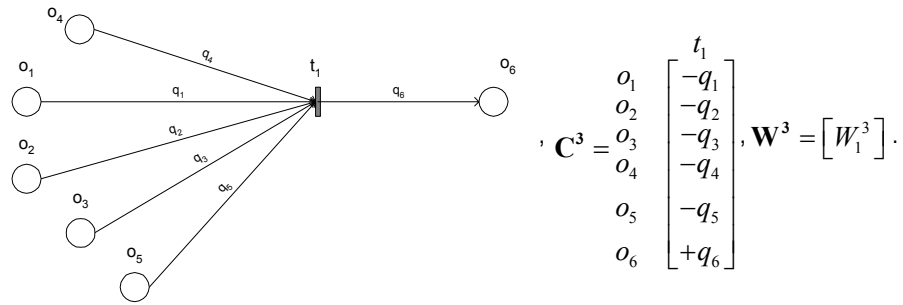


Figura 32: Rede de Petri referente ao nível de agregação  $v=l=3$ .

**i.ii Nível  $v=l-1=2$**

Por aplicação do algoritmo de Configuração Genérica, secção 2.1.4.i, cria-se a Rede de Petri do Nível 2 representado na Figura 33. A transformação  $t_1$  da Figura 33 agrega as transformações  $t_1$  a  $t_3$  do sistema de produção e a transformação  $t_2$  da Figura 33 corresponde à transformação  $t_4$  da Figura 31. Os objectos  $o_1$  a  $o_7$  da Figura 33 coincidem, respectivamente, com os objectos  $o_1, o_2, o_4, o_6, o_7, o_8$  e  $o_9$  da Figura 31.

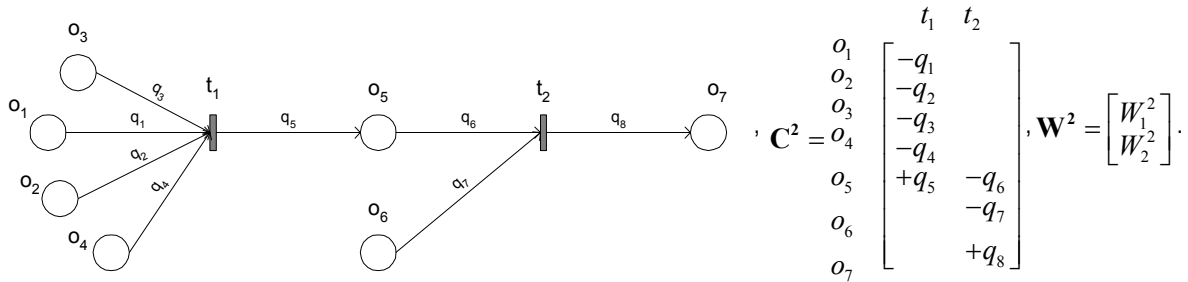


Figura 33: Rede de Petri referente ao nível de agregação  $v=l-1=2$ .

**i.iii Nível  $v=l-2=1$**

Como a Rede de Petri do Nível 2 é diferente da Rede de Petri referente à estrutura do produto, não se verifica o critério de paragem do algoritmo de Configuração Genérica. Por esta razão cria-se a Rede de Petri do Nível 3, como resultado da execução de mais um processo de desagregação, representada na Figura 34.

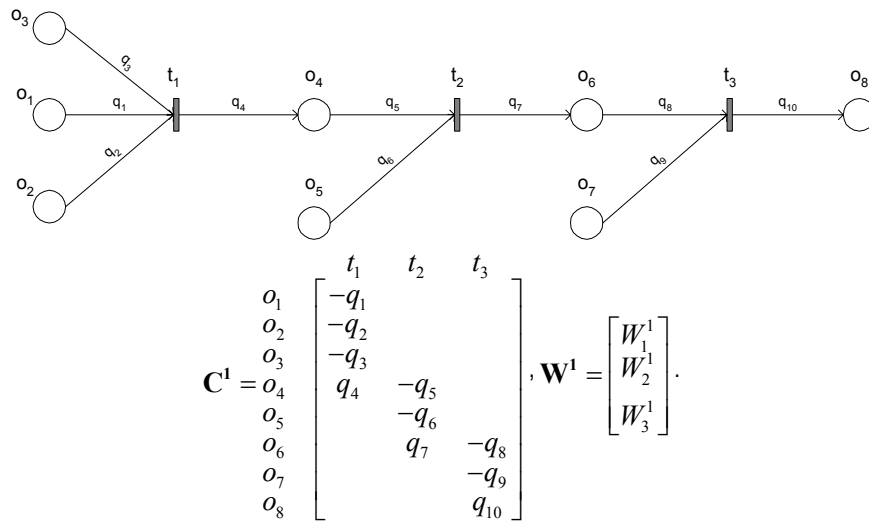


Figura 34: Rede de Petri referente ao nível de agregação  $v=l-2=1$ .

A transformação  $t_1$  da Figura 34 agrega as transformações  $t_1$  e  $t_2$  do sistema de produção e as transformações  $t_2$  e  $t_3$  da Figura 34 correspondem, respectivamente, às transformações  $t_3$  e  $t_4$  da Figura 31. Os objectos  $o_1, o_2, o_3, o_4, o_5, o_6, o_7$ , e  $o_8$  da Figura 34 coincidem, respectivamente, com os objectos  $o_1, o_2, o_4, o_5, o_6, o_7, o_8$  e  $o_9$  da Figura 31.

#### **i.iv Nível inferior $v=0$**

Neste nível não existe agregação, nem de transformações, nem de objectos, correspondendo ao nível elementar, das operações de produção, do sistema de produção. A Rede de Petri deste nível corresponde à Rede de Petri da Figura 31. As matrizes da Rede de Petri e da quantidade de trabalho correspondem às matrizes da secção III - 3.2:  $C^0 = C; W^0 = W$ .

#### **ii. Modelação da Transição entre Níveis de Agregação**

As matrizes de transição entre níveis permitem relacionar as quantidades de objecto e as quantidades de trabalho das transformações entre dois níveis diferentes. A matriz de transição  $\mathbf{M}$  relaciona os objectos dos dois níveis de agregação. A matriz de transição  $\mathbf{N}$  permite determinar, para um nível, a quantidade de trabalho a executar para cada transformação a partir das quantidades de trabalho determinadas para o nível inferior. Além disso permitem determinar as quantidades de utilização de objecto, i.e. pesos das Redes de Petri, num nível de maior agregação a partir de um nível de menor agregação. Deve ser notado que as matrizes  $\mathbf{M}_v^{v+1}$  e  $\mathbf{N}_v^{v+1}$  se deduzem por análise das Redes de Petri dos dois níveis,  $v$  e  $v+1$ , resultando a primeira matriz da verificação do mapeamento entre objectos desses níveis, e a segunda é uma matriz binária que representa a agregação de transformações no nível  $v+1$ .

Neste caso verifica-se, pela Figura 31 e pela Tabela 8, que existem dois objectos, i.e.  $o_5 (P_{31})$  e  $o_7 (P_{41})$  que são produzidos e consumidos em quantidades diferentes. Se estes objectos forem embebidos em alguma transformação agregada, então a determinação das quantidades de utilização dos objectos no nível de agregação, depende do factor de conversão. Dessa forma, para relacionamento entre níveis de agregação, aplica-se a Equação 12 do modelo MSDP desenvolvido neste trabalho.

As matrizes de transição entre níveis  $\mathbf{M}_v^{v+1}$  e  $\mathbf{N}_v^{v+1}$ , com  $v=0..l-1$ , sendo o Nível Superior  $l=3$ , são obtidas e utilizadas nas secções seguintes.

#### **ii.i Transição entre o Nível $v+1=1$ e o Nível Inferior $v=0$**

Por análise das Redes de Petri do Nível 0 (Figura 31) e do Nível 1 (Figura 34) verifica-se que apenas o objecto  $o_3$ , i.e. produto  $P_{21}$ , do Nível 0 está embebido no Nível 1, logo não tem mapeamento na coluna 3 da matriz  $\mathbf{M}$ . A matriz  $\mathbf{N}$  resulta da verificação de que as transformações  $t_2$  e  $t_3$  do Nível 1 são transformações simples (não agregadas) e que a transformação  $t_1$  do Nível 1 agrega duas transformações do Nível 0, respectivamente  $t_1$  e  $t_2$ . As matrizes de transição entre estes níveis são formuladas da seguinte forma:



$$\cdot fc_{jj,j,i} = q_i^* / q_i \Leftrightarrow fc_{1,2,4} = q_{o_4}^* / q_{o_4} \Leftrightarrow fc_{1,2,4} = q_5 / q_4 = 3/2.$$

Por aplicação da Equação 10, apresentada na secção III - 2.1.4.ii, determinam-se as seguintes quantidades do Nível 2:

$$\begin{aligned} \cdot q_{ii}' &= q_{ii} \cdot fc_{jj,j,i} \Leftrightarrow q_1^2 = q_1^1 \cdot fc_{1,2,4} \Leftrightarrow q_1^2 = 2 \cdot 3/2 = 3; \\ \cdot q_{ii}' &= q_{ii} \cdot fc_{jj,j,i} \Leftrightarrow q_2^2 = q_2^1 \cdot fc_{1,2,4} \Leftrightarrow q_2^2 = 3 \cdot 3/2 = 4.5; \\ \cdot q_{ii}' &= q_{ii} \cdot fc_{jj,j,i} \Leftrightarrow q_3^2 = q_3^1 \cdot fc_{1,2,4} \Leftrightarrow q_3^2 = 2 \cdot 3/2 = 3. \end{aligned}$$

A Tabela 11 representa as quantidades  $q_Q$  do Nível 2 de agregação, que resultam da conversão de quantidades entre níveis. Conforme determinado acima, as quantidades de objecto consumidas pela transformação  $t_1$  do Nível 1 são afectadas pela agregação do objecto  $o_4$ , que é consumido pela transformação  $t_2$ , através do factor de conversão  $fc_{1,2,4}$ . Todas as outras quantidades do Nível 2 de agregação têm valores iguais a quantidades do Nível 1.

Tabela 11: Transição entre o Nível 1 e 2 - conversão de quantidades utilizadas pelo sistema.

Nível 2	Produtos	$P_{11}$	$P_{12}$			$P_{22}$			$P_{32}$	$P_{41}$	$P'_{41}$	$P_{42}$	$P_5$
	Objectos	$o_1$	$o_2$			$o_3$			$o_4$	$o_5$	$o'_5$	$o_6$	$o_7$
	Quantidades	$q_1$	$q_2$			$q_3$			$q_6$	$q_7$	$q_8$	$q_9$	$q_{10}$
		3	4.5			3			4	1	2	2	1
Nível 1		2	3			2	2	3	4	1	2	2	1
	Quantidades	$q_1$	$q_2$			$q_3$	$q_4$	$q_5$	$q_6$	$q_7$	$q_8$	$q_9$	$q_{10}$
	Objectos	$o_1$	$o_2$			$o_3$	$o_4$	$o'_4$	$o_5$	$o_6$	$o'_6$	$o_7$	$o_8$
	Produtos	$P_{11}$	$P_{12}$			$P_{22}$	$P_{31}$	$P'_{31}$	$P_{32}$	$P_{41}$	$P'_{41}$	$P_{42}$	$P_5$

### ii.iii Transição entre o Nível $v=l-1=2$ e o Nível Superior $v+1=l=3$

Por análise das Redes de Petri do Nível 2 (Figura 33) e do Nível 3 (Figura 32) deduzem-se as seguintes matrizes de transição entre níveis:

$$\mathbf{M}_2^3 = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & 0 & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}, \mathbf{N}_2^3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

A matriz  $\mathbf{M}$  resulta do facto de apenas o objecto  $o_5$ , i.e. produto  $P_{41}$ , do Nível 2 estar embebido no Nível 3, logo não ter mapeamento na coluna 5. A matriz  $\mathbf{N}$  resulta da verificação de que a transformação  $t_1$  do Nível 2 agrega duas transformações do Nível 3, respectivamente  $t_1$  e  $t_2$ .

Nesta transição a conversão (Tabela 12) de quantidades  $q_Q$  entre níveis não é directa para os objectos  $o_1$  a  $o_4$ , i.e.  $P_{11}$ ,  $P_{12}$ ,  $P_{22}$ , e  $P_{32}$ , porque o único objecto embebido, i.e. objecto  $o_5$  do Nível 2, não é utilizado na mesma quantidade em que é produzido. O factor de conversão para estes objectos é igual à relação entre o consumo e a produção de  $o_5$  ( $P_{41}$ ) no Nível 2:

$$\cdot fc_{jj,j,i} = q_i^* / q_i \Leftrightarrow fc_{1,2,5} = q_{o_5}^* / q_{o_5} \Leftrightarrow fc_{1,2,5} = q_6^2 / q_5^2 = 2.$$

A Tabela 12 representa as quantidades  $q_Q$  do Nível 3 de agregação, que resultam da conversão de quantidades entre níveis. As quantidades de objecto consumidas pela transformação  $t_1$  do Nível 2 são afectadas pela agregação do objecto  $o_5$ , que é consumido pela transformação  $t_2$ , através do factor de conversão  $fc_{1,2,5}$ . Todas as outras quantidades do Nível 3 de agregação têm valores iguais a quantidades do Nível 2.

Tabela 12: Transição entre o Nível 2 e 3 - conversão de quantidades utilizadas pelo sistema.

Nível 3	Produtos	$P_{11}$	$P_{12}$			$P_{22}$			$P_{32}$			$P_{42}$	$P_5$
	Objectos	$o_1$	$o_2$			$o_3$			$o_4$			$o_5$	$o_6$
	Quantidades	$q_1$	$q_2$			$q_3$			$q_4$			$q_5$	$q_6$
		6	9			6			8			2	1
		3	4.5			3			4	1	2	2	1
Nível 2	Quantidades	$q_1$	$q_2$			$q_3$			$q_4$	$q_5$	$q_6$	$q_7$	$q_8$
	Objectos	$o_1$	$o_2$			$o_3$			$o_4$	$o_5$	$o'_5$	$o_6$	$o_7$
	Produtos	$P_{11}$	$P_{12}$			$P_{22}$			$P_{32}$	$P_{41}$	$P'_{41}$	$P_{42}$	$P_5$

A Equação 12 formaliza a transição entre níveis, relacionando os vectores de variação das quantidades armazenadas de objectos, os vectores de quantidade de trabalho a executar por cada transformação e as matrizes de quantidades de consumo de objectos pelas transformações. As quantidades de consumo de objectos foram determinadas pela aplicação directa dos factores de conversão, mas poderiam ter sido determinadas pela aplicação da Equação 12, conforme se mostra a seguir.

O relacionamento entre quantidades de objecto necessárias num nível com as quantidades determinadas para o nível inferior resulta da seguinte equação, que mostra a igualdade da variação das quantidades armazenadas dos produto nos dois níveis:

$$\begin{aligned} \delta \mathbf{S}^{v+1} = \mathbf{M}_v^{v+1} \cdot \delta \mathbf{S}^v \Leftrightarrow \delta \mathbf{S}^3 = \mathbf{M}_2^3 \cdot \delta \mathbf{S}^2 \Leftrightarrow [\delta s_1^3, \dots, \delta s_6^3]^T &= \mathbf{M}_2^3 \cdot [\delta s_1^2, \dots, \delta s_7^2]^T \Leftrightarrow \\ \Leftrightarrow \begin{bmatrix} \delta s_1^3 \\ \delta s_2^3 \\ \delta s_3^3 \\ \delta s_4^3 \\ \delta s_5^3 \\ \delta s_6^3 \end{bmatrix} &= \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & 0 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix} \times \begin{bmatrix} \delta s_1^2 \\ \delta s_2^2 \\ \delta s_3^2 \\ \delta s_4^2 \\ \delta s_5^2 \\ \delta s_6^2 \\ \delta s_7^2 \end{bmatrix} \Leftrightarrow \begin{cases} \delta s_1^3 = \delta s_1^2 \\ \delta s_2^3 = \delta s_2^2 \\ \delta s_3^3 = \delta s_3^2 \\ \delta s_4^3 = \delta s_4^2 \\ \delta s_5^3 = \delta s_6^2 \\ \delta s_6^3 = \delta s_7^2 \end{cases} \end{aligned}$$

O relacionamento entre quantidades de trabalho a executar, para cada transformação, num nível com as quantidades determinadas para o nível inferior, resultam da seguinte equação, que mostra a igualdade de quantidade de trabalho a executar pela única transformação do Nível 3 com as duas do Nível 2:

$$\mathbf{W}^v = \mathbf{N}_v^{v+1} \cdot \mathbf{W}^{v+1} \Leftrightarrow \mathbf{W}^2 = \mathbf{N}_2^3 \cdot \mathbf{W}^3 \Leftrightarrow \begin{bmatrix} W_1^2 \\ W_2^2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot [W_1^3] \Leftrightarrow \begin{bmatrix} W_1^2 \\ W_2^2 \end{bmatrix} = \begin{bmatrix} W_1^3 \\ W_1^3 \end{bmatrix}$$

As matrizes de transição  $\mathbf{M}$  e  $\mathbf{N}$ , permitem relacionar as matrizes  $\mathbf{C}$  de utilização de objectos dos dois níveis, e dessa forma relacionar as quantidades consumidas e geradas por cada transformação, através da Equação 12.

Sendo a matriz  $\mathbf{Fc}$ :

$$\mathbf{Fc}_2^3 = \begin{bmatrix} fc_{jj,j,i} \\ 1 \end{bmatrix} = \begin{bmatrix} fc_{1,2,5} \\ 1 \end{bmatrix} = \begin{bmatrix} q_6/q_5 \\ 1 \end{bmatrix} = \begin{bmatrix} 2/1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

A matriz  $\mathbf{N}'$  resulta da seguinte equação:

$$\mathbf{N}'_2^3 = \mathbf{Fc}_2^3 \circ \mathbf{N}_2^3 \Leftrightarrow \mathbf{N}'_2^3 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Leftrightarrow \mathbf{N}'_2^3 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

A matriz de quantidades de utilização de objectos do Nível 3 resulta da aplicação Equação 12:

$$\mathbf{C}^{v+1} = \mathbf{M}_v^{v+1} \times \mathbf{C}^v \times (\mathbf{Fc}_v^{v+1} \circ \mathbf{N}_v^{v+1}) \Leftrightarrow \mathbf{C}^3 = \mathbf{M}_2^3 \times \mathbf{C}^2 \times (\mathbf{Fc}_2^3 \circ \mathbf{N}_2^3) \Leftrightarrow \mathbf{C}^3 = \mathbf{M}_2^3 \times \mathbf{C}^2 \times \mathbf{N}'_2^3 \Leftrightarrow$$

$$\begin{bmatrix} -q_1^3 \\ -q_2^3 \\ -q_3^3 \\ -q_4^3 \\ -q_5^3 \\ +q_6^3 \end{bmatrix} = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix} \times \begin{bmatrix} -q_1^2 \\ -q_2^2 \\ -q_3^2 \\ -q_4^2 \\ +q_5^2 \\ -q_6^2 \\ -q_7^2 \\ +q_8^2 \end{bmatrix} \times \begin{bmatrix} 2 \\ 1 \end{bmatrix} \Leftrightarrow \begin{cases} -q_1^3 = -2q_1^2 \\ -q_2^3 = -2q_2^2 \\ -q_3^3 = -2q_3^2 \\ -q_4^3 = -2q_4^2 \\ -q_5^3 = -q_7^2 \\ q_6^3 = q_8^2 \end{cases} \Leftrightarrow \begin{cases} q_1^3 = 2q_1^2 \\ q_2^3 = 2q_2^2 \\ q_3^3 = 2q_3^2 \\ q_4^3 = 2q_4^2 \\ q_5^3 = q_7^2 \\ q_6^3 = q_8^2 \end{cases}$$

Este resultado confirma os resultados deduzidos anteriormente para todas as quantidades de utilização de objectos pelas transformações do Nível 3.

### 3.3 Restrições de Precedência de Processos

As restrições de precedência de processos, formuladas na Equação 13, para o Nível Inferior  $v=0$  são dadas pela seguinte matriz  $\mathbf{P}^*$ , em que as colunas indicam transformações que precedem as que são indicadas pelas linhas. Verifica-se nesta matriz que as transformações  $t_1$ ,  $t_2$  e  $t_3$  precedem, respectivamente,  $t_2$ ,  $t_3$  e  $t_4$ .

$$\mathbf{P} = \mathbf{Pre}^T \cdot \mathbf{Post} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ q_3 \cdot q_4 & 0 & 0 & 0 \\ 0 & q_6 \cdot q_7 & 0 & 0 \\ 0 & 0 & q_9 \cdot q_{10} & 0 \end{bmatrix}, \text{ e } \mathbf{P}^* : P_{ij}^* = \text{Min}\{1, P_{ij}\} \Leftrightarrow \mathbf{P}^* = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Determina-se o seguinte vector  $\mathbf{T}^-$  dos tempos de início de cada transformação, pela aplicação da Equação 14. Este vector indica que a transformação  $t_1$  pode iniciar-se no instante zero,  $t_2$  no instante de conclusão de  $t_1$ ,  $t_3$  no instante de conclusão de  $t_2$ , e  $t_4$  no instante de conclusão de  $t_3$ . Estes resultados confirmam a leitura da Rede de Petri apresentada na Figura 31.

$$\mathbf{T}^- = \mathbf{P}^* \otimes \mathbf{T}^+ \Leftrightarrow T_i^- = \text{Max}_j(p_{ij}^* \cdot T_j^+) \Leftrightarrow \mathbf{T}^- = \begin{bmatrix} 0 \\ T_1^+ \\ T_2^+ \\ T_3^+ \end{bmatrix}.$$

### 3.4 Modelação de Recursos

A representação formal do Sistema Distribuído de Produção permite construir um modelo do sistema baseado em níveis de agregação. Este modelo representa as transformações de cada nível do sistema de produção e os objectos consumidos e produzidos, isto é, representa configurações genéricas de cada nível. Esta configuração não representa os recursos de produção utilizados. Existe assim a necessidade de representar esses recursos, conforme secção III - 2, e além disso, determinar a quantidade de trabalho atribuída a cada recurso, isto é, determinar uma solução para o problema de afectação de recursos.

No caso em estudo, AZ\_FFS, com o objectivo de determinação de uma solução de afectação e de verificação da aplicabilidade do modelo MSDP para encontrar uma solução, utilizam-se apenas duas restrições: de procura e de disponibilidade. Estas restrições são as únicas referidas por Lecompte, Deschamps and Bourrières (2000), na formulação da Programação Generalizada, e são as que permitem verificar a existência de capacidade disponível para satisfação da encomenda. Não estão definidas nem restrições, nem critérios, que permitam determinar a quantidade de trabalho atribuída aos recursos, considerando a sua agenda de ocupação e as restrições tecnológicas impostas pelo processo de produção.

No exemplo em estudo, existem 11 recursos disponíveis para o processamento de 4 transformações, conforme se pode confirmar pela Figura 29. No entanto, existem recursos que não podem ser utilizados para a execução de determinadas transformações, pelo que terão quantidade de trabalho nessa transformação igual a zero:

- $w_{1k} = 0, k \in \{4, \dots, 11\}$ ;
- $w_{2k} = 0, k \in \{1, 2, 3, 6, \dots, 11\}$ ;
- $w_{3k} = 0, k \in \{1, \dots, 5, 10, 11\}$ ;
- $w_{4k} = 0, k \in \{1, \dots, 9\}$ .

As incógnitas do sistema são as quantidades de trabalho a atribuir a cada recurso relativas a cada transformação, que podem ser representadas pelo seguinte vector:

$$\begin{aligned} \Theta &= \left[ w_{1,1}, \dots, w_{1,k}, \dots, w_{1,r}, \dots, w_{j,1}, \dots, w_{j,k}, \dots, w_{j,r}, \dots, w_{n,1}, \dots, w_{n,k}, \dots, w_{n,r} \right]^T \\ &= \left[ w_{1,1}, \dots, w_{1,k}, \dots, w_{1,11}, \dots, w_{j,1}, \dots, w_{j,k}, \dots, w_{j,11}, \dots, w_{4,1}, \dots, w_{n,k}, \dots, w_{4,11} \right]^T \end{aligned}$$

As restrições de procura (Equação 15) e disponibilidade (Equação 16) são utilizadas para a selecção de recursos, conduzindo ao seguinte sistema de equações:

$$\begin{array}{l} \text{Restrições de Procura} \\ \text{Restrições de Disponibilidade} \end{array} \quad \begin{cases} \mathbf{A} \times \Theta = \mathbf{W} \\ \boldsymbol{\gamma} \times \Theta \leq \mathbf{1} \end{cases} \quad \xrightarrow{\text{transformado em}} \quad \begin{cases} \mathbf{A} \times \Theta = \mathbf{W} \\ \boldsymbol{\gamma} \times \Theta + \Upsilon = \mathbf{1} \end{cases}$$

A quantidade de trabalho disponível  $\alpha_{jk}$  realizável pelo recurso  $k$  na transformação  $j$ , corresponde à quantidade de trabalho disponível do recurso  $k$  relativa à transformação  $j$ . Esta quantidade de trabalho disponível está representada na coluna de "Capacidade Disponível" da Tabela 9. A matriz  $\boldsymbol{\gamma}$  é composta por vários elementos cujo valor é igual ao inverso das capacidades disponíveis. O vector da capacidade excedente de cada recurso, isto é, da capacidade disponível após afectação de recursos corresponde a:

$$\Upsilon = [\Upsilon_1, \dots, \Upsilon_k, \dots, \Upsilon_r], \Upsilon_k \in [0, 1].$$

Assim, obtém-se um sistema de equações que permitirá determinar a quantidade de trabalho de cada recurso relativa a cada transformação,  $\Theta$ , e a capacidade que restará de cada recurso,  $\Upsilon$ :

$$\hat{\mathbf{A}} \cdot \mathbf{Z} = \hat{\mathbf{W}}, \text{ com: } \hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \boldsymbol{\gamma} & \mathbf{1} \end{bmatrix}, \hat{\mathbf{W}} = \begin{bmatrix} \mathbf{W} \\ \mathbf{1} \end{bmatrix}, \mathbf{Z} = \begin{bmatrix} \Theta \\ \Upsilon \end{bmatrix}$$

A dimensão vertical de  $\hat{\mathbf{A}}$  será igual à soma das dimensões verticais de  $\mathbf{A}$ ,  $\boldsymbol{\gamma}$  e a dimensão horizontal será igual à soma das dimensões verticais de  $\Theta$ ,  $\Upsilon$ :

$$\dim_{\text{vertical}}(\hat{\mathbf{A}}) = \dim_{\text{vertical}}(\mathbf{A}) + \dim_{\text{vertical}}(\boldsymbol{\gamma}) = n + r = 15;$$

$$\dim_{\text{horizontal}}(\hat{\mathbf{A}}) = \dim_{\text{vertical}}(\Theta) + \dim_{\text{vertical}}(\Upsilon) = nr + r = (n + 1) \cdot r = 55.$$

A matriz das incógnitas é composta pelas variáveis de quantidade de trabalho a atribuir a cada recurso por cada transformação e pela quantidade de capacidade restante após atribuição aos recursos. Esta matriz será assim constituída por 4 grupos (transformações) de 11 variáveis (recursos) de quantidade de trabalho a atribuir a cada recurso, mais 11 variáveis de quantidade de capacidade restante após atribuição aos recursos. Esta matriz é representada em seguida:

$$\mathbf{Z} = \begin{bmatrix} \Theta \\ \Upsilon \end{bmatrix} = \begin{bmatrix} w_{11} & \dots & w_{1r} & \dots & w_{j1} & \dots & w_{jr} & \dots & w_{n1} & \dots & w_{nr} & \Upsilon_1 & \dots & \Upsilon_k & \dots & \Upsilon_r \end{bmatrix}^T$$

$$\mathbf{Z} = \begin{bmatrix} 1 & 2 & 3 & 4 & \dots & 11 & | & 1 & 2 & 3 & 4 & 5 & 6 & \dots & 11 & | & 1 & \dots & 5 & 6 & 7 & 8 & 9 & 10 & 11 & | & 1 & \dots & 9 & 10 & 11 & | & 1 & \dots & 11 \\ w_{1,1} & w_{1,2} & w_{1,3} & 0 & \dots & 0 & | & 0 & 0 & 0 & w_{2,4} & w_{2,5} & 0 & \dots & 0 & | & 0 & \dots & 0 & w_{3,6} & w_{3,7} & w_{3,8} & w_{3,9} & 0 & 0 & | & 0 & \dots & 0 & w_{4,10} & w_{4,11} & | & \Upsilon_1 & \dots & \Upsilon_r \end{bmatrix}^T$$

A matriz da quantidade de trabalho, resulta da quantidade de trabalho requerida por cada transformação e de um vector unidade com  $r$  elementos:

$$\hat{\mathbf{W}} = \begin{bmatrix} \mathbf{W} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} W_1 & \dots & W_n & 1_1 & \dots & 1_r \end{bmatrix}^T$$

$$= \begin{bmatrix} W_1 & \dots & W_4 & 1_1 & \dots & 1_{11} \end{bmatrix}^T = \begin{bmatrix} 300 & 300 & 200 & 100 & 1_1 & \dots & 1_{11} \end{bmatrix}^T$$

A matriz dos parâmetros do sistema de equações toma a seguinte forma:

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \boldsymbol{\gamma} & \mathbf{1} \end{bmatrix} = \begin{array}{c} W_1 \\ \dots \\ W_j \\ \dots \\ W_n \\ \hline 1_1 \\ \dots \\ 1_r \end{array} \begin{array}{cccccccc|cccc} w_{11} & \dots & w_{1r} & \dots & w_{j1} & \dots & w_{jr} & \dots & w_{n1} & \dots & w_{nr} & \Upsilon_1 & \dots & \Upsilon_k & \dots & \Upsilon_r \\ \hline 1 & \dots & 1 & 0 & & & & & & & 0 & 0 & & & & & \\ \dots & & & & & & & & & & & & \dots & & & & \\ W_j & 0 & \dots & 0 & 1 & \dots & 1 & 0 & & & \dots & 0 & & & & & \\ \dots & & & & & & & & & & & & & & & & \\ W_n & 0 & & & & & & & 0 & 1 & \dots & 1 & & & & & 0 \\ \hline 1_1 & 1/\alpha_{11} & & & 1/\alpha_{j1} & & & & 1/\alpha_{n1} & & & 1 & & & & & \\ \dots & & & & & & & & & & & \dots & & & & & \\ 1_r & & 1/\alpha_{1r} & & & & 1/\alpha_{jr} & & & & 1/\alpha_{nr} & & & & & & 1 \end{array}$$

$n \cdot r = 44$   $r = 11$

Este sistema de equações representa um conjunto de restrições do sistema, com mais equações do que variáveis, estando a obtenção de soluções dependente dos critérios utilizados.



### 3.5 Afecção de Recursos

Para verificação, sumária, da possibilidade de utilização do modelo formal do MSDP para resolução de um problema de afecção de recursos, pretende-se determinar a quantidade de trabalho a atribuir a cada recurso, que minimiza o tempo total de utilização de todos os recursos. Este objectivo é definido para exemplificação da possibilidade de afecção. Em termos formais, pretende-se determinar a quantidade de trabalho  $w_{jk}$  executada pelo recurso  $k$  na transformação  $j$ , com  $1 \leq j \leq n=4$  e  $1 \leq k \leq r=11$ , minimizando o tempo total de utilização de todos os recursos. Isto significa que se vão beneficiar os recursos que forem mais rápidos no processamento de uma transformação, atribuindo-lhe o máximo de quantidade de trabalho possível. O tempo total de utilização dos recursos é definido da seguinte da forma:

Tempo Total de  
Utilização dos Recursos

$$\sum_{j=1}^n \sum_{k=1}^r c_{jk} \cdot w_{jk}$$

A relação entre os recursos e as quantidades de trabalho a executar está especificada no sistema de equações  $\hat{\mathbf{A}} \cdot \mathbf{Z} = \hat{\mathbf{W}}$ , com mais equações do que variáveis. Este sistema reflecte as restrições de procura e de disponibilidade definidas na Equação 15 e na Equação 16. Utilizando estas restrições é possível formular o problema de afecção de recursos, como de minimização do tempo total de utilização dos recursos da seguinte forma.

**Função Objectivo:**

Minimizar o Tempo Total de  
Utilização dos Recursos

minimizar  $\sum_{j=1}^n \sum_{k=1}^r c_{jk} \cdot w_{jk}$

**Restrições:**

Procura

$$\forall j \in \{1, \dots, n\} : \sum_{k=1}^r w_{jk} = W_j$$

Disponibilidade

$$\forall k \in \{1, \dots, r\} : \sum_{j=1}^n \frac{w_{jk}}{\alpha_{jk}} \leq 1$$

Variáveis

$$\forall j \in \{1, \dots, n\}, k \in \{1, \dots, r\} : w_{jk} \in \mathbb{Z}_0^+$$

A solução obtida para este problema, considerando os dados apresentados na Tabela 9, está representada na Tabela 13. Nesta solução, pode verificar-se a afecção da capacidade disponível máxima, até ao limite da capacidade requerida, dos recursos mais rápidos a executar cada transformação.

Tabela 13: Quantidade de trabalho atribuído a cada recurso.

Quantidade de Trabalho	$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	$w_{2,4}$	$w_{2,5}$	$w_{3,6}$	$w_{3,7}$	$w_{3,8}$	$w_{3,9}$	$w_{4,10}$	$w_{4,11}$
	210	90	0	240	60	0	80	120	0	10	90

Nesta secção 3, apresentou-se a formulação do modelo de um sistema do tipo linha flexível de fabrico, com representação dos objectos processados nessa linha, das transformações executadas e dos recursos disponíveis. A representação formal do sistema permitiu modelar o problema de afecção de recursos como

um problema de otimização linear. A resolução deste problema mostrou a possibilidade de obtenção de uma solução de configuração para o sistema, a partir da afectação de recursos para a execução de cada transformação. Esta afectação, depende da aptidão, i.e. da adequabilidade tecnológica, da capacidade disponível de cada recurso e do critério de afectação utilizado. O critério utilizado neste caso foi o de minimização do tempo total de utilização dos recursos, que favoreceu a atribuição da máxima quantidade de trabalho possível aos recursos mais rápidos.

#### **4. CONSIDERAÇÕES FINAIS**

Neste capítulo, apresenta-se um modelo de Sistemas Distribuídos de Produção apoiado na definição proposta neste trabalho, em conceitos de Sistemas de Produção Orientados ao Produto e na teoria de modelação de sistemas Orientados aos Objectos. O modelo permite configurar ou reconfigurar sistemas de produção para produzir uma certa quantidade de um produto encomendado.

A definição de produto apresentada neste trabalho estende-se a todos os elementos da estrutura de um produto final. Cada produto processado no sistema, resulta da mudança de estado de outros produtos. O conceito de mudança de estado, introduzido neste trabalho, permite modelar os processos de produção do sistema, tornando-se um dos conceitos fundamentais do modelo de Sistemas Distribuídos de Produção. Adicionalmente, de acordo com uma visão fractal do sistema, é possível visualizar produtos dentro de produtos, resultantes de mudanças de estado, e sistemas dentro de sistemas que satisfazem encomendas daqueles produtos. Isto significa que alguns sistemas agregam outros sistemas.

O modelo de Sistemas Distribuídos de Produção é, assim, constituído por sistemas dentro de sistemas, e cada um destes corresponde a uma célula virtual autónoma, que existe no âmbito de cada SDP. Cada SDP tem a responsabilidade de satisfação da encomenda de um dado produto final. Cada célula tem a responsabilidade de satisfação da encomenda de um elemento da estrutura desse produto final. As células autónomas formam a configuração genérica do Sistema Distribuído de Produção. A autonomia das células baseia-se na sua capacidade de execução de transformações de mudança de estado dos elementos da estrutura do produto. A célula será composta por um ou mais recursos reais autónomos, isto é, recursos com capacidade de executar a transformação que a célula tem que executar para satisfazer a encomenda. A afectação de recursos de produção a cada célula, para cada encomenda, constitui uma configuração específica do SDP.

A criação de uma configuração genérica e posterior configuração específica, de Sistemas Distribuídos de Produção, para cada produto encomendado, torna dinâmica a configuração e portanto pode ser vista como um processo que pode ser aplicado para reconfiguração de sistemas. Neste processo de configuração de SDP, os recursos de produção podem pertencer simultaneamente a diversas células virtuais autónomas de produção, associadas a várias encomendas de vários produtos.

A formalização do modelo MSDP desenvolvido, efectua-se com base na representação formal de Programação Generalizada, inicialmente apresentada por Lecompte, Deschamps and Bourrières (2000) e estendida neste trabalho. Esta representação formal de SDP, baseada em Redes de Petri, permite especificar o significado preciso de cada elemento do modelo e determinar os requisitos de produção a partir da

encomenda e de restrições de recursos e de materiais. Desenvolveu-se um Algoritmo de Configuração Genérica para criação do modelo do Sistema Distribuído de Produção com os respectivos sistemas dentro de sistemas, isto é, com agregação de transformações. Esta agregação, além de permitir modelar sistemas com base em replicação de estruturas similares, permite reduzir a dimensão de problemas, de determinação de necessidade de materiais e quantidade de trabalho, em níveis de maior agregação. Nestes níveis é possível determinar as necessidades de materiais a partir das necessidades de produtos finais com base na quantidade de trabalho de uma única transformação. As necessidades de materiais e de quantidade de trabalho em todos os níveis de agregação são relacionadas entre si.

A configuração genérica do sistema permite efectuar uma primeira abordagem à formalização do problema de afectação de recursos para execução das transformações, isto é, para as células. Este problema de afectação relaciona as quantidades de trabalho requeridas por cada transformação da configuração genérica com as quantidades de trabalho de cada recurso, através de restrições de Procura e Disponibilidade. Esta formalização, no entanto, não é suficiente para determinar soluções sendo necessário adicionar objectivos ou critérios para a afectação de recursos. A aplicação da representação formal do modelo MSDP a um caso de exemplo permite concluir que é possível utilizar este modelo MSDP para determinar as quantidades de materiais e de trabalho em vários níveis de agregação. Além disso, efectuou-se uma abordagem sumária ao problema de afectação de recursos aplicada ao caso de exemplo, com o critério de minimização do tempo total de utilização dos recursos, que mostrou a sua aplicabilidade para criação de uma configuração específica do Sistema Distribuído de Produção. Esta afectação de recursos não considerou restrições de precedências de processos nem de agenda de recursos, permitindo apenas atribuir capacidade a cada recurso. O desenvolvimento de processos de criação de configurações específicas deve, no âmbito dos objectivos desta dissertação, basear-se na filosofia de Produção Simultânea. Dessa forma será possível criar um modelo do Sistema Distribuído de Produção em Ambiente de Produção Simultânea, em que as instâncias do SDP sejam determinadas de acordo com a filosofia de PS.



## **IV - PRODUÇÃO SIMULTÂNEA**

Para reduzir tempos de desenvolvimento de produtos e aumentar a interacção entre diversos tipos de conhecimento existentes na empresa aplicam-se princípios de Engenharia Concorrente, ou Engenharia Simultânea. Da mesma forma, e com o intuito de reduzir tempos de entrega, a Produção Simultânea aplica conceitos de processamento simultâneo a ordens ou lotes de produção. Desta forma concentra os recursos, i.e. os meios de produção, no fabrico de uma ordem de produção, ou encomenda, de cada vez, como estratégia para diminuir o seu tempo em curso de fabrico e contribuir para a redução do prazo de entrega. Neste processo, o processamento paralelo da mesma ordem de produção é adoptado sempre que possível.



A selecção de recursos de produção e a respectiva afectação de quantidades de trabalho, para projecto de um sistema distribuído de produção, depende dos critérios e estratégias de selecção utilizados. Neste capítulo explora-se a utilização da Produção Simultânea (PS) como estratégia para formulação do problema de afectação de recursos, no modelo de Sistemas Distribuídos de Produção desenvolvido neste trabalho. Para formulação do problema de afectação de recursos de acordo com a PS, desenvolve-se a definição de Produção Simultânea e introduzem-se parâmetros de medida do Grau de Simultaneidade do sistema de produção além de princípios de simultaneidade. Para estudo do Sistema Distribuído de Produção em ambiente de Produção Simultânea definem-se possibilidades de configuração do sistema.

## **1. DEFINIÇÃO**

Uma encomenda pode ser composta por vários produtos ou por um único produto, requeridos em diferentes quantidades. Uma encomenda pode, ainda, ser agregada com outras encomendas e as funções de planeamento, programação e controlo da produção serem executadas para o total das encomendas. No entanto, estas funções do sistema de produção podem executar-se para uma única encomenda de um único produto de cada vez. Este trabalho enquadra-se nesta estratégia de processamento de encomendas, subjacente à Produção Simultânea. Na Produção Simultânea preconiza-se, na medida do possível, a utilização simultânea de todos os recursos do sistema na produção de uma encomenda. Esta estratégia deve ser considerada para o planeamento e programação total de uma encomenda. A sua implementação aponta no sentido da divisão da encomenda para aumento da simultaneidade ou paralelismo do processamento. Tal simultaneidade levará, previsivelmente, à diminuição do prazo de fabrico associado à encomenda e por conseguinte, contribuirá para a diminuição do seu prazo de entrega.

### **1.1 Breve Abordagem à Engenharia Concorrente / Simultânea**

Uma breve abordagem a conceitos de Engenharia Concorrente / Simultânea permite apontar algumas das razões inerentes à aplicação de princípios de simultaneidade processual. As definições de Engenharia Concorrente apontam para a realização simultânea de actividades de projecto do produto e do processo:

- Para Kusiak and Wang (1993) Engenharia Concorrente corresponde à abordagem sistemática ao projecto simultâneo e integrado de produtos e processos, incluindo produção e suporte.
- Wentorf and Shephard (1993) definem Engenharia Concorrente como sendo a realização em paralelo das actividades de Engenharia relacionadas com o desenvolvimento do produto ao longo do processo de projecto.

Algumas das vantagens, apontadas por Gertsen (1991) e Wentorf and Shephard (1993), que se pretendem obter com a aplicação de Engenharia Concorrente são:

- Aumento da cooperação entre diferentes unidades funcionais da empresa.
- Aumento da qualidade do produto em todo ciclo de vida.
- Diminuição dos custos produtivos em todo ciclo de vida.
- Aumento da produtividade.
- Diminuição do tempo de percurso de fabrico<sup>34</sup>.

---

<sup>34</sup> "Throughput time".

Um dos factores de avaliação de um Sistema de Produção referidos por vários autores da Engenharia Concorrente é o tempo de percurso de fabrico. Herrmann and Chincholkar (2001) definem o tempo de percurso de fabrico<sup>35</sup> como o período de tempo para o desempenho de todas as operações necessárias para completar uma ordem de fabrico no sistema de produção. Os tempos incluídos neste período incluem tempos de movimentação, filas de espera, preparação e processamento. Ainda segundo este autor, a redução do tempo de percurso de fabrico tem vantagens que incluem a redução da quantidade armazenada de produtos, redução de custos, aumento da flexibilidade e um menor tempo para o mercado porque o primeiro lote de produtos fabricados pode chegar mais depressa ao consumidor.

## **1.2 Produção Simultânea**

O conceito ou filosofia designada por Produção Simultânea<sup>36</sup> (PS), introduzida por Silva, S. C. and Putnik (1995), corresponde à produção de um produto por utilização de todos os meios existentes com o objectivo de minimizar o tempo total de produção. Este conceito de produção pode levar à exploração intensa do processamento paralelo de componentes do produto a produzir.

Neste tipo de produção uma encomenda / lote de produção é escalonada nos meios de produção disponíveis de forma a minimizar o tempo total de processamento da encomenda, pela redução do seu tempo de percurso de fabrico. As encomendas são ordenadas previamente e depois escalonadas seguindo a filosofia de PS. Neste processo são definidos esquemas de partição ou sobreposição dos lotes de produção para explorar o processamento simultâneo de cada encomenda nos meios de produção do sistema.

A redução do tempo de percurso de fabrico de uma encomenda, contribui para uma possível libertação dos recursos de produção mais cedo. A libertação dos recursos permitirá, de acordo com a filosofia de PS, a sua utilização noutra encomenda, que vai ser escalonada de forma a minimizar o tempo de percurso de fabrico. Este processo de sucessiva redução de tempo de percurso de fabrico conduz, previsivelmente, a um aumento do ritmo ao qual se satisfazem encomendas, isto é, ao ritmo de conclusão de encomendas. Aumentando este ritmo, pode-se aumentar a satisfação dos clientes, pela redução do prazo de entrega das encomendas. Assim, a filosofia de PS pode ser utilizada para redução do prazo de fabrico de cada encomenda, com o objectivo de aumentar a satisfação dos clientes. Torna-se, no entanto, necessário analisar o efeito da utilização de PS sobre um conjunto de encomendas sucessivas.

Outras abordagens utilizadas nos sistemas de produção têm realçado a importância de alguns parâmetros de débito, apontando-os como devendo ser alvo de maximização, nomeadamente:

- Na teoria das restrições (Goldratt and Fox (1986); Goldratt (1990)) é utilizada a noção de *débito*<sup>37</sup> que é o ritmo ao qual o sistema gera dinheiro através da venda de produtos.
- Em trabalhos de Engenharia Concorrente também designada por alguns autores, nomeadamente por Gertsen (1991), como Engenharia Simultânea, em que existe uma abordagem integrada ao projecto do produto e do processo, para diminuição do tempo para o mercado e, também, do tempo de percurso de fabrico.

---

<sup>35</sup> Herrmann and Chincholkar (2001) – “Note that many authors use the terms *manufacturing cycle time* or *flow time* instead of *throughput time*”.

<sup>36</sup> “SM - Simultaneous Manufacturing”.

<sup>37</sup> “Throughput”.



A atribuição simultânea dos meios de produção disponíveis a uma encomenda de cada vez, não garante a minimização do prazo de entrega para todas as encomendas num período de tempo. A atribuição de todos os meios de produção ao processamento duma encomenda, procura apenas diminuir o tempo que essa encomenda demora a atravessar o sistema, isto é, o tempo de percurso de fabrico da encomenda. Tal solução, pode ocupar um processador num momento em que poderia ser utilizado, de uma forma mais vantajosa, por outra encomenda, à luz de outros critérios.

No caso de sistemas distribuídos de produção, uma célula autónoma propõe um prazo de entrega para uma encomenda, devendo garantir o processamento que disponibiliza. Neste caso a Produção Simultânea pode ser aplicada na sua vertente de paralelismo de processamento de componentes. Assim sendo, uma encomenda de uma determinada quantidade  $q$  pode ser dividida em várias encomendas de quantidades  $qq$ , sendo  $q$  múltiplo de  $qq$ .

Deve notar-se que o processamento simultâneo poderá levar ao aumento de tempos de preparação e ao aumento do número de recursos envolvidos no processamento de uma encomenda. É previsível, neste caso, o aumento do número de actividades de transporte, podendo adquirir um peso relativo importante. Se este peso relativo for importante, então deveria ser considerado no processo de modelação e afectação de recursos. Não foi definido como objectivo deste trabalho a modelação de actividades de transporte. No entanto, esta modelação poderia ser efectuada pela consideração de transformações associadas a essas actividades, e por uma caracterização adequada desse tipo de actividade.

O custo de posse, por aumentar com o aumento do tempo de armazenamento, terá tendência a diminuir por acção do previsível aumento de rapidez de entrega de encomendas, resultante da PS. A consideração conjunta do custo de posse e do custo de preparação poderá levar à determinação de um “grau” de simultaneidade de utilização de recursos, com redução do instante de conclusão da encomenda e com custos conjuntos relativamente baixos. Este grau de simultaneidade poderá ser um bom indicativo da configuração do SDP.

### **1.3 Medida de Simultaneidade**

Medir a simultaneidade de utilização dos recursos permite estabelecer uma medida do Grau de Simultaneidade de um sistema de produção. Para a satisfação de uma única encomenda pode projectar-se um sistema de produção supondo a possibilidade de utilização de todos os recursos disponíveis. Se forem utilizados todos os recursos simultaneamente, então teremos um Grau de Simultaneidade máximo, por oposição com a utilização de apenas um recurso.

#### **1.3.1. Grau de Simultaneidade do Sistema**

Neste trabalho, o Grau de Simultaneidade do sistema corresponde a um índice da utilização simultânea de recursos, para uma encomenda. Este índice é tanto mais elevado quanto mais recursos forem utilizados para a execução de todas as transformações do sistema, até ao limite de todos os recursos candidatos para todas as transformações. Um recurso será candidato à execução de uma transformação, conforme representação da Equação 24, se puder executar aquela transformação, isto é, se o tempo de execução unitário  $c_{jk}$ , de uma unidade de quantidade de trabalho, para uma transformação for superior a zero.

$$\begin{array}{l}
 \text{Recurso candidato a uma transformação} \\
 \text{Recurso candidato a uma transformação}
 \end{array}
 \quad
 \begin{array}{l}
 cr_{jk} : \mathbb{Z}_0^+ \rightarrow \{0,1\} \\
 cr_{jk}(c_{jk}) = \begin{cases} 0 & \text{se } c_{jk} = 0 \\ 1 & \text{se } c_{jk} > 0 \end{cases}
 \end{array}
 \quad
 \text{Equação 24}$$


---

Considerando a definição de recurso candidato a uma transformação apresentada na Equação 24, em que  $cr_{jk}$  indica se o recurso  $k$  é candidato para a execução da transformação  $j$ , então o número total de recursos candidatos para todas as transformações do sistema é representado pela Equação 25.

$$\begin{array}{l}
 \text{Número de Recursos Candidatos do SDP}
 \end{array}
 \quad
 \sum_{j=1}^n \sum_{k=1}^r cr_{jk}
 \quad
 \text{Equação 25}$$


---

Considera-se que um recurso é seleccionado para o processamento de uma encomenda se a quantidade de trabalho que lhe é atribuída para execução de uma transformação for superior a zero.

$$\begin{array}{l}
 \text{Recurso seleccionado}
 \end{array}
 \quad
 \begin{array}{l}
 wz_{jk} : \mathbb{Z}_0^+ \rightarrow \{0,1\} \\
 wz_{jk}(w_{jk}) = \begin{cases} 0 & \text{se } w_{jk} = 0 \\ 1 & \text{se } w_{jk} > 0 \end{cases}
 \end{array}
 \quad
 \text{Equação 26}$$


---

Considerando  $wz_{jk}$  como a variável que representa a quantidade de trabalho atribuída ao recurso  $k$  para a execução da transformação  $j$ , o número de recursos utilizados para o processamento de uma encomenda  $WZ$ , resulta do somatório de todos os recursos utilizados em todas as transformações, i.e. de todos os recursos com quantidade de trabalho superior a zero em todas as transformações.

$$\begin{array}{l}
 \text{Número de Recursos Seleccionados}
 \end{array}
 \quad
 WZ = \sum_{j=1}^n \sum_{k=1}^r wz_{jk}
 \quad
 \text{Equação 27}$$


---

A formalização do Grau de Simultaneidade do sistema  $GS$ , representado pela Equação 28, resulta do quociente entre o número de recursos utilizados para a execução de todas as transformações da encomenda e o número total de recursos candidatos.

$$\begin{array}{l}
 \text{Grau de Simultaneidade do Sistema}
 \end{array}
 \quad
 GS = \frac{\sum_{j=1}^n \sum_{k=1}^r wz_{jk}}{\sum_{j=1}^n \sum_{k=1}^r cr_{jk}}
 \quad
 \text{Equação 28}$$


---

Este índice terá um limite superior igual a um (1) e um limite inferior superior a zero. Este limite inferior corresponde à selecção de, apenas, um recurso em cada transformação.

$$\begin{array}{l}
 \text{Limites}
 \end{array}
 \quad
 n \cdot \left( \sum_{j=1}^n \sum_{k=1}^r cr_{jk} \right)^{-1} \leq GS \leq 1
 \quad
 \text{Equação 29}$$


---

### 1.3.2. Grau de Simultaneidade de uma Transformação

O Grau de Simultaneidade de uma transformação corresponde a um índice da utilização de recursos simultânea para uma transformação, da configuração genérica do sistema gerada para uma encomenda. Este índice será tanto mais elevado quanto mais recursos forem utilizados para a execução dessa transformação,

até ao limite de todos os recursos candidatos. Um recurso será candidato se puder executar aquela transformação, conforme formulado na Equação 24. Considerando esta definição de recurso candidato, com  $cr_{jk}$  representando o recurso  $k$  como candidato para a execução da transformação  $j$ , o número total de recursos candidatos para o processamento de uma transformação será determinado pela Equação 30.

---

Número de Recursos Candidatos	$\forall j \in \{1, \dots, n\} : \sum_{k=1}^r cr_{jk}$	Equação 30
-------------------------------	--	------------

---

Considerando a definição de recurso seleccionado apresentada na Equação 26, o número de recursos utilizados para o processamento de uma transformação, resulta do somatório de todos os recursos com quantidade de trabalho superior a zero para essa transformação.

---

Número de Recursos por Transformação	$\forall j \in \{1, \dots, n\} : WZ_j = \sum_{k=1}^r wz_{jk}$	Equação 31
--------------------------------------	---	------------

---

A formalização do Grau de Simultaneidade de uma transformação  $j$  conduz ao parâmetro  $GS_j$ , representado pela Equação 32, relativa ao quociente entre o número de recursos utilizados para a execução de uma transformação e o número total de recursos candidatos a essa execução.

---

Grau de Simultaneidade de uma Transformação	$\forall j \in \{1, \dots, n\} : GS_j = \frac{\sum_{k=1}^r wz_{jk}}{\sum_{k=1}^r cr_{jk}}$	Equação 32
---	--	------------

---

Este parâmetro terá um limite superior de um (1) e um limite inferior superior a zero.

---

Limites	$\left( \sum_{k=1}^r cr_{jk} \right)^{-1} \leq GS_j \leq 1$	Equação 33
---------	---	------------

---

## 1.4 Princípios de Simultaneidade

Segundo a definição de Produção Simultânea, a afectação de recursos é efectuada para todas as transformações de uma encomenda, como se nenhuma outra existisse, pelo que se diz que se utilizam simultaneamente todos os recursos do sistema para uma encomenda. Assim sendo, a programação de produção para a satisfação de uma encomenda só se pode executar considerando a capacidade livre dos recursos após programação das encomendas anteriores. Como foi referido atrás, neste trabalho consideramos o lote de produção equivalente a uma encomenda de um produto. Assim, em princípio, a quantidade do lote será igual à quantidade encomendada. Se for definida alguma política de dimensionamento de lotes, então a encomenda será dividida em encomendas com quantidades equivalentes às dos lotes.

Como lote de processamento vamos considerar um conjunto de objectos que têm que ser processados numa determinada transformação  $t_j$ . A utilização simultânea de todos os recursos, para uma encomenda de cada vez, depende da forma de processamento do lote que pode ser sem partição do lote, com partição do lote (“batch splitting”) ou com sobreposição de lotes (“batch overlapping”), como indicado na Tabela 14.

Tabela 14: Produção Simultânea – Princípios de simultaneidade de processamento da encomenda

	Princípio de Processamento	Descrição
i.	Inteiro	Sem partição do lote - Se não houver partição do lote, o recurso seleccionado para a execução de uma transformação, processa completamente o lote e só depois é que se pode iniciar a execução de outra transformação noutra recurso.
ii.	Partição	Separar o lote em mais do que uma unidade de processamento.
(a)	Interrupção	Interrupção temporal do processamento do lote numa transformação, em unidades que serão processadas em diferentes períodos temporais.
(b)	Separação	Partição do lote em unidades separadas de processamento, que serão processadas em diferentes recursos alternativos.
iii.	Sobreposição	A sobreposição de lotes existe quando algumas unidades de um lote, depois de processadas, são transferidas para processamento em outro recurso. No entanto, o lote é totalmente processado, sem interrupção, em cada recurso.

O princípio de processamento inteiro de lote representa o processamento de um conjunto de objectos numa transformação por um recurso, sem interrupção do processamento e sem transferência de objectos para outros recursos enquanto não terminar o processamento neste. A interrupção temporal de execução do lote pode existir entre transformações, existindo assim, algum tempo de espera entre transformações no processamento desse conjunto de objectos. O processamento do lote apenas se pode executar em paralelo em transformações, do processo de produção, que possam ser executadas em paralelo. Este será o princípio com menor exigência de determinação de uma solução, de sincronização dos lotes e de controlo dos trabalhos.

A aplicação do princípio de partição de lotes com interrupção do processamento permite que o lote seja processado no recurso em períodos não contíguos. Pode mesmo, ser considerado o não processamento completo de uma peça, que terá que continuar num instante posterior. Entretanto o recurso poderá processar outros lotes. As soluções obtidas poderão ser extremamente fragmentadas, obrigando a mudanças frequentes do processamento executado pelo recurso, e à introdução de processos de controlo complexos. A interrupção do trabalho poderá obrigar à existência de um mecanismo de *congelamento* do estado de processamento, se possível, para posterior recomeço no ponto de paragem. Este mecanismo seria similar aos mecanismos de interrupção, utilizados na partilha de um processador, nos sistemas operativos de computadores. Serão, em princípio, soluções com grandes implicações ao nível dos tempos de preparação de trabalhos e com elevada complexidade de sincronização e controlo da produção.

O princípio de partição com separação dos lotes permite enviar conjuntos de objectos para processamento em qualquer recurso. Poderão obter-se soluções equivalentes às do princípio anterior no caso de se considerar partições unitárias. No entanto, não haverá necessidade de mecanismos de congelamento do estado de processamento, porque cada partição será um novo lote. Com excepção desta diferença, as exigências de sincronização e controlo serão idênticas às do caso anterior. Em ambos os casos se poderá explorar o processamento paralelo de um lote por recursos alternativos.

A sobreposição do processamento de lotes permite, antes da conclusão do processamento de todas as unidades do lote, transferir uma parte dos objectos já processados de um lote num recurso, para processamento num recurso de uma transformação posterior. A quantidade de transferência é inferior ao tamanho do lote. A aplicação deste princípio terá que considerar, para os recursos, a existência de armazéns<sup>38</sup> de entrada e de saída, que permitam sincronizar os trabalhos de processamento de um lote (ver

---

<sup>38</sup> "Buffers".

trabalho de Almeida (2002)). Poderão obter-se soluções com processamento simultâneo de um lote em diferentes recursos, mas o recurso processará ininterruptamente cada lote. Estas soluções, trarão, em princípio, menores exigências do ponto de vista da sincronização e controlo dos trabalhos do que a aplicação de princípios de partição e não incorrem em aumento de custos de preparação.

Se for possível, também se pode aplicar simultaneamente mais do que um princípio, desde que haja vantagem relativamente ao instante de conclusão da encomenda, ou outro qualquer critério que se considere relevante.

## **2. CONFIGURAÇÃO DINÂMICA DO SISTEMA**

O satisfação de uma encomenda de um produto pressupõe a entrega desse produto num tempo, com um custo e com a qualidade considerados aceitáveis. Um dos pressupostos da satisfação da encomenda é o de entrega do produto, que só é possível pelo recurso a um sistema de produção. O sistema de produção é, neste trabalho, considerado distribuído, sendo necessário efectuar o seu projecto, aqui denominado de configuração dinâmica. A configuração é dinâmica porque está associada à estrutura do produto encomendado, podendo variar para cada encomenda colocada, mesmo que sejam do mesmo produto e nas mesmas quantidades. Esta configuração resulta dos recursos seleccionados para a execução de cada transformação de mudança de estado do produto, não estando, portanto, relacionada com a implantação física do sistema. Por não estar relacionada com a implantação física é um conceito equivalente ao utilizado nas células virtuais conforme descrito por Mclean and Brown (1987).

As cinco etapas fundamentais da resolução de problemas de atribuição de recursos distribuídos referidos por Tharumarajah (2001) são apresentados na Tabela 15. Na etapa 1 decompõem-se as encomendas em operações, que são relacionadas com as máquinas na etapa 2, verificando a possibilidade de execução. Na etapa 3 selecciona-se uma máquina para a execução da cada operação. Na etapa 4 envia-se e confirma-se a afectação das operações às máquinas. A coordenação da afectação às várias máquinas resulta da etapa 5, que inclui a construção do programa de produção. Na realidade, os problemas resolvidos nestas etapas, têm que ser solucionados sempre que é necessário afectar trabalhos a recursos e construir um programa de produção.

Tabela 15: Etapas de afectação de recursos distribuídos, segundo Tharumarajah (2001).

Etapas	Acções
1	Decompor encomendas em operações.
2	Relacionar operações com máquinas que as possam executar.
3	Seleccionar máquinas para executar as operações.
4	Afectar operações a máquinas.
5	Coordenar a afectação e construir o programa de produção.

Relativamente à etapa 1, no modelo de Sistemas Distribuídos de Produção que se apresenta neste trabalho, considera-se que a encomenda diz respeito apenas a um produto, e a sua decomposição em operações está associada à estrutura do produto e às transformações de mudanças de estado dos elementos dessa estrutura, de acordo com a descrição apresentada na Secção III - 1.2. Nesta etapa é criada uma *configuração genérica* do sistema, com construção do modelo agregado, fractal, do sistema, em que se definem as transformações a executar para satisfação de cada encomenda e se faz a sua representação por

Redes de Petri, de acordo com a formalização apresentada na secção III - 2.1 e em particular na secção III - 2.1.4.i - Configuração Genérica.

No modelo MSDP, a etapa 2 corresponde à definição do domínio dos recursos disponíveis como o conjunto de todos os recursos distribuídos, que tenham habilidade para executar alguma(s) transformação(s) e que se declarem candidatos. Na etapa 3, seleccionam-se recursos para a execução das transformações, com determinação do período de execução, em função de critérios de atribuição predefinidos. Desta forma, também se resolve o problema de programação da produção, já que se determinam os instantes de início e conclusão da execução de uma transformação por parte de um recurso, executando-se assim parte do passo 5. As etapas 3 e 5 podem resultar da aplicação genérica dos processos de selecção local ou global, descritos na secção III - 1.2.3, de criação de uma *configuração específica* do sistema. Estes processos carecem, no entanto, de uma definição que permita a sua formalização. Uma abordagem sumária a esta formalização efectuou-se através do caso AZ\_FFS, secção III - 3.5, de exemplificação da aplicabilidade do modelo MSDP.

A etapa 4 e a coordenação da afectação e subsequente controlo da execução das operações, referidos na etapa 5, não são um problema tratado no âmbito do modelo do Sistema Distribuído de Produção nem da Produção Simultânea, porque se consideram dependentes das estratégias operacionais, devendo ser tratadas ao nível de operação de um sistema. O modelo MSDP pode, no entanto, ser utilizado na malha de realimentação de um sistema de controlo, como instrumento de reconfiguração do sistema.

## **2.1 Configuração Específica**

A configuração específica do Sistema Distribuído de Produção corresponde a uma configuração do sistema baseada nos recursos de produção que podem satisfazer a encomenda em causa, isto é, corresponde a uma afectação de recursos específicos às células autónomas genéricas. A criação desta configuração específica requer, portanto, uma estratégia de afectação de recursos, relacionada fundamentalmente com: critérios para selecção de recursos e modelos de afectação a adoptar, nomeadamente o modelo local ou o modelo global que adiante se clarificam.

A configuração dependerá então do objectivo a atingir, e dos critérios estabelecidos. Podem aplicar-se inúmeros critérios simples ou combinados, relacionados, por exemplo, com o prazo de entrega da encomenda, com a utilização dos recursos, com os custos de produção, com os custos de posse e de preparação ou com prioridades estabelecidas.

O estabelecimento de uma configuração específica dependerá ainda de se considerarem modelos de afectação globais, ou locais. Pela aplicação de um *modelo global* aborda-se o problema de afectação de recursos, considerando simultaneamente todos os recursos candidatos e todas as transformações envolvidas na produção do produto encomendado. Pela aplicação de um *modelo local* aborda-se o problema de afectação de recursos por aplicação sucessiva de critérios locais, isto é, critérios de afectação de recursos para uma transformação de cada vez.

A aplicação destes critérios em ambiente de Produção Simultânea exige a aplicação de princípios de simultaneidade descritos na secção 1.4 deste capítulo. Várias soluções de configurações específicas de acordo com estes princípios são exemplificadas a seguir.

### 2.1.1. Exemplo

#### CONFIGURAÇÃO GENÉRICA

Para exemplificar o processos de obtenção de uma configuração genérica, secção III - 2.1.4.i e para apresentar várias configurações específicas, que estão de acordo com os princípios de Produção Simultânea, vamos utilizar uma estrutura de produto simples, representada na Figura 35. Estes exemplos servem de ilustração para os princípios de Produção Simultânea, não se apresentando modelos de obtenção das configurações específicas, que serão desenvolvidos posteriormente.

Neste exemplo, o produto resulta de uma operação de montagem de P3 e P4 em P5 e duas transformações de P1 e P2, respectivamente, em P3 e P4. A quantidade utilizada de cada objecto para a produção do objecto ascendente é indicada por Q.

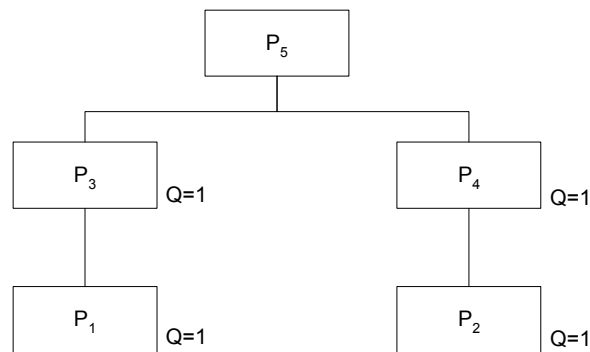


Figura 35: Configuração genérica: estrutura simples de um produto.

A aplicação do procedimento 1 do algoritmo de Configuração Genérica (página 65) resulta na Rede de Petri apresentada na Figura 36. Esta rede e todas as que são geradas por este algoritmo, têm uma formalização algébrica equivalente à que foi apresentada no capítulo anterior. Considerando o número de objectos  $m=5$ , os objectos  $o_i, i=1..m$  correspondem aos produtos  $P_{\#}$  da estrutura do produto. Considerando, também, o número de transformações  $n=3$ , as transformações  $t_j, j=1..n$  correspondem às mudanças de estado de um produto para outro da estrutura da Figura 35.

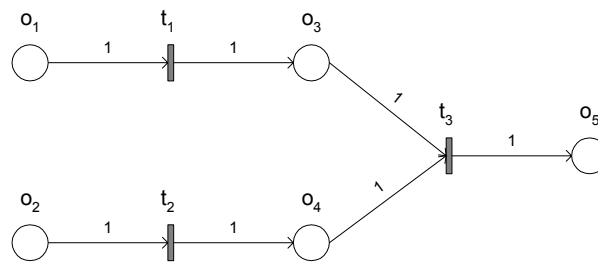


Figura 36: Configuração genérica: Rede de Petri do passo 1.

A aplicação do procedimento 2 do algoritmo de configuração genérica resulta na Rede de Petri apresentada na Figura 37. Os objectos  $o_1, o_2$  e  $o_3$  são equivalentes aos objectos  $o_1, o_2$  e  $o_5$  da figura anterior. A transformação  $t_1$  agrega todas as transformações anteriores. A única transformação da Figura 37 entrega o produto P5, e a produção de P5, exige a disponibilidade de P3 e P4, conforme se pode verificar na Figura 35.

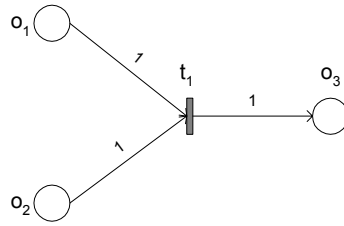


Figura 37: Configuração genérica: Rede de Petri do passo 2.

A execução do procedimento 4.1.1 resulta na expansão da única transformação da Figura 37 nas transformações que entregam os produtos imediatamente necessários para a produção de P5. Esses produtos são P3 e P4, que resultam da execução das transformações  $t_1$  e  $t_2$ . Assim sendo, em 4.1.2 obtém-se um subsistema de produção que corresponde ao nível mais baixo de agregação, portanto idêntico ao representado na Figura 36.

Neste exemplo só existem dois níveis de agregação que se obtém numa única iteração do algoritmo. Se existissem mais níveis seria necessário executar mais iterações.

### CONFIGURAÇÃO ESPECÍFICA

A obtenção de configurações específicas para este exemplo depende das quantidades encomendadas do produto, dos dados sobre os recursos, dos princípios de Produção Simultânea utilizados e do modelo de afectação de recursos.

Se houver uma encomenda de 5 unidades do produto P5, da Figura 35, para satisfazer, é necessário obter 5 unidades de todos os objectos representados na Rede de Petri da Figura 36. Para tal, é necessário seleccionar recursos para executar as transformações  $t_1$  a  $t_3$ . Neste exemplo consideram-se três recursos  $r_1$  a  $r_3$  como candidatos para executar as três transformações. Na Tabela 16 apresenta-se o tempo que cada recurso demora a executar uma unidade de quantidade de trabalho de cada transformação. Neste caso, executar uma unidade de quantidade de trabalho equivale a entregar uma unidade de objecto. Assim sendo, o recurso  $r_1$  demora 10 unidades de tempo para entregar 5 unidades do produto P5 como resultado da execução da transformação  $t_3$ . A agenda do recurso é representada por um conjunto de pares de elementos correspondentes aos instantes de início e conclusão de cada período de ocupação do recurso.

Tabela 16: Tempos unitários de processamento e agenda de ocupação dos recursos.

	$t_1$	$t_2$	$t_3$	Agenda de períodos ocupados em unidades de tempo
$r_1$	2		2	$Ag_1 = \{(0,5), (20,25), (35,50)\}$
$r_2$	4	2	1	$Ag_2 = \{(5,10), (20,25), (30,45)\}$
$r_3$	3	2	2	$Ag_3 = \{(0,15), (45,50)\}$

Considerando a agenda de ocupação dos recursos, representada no gráfico de Gantt da Figura 38 entre os instantes de tempo 0 e 50, podem obter-se inúmeras soluções alternativas dependentes do princípio de simultaneidade utilizado. Nas figuras seguintes, em que os períodos com fundo mais escuro representam os períodos de ocupação dos recursos, dão-se exemplos de soluções que se podem obter com aplicação de princípios de simultaneidade alternativos, não se apresentando métodos para atingir essas soluções.



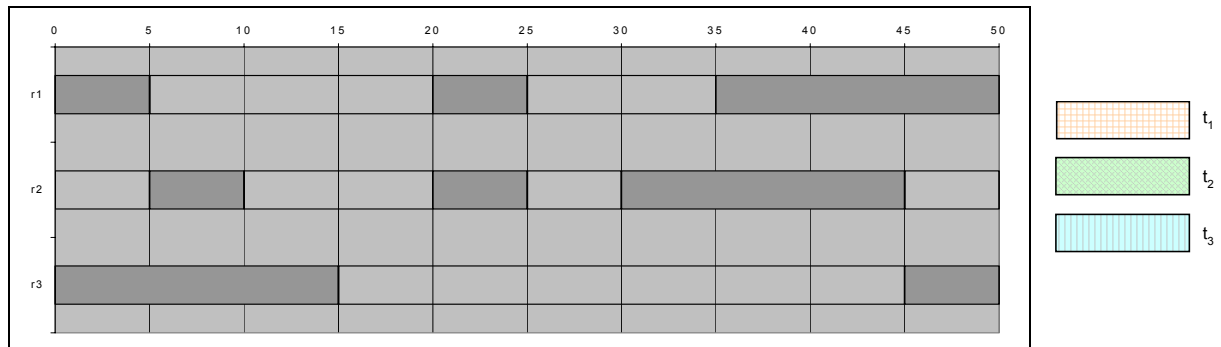


Figura 38: Gráfico de Gantt - Agenda dos recursos candidatos.

Aplicando o princípio “i” da Tabela 14, secção IV - 1.4, pode obter-se uma solução apresentada no gráfico de Gantt da Figura 39, com processamento completo de cada lote por cada recurso. A execução da transformação  $t_2$ , seja por  $r_2$  ou por  $r_3$ , tem sempre a duração de 2 UT. A conclusão dessa transformação o mais cedo possível resulta da sua afectação a  $r_2$ . Em função desta afectação é possível terminar o processamento da encomenda no instante 30.

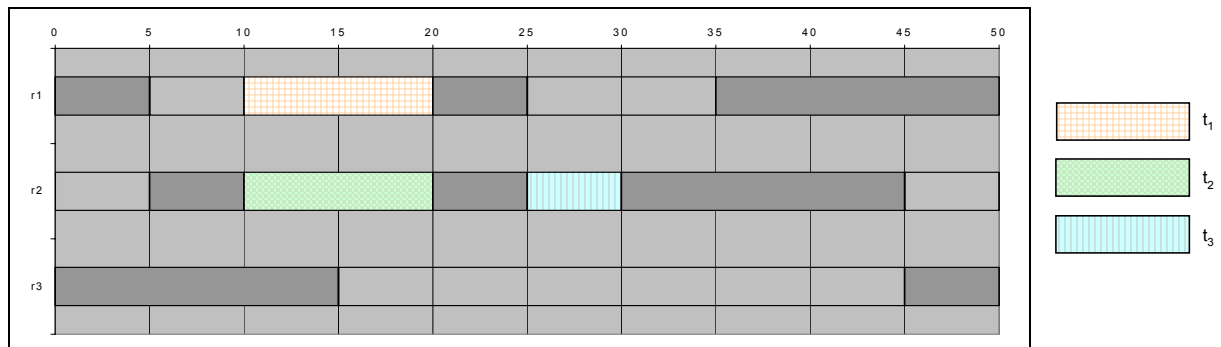


Figura 39: Gráfico de Gantt – Aplicação do princípio (i).

Pela aplicação do princípio de interrupção do lote pode-se obter a solução representada pelo gráfico de Gantt da Figura 40. O processamento da transformação  $t_2$ , com cor verde, no recurso  $r_2$  é interrompido no período 5 a 10, permitindo o início do processamento de  $t_3$  no instante 15 e a conclusão do processamento de todos os lotes no instante 20. O processamento de uma unidade da transformação  $t_2$  no recurso  $r_2$  demora 2 unidades de tempo. Logo, o processamento da terceira unidade do lote de 5 unidades da transformação  $t_2$  é interrompido a meio e recomeça no instante 10.

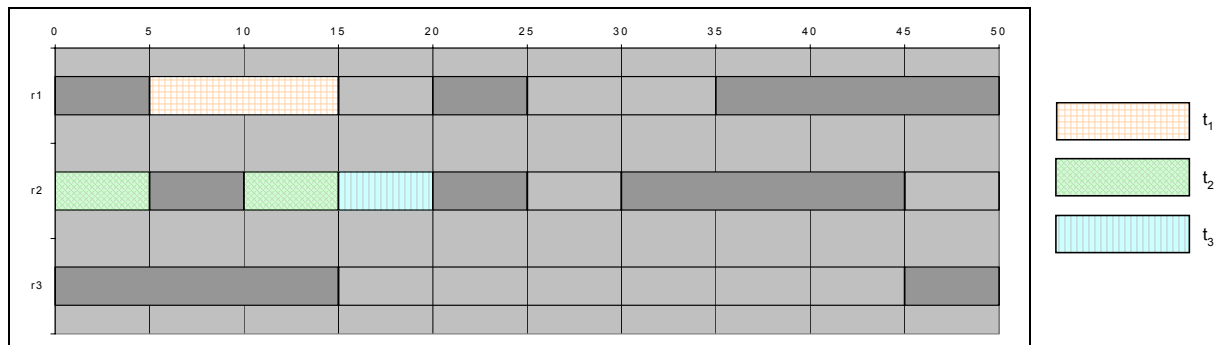


Figura 40: Gráfico de Gantt – Aplicação do princípio (ii.a).

No gráfico de Gantt da Figura 41 pode verificar-se a exploração do processamento paralelo da mesma transformação em recursos alternativos, nomeadamente de  $t_2$  em  $r_2$  e  $r_3$  e de  $t_3$  em todos os recursos. Uma

unidade de quantidade de trabalho de  $t_3$  é executada em duas unidades de tempo do recurso  $r_1$  e nos recursos  $r_2$  e  $r_3$  processam-se duas unidades de quantidade de trabalho em cada um, utilizando, respectivamente, duas e quatro unidades de tempo. Com esta solução conclui-se o processamento de todas as unidades no instante de tempo 22.

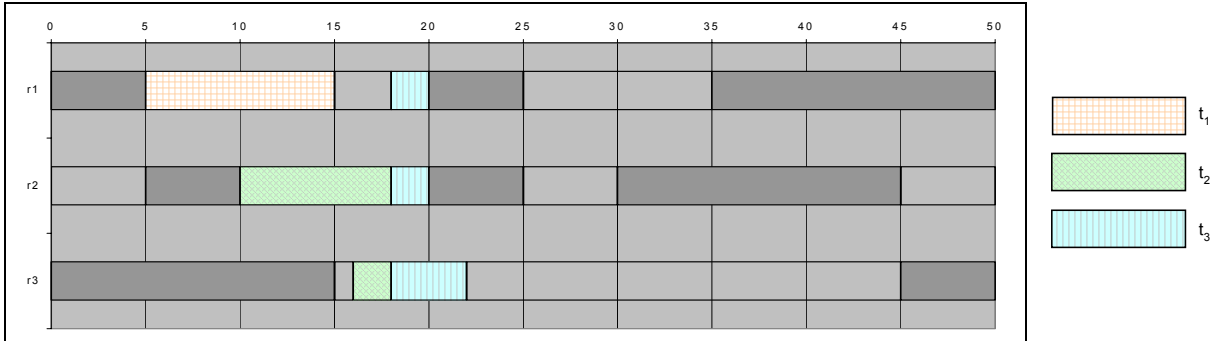


Figura 41: Gráfico de Gantt – Aplicação do princípio (ii.b).

O gráfico de Gantt da Figura 42 representa uma solução com sobreposição do processamento de lotes, em que se define, para exemplificação, um lote de transferência de 3 unidades. Dessa forma, no instante de conclusão do processamento de 3 unidades nas transformações  $t_1$  e  $t_2$  em  $r_1$  e  $r_2$ , respectivamente, inicia-se o processamento de  $t_3$  em  $r_3$ . Dessa forma é possível concluir o processamento das 5 unidades no instante de tempo 26.

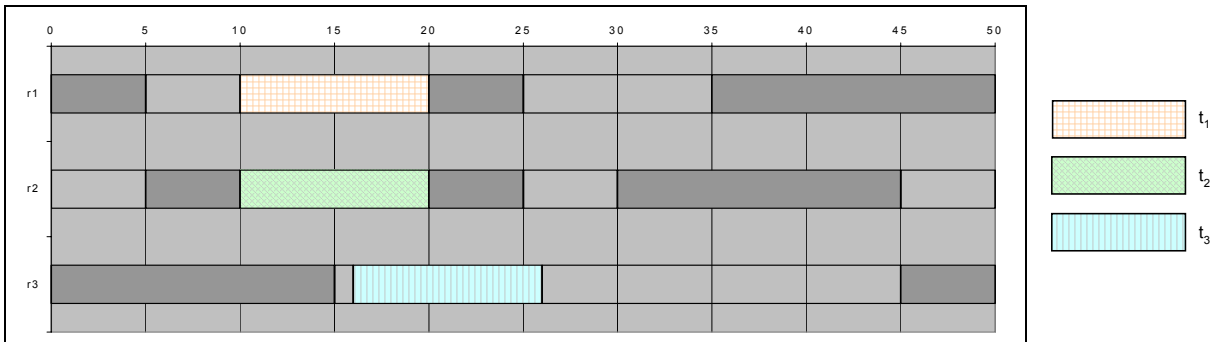


Figura 42: Gráfico de Gantt – Aplicação do princípio (iii).

## 2.2 Possibilidades de Configuração do Sistema

O modelo MSDP, de acordo com a PS, deve permitir analisar a influência da Produção Simultânea na configuração do sistema distribuído de produção, permitindo dar resposta a questões como:

- De que forma a aplicação da Produção Simultânea afectará a configuração do sistema?
- De que forma se comportará o prazo de entrega relativamente a custos dos sistema?

A configuração do sistema depende dos critérios utilizados e da aplicação de modelos de afectação locais ou globais. Serão, nesta secção, definidas algumas possibilidades de configuração e comportamento do sistema, relativas à afectação de recursos para uma transformação de cada vez, modelo local, e para todas as transformações simultaneamente, modelo global. Os modelo local e o modelo global são baseados nos processos de configuração do SDP, apresentados na secção III - 1.2.3.i, e serão especificados mais adiante neste capítulo.

### 2.2.1. Modelo Local de Afecção de Recursos

No modelo local de afecção de recursos para a criação de uma configuração específica para o Sistema Distribuído de Produção, consideram-se as necessidades de quantidade de trabalho de uma transformação de cada vez. A minimização do instante de conclusão de cada ordem de produção em cada transformação pressupõe, na maioria dos casos, a utilização simultânea de mais do que um recurso para execução em paralelo das unidades do produto encomendadas. Estaremos, assim, a recorrer ao princípio de simultaneidade de partição de encomendas (ii.b). Pretende-se analisar a forma de variação do instante de conclusão de cada transformação em relação à variação do Grau de Simultaneidade, sendo este directamente proporcional ao número de recursos utilizados nessa transformação, como se definiu na secção IV - 1.3.2.

Para análise da variação do instante de conclusão, considere-se o seguinte cenário:

- Uma encomenda de  $m$  produtos.
- Um conjunto de  $r$  recursos.
- Todos os recursos têm a agenda totalmente livre.
- Uma transformação para processar os  $m$  produtos e que produz uma unidade desse produto por cada unidade de trabalho executada; assim, para produzir  $m$  produtos é necessário executar uma quantidade de trabalho  $w = m$  nesta transformação.
- Cada recurso demora o mesmo tempo  $c$  a processar cada produto na transformação, e  $m \cdot c$  se processar todos os produtos.

Neste cenário podem verificar-se três situações diferentes:  $m < r$ ,  $m = r$  e  $m > r$ . Na primeira situação podem utilizar-se de um até  $m$  recursos, passando por  $m/2$ , obtendo-se instantes de conclusão respectivamente iguais a  $m \cdot c$ ,  $m \cdot c/m$  e  $m \cdot c/(m/2)$ . Nesta situação é possível utilizar, no máximo, um número de recursos igual ao número de objectos a produzir, sendo os restantes não utilizados. Na segunda situação obtém-se valores iguais aos da primeira situação, podendo utilizar-se todos os recursos. Na terceira situação podem utilizar-se de um, até  $r$  recursos passando por  $r/2$ , obtendo-se instantes de conclusão respectivamente iguais a  $m \cdot c$ ,  $m \cdot c/r$  e  $m \cdot c/(r/2)$ , desde que o número de objectos seja divisível por  $r$  e por  $r/2$ . Nesta situação, aumentando o número de recursos utilizados obtém-se reduções do instante de conclusão, até valores que não serão tão reduzidos como nas duas situações anteriores. Considera-se, simplificando, que o prazo de entrega da encomenda é definido pelo instante de conclusão do processamento da mesma. Analisando e extrapolando estes resultados é possível estabelecer a curva de variação do instante de conclusão, isto é, prazo de entrega (PE), em função da variação do número de recursos utilizados  $wz$ :

Na primeira e segunda situações	$PE = c \cdot w/wz \quad \text{com } wz \in \{1, \dots, m\}$
Na terceira situação	$PE = c \cdot w/wz \quad \text{com } wz \in \{1, \dots, r\}$

No cenário em análise a variação do instante de conclusão é do tipo da função  $1/x$ , apresentada na Figura 43. Neste tipo de curva dá-se uma redução elevada para valores baixos da variável, passando a reduções menos acentuadas para valores mais elevados da variável. Isto reflecte-se num decréscimo de ganhos conforme se aumenta o número de recursos utilizados.

Com condições similares à do cenário descrito anteriormente, mas com duas transformações em paralelo ainda se obtém uma curva de variação do instante de conclusão do tipo  $1/x$ , tendo as transformações,

recursos alternativos dedicados. Com condições similares à do cenário descrito anteriormente, com recursos alternativos dedicados para duas transformações em sequência obtém-se uma curva de variação do instante de conclusão do tipo da função  $1/x$ . Neste caso, esta curva é igual a  $2 \cdot m \cdot c / wz$ , com aumento do instante de conclusão, devido à exigência de execução da segunda transformação após o término da primeira.

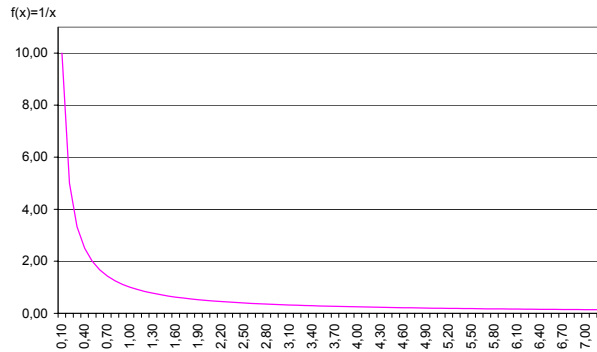


Figura 43: Curva de variação da função  $1/x$ .

Em condições mais próximas das condições impostas por ambientes industriais, a utilização dos recursos depende da sua agenda e da possibilidade de utilização em diferentes transformações. No entanto, prevê-se uma variação da função de minimização do instante de conclusão similar ao deduzido anteriormente. A possibilidade de variação do prazo de entrega em função da alteração do número de recursos, pela aplicação do princípio de simultaneidade (ii.b) de partição de encomendas, está representada na curva da Figura 44. Nesta curva o número de recursos na transformação  $j$  é representado por  $wz_j$  e o resultado da minimização do prazo de entrega por  $PE$ . Se as restrições de agenda o permitirem, a utilização de um único recurso de produção por transformação ( $wz_{jmin} = 1$ ) reflecte-se no máximo prazo de entrega  $PE_{max}$  da encomenda associada a essa transformação. Se um único recurso não puder executar a quantidade de trabalho requerida pela transformação então o máximo prazo de entrega resulta da utilização de  $wz_{jmin}$  recursos. Como foi referido prevê-se que esta curva seja similar à função  $1/x$ . Nestas curvas, a partir de determinado ponto, obtêm-se pequenas reduções de prazo de entrega em direcção ao mínimo  $PE_{min}$ , com o aumento do número de recursos envolvidos.

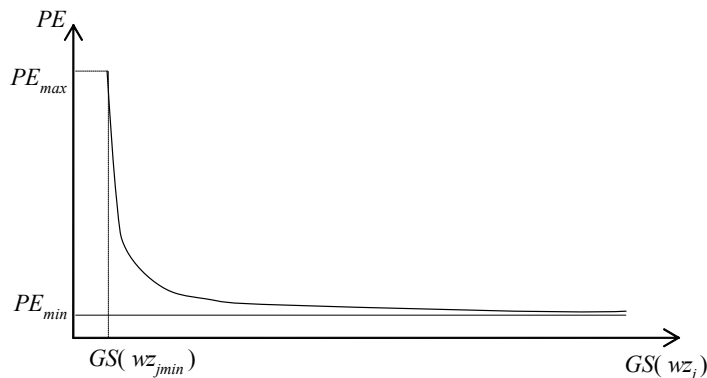


Figura 44: Modelo local - Prazo de Entrega vs Grau de Simultaneidade.

Não sendo objectivo deste trabalho, a modelação do custo dos recursos envolvidos, é no entanto, importante efectuar considerações acerca desse custo, que servem para aprofundar o conhecimento do

comportamento dos sistemas de produção perante a Produção Simultânea. Supõe-se que cada recurso de produção propõe um custo que considera o custo de obtenção dos seus componentes, o custo de produção interna e o custo de distribuição. O tempo de entrega que propõe considera o tempo de obtenção dos seus componentes, o tempo de preparação, o tempo de processamento interno e o tempo de entrega à célula que encomendou aquele produto.

Se considerarmos que cada recurso apresenta o custo de entrega da encomenda no local do cliente, os custos de preparação e de distribuição terão tendência a aumentar com a diminuição do prazo de entrega, devido ao aumento do número de recursos envolvidos. Isto deve-se ao facto de se aumentar o número de operações de distribuição, e ao facto de poderem existir operações de preparação de produção em maior número. Além do aumento de operações de distribuição, supõe-se que a diminuição do tempo de entrega pelo aumento do número de recursos, provavelmente, aumentará a distância entre os recursos envolvidos no sistema, contribuindo para o aumento do custo de distribuição devido ao aumento da distância. No entanto, poderão existir vantagens relacionadas com a diminuição do prazo de entrega que se podem reflectir sobre os custos do recurso de produção, nomeadamente: diminuição dos custos relacionados com a posse dos produtos, por diminuição do tempo de posse; aumento da satisfação dos clientes, pela possibilidade de redução do prazo de entrega.

### **2.2.2. Modelo Global de Afectação de Recursos**

A selecção de recursos simultâneos para mais do que uma transformação, tem como requisito a utilização de um modelo de optimização que permita respeitar critérios globais, i.e. em relação ao sistema todo ou a todas as transformações consideradas em conjunto. Neste caso, haverá necessidade de conhecimento das agendas dos recursos envolvidos, e estaremos perante um sistema de células com menor autonomia do que no caso do modelo local. As células terão que aceitar disponibilizar as suas agendas para determinação de um programa de produção de uma encomenda.

A afectação global de recursos, isto é, afectação simultânea para todas as transformações, deve considerar restrições relacionadas com a agenda dos recursos e com a precedência e paralelismo de execução das transformações. Estas restrições impõem dependências entre quantidades de trabalho atribuídas aos recursos que afastam este modelo do modelo anterior, e do cenário mais simples equacionado na secção anterior. No entanto, pode-se admitir, que o prazo de entrega terá um comportamento similar, diminuindo desde um valor máximo  $PE_{max}$  até um determinado limite  $PE_{min}$ , com o aumento do número de recursos  $WZ$  seleccionados. Assim sendo, previsivelmente, a variação da minimização do prazo de entrega  $PE$  em função do Grau de Simultaneidade  $GS$ , que é, por definição, directamente proporcional ao número de recursos utilizados  $WZ$ , terá um comportamento similar ao do modelo local. O número de recursos para o qual  $PE$  é máximo é designado por  $WZ_{min}$ . A variação de  $PE$  será assim, similar ao de uma função do tipo  $1/x$ , conforme ilustração da Figura 45. Esta curva terá uma tendência de variação tal que, a partir de determinado ponto, o aumento do número de recursos envolvidos provocam, substancialmente, menores reduções de prazo de entrega, isto é, existe uma diminuição dos ganhos na redução do prazo de entrega.

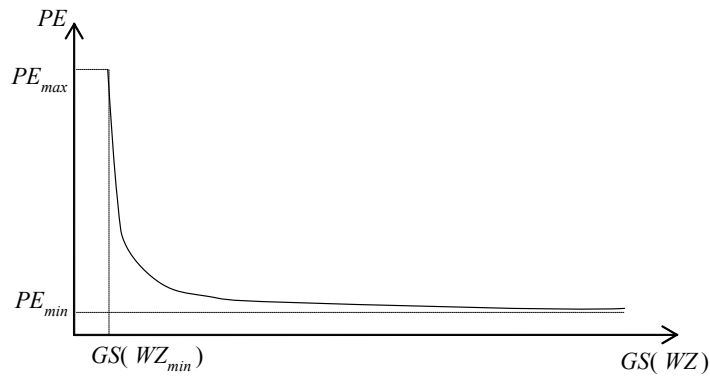


Figura 45: Modelo global - Prazo de Entrega vs Grau de Simultaneidade.

Num sistema de produção, as actividades de transporte e preparação da produção estão relacionadas com a implantação física estabelecida e com os roteiros de fabrico. Considera-se, neste modelo, o estudo de sistemas de produção com actividades de transporte e preparação negligenciáveis face a outras variáveis, e dessa forma aborda-se o problema de selecção dinâmica de recursos para redução de prazos de entrega, sem considerar a modelação explícita destas actividades.

Não existindo o objectivo de modelação dos custos de cada célula e do sistema, pretende-se analisar de uma forma simplificada os custos de preparação e posse, nos sistemas formados pela aplicação de um modelo global. Neste análise, admite-se que as actividades de transporte e preparação não são influenciadas pela ordem de utilização dos recursos, e que a decomposição de uma encomenda em vários lotes a fabricar em diferentes recursos, previsivelmente aumentará os custos de transporte e de preparação. Admite-se portanto, que os custos de preparação e de transporte serão proporcionais ao número de recursos utilizados, logo ao Grau de Simultaneidade  $GS$ . Ambos os custos de transporte e de preparação, neste modelo simplificado, são representados por  $C_s$ , e referidos por custo de preparação. A variação do custo de preparação  $C_s$  com o Grau de Simultaneidade  $GS$  pode ser representada por curvas similares à curva da esquerda da Figura 46.

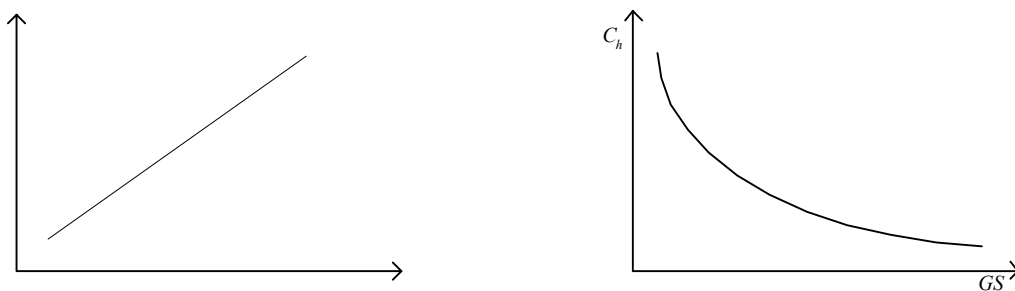


Figura 46: Modelo global - Custos de Preparação e de Posse vs Grau de Simultaneidade.

Por sua vez, os custos de posse estão relacionados, entre outros, com o número de unidades, o tempo em que as unidades estão armazenadas e o custo unitário de armazenamento. O custo unitário de armazenamento corresponde ao custo de armazenar, durante uma unidade de tempo, uma unidade de um objecto do sistema de produção, i.e. matérias primas, componentes ou produtos finais. Este custo unitário depende, entre outros, dos custos de gestão do armazenamento e do valor do produto armazenado. Supondo, para modelação simplificada do custo de posse, que o custo de gestão do armazenamento é constante, então

o custo de posse pode modelar-se como uma função do valor do objecto armazenado, do tempo de armazenamento e do número de objectos. Nesta modelação, considera-se que o valor de cada objecto aumentará com a execução de cada transformação, que terá de ser contabilizado na determinação do custo de posse da encomenda. O aumento do número de recursos envolvidos na produção do produto para satisfação de uma encomenda deverá diminuir, previsivelmente, o prazo de entrega, logo o tempo de armazenamento dos objectos e dessa forma o custo de posse. A curva de variação do custo de posse  $C_h$  com o Grau de Simultaneidade  $GS$  será previsivelmente do tipo da curva representada na Figura 46.

### **3. FORMULAÇÃO DO PROBLEMA DE SELECÇÃO DE RECURSOS**

Num sistema orientado à satisfação da encomenda de um produto, a afectação de recursos para a execução das transformações necessárias, é efectuada para uma encomenda de cada vez. Num ambiente de Produção Simultânea, além da satisfação de uma encomenda de cada vez, a selecção de recursos deve respeitar algum princípio de simultaneidade. Neste trabalho explora-se, fundamentalmente, a aplicação da partição com separação de lotes, que corresponde ao princípio de simultaneidade (ii.b) apresentado na Tabela 14. Além da utilização deste princípio de simultaneidade também se formula o problema de afectação de uma única transformação segundo o princípio (i).

Para satisfação de uma encomenda é necessário executar um conjunto de transformações e para execução destas transformações é necessário seleccionar recursos para todas elas. A afectação de recursos pode efectuar-se para uma transformação de cada vez, através do modelo local, ou para todas as transformações simultaneamente, através do modelo global. Assim sendo, nesta secção apresenta-se a formulação, para modelos locais ou globais de afectação de recursos, do problema de optimização da afectação de quantidades de trabalho de execução de transformações a recursos distribuídos.

#### **3.1 Modelo Local de Afectação de Recursos**

Para execução de uma transformação de um lote de produção, aplicando os princípios de simultaneidade (i) e (ii.b) apresentado na Tabela 14, pode seleccionar-se um recurso que execute completamente o lote ou então, atribuir partes do lote a recursos diferentes. Assim sendo, teremos duas alternativas a analisar, que podem ser descritas da seguinte forma:

- i.  $1 \times 1$  - Uma transformação executada por um recurso de produção;
- ii.  $1 \times q$  - Uma encomenda dividida em  $q$  unidades e a respectiva transformação executada por  $q$  recursos;

##### **3.1.1. Estratégia de Selecção de Recursos $1 \times 1$**

Conforme descrito nos objectivos do trabalho, para efeitos de afectação de recursos aplica-se o critério principal de minimização do prazo de entrega  $PE$ . Determina-se o *prazo de entrega* pelo cálculo da diferença entre o instante de tempo em que se entrega o produto encomendado nas quantidades  $q$  encomendadas e o instante que se define como de início do processamento. A selecção de um único recurso para execução de uma transformação baseia-se neste critério. Embora seja dado realce ao critério de prazo de entrega, pretende-se também formular o processo de selecção de recursos relativamente ao custo total de encomenda

$Cte$ , definido como o custo de executar a produção e entregar a referida encomenda, determinado por cada recurso. Este custo total de encomenda é diferente do custo de encomenda utilizado nos modelos de gestão de quantidades de objecto armazenadas, associado ao custo de preparação ou de efectuar a encomenda.

Cada transformação  $j$  pode ser executada por um determinado número máximo de recursos  $r_j < r$ , com o primeiro desses recursos representado pelo índice  $k_{j,1}$  e o último por  $k_{j,r_j}$ . O conjunto dos recursos candidatos à execução da transformação  $j$  é um subconjunto de todos os recursos do sistema, conforme definição da Equação 34.

$$\{k_{j,1}, \dots, k_{j,r_j}\} \subseteq \{1, \dots, r\} \quad \text{Equação 34}$$

### **i. Menor Prazo de Entrega**

A aplicação do objectivo de minimização do prazo de entrega, leva à selecção do recurso  $k$  que apresentar o menor instante de conclusão para a transformação  $j$ . O prazo de entrega será, então, dado pela Equação 35 e o custo da encomenda será dado pela Equação 36.

$$PE = \min(PE_k) \quad \text{com } k \in \{k_{j,1}, \dots, k_{j,r_j}\} \quad \text{Equação 35}$$

$$Cte = Cte_k \quad \text{com } k \in \{k_{j,1}, \dots, k_{j,r_j}\} \quad \text{Equação 36}$$

O prazo de entrega e o custo apresentados por um recurso para execução de uma transformação, dependem dos objectivos do recurso autónomo, assumindo-se no entanto, que foram definidos em função da capacidade disponível e da agenda do recurso.

### **ii. Menor Custo**

O recurso que apresentar o menor custo de encomenda para a transformação  $j$ , será seleccionado para executar essa transformação.

$$PE = PE_k \quad \text{com } k \in \{k_{j,1}, \dots, k_{j,r_j}\} \quad \text{Equação 37}$$

$$Cte = \text{Min}(Cte_k) \quad \text{com } k \in \{k_{j,1}, \dots, k_{j,r_j}\} \quad \text{Equação 38}$$

A utilização de critérios mistos, estabelecendo um valor limite para um dos critérios, poderá levar a soluções mais interessantes para as empresas. Dessa forma aplicam-se os mesmos critério de minimização, com restrição do outro critério. Após recepção das propostas, restringe-se a selecção aos recursos que cumpram com o valor limite imposto.

### **3.1.2. Estratégia de Selecção de Recursos 1xq**

Uma encomenda de  $qt$  unidades de um determinado produto pode dividir-se em encomendas de  $q$  unidades, com  $q \leq qt$ . Cada uma destas encomendas pode ser entregue a um recurso diferente, estabelecendo assim a possibilidade de processamento da encomenda no mesmo instante em vários recursos. Conforme descrito anteriormente, aplica-se o critério de minimização do prazo de entrega. Embora seja dado



realce a este critério, pretende-se também formular o processo de selecção de recursos relativamente ao custo total da encomenda  $C_{te}$ , definido como o custo de executar e entregar a referida encomenda, determinado por cada recurso.

Supondo que  $qt$  é múltiplo de  $q$  então o número de encomendas  $ne$  será dado pela Equação 39. Sendo cada encomenda processada num recurso, o número de recursos seleccionados é também igual a  $ne$ .

---


$$ne = qt/q \quad \text{Equação 39}$$

### **i. Menor Prazo de Entrega**

A ordenação dos recursos por ordem crescente dos prazos de entrega propostos, permite seleccionar os recursos como sendo os correspondentes aos primeiros  $ne$  da lista ordenada. O prazo de entrega da encomenda é igual ao prazo de entrega do elemento  $ne$  da lista, isto é, ao maior prazo de entrega dos primeiros  $ne$  elementos. O custo da encomenda é igual ao somatório dos custos dos primeiros  $ne$  elementos da lista.

---


$$PE = PE_{ne} \quad \text{Equação 40}$$

$$Ce = \sum_{k=1}^{ne} Ce_k \quad \text{Equação 41}$$


---

### **ii. Menor Custo**

A ordenação dos recursos por ordem crescente dos custos de encomenda propostos, permite seleccionar os recursos como sendo os correspondentes aos primeiros  $ne$  da lista ordenada. O custo de encomenda é igual ao somatório dos custos do primeiros  $ne$  elementos da lista. O prazo entrega da encomenda é igual ao máximo dos prazos de entrega dos primeiros  $ne$  elementos da lista.

---


$$PE = \max_{k=1..ne} (PE_k) \quad \text{Equação 42}$$

$$Ce = \sum_{k=1}^{ne} Ce_k \quad \text{Equação 43}$$


---

A utilização de critérios mistos, permite utilizar, nesta estratégia de selecção de recursos  $1 \times q$ , um dos dois critérios de minimização, com restrição do outro critério. Após recepção das propostas, restringe-se a selecção aos recursos que cumpram com o valor limite imposto.

## **3.2 Modelo Global de Afectação de Recursos**

A afectação simultânea de recursos a mais do que uma transformação resulta na necessidade de formular o problema de uma forma global. Um pressuposto na utilização desta formulação, é o de disponibilização das agendas e de dados sobre a capacidade produtiva por parte dos recursos, que possibilita uma tomada de decisão centralizada para todas as transformações e recursos. Esta forma de decisão permite afectar parte das agendas dos recursos ao trabalho a programar. Assim sendo, este pressuposto leva a caracterizar os elementos do Sistema Distribuído de Produção com menor autonomia do que no caso de tomada de decisão

local. A formulação do problema desta forma permitirá estudar o comportamento do modelo em ambiente de Produção Simultânea, e poderá servir como referência para soluções obtidas sem centralização do processo de decisão.

O modelo MSDP desenvolvido no capítulo anterior permite representar sistemas de produção, os seus objectos, transformações e recursos. Na modelação dos recursos e na respectiva ligação com as transformações (secção III - 2.2) impõem-se apenas duas restrições: a de satisfação da procura por parte das transformações e a de não exceder a capacidade disponível dos recursos. Esta modelação é utilizada num exemplo, em que se verifica a possibilidade de obtenção de uma configuração específica para o sistema por formulação de um problema de optimização linear (secção III - 3.5). Nesta formulação não se considera nem a agenda dos recursos, nem as restrições tecnológicas do processo de produção, mas apenas a sua capacidade livre. Neste caso, não se definem, nem verificam, períodos de execução das transformações em cada recurso, resolvendo-se, portanto, o problema de planeamento de capacidade, que permite verificar a existência de capacidade livre para execução da encomenda.

A afectação de recursos, considerando as restrições das agendas dos recursos e as restrições tecnológicas do processo, corresponde a resolver o problema de programação da produção, associado a uma encomenda, isto é, o de afectação e calendarização de recursos. Neste modelo global de afectação de recursos apresenta-se abaixo uma formulação que permite resolver este problema, considerando simultaneamente estas restrições para todas as transformações de uma encomenda. Nas secções seguintes apresentam-se as restrições do problema ( $R\#$ , em que  $\#$  representa um número sequencial) e a definição dos critérios de optimização. As restrições do problema são agrupadas em: restrições genéricas e restrições de agenda. Na secção referente às restrições genéricas apresenta-se: variáveis de decisão do problema ( $V\#$ ), o seu inter-relacionamento e os respectivos limites ( $UB\#$ ); restrições de procura e disponibilidade já definidas no capítulo anterior; restrições relacionadas com as transformações precedentes e com as transformações paralelas. Na secção referente às restrições de agenda apresenta-se a definição da agenda dos recursos, as restrições não lineares da agenda e a transformação destas em restrições lineares.

### **3.2.1. Restrições Genéricas**

Nesta secção apresentam-se as definições de variáveis de decisão para o processo de optimização e as restrições relacionadas com estas variáveis, isto é, apresentam-se tipos de variáveis e os respectivos limites. Também se apresentam restrições que não dependem da consideração da agenda dos recursos.

#### ***i. Variáveis de Decisão***

Para determinar a capacidade necessária para satisfação da encomenda em análise utiliza-se a quantidade de trabalho  $w_{jk}$  executada por um recurso  $k$  numa transformação  $j$ , como variável de decisão. Esta quantidade de trabalho tem que tomar valores inteiros, porque não se admite a possibilidade de uma parte de um produto ser processado num recurso e outra parte noutra. É importante recordar que, neste modelo, produto representa qualquer elemento da estrutura do produto final, i.e. objectos do sistema, podendo corresponder a materiais, componentes, peças, produtos semi-acabados ou produtos finais. A minimização do instante de conclusão de uma ordem de produção está associada aos instantes de início  $T_j^-$ , de conclusão

$T_j^+$ , e ao tempo de execução  $T_j$  de cada transformação. Estes instantes de tempo dependem do início, conclusão e tempo de execução de cada um dos recursos nessas transformações.

O tempo de execução  $T_j$  de uma transformação formulado na Equação 44, é determinado pela diferença entre o instante de início da transformação e o instante de conclusão, tomando sempre valores não negativos.

$$\forall j \in \{1, \dots, n\} : T_j = T_j^+ - T_j^- \quad \text{Equação 44}$$

O tempo de execução  $T_{jk}$  da transformação  $j$  pelo recurso  $k$ , apresentada na Equação 45, obtém-se pelo produto entre a quantidade de trabalho executada na transformação  $w_{jk}$ , pelo tempo de execução de uma unidade  $c_{jk}$  dessa quantidade de trabalho. A variável  $w_{jk}$  e o tempo de execução de uma unidade  $c_{jk}$  foram definidos na secção III - 2.2 - Modelação de Recursos.

$$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_{jk} = c_{jk} \cdot w_{jk} \quad \text{Equação 45}$$

▪ **Variáveis de decisão do modelo**

As seguintes variáveis de decisão podem ser utilizadas em determinadas formulações, podendo ser uma função de outras variáveis em outras formulações. Se forem uma função de outras, então podem ser uma função que se pretende otimizar ou simplesmente um resultado da aplicação do modelo.

V1 – Quantidade de Trabalho	$\forall j \in \{1, \dots, n\}, k \in \{1, \dots, r\} : w_{jk} \in \mathbb{Z}_0^+$	Equação 46
V2 – Instante de início de uma transformação	$\forall j \in \{1, \dots, n\} : T_j^- \geq 0$	Equação 47
V3 – Instante de conclusão da transformação	$\forall j \in \{1, \dots, n\} : T_j^+ \geq 0$	Equação 48
V4 – Tempo de execução da transformação	$\forall j \in \{1, \dots, n\} : T_j \geq 0$	Equação 49
V5 – Instante de início de uma transformação executada por um recurso	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_{jk}^- \geq 0$	Equação 50
V6 – Instante de conclusão de uma transformação executada por um recurso	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_{jk}^+ \geq 0$	Equação 51
V7 – Tempo de execução de uma transformação por um recurso	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_{jk} \geq 0$	Equação 52
V8 – Recurso seleccionado	$\forall j \in \{1, \dots, n\}, k \in \{1, \dots, r\} : wz_{jk} \in \{0, 1\}$	Equação 53

## ii. Limites Superiores

A determinação de limites superiores para as variáveis utilizadas poderá simplificar o processo de optimização pela redução do domínio do problema.

A quantidade de trabalho disponível  $\alpha_{jk}$  de um recurso para a execução de uma transformação, está formulada na Equação 17, e depende do tempo livre  $T_{jk}^f$  do recurso e do tempo de execução unitário  $c_{jk}$ . A quantidade de trabalho  $w_{jk}$  executada por cada recurso numa determinada transformação é não superior à quantidade de trabalho  $W_j$  requerida por aquela transformação e à quantidade de trabalho disponível  $\alpha_{jk}$ .

$$\text{UB1} \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : w_{jk} \leq \min(W_j, \alpha_{jk}) \quad \text{Equação 54}$$

O tempo de execução de uma transformação nunca excederá o tempo que o recurso mais lento, com possibilidade de executar essa transformação, demorará a executar a quantidade de trabalho requerida por essa transformação, conforme representado na Equação 55.

$$\text{UB2} \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_j \leq \max_k(c_{jk}) \cdot W_j \quad \text{Equação 55}$$

O tempo que um recurso demora a executar uma transformação nunca excederá o tempo que o recurso demoraria se executasse completamente essa transformação, isto é, se executasse o total da quantidade de trabalho dessa transformação.

$$\text{UB3} \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_{jk} \leq c_{jk} \cdot W_j \quad \text{Equação 56}$$

Considerando o recurso que demora mais tempo a executar qualquer transformação e a afectação desse recurso para execução de todas as transformações, consideradas em sequência, determina-se um tempo de execução UB4, que é um limite superior para todos os tempos de execução. Esta situação considera a afectação de recursos a um conjunto de transformações que dá o resultado mais demorado para o tempo de execução. No modelo desenvolvido, considera-se que no instante zero se inicia a execução das primeiras transformações do sistema, i.e. aquelas que não têm transformações precedentes. A afectação do recurso mais lento e o facto de se considerar o instante zero de início da execução da encomenda, faz com que todos os instantes de início e conclusão de cada transformação não excedam o limite UB4.

$$\text{UB4} \quad \text{UB4} = \max_{j,k} (c_{jk}) \cdot \sum_{j=1}^n W_j \Rightarrow \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : \begin{cases} T_{jk}^+ \leq \text{UB4} \\ T_{jk}^- \leq \text{UB4} \\ T_j^+ \leq \text{UB4} \\ T_j^- \leq \text{UB4} \end{cases} \quad \text{Equação 57}$$

## iii. Número de recursos

Considera-se que um recurso é seleccionado para a execução de uma transformação se a quantidade de trabalho executado por esse recurso nessa transformação for superior a zero. A variável  $w_{z_{jk}}$  binária representativa da selecção de um recurso para execução de uma transformação depende do valor de

quantidade de trabalho atribuído  $w_{jk}$ , da forma representada na Equação 26, secção IV - 1.3.1. Como esta formulação é não linear, é necessário alterar a formulação para incluir num modelo de optimização linear.

O quociente entre a quantidade de trabalho atribuída  $w_{jk}$  e a capacidade disponível  $\alpha_{jk}$  é superior a zero se  $w_{jk}$  for superior a zero. Assim sendo, e como  $wz_{jk}$  é uma variável binária é possível determinar o seu valor a partir da Equação 58. Se um recurso não tiver aptidão de execução de uma transformação, as variáveis associadas à execução da transformação pelo recurso não existem.

$$\text{R0} \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : wz_{jk} \geq w_{jk} / \alpha_{jk} \quad \text{Equação 58}$$

#### **iv. Procura**

A quantidade de trabalho total executada por todos os recursos numa transformação é igual à quantidade de trabalho requerida por essa transformação, conforme formulação apresentada na Equação 59. Esta restrição é igual à restrição apresentada na Equação 15, secção III - 2.2.1.

$$\text{R1} \quad \forall j \in \{1, \dots, n\} : \sum_{k=1}^r w_{jk} = W_j \quad \text{Equação 59}$$

#### **v. Disponibilidade**

O tempo total de execução de trabalhos por um recurso  $k$  não pode exceder o seu tempo livre  $Tf_k$ , conforme formulação apresentada na Equação 60. Esta restrição é igual à restrição apresentada na Equação 18, secção III - 2.2.2.

$$\text{R2} \quad \forall k \in \{1, \dots, r\} : \sum_{j=1}^n c_{jk} \cdot w_{jk} \leq Tf_k \quad \text{Equação 60}$$

#### **vi. Tempos de recursos – relação entre execução e instante de conclusão**

O tempo de execução  $T_{jk}$  da transformação  $j$  pelo recurso  $k$ , não pode ser interrompido e é obtido pelo produto entre a quantidade de trabalho executado na transformação e o tempo de execução de uma unidade dessa quantidade de trabalho (Equação 61). O instante de conclusão  $T_{jk}^+$  da execução de uma transformação  $j$  por um recurso  $k$  é igual ao tempo de execução  $T_{jk}$  adicionado ao instante de início  $T_{jk}^-$  da execução (Equação 62).

$$\text{R3.1} \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_{jk} = c_{jk} \cdot w_{jk} \quad \text{Equação 61}$$

$$\text{R3.2} \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_{jk}^+ = T_{jk}^- + T_{jk} \quad \text{Equação 62}$$

#### **vii. Tempos de transformações – relação com os recursos**

O instante de conclusão de uma transformação é igual ao maior instante de conclusão do processamento dessa transformação por parte de qualquer recurso (Equação 63). O instante de início de uma transformação

é igual ao *menor* instante de início do processamento dessa transformação por parte de qualquer recurso (Equação 64).

$$R4.1 \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_j^+ = \max_k (T_{jk}^+) \quad \text{Equação 63}$$

$$R4.2 \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_j^- = \min_k (T_{jk}^-) \quad \text{Equação 64}$$

No caso da função objectivo considerar a minimização de uma função linearmente dependente de  $T_j^+$ , então a restrição R4.1 é transformada na restrição linear da Equação 65.

$$R4.1 \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_j^+ \geq T_{jk}^+ \quad \text{Equação 65}$$

### viii. Transformações precedentes

Considere-se  $prec_j$  o conjunto de transformações que são imediatamente precedentes da transformação  $j$ , com  $npr_j$  elementos. Se todas as transformações forem paralelas seguidas pela execução de  $j$ , e considerando que uma transformação não é imediatamente precedente de si própria, então o número de elementos de  $prec_j$  é sempre não superior ao número de transformações menos uma, isto é,  $npr_j \leq n-1$ .

$$\forall j \in \{1, \dots, n\} : (j \notin prec_j) \wedge (prec_j \subset \{1, \dots, n\}) \quad \text{Equação 66}$$

O instante de início da execução de qualquer transformação é igual ao maior instante de conclusão de todas as transformações que a precedem.

$$R5 \quad \forall j \in \{1, \dots, n\}, \forall pr_j \in prec_j : T_j^- = \max_{pr_j} (T_{pr_j}^+) \quad \text{Equação 67}$$

No caso da função objectivo considerar a minimização de uma função linear directamente proporcional aos instantes de conclusão das transformações, então a restrição R6 pode ser representada pela seguinte restrição linear.

$$R5 \quad \forall j \in \{1, \dots, n\}, \forall pr_j \in prec_j : T_j^- \geq T_{pr_j}^+ \quad \text{Equação 68}$$

A Equação 64, permite reformular a restrição R5 da Equação 68, aplicando a restrição aos instantes de início de execução de uma transformação por qualquer recurso. A restrição R5 pode ser assim representada pela Equação 69.

$$R5 \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}, \forall pr_j \in prec_j : T_{jk}^- \geq T_{pr_j}^+ \quad \text{Equação 69}$$

### ix. Transformações paralelas

O conjunto  $paral_j$  de transformações que podem ser executadas em paralelo com a transformação  $j$  tem  $npar_j$  elementos. Considerando  $j$  paralela com todas as outras transformações e que estas têm que preceder uma transformação final de montagem, então o número de elementos de  $paral_j$  é sempre não superior ao número total de transformações menos duas, isto é, então  $npar_j \leq n-2$ .

$$\forall j \in \{1, \dots, n\} : (j \notin \text{paral}_j) \wedge (\text{paral}_j \subset \{1, \dots, n\}) \quad \text{Equação 70}$$

Se um recurso tiver a possibilidade de executar mais do que uma transformação, não o pode fazer simultaneamente. Assim sendo, um período de execução de uma transformação deverá concluir-se antes de outro período, ou iniciar-se depois.

$$\begin{aligned} \text{R6} \quad & \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}, \forall \text{par}_j \in \text{paral}_j : \\ & \left( T_{\text{par}_j, k}^+ \leq T_{jk}^- \right) \vee \left( T_{\text{par}_j, k}^- \geq T_{jk}^+ \right) \quad \text{Equação 71} \end{aligned}$$

Uma restrição com o operador lógico *ou* pode transformar-se num conjunto de duas restrições lineares com recurso a variáveis binárias  $b$  e à utilização duma constante de valor suficientemente elevado  $M$ , conforme descrito por Hillier and Lieberman (1989). Assim sendo, a restrição relativa a restrições paralelas, formulada na Equação 71 é equivalente à restrição formulada na Equação 72.

$$\begin{aligned} \text{R6} \quad & \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}, \forall \text{par}_j \in \text{paral}_j : \\ & T_{\text{par}_j, k}^+ \leq T_{jk}^- + b_{\text{par}_j, k} \cdot M \wedge T_{\text{par}_j, k}^- \geq T_{jk}^+ - (1 - b_{\text{par}_j, k}) \cdot M \quad \text{Equação 72} \end{aligned}$$

### 3.2.2. Restrições de Agenda

Nesta secção apresentam-se as restrições relacionadas com as agendas dos recursos.

#### i. Definição da Agenda

Para cada recurso  $r_k$  existe um conjunto  $free_k$  de  $nf_k$  períodos livres, representados por pares de instantes de início e conclusão de cada período, que representam a sua agenda livre.

$$\forall k \in \{1, \dots, r\}, \exists nf_k \in \mathbb{Z}_0^+ : free_k = \left\{ (Tf_{k,1}^-, Tf_{k,1}^+), \dots, (Tf_{k,nf_k}^-, Tf_{k,nf_k}^+) \right\} \quad \text{Equação 73}$$

Cada um dos  $nf_k$  períodos livres dum recurso também pode ser representado por  $free_{k,fk}$ .

$$\forall k \in \{1, \dots, r\}, \forall fk \in \{1, \dots, nf_k\} : free_{k,fk} = [Tf_{k,fk}^-, Tf_{k,fk}^+] \quad \text{Equação 74}$$

#### ii. Execução sem interrupção num período livre da agenda

Respeitando o princípio de simultaneidade ii.b representado na Tabela 14, os instantes de início e conclusão do processamento de uma transformação por um recurso terão que estar contidos num único período livre do recurso. Esta necessidade traduz-se na formulação apresentada na seguinte equação.

$$\begin{aligned} \text{R7} \quad & \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}, \exists fk \in \{1, \dots, nf_k\} : \\ & T_{jk}^- \in free_{k,fk} \wedge T_{jk}^+ \in free_{k,fk} \quad \text{Equação 75} \\ & \Leftrightarrow Tf_{k,fk}^- \leq T_{jk}^- \leq Tf_{k,fk}^+ \wedge Tf_{k,fk}^- \leq T_{jk}^+ \leq Tf_{k,fk}^+ \end{aligned}$$

A aplicação das restrições R3.1 e R3.2 transformam a restrição R7 na Equação 76.

$$R7 \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}, \exists fk \in \{1, \dots, nf_k\} : \quad \text{Equação 76}$$

$$\left( Tf_{k, fk}^- \leq T_{jk}^- \right) \wedge \left( T_{jk}^+ \leq Tf_{k, fk}^+ \right)$$

Na restrição R7, ambos os instantes de início e conclusão da execução de uma transformação pelo recurso, têm que estar contidos num único dos períodos livres do recurso. Uma restrição com uma condição que só se pode verificar num caso pode transformar-se num conjunto de restrições lineares com recurso a variáveis binárias *bin*. Considerando que todos os recursos podem executar todas as transformações, então é necessário considerar, para cada combinação transformação-recurso, um conjunto de tantas variáveis binárias quantos os períodos livres do recurso. Apenas uma destas variáveis pode tomar o valor 1, que corresponde à selecção de um determinado período livre para execução da transformação por parte do recurso. Assim sendo, a restrição R7 formulada na Equação 76 é equivalente à Equação 77.

$$R7 \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}, \forall fk \in \{1, \dots, fr_k\} : \quad \text{Equação 77}$$

$$\sum_{fk=1}^{fr_k} bin_{j, k, fk} \leq 1 \wedge \sum_{fk=1}^{fr_k} (bin_{j, k, fk} \cdot Tf_{k, fk}^-) \leq T_{jk}^- \wedge T_{jk}^+ \leq \sum_{fk=1}^{fr_k} (bin_{j, k, fk} \cdot Tf_{k, fk}^+)$$

### 3.2.3. Optimização da Selecção de Recursos

A selecção de um recurso  $r_k$  para execução de uma transformação pressupõe que este recurso tem habilidade e tempo livre para executar essa transformação. Se não houver consideração da agenda, então apenas se considera a quantidade de tempo livre do recurso. A consideração de restrições de agenda pressupõe a selecção deste recurso  $r_k$ , apenas nos períodos livres da agenda. Teremos assim, restrições genéricas, restrições de sequência de processamento e restrições de agenda.

#### i. Minimização do tempo total de execução

A minimização do tempo total de execução de todas as transformações por todos os recursos corresponde ao problema de minimização representado pela Equação 78. Esta formulação foi introduzida inicialmente na secção III - 3.5. Com esta formulação o processo de selecção de recursos favorece os recursos com tempos de processamento mais rápidos, atribuindo-lhes trabalho durante mais tempo. Assim sendo, só existe selecção de recursos alternativos se não existir capacidade disponível suficiente no recurso mais rápido.

$$\text{minimizar } \sum_{j=1}^n \sum_{k=1}^r c_{jk} \cdot w_{jk} \quad \text{Equação 78}$$

A solução obtida com esta formulação pode conduzir a sistemas de produção com um reduzido número de mudanças de recurso, isto é, com um reduzido número de trabalhos de preparação e de distribuição ou transporte, facilitando por isso, o processo de sincronização e controlo da execução dos trabalhos.

#### ii. Minimização do número de recursos

A minimização do número de recursos utilizado, formulado na Equação 79, está associada à quantidade de trabalho atribuída a cada recurso, isto é, à atribuição da maior quantidade de trabalho possível ao mesmo



recurso. No entanto, não há garantia de obtenção de soluções com tempos baixo de conclusão dos trabalhos, nem de boa utilização dos recursos.

$$\text{minimizar } \sum_{j=1}^n \sum_{k=1}^r w z_{jk} \quad \text{Equação 79}$$

A solução obtida com esta formulação conduz, obrigatoriamente, a sistemas de produção com um reduzido número de mudanças de recurso, isto é, com um reduzido número de trabalhos de preparação e operações de distribuição e transporte, facilitando assim também o processo de sincronização e controlo da execução dos trabalhos.

Esta formulação pode ser combinada com a imposição de restrições adicionais, nomeadamente, impondo um tempo de conclusão de processamento que permita *respeitar um prazo de entrega negociado*. Neste caso é possível obter soluções cujo critério é a minimização do número de recursos e sujeito ao cumprimento de prazos de entrega.

### iii. **Minimização do instante de conclusão do processamento**

O instante de conclusão de uma encomenda corresponde ao instante de conclusão da transformação responsável pela entrega do produto final. Esta transformação corresponde à única transformação do nível superior dos níveis de agregação. No exemplo utilizado neste capítulo, na secção III - 2.1.4.i, esta transformação está representada na Figura 37. Se se considerar a possibilidade de início do processamento no instante zero, então o tempo de execução desta única transformação, é equivalente ao tempo de percurso de fabrico, aqui denominado por  $FT^{99}$ , como representado na Equação 80. O tempo de percurso de fabrico definido desta forma tem o mesmo significado do prazo de entrega  $PE$  utilizado na secção IV - 3.1, com a diferença de  $PE$  ser definido pelo recurso enquanto  $FT$  é determinado pelo método de selecção com base na agenda do recurso.

---

Nível de Agregação Superior	$FT = T_1$	Equação 80
-----------------------------	------------	------------

No nível inferior dos níveis de agregação, esta transformação corresponde à transformação de índice superior,  $j=n$ , que é a transformação que não precede nenhuma outra. No exemplo utilizado neste capítulo, na secção III - 2.1.4.i, esta transformação está representada na Figura 36 por  $t_3$ . Se se considerar a possibilidade de início do processamento no instante zero, então o instante de conclusão do processamento é equivalente ao tempo de percurso de fabrico, como representado na Equação 81.

---

Nível de Agregação Inferior	$FT = T_n^+$	Equação 81
-----------------------------	--------------	------------

A minimização do instante de conclusão do processamento de uma encomenda nas condições aqui descritas é equivalente à minimização do tempo em percurso de fabrico, correspondendo ao problema de

---

<sup>99</sup> FT – “Flow Time”.

minimização representado pela Equação 82. As restrições impostas a este problema podem ser todas as que estão descritas nas secções IV - 3.2.1 - Restrições Genéricas e IV - 3.2.2 - Restrições de Agenda, obtendo-se assim a formulação de um problema de programação da produção. Se, nesta formulação, não se considerarem as restrições de agenda dos recursos, então as soluções obtidas supõem a possibilidade de utilização da capacidade livre dos recursos em qualquer instante requerido.

$$\text{minimizar } T_n^+ \quad \text{Equação 82}$$

As restrições de precedência entre transformações, formuladas na Equação 13 e na Equação 14 da secção III - 2.1.5, permitem obter uma matriz de precedências  $\mathbf{P}^*$ , em que as transformações que não precedem nenhuma outra podem ser identificadas por terem a coluna da matriz composta por zeros. O início de processamento de qualquer transformação  $T_j^-$  só se pode efectuar quando todas as transformações que a precedem tiverem terminado. Resumindo, considera-se a possibilidade de partição do lote, em cada transformação, enviando as várias partições para recursos diferentes. Esta forma de cálculo não considera a possibilidade de concluir o processamento de alguns objectos e enviá-los imediatamente para a transformação seguinte, isto é, não considera a possibilidade de sobreposição de lotes.

Pela aplicação do princípio de simultaneidade ii.b representado na Tabela 14, o processamento de uma transformação pode ser efectuado em vários processadores alternativos. Assim sendo, considerando a Equação 61, o instante de conclusão de uma transformação, formula-se na Equação 83, como sendo igual ao maior dos instantes de conclusão dos processadores que executam essa transformação.

$$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}: T_j^+ = \max_k (T_{jk}^+) \Leftrightarrow T_j^+ = \max_k (T_{jk}^- + c_{jk} \cdot w_{jk}) \quad \text{Equação 83}$$

Pela aplicação da Equação 83 é possível reformular a Equação 82 de minimização do instante de conclusão do processamento de uma encomenda  $T_n^+$  na Equação 84. Esta formulação relaciona  $T_n^+$  com a quantidade de trabalho executada por todos os recursos na transformação  $n$ . As restrições de precedência formuladas na Equação 69 permitem relacionar  $T_n^+$  com os instantes de conclusão das transformações imediatamente precedentes  $T_{pj}^+$  e dessa forma, recursivamente, com todas as quantidades de trabalho executadas em todas as transformações por todos os recursos.

$$\text{minimizar } T_n^+ = \max_k (T_{nk}^- + c_{nk} \cdot w_{nk}) \quad \text{Equação 84}$$

### iii.i Exemplo

Para o exemplo seguido neste capítulo, Figura 36, é possível determinar as Restrições de Precedência, através da Equação 13, secção III - 2.1.5, obtendo-se a matriz de precedências representada na Equação 85. Nesta matriz é possível verificar que a transformação  $t_3$  não precede nenhuma outra porque todos os elementos da 3ª coluna são nulos e que  $t_1$  e  $t_2$  precedem  $t_3$ .

$$\mathbf{P} = \mathbf{Pre}^T \times \mathbf{Post} = \begin{matrix} t_1 & t_2 & t_3 \\ t_2 & 0 & 0 \\ t_3 & 0 & 0 \\ t_3 & 1 & 1 & 0 \end{matrix} = \mathbf{P}^* \quad \text{Equação 85}$$

A Equação 14 da secção III - 2.1.5 permite determinar o vector dos tempos de início de cada transformação, representada na Equação 86.

$$\begin{aligned} \mathbf{T}^- &= \mathbf{P}^* \otimes \mathbf{T}^+ \Leftrightarrow \forall i, j \in \{1, \dots, n\} : T_i^- = \text{Max}_j (P_{ij}^* \cdot T_j^+) \\ &\Leftrightarrow \mathbf{T}^- = \left[ 0 \quad 0 \quad \text{Max}(T_1^+, T_2^+) \right]^T \end{aligned} \quad \text{Equação 86}$$

O instante de conclusão do processamento da encomenda é dado pela Equação 87, com  $k = 1, \dots, r$ .

$$\begin{aligned} T_n^+ &= \max_k (T_{nk}^- + c_{nk} \cdot w_{nk}) \Leftrightarrow T_3^+ = \max_k (T_{3k}^- + c_{3k} \cdot w_{3k}) \\ &\text{com } T_{3k}^- \geq T_1^+ \wedge T_{3k}^- \geq T_2^+ \end{aligned} \quad \text{Equação 87}$$

Considerando que não existem restrições de agenda e sabendo que o instante de início do processamento das transformações que não têm nenhuma outra precedente é zero, é possível formular  $T_n^+$  em função da quantidade de trabalho, obtendo-se o resultado apresentado na Equação 88, com  $k = 1, \dots, r$ .

$$\begin{aligned} T_3^+ &= \max_k (T_{3k}^- + c_{3k} \cdot w_{3k}) \Leftrightarrow T_3^+ = \max_k (\max(T_1^+, T_2^+) + c_{3k} \cdot w_{3k}) \Leftrightarrow \\ T_3^+ &= \max(T_1^+, T_2^+) + \max_k (c_{3k} \cdot w_{3k}) \Leftrightarrow \\ T_3^+ &= \max \left( \max_k (T_{1k}^- + c_{1k} \cdot w_{1k}), \max_k (T_{2k}^- + c_{2k} \cdot w_{2k}) \right) + \max_k (c_{3k} \cdot w_{3k}) \Leftrightarrow \\ T_3^+ &= \max \left( \max_k (c_{1k} \cdot w_{1k}), \max_k (c_{2k} \cdot w_{2k}) \right) + \max_k (c_{3k} \cdot w_{3k}) \end{aligned} \quad \text{Equação 88}$$

### iii.ii Formulação Completa

A formulação do problema de optimização resulta da representação da função objectivo, e de algumas das restrições, variáveis e dos respectivos limites, que foram introduzidos neste capítulo. Outras variáveis e restrições introduzidas poderiam ser utilizadas em formulações adicionais. No caso da Produção Simultânea baseada na partição de lotes, para minimizar o instante de conclusão de uma encomenda, obtém-se:

Função Objectivo:  $\text{minimizar } T_n^+ = \max_k (T_{nk}^- + c_{nk} \cdot w_{nk})$

Restrições:

$$\text{R1} \quad \forall j \in \{1, \dots, n\} : \sum_{k=1}^r w_{jk} = W_j$$

$$\text{R2} \quad \forall k \in \{1, \dots, r\} : \sum_{j=1}^n c_{jk} \cdot w_{jk} \leq Tf_k$$

$$\text{R3.1} \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_{jk}^- = c_{jk} \cdot w_{jk}$$

R3.2	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_{jk}^+ = T_{jk}^- + T_{jk}$
R4.1	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_j^+ \geq T_{jk}^+$
R5	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}, \forall pr_j \in prec_j : T_{jk}^- \geq T_{pr_j}^+$
R6	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}, \forall par_j \in paral_j : T_{par_j,k}^+ \leq T_{jk}^- + b_{par_j,k} \cdot M \wedge T_{par_j,k}^- \geq T_{jk}^+ - (1 - b_{par_j,k}) \cdot M$
R7	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}, \forall fk \in \{1, \dots, fr_k\} : \sum_{fk=1}^{fr_k} bin_{j,k,fk} \leq 1 \wedge \sum_{fk=1}^{fr_k} (bin_{j,k,fk} \cdot Tf_{k,fk}^-) \leq T_{jk}^- \wedge T_{jk}^+ \leq \sum_{fk=1}^{fr_k} (bin_{j,k,fk} \cdot Tf_{k,fk}^+)$

Variáveis:

V1	$\forall j \in \{1, \dots, n\}, k \in \{1, \dots, r\} : w_{jk} \in \mathbb{Z}_0^+$
V3, V5, V6	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : T_j^+, T_{jk}^-, T_{jk}^+ \geq 0$
V Binárias	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\}, \forall par_j \in paral_j, \forall fk \in \{1, \dots, fr_k\} : b_{par_j,k}, bin_{j,k,fk} \in \{0, 1\}$

Limites:

UB1	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : w_{jk} \leq \min(W_j, \alpha_{jk})$
UB4	$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : UB4 = \max_{j,k} (c_{jk}) \cdot \sum_{j=1}^n W_j \Rightarrow (T_{jk}^+ \leq UB4) \wedge (T_{jk}^- \leq UB4) \wedge (T_j^+ \leq UB4) \wedge (T_j^- \leq UB4)$

#### iv. Formulações Mistas

A utilização de uma única formulação anterior não permite controlar o resultado relativamente a outros critérios. A formulação de minimização do tempo total de processamento de todas as transformações por todos os recursos apresentada na Equação 78, não garante um tempo de conclusão aceitável, nem nenhum tipo de controlo sobre o número de recursos utilizados. A minimização do número de recursos, formulado na Equação 79, também não garante um tempo de conclusão aceitável, podendo dar resultados elevados de utilização dos recursos. A minimização do instante de conclusão do processamento de uma encomenda, formulado na Equação 82, pode dar resultados elevados de utilização e do número de recursos. A utilização mista de vários critérios pode permitir efectuar estudos do comportamento dos sistemas de produção em ambiente de Produção Simultânea de uma forma mais controlada.

Com o objectivo de controlar um destes parâmetros, quando se otimiza outro, é possível adicionar restrições (RAD#) que permitam impor algum grau de controlo. Uma forma de aumentar o controlo, resulta da resolução de um dos problemas de optimização obtendo um valor mínimo para um dos parâmetros, por exemplo o instante de conclusão de processamento da encomenda  $T_n^+$ , e em seguida resolver outro problema, por exemplo o de minimização do número de recursos  $WZ$ , impondo para aquele parâmetro ( $T_n^+$ ) uma restrição relacionada com o mínimo obtido.

Uma formulação mista pode resultar da utilização de uma função linear do resultado da minimização do número de recursos  $g(wz)$ , por adição da restrição formulada na Equação 89, ao problema de minimização do instante de conclusão de processamento.

$$\text{RAD1} \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, r\} : \sum_{j=1}^n \sum_{k=1}^r wz_{jk} \leq g \left( \min_{\text{anterior}} \left( \sum_{j=1}^n \sum_{k=1}^r wz_{jk} \right) \right) \quad \text{Equação 89}$$

Considerando  $f(t)$  uma função linear do resultado da minimização do instante de conclusão, pode adicionar-se a restrição formulada na Equação 90, aos problemas de minimização do tempo total de processamento de todas as transformações por todos os recursos e ao problema de minimização do número de recursos. Obtém-se assim outras duas formulações mistas de problemas de optimização.

$$\text{RAD2} \quad \forall j \in \{1, \dots, n\} : T_n^+ \leq f \left( \min_{\text{anterior}} \left( T_n^+ \right) \right) \quad \text{Equação 90}$$

#### **4. CONSIDERAÇÕES FINAIS**

A Produção Simultânea é um conceito similar ao da Engenharia Concorrente / Simultânea aplicado à produção, isto é, à fase de transformação das matérias primas em produtos finais. A definição de Produção Simultânea serve de base para o desenvolvimento dos conceitos que se pretendem aplicar para a criação de configurações de Sistemas Distribuídos de Produção. Esta definição funciona como fundação para o aprofundamento de conceitos de utilização simultânea de recursos e para o desenvolvimento de métodos de criação das referidas configurações de sistemas. No modelo de Sistemas Distribuídos de Produção, desenvolvido neste trabalho, gerido em ambiente de Produção Simultânea, os sistemas são orientados à satisfação de uma encomenda de um produto de cada vez. Isto significa que só depois de programada uma encomenda de um produto se pode considerar outra encomenda para programação, com vista à sua execução face às restrições de agenda dos recursos. Por definição, a Produção Simultânea não se refere à utilização simultânea de recursos para a satisfação de um grupo de encomendas integradas, mas sim, à utilização simultânea de recursos em encomendas sucessivas, i.e. sucessivamente programadas.

Neste capítulo apresentam-se medidas de simultaneidade de um sistema produtivo. Estas medidas permitem aferir a utilização simultânea dos recursos do sistema para cada encomenda. Quanto maior for essa utilização, maior será o Grau de Simultaneidade, até um máximo de uma unidade. Apresentam-se duas medidas, uma global, referida ao total das transformações, e outra local, referida a cada transformação. Assim, pode aferir-se a simultaneidade de utilização de recursos para todo o sistema e a simultaneidade de utilização de recursos para subsistemas de execução de cada transformação.

Os princípios de simultaneidade desenvolvidos permitiram caracterizar a Produção Simultânea em relação ao tipo de simultaneidade que se pode explorar. Neste trabalho explora-se a simultaneidade por partição dos lotes e a separação completa do processamento de cada um. Com este princípio obtém-se soluções com sobreposição do processamento dos lotes, com maior flexibilidade de exploração da agenda dos recursos, mas também com maior exigência do ponto de vista do controlo da produção no sistema.

A criação de uma configuração de um Sistema Distribuído de Produção, de acordo com o modelo MSDP, baseia-se na configuração genérica e na configuração específica do sistema. A configuração genérica, que

permite obter uma descrição formal das transformações que se têm que executar para satisfazer uma encomenda, foi apresentada no capítulo anterior. A criação do sistema impõe a selecção de recursos físicos capazes de executar aquelas transformações em períodos determinados, isto é, a obtenção de uma configuração específica. Esta configuração depende, neste trabalho, da aplicação de modelos de selecção de recursos. Estes modelos baseiam-se na aplicação do princípio de simultaneidade de partição e separação, em cada transformação, das quantidades encomendadas. Estes modelos podem ser de: optimização global, com a afectação simultânea de todos os recursos, considerando que eles disponibilizam a sua agenda de tempo livre para processamento da encomenda; optimização local, com a afectação de recursos para a execução de uma transformação de cada vez, considerando que eles efectuem ofertas para execução da transformação. O modelo global considera que os recursos, relativamente ao modelo local, terão uma autonomia inferior, porque terão que disponibilizar a sua agenda e dados relacionados com a sua capacidade de produção.

Analisa-se possibilidades de configuração do SDP em função do Grau de Simultaneidade, que permitem analisar a variação do menor prazo de entrega das encomendas, isto é, instante de conclusão do processamento de todas as transformações da encomenda, em função do Grau de Simultaneidade.

Finalmente, descreve-se a formulação dos problemas de afectação e calendarização de recursos que permitem obter soluções óptimas locais ou globais. A utilização destas formulações permite obter soluções de selecção de recursos para projecto dinâmico do Sistema Distribuído de Produção em ambiente de Produção Simultânea. Será assim possível, analisar as possibilidades de configuração apresentadas e as soluções em comparação com outros métodos.

## **V - EXPERIMENTAÇÃO DO MODELO MSDP EM AMBIENTE DE PS**

Na formalização de conceitos de Produção Simultânea aplicada ao modelo de Sistemas Distribuídos de Produção efectua-se algumas considerações sobre o comportamento destes sistemas que se pretende analisar em alguns exemplos. A possibilidade de aplicação do modelo para programação da produção de várias encomendas sucessivas é demonstrada usando vários casos de teste. Em tal demonstração o critério de optimização é o instante de conclusão de cada encomenda a processar. Os resultados da programação de várias encomendas são depois, avaliados à luz de medidas de desempenho relevantes para as empresas e relacionadas com a filosofia de Produção Simultânea. Em particular, avalia-se o desempenho em função de uma medida designada de “Qualidade de Serviço de Produção”.





## **1. POSSIBILIDADES DE CONFIGURAÇÃO DO SISTEMA**

No capítulo dedicado à Produção Simultânea definiram-se algumas possibilidades de configuração do sistema relativas ao comportamento de algumas variáveis, em função do Grau de Simultaneidade do sistema. A configuração do sistema depende da afectação de recursos para a execução das transformações que permitem satisfazer uma encomenda. Daqui resulta o estudo de possibilidades de configuração, estabelecidas na secção IV - 2.2, associadas a dois modelos de afectação de recursos: modelo local - configurações obtidas para uma transformação de cada vez; modelo global - configurações obtidas para todas as transformações equacionadas simultaneamente.

O estudo das possibilidades de configuração do sistema baseia-se num caso de teste que representa um sistema genérico simples de produção. Estas possibilidades resultam da aplicação dos modelos local e global de afectação de recursos. A experimentação realizada resulta da aplicação de diversos testes com restrição do número de recursos para cada um destes modelos, utilizando o mesmo caso de teste para ambos. Em seguida descreve-se o plano de experimentação para estudo das possibilidades de configuração, apresenta-se o caso de teste e para cada um dos modelos de afectação de recursos é descrita a forma de realização da experimentação e apresentados e analisados os resultados obtidos.

### **1.1 Plano de Experimentação**

As possibilidades de configuração estão formuladas para dois modelos de afectação de recursos, sendo, para cada um deles, executados processos de *experimentação* compostos por um conjunto de testes. Cada *teste* executado corresponde à aplicação do modelo de afectação de recursos em análise, para determinação de uma solução de programação de produção específica para o caso de teste, com variação de parâmetros de teste.

O *caso de teste* para estudo das possibilidades de configurações do sistema, corresponde a uma configuração genérica de um Sistema Distribuído de Produção para o qual se procuram soluções de programação da produção, e dessa forma configurações específicas do SDP. A determinação de uma configuração específica, resulta da aplicação de modelos de afectação de recursos com estabelecimento de critérios de optimização. O estudo das referidas possibilidades efectua-se para um caso genérico de teste, utilizado em ambos os modelos, local e global, de afectação.

As possibilidades de configuração correspondem a instâncias do SDP, resultantes da aplicação de modelos de afectação de recursos, aplicando o critério de minimização do instante de conclusão de uma encomenda para cada Grau de Simultaneidade definido. Nesta trabalho de experimentação, considera-se o prazo de entrega da encomenda igual ao instante de conclusão do processamento da encomenda. Assim sendo, o *critério* de optimização utilizado para cada teste é o de minimização do instante de conclusão e o *parâmetro de teste* é o Grau de Simultaneidade. Como o Grau de Simultaneidade é directamente proporcional ao número de recursos seleccionados pelo modelo de afectação, este número é utilizado como *parâmetro de teste* equivalente. Isto significa que, para cada teste, se determina uma configuração específica para o sistema, que optimiza o instante de conclusão da encomenda sujeita à utilização de um determinado número

de recursos. Esta restrição, similar à restrição RAD1 especificada na secção IV - 3.2.3.iv, é adicionada às restrições genéricas e de agenda, especificadas nas secções IV - 3.2.1 e IV - 3.2.2. Deve notar-se que, no caso do modelo local, impõem-se um número de recursos igual para todas as transformações, e no caso do modelo global esta restrição aplica-se ao número total de recursos utilizados no sistema. Desta forma, obtém-se uma curva de variação do instante de conclusão em função do Grau de Simultaneidade, que está directamente relacionado com o número de recursos. No modelo local, considera-se a variação do Grau de Simultaneidade por transformação e no modelo global, considera-se a variação do Grau de Simultaneidade do sistema.

Na aplicação do modelo *global* de afectação de recursos, estabeleceram-se duas possibilidades de variação dos custos em função do Grau de Simultaneidade. Por essa razão, utilizam-se duas *variáveis* de custos, que dependem do programa de produção obtido pela aplicação do modelo global de afectação de recursos. Isto significa que, como resultado da execução dos testes, se obtém uma curva de variação de custos em função do Grau de Simultaneidade.

## 1.2 Caso de Teste

A configuração genérica de Sistemas Distribuídos de Produção orientados ao produto descritos neste trabalho, pode ser composta, fundamentalmente por duas estruturas, denominadas estruturas genéricas de base, representadas na Figura 47:

- Estrutura 1 – estrutura genérica constituída por duas transformações sequenciais.
- Estrutura 2 – estrutura genérica constituída por uma transformação precedida por duas transformações paralelas.

Estas estruturas podem resultar de estruturas elementares mais simples com uma única transformação, correspondentes às estruturas que estão representadas no nível com agregação do Caso 1 e 2, da Figura 24, página 67. No entanto, a estrutura produtiva de produtos complexos, modelados da forma considerada neste trabalho, pode resultar da combinação das estruturas genéricas de base. Estas estruturas genéricas têm mais de uma transformação, permitindo, por essa razão, uma análise de comportamento do sistema com relacionamento entre transformações.

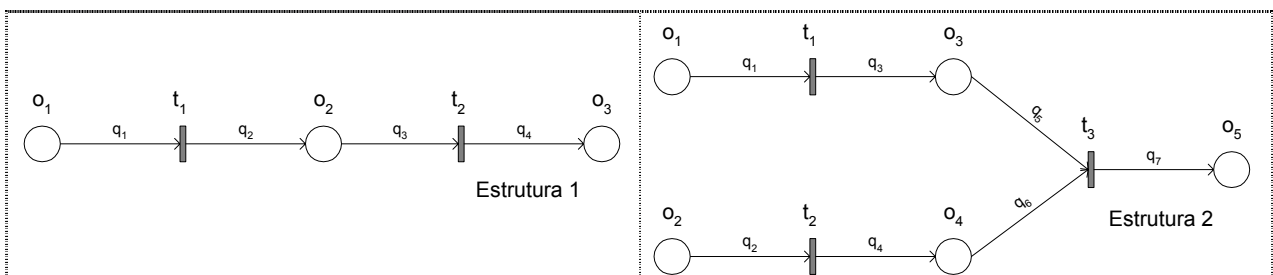


Figura 47: Estruturas genéricas de base.

A primeira estrutura genérica de base tem transformações precedentes, sendo, neste caso,  $t_2$  precedida por  $t_1$ . A segunda estrutura genérica de base, além de ter transformações precedentes, sendo, neste caso,  $t_3$  precedida por  $t_1$  e  $t_2$ , também inclui transformações paralelas,  $t_1$  e  $t_2$ , que não podem ser executadas simultaneamente pelos mesmos recursos. A Estrutura 2, por ter transformações precedentes e paralelas, é

considerada mais geral do ponto de vista da análise, sendo por isso utilizada como caso de teste para análise das possibilidades de configuração relativas a ambos os modelos de afectação de recursos.

Como se pode depreender da análise efectuada na secção IV - 2.2.1, no cenário referido, se existir um número suficientemente elevado de recursos com capacidade disponível relativamente ao número de objectos encomendados, a afectação de um recurso a uma transformação não influencia a execução de outra transformação, porque pode utilizar outro recurso. Neste caso, considerando recursos nas mesmas condições, com valores idênticos de capacidade livre e de tempos unitários de execução de uma transformação, a curva do instante de conclusão mínimo relativamente ao Grau de Simultaneidade, para transformações paralelas, é igual à de uma única transformação. Considerando transformações imediatamente precedentes, os instantes de conclusão de cada transformação têm que ser adicionados para se obter o instante de conclusão da execução das duas transformações. Assim sendo, considerando as condições do cenário referido, obtêm-se instantes de conclusão iguais à soma dos instantes de conclusão das duas transformações, neste caso, igual ao dobro do instante de conclusão de uma delas.

Para melhor analisar as possibilidades de configuração, é necessário que o caso de teste seja afectado por interacções entre os recursos. Tal interacção implica que por razões de número de recursos disponíveis a melhor solução tenha que considerar, para a maioria dos testes, a afectação do mesmo recurso para a execução de várias unidades de quantidade de trabalho de mais do que uma transformação. Como forma de garantir esta interacção entre recursos, determina-se como *factor* de teste que o número de recursos disponíveis tem que ser não superior à quantidade de trabalho a executar na transformação de produção da quantidade do produto encomendado.

Nestes testes não se pretende analisar a influencia da capacidade livre e/ou dos tempos unitários de execução sobre a variação do instante de conclusão face ao Grau de Simultaneidade. Por esta razão consideram-se recursos com valores idênticos e suficientemente elevados de capacidade livre relativamente à procura. Considera-se ainda, que todos os recursos podem executar qualquer transformação, demorando o mesmo tempo. Por razões similares, estabelecem-se quantidades de utilização  $q_Q$ ,  $Q=1, \dots, 7$ , de objectos da Rede de Petri da Estrutura 2 iguais a uma unidade. Como todas as quantidades de utilização dos objectos são iguais a 1, consideram-se as quantidades armazenadas tais que, o número de objectos a produzir coincide com a quantidade de trabalho a executar.

O modelo de afectação global desenvolvido, é um modelo de optimização linear para o qual não se desenvolveu uma análise de complexidade detalhada. No entanto, realizaram-se testes sumários que mostraram tempos elevados de resolução com o aumento da dimensão do problema, isto é, com o aumento do número de objectos encomendados, número de transformações e número de recursos. Este facto impõe algumas restrições na definição da dimensão do problema a testar, nomeadamente no que respeita à quantidade de objectos encomendados e ao número de recursos. O factor de teste atrás definido, estabelece uma relação entre o número de recursos disponível e a quantidade de trabalho a executar. Fixando a quantidade de objectos encomendados em 10 unidades, o número de recursos deve ser não superior a este valor. Os testes de complexidade sumários referidos, conduziram à definição de 7 recursos para este caso de

teste. É importante notar que se realizaram muitos testes com 5 recursos e alguns com 10 recursos, sempre com resultados similares aos que vão ser descritos no âmbito deste plano de experimentação.

Pelas razões descritas, considera-se a existência de 7 recursos  $r_k$  candidatos à execução das transformações do caso de teste, assumindo que qualquer dos recursos tem aptidão para executar qualquer das transformações, estando os seus dados representados na Tabela 17. Na agenda de cada recurso estão representados os pares de instantes de início e conclusão dos períodos livres iniciais, i.e. períodos de disponibilidade do recurso. Neste caso, no início, todos os recursos têm 100 UT<sup>40</sup> livres.

Tabela 17: Tempos unitários de processamento  $c_{jk}$  e agenda livre dos recursos.

	$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$	$r_7$	Agenda de períodos livres em unidades de tempo
$t_1$	1	1	1	1	1	1	1	$\forall k \in \{1, \dots, r\}: \text{AgL}_k = \{(0, 100)\}$
$t_2$	1	1	1	1	1	1	1	
$t_3$	1	1	1	1	1	1	1	

Da análise destes dados resulta que, no máximo, se podem afectar quantidades de trabalho simultaneamente a 7 recursos nas 3 transformações, correspondendo a um número máximo de 21 recursos utilizados no sistema. Desta mesma análise resulta que, no mínimo, se pode afectar 1 recurso a cada uma das 3 transformações o que corresponde a um número mínimo de recursos do sistema igual a 3. No modelo local, podem afectar-se quantidades de trabalho simultaneamente a 7 recursos por cada transformação, correspondendo a um número máximo de 7 recursos utilizados por transformação. Neste modelo, o número mínimo de recursos por transformação é igual a 1.

Conforme foi referido atrás, definiu-se uma encomenda de 10 unidades. Por se adoptarem as quantidades de utilização iguais a 1, i.e.  $q_Q=1, Q=1, \dots, 7$ , para a Rede de Petri do caso de teste, Estrutura 2 da Figura 47, a execução de uma unidade de quantidade de trabalho de cada transformação, consome uma unidade de cada um dos objectos de entrada e produz uma unidade do objecto de saída. Assim sendo, para uma encomenda de 10 unidades do objecto  $o_5$  é necessário executar 10 unidades de quantidade de trabalho  $W_j$  de todas as transformações  $t_1$  a  $t_3$ , conforme apresentado na Tabela 18.

Tabela 18: Quantidade de trabalho requerida por cada transformação.

	$W_i$
$t_1$	10
$t_2$	10
$t_3$	10

Em seguida apresenta-se a experimentação realizada, isto é, o conjunto de testes realizados, de estudo da variação, no caso de teste descrito, do prazo de entrega de uma encomenda em função do Grau de Simultaneidade. Esta experimentação é apresentada em duas secções, uma referente ao modelo local de afectação de recursos e outra ao modelo global.

<sup>40</sup> Unidades de Tempo.

### **1.3 Modelo Local de Afecção de Recursos**

A afecção de recursos para uma transformação resulta da afecção de um ou mais recursos que correspondam a critérios de desempenho seleccionados. Sendo o critério utilizado a minimização do instante de conclusão, então favorece-se a selecção do recurso que mais depressa entregar, i.e. processar, a encomenda. Para diminuir o instante de conclusão é possível aumentar a simultaneidade de utilização de recursos do sistema, dividindo as quantidades a processar, i.e. fabricar, por vários recursos.

No modelo local assume-se a autonomia dos recursos, que estabelecem os seus tempos de entrega, isto é, instantes de conclusão do processamento de cada transformação. Supondo, em consonância com a filosofia de PS, que todos os recursos estão interessados em entregar a encomenda logo que possam, pode utilizar-se a agenda dos recursos para determinar os instantes de tempo de entrega. Assim sendo, utiliza-se o modelo global, que assume uma menor autonomia dos recursos, aplicando-o a uma transformação de cada vez, através da minimização do instante de conclusão da transformação, sujeita às restrições referidas no plano de teste: genéricas, de agenda e RAD1. A aplicação do modelo global a uma transformação de cada vez, com actualização das agendas dos recursos entre transformações, permite obter configurações específicas do sistema por minimização do instante de conclusão de cada transformação. Desta forma, pode verificar-se o comportamento do instante de conclusão relativamente à variação do número de recursos utilizados simultaneamente em cada transformação. A cada número de recursos considerado para utilização em cada transformação, corresponde um Grau de Simultaneidade para o qual se obtém um instante mínimo de conclusão de cada transformação.

#### **1.3.1. Descrição da Experimentação**

Considera-se, de acordo com a descrição efectuada anteriormente, que se coloca uma encomenda por transformação, utilizando o modelo de optimização global com minimização do instante de conclusão da transformação. Desta minimização resulta uma afecção de quantidades de trabalho aos recursos em períodos de tempo bem definidos, isto é, resulta um programa de produção e a respectiva configuração específica para cada transformação. A actualização das agendas dos recursos, entre aplicações do modelo de optimização a cada transformação, permite encadear a execução das várias transformações pelos recursos, obtendo uma configuração específica para o SDP, de forma a produzir as quantidades de produto final encomendadas.

Para o caso de teste, Estrutura 2 da Figura 47, é necessário começar por colocar uma encomenda para o objecto  $o_3$ . Inicialmente considera-se que todos os recursos têm a sua agenda completamente livre, conforme representado na Tabela 17. Depois de seleccionados os recursos para a produção desta encomenda é necessário actualizar a sua agenda. Em seguida, coloca-se uma encomenda para o objecto  $o_4$  e volta-se a actualizar a agenda. As transformações  $t_1$  e  $t_2$  podem iniciar-se no instante zero, mas  $t_3$  só se pode iniciar depois de obtidos os objectos  $o_3$  e  $o_4$ . Desta forma, o instante de início de  $t_3$  não é inferior ao máximo instante de conclusão das encomendas destes objectos. A experimentação é composta pela aplicação sucessiva de processos de minimização do instante de conclusão para as três transformações, com variação do número de recursos máximo que se admite para cada uma delas. De acordo com a descrição apresentada do caso de teste, este número pode variar entre 1 e 7 recursos por transformação.

Os dados do caso de teste são introduzidos em folhas de cálculo Excel. A execução dos testes é realizada em Excel com recurso a alguns procedimentos desenvolvidos em VBA<sup>41</sup> e a processos externos de optimização. Para execução dos processos de optimização foi utilizado o programa Xpress-MP 2003 com o código Mosel indicado em anexo. Os processos de optimização executados no programa Xpress-MP recebem, a partir de um ficheiro de texto exemplificado em anexo, os seguintes dados de entrada:

- O número de transformações e de recursos do sistema de produção. Neste caso, temos sempre uma transformação e 7 recursos.
- A quantidade de trabalho requerida na transformação para obtenção do número de unidades de produto encomendadas, i.e. 10 unidades.
- Agenda livre de cada recurso, por indicação de pares de valores indicando os instantes iniciais e finais de cada período livre.
- Alguns valores limite para as restrições do modelo de optimização.
- O tipo de optimização que se pretende. Neste caso, pretende-se executar a minimização do instante de conclusão com a restrição adicional do número de recursos por transformação.

O processo de experimentação representado no diagrama de fluxo da Figura 48, é composto pela inicialização dos dados, seguida da execução sucessiva de processos de minimização do instante de conclusão com restrição do número de recursos. Começa-se com o número mínimo de um recurso por transformação e incrementa-se esse valor até ao número máximo de 7 recursos por transformação. No diagrama da Figura 48 o número de recursos é representado pela variável “res” e o instante de conclusão pela variável “ $T_n^+$ ”.

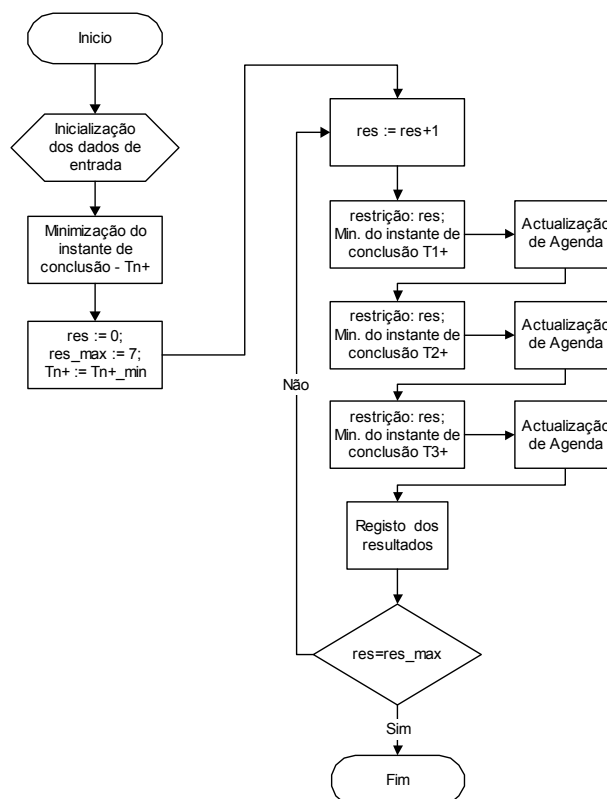


Figura 48: Diagrama de Fluxo do processo de experimentação para modelo local.

<sup>41</sup> “Visual Basic for Applications”.

### 1.3.2. Resultados

Nesta experimentação efectua-se 7 testes, considerando em cada um deles um número variável de recursos, entre 1 e 7 recursos por transformação. Isto corresponde a uma variação do Grau de Simultaneidade de cada transformação entre 0.14 e 1.00. Por definição de Grau de Simultaneidade (secção IV - 1.3) do sistema e de uma transformação, se o número de recursos candidatos for igual em todas as transformações e a utilização de recursos por transformação também for igual, então o Grau de Simultaneidade do sistema vai ser igual ao Grau de Simultaneidade de cada transformação. Neste caso de teste impõe-se a utilização de um número igual de recursos em todas as transformações, que são seleccionados a partir do mesmo conjunto de recursos candidatos, pelo que se verifica a igualdade entre os Grau de Simultaneidade do sistema e de todas as transformações.

Na Tabela 19 apresentam-se os resultados dos testes realizados, com afectação de recursos para cada transformação. A coluna “res” apresenta o número de recursos máximo imposto para cada teste, “GS” o respectivo Grau de Simultaneidade e “T1+”, “T2+” e “T3+” os instantes de conclusão de cada transformação.

Tabela 19: Resultados – Modelo Local de Afectação de Recursos.

Teste	Parâmetro		Resultados		
	GS	res	T1+	T2+	T3+
1	0,14	1	10	10	20
2	0,29	2	5	5	10
3	0,43	3	4	4	8
4	0,57	4	3	3	6
5	0,71	5	2	4	6
6	0,86	6	2	3	5
7	1,00	7	2	3	5

Na Figura 49 representam-se os instantes de conclusão de cada transformação e as curvas de aproximação destes instantes em função do Grau de Simultaneidade. É possível verificar que o instante de conclusão de processamento da encomenda, T3+, tem uma variação similar ao da função  $1/x$ . Uma pequena variação do Grau de Simultaneidade para valores baixos tem um maior impacto na diminuição do instante de conclusão do que a mesma variação para valores próximos da unidade. Pode-se afirmar que se obtém ganhos maiores para valores menores de GS.

Nesta experimentação, o instante de conclusão varia 15 UT, entre 20 e 5 UT, para graus de simultaneidade entre 0.14 e 1.00. No entanto, entre graus de simultaneidade de 0.43 a 1.00, os instantes de conclusão só variam 3 UT entre 8 e 5 UT. A utilização, para este caso, de um Grau de Simultaneidade de 0.43 permite obter instantes de conclusão próximos do mínimo.

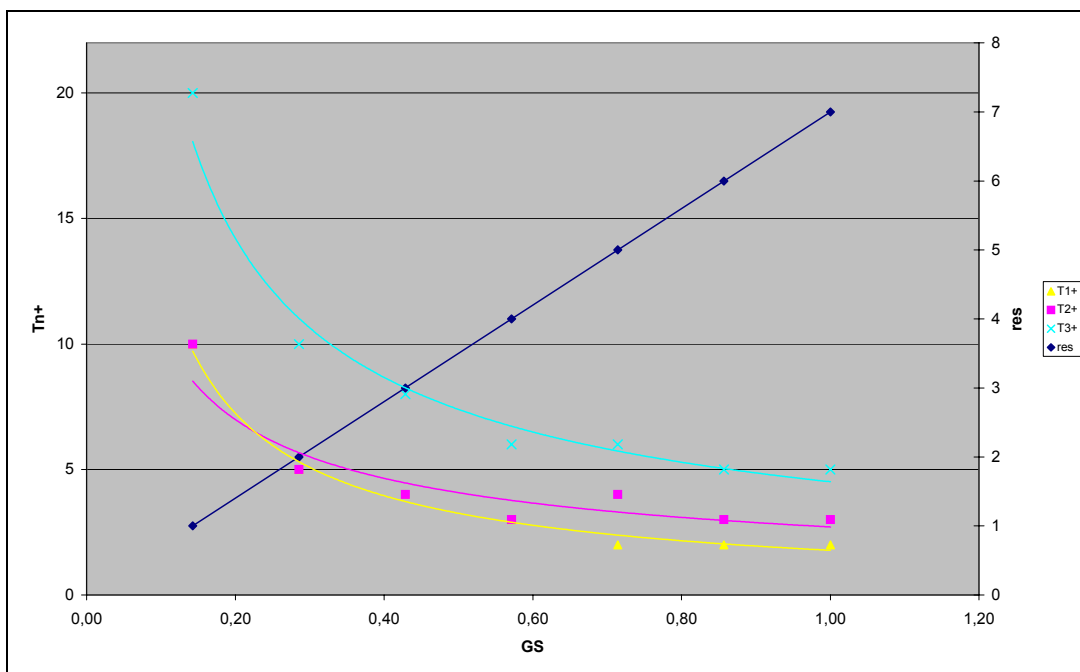


Figura 49: Curvas de Variação dos Instantes de Conclusão das Transformações

#### 1.4 Modelo Global de Afecção de Recursos

A aplicação do modelo global de afectação de recursos permite construir um programa de produção, com afectação de trabalho a recursos, equacionada em simultâneo, para todas as transformações de uma encomenda. No modelo local, utilizado na secção anterior V - 1.3, esta afectação de recursos realiza-se para uma transformação de cada vez. Na possibilidade de configuração associada ao modelo global, descrita na secção IV - 2.2.2, apresenta-se uma relação de variação do mínimo instante de conclusão de uma encomenda em função do Grau de Simultaneidade. Além desta variação, também se apresentam possibilidades de variação dos custos de posse e preparação, modelados de uma forma simplificada. Assim sendo, pretende-se verificar, em função do Grau de Simultaneidade, a forma como varia o instante de conclusão e a forma como varia o custo de posse e o custo de preparação. Para este efeito, é necessário definir uma forma de cálculo do custo de posse e do custo de preparação, sendo, neste processo de análise, considerados modelos simples de cálculo destas variáveis.

Assume-se, de acordo com a secção IV - 2.2.2, que o custo de preparação apenas depende do número de recursos envolvidos, desprezando-se, neste caso, o relacionamento entre operações executadas no recurso. Assim sendo, o custo de preparação é directamente proporcional ao número de recursos envolvidos no processamento da mesma encomenda. Nesta análise, para simplificação de cálculo, assume-se que o custo de preparação é idêntico ao número de recursos. Assume-se ainda, que o custo de posse depende do valor investido no produto, do número de unidades, e do tempo de armazenamento. Assume-se também, que um produto tem um valor unitário, que depois de transformado passa a ter um valor superior, resultante da adição de mais uma unidade, e que uma transformação que suceda a duas outras deve considerar a soma do valor dos produtos depois das duas transformações.



### **1.4.1. Descrição da Experimentação**

Considera-se, de acordo com a descrição efectuada anteriormente, que se coloca uma encomenda para todo o sistema, isto é, para todas as transformações. Utiliza-se o modelo de optimização global com minimização do instante de conclusão da encomenda para afectação global de recursos às transformações. Desta minimização resulta uma afectação de quantidades de trabalho de todas as transformações aos recursos, em períodos de tempo bem definidos, isto é, resulta um programa de produção, e a respectiva configuração específica para o SDP.

A experimentação, conforme foi referido, consiste em executar um conjunto de testes, determinando-se, em cada um destes, uma configuração específica para o SDP. Esta experimentação executa-se de acordo com o Diagrama de Fluxo representado na Figura 50. A configuração específica para o SDP, resulta da aplicação do modelo de optimização global ao caso de teste, com restrições do número de recursos sucessivamente diferentes.

Os dados do caso de teste, tal como no caso do modelo local, são introduzidos em folhas de cálculo Excel. A execução dos testes é realizada em Excel com recurso a alguns procedimentos desenvolvidos em VBA e a processos externos de optimização. Para execução dos processos de optimização foi utilizado o programa Xpress-MP 2003 com o código Mosel indicado em anexo. Os processos de optimização executados no programa Xpress-MP recebem, a partir de um ficheiro de texto exemplificado em anexo, os seguintes dados de entrada:

- O número de transformações e de recursos do sistema de produção. Neste caso de teste, temos 3 transformações e 7 recursos que podem ser utilizados por qualquer das transformações.
- A estrutura do sistema de produção, através da indicação das transformações que precedem outras e das que são paralelas a outras. Neste caso de teste, temos duas precedências, das transformações  $t_1$  para  $t_3$  e de  $t_2$  para  $t_3$ . Temos ainda uma condição de paralelismo entre as transformações  $t_1$  e  $t_2$ .
- A quantidade de trabalho requerida na transformação para obtenção do número de unidades de produto encomendadas, i.e. 10 unidades.
- Agenda livre de cada recurso, por indicação de pares de valores indicando os instantes iniciais e finais de cada período livre.
- Alguns valores limite para as restrições do modelo de optimização.
- O tipo de optimização que se pretende, podendo considerar-se os seguintes critérios: minimização do instante de conclusão, minimização da quantidade total de trabalho, minimização do número de recursos, e ainda com adição de uma restrição para um destes critérios seguido de minimização para outro. Neste caso de teste, pretende-se executar a minimização do instante de conclusão com a restrição adicional do número de recursos por transformação.

O processo de experimentação, especificado na Figura 50, resulta da aplicação sucessiva de processos de minimização do instante de conclusão da encomenda, variando o número máximo de recursos admitidos, conforme descrição anterior. Este processo permite, por variação do número máximo de recursos, construir uma curva de variação do instante de conclusão em função do número recursos utilizado. Como o Grau de Simultaneidade é directamente proporcional ao número de recursos utilizado, a referida curva também estabelece a relação com o Grau de Simultaneidade.

O primeiro passo da experimentação é o de execução de um processo de inicialização dos dados, seguido de uma minimização do número de recursos e minimização do instante de conclusão. A minimização do instante de conclusão sem restrição do número de recursos, permite obter uma solução para a configuração do sistema com um determinado número de recursos que se vai considerar máximo. Este valor considera-se máximo, porque com o aumento do número de recursos para além desse valor, não se consegue diminuir o instante de conclusão da encomenda. Em seguida executam-se sucessivamente processos de minimização, recorrendo ao programa Xpress-MP, do instante de conclusão com variação do número de recursos admitido para a execução da encomenda. Inicia-se o processo com o número mínimo e incrementa-se até que o número de recursos seja igual ao número que se estabeleceu como máximo. No diagrama da Figura 50 o número de recursos utilizados por todo o sistema, i.e. por todas as transformações do sistema, é representado pela variável “res” e o instante de conclusão pela variável “ $T_n$ ”.

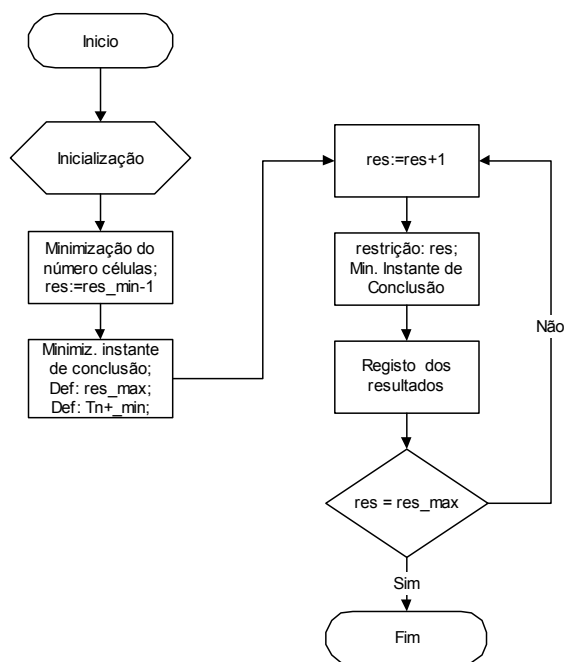


Figura 50: Diagrama de Fluxo do processo de experimentação para o modelo global.

### **1.4.2. Resultados**

Nesta experimentação executam-se tantos testes quantas as restrições impostas ao número admitido de recursos para a execução da encomenda. A minimização do instante de conclusão da encomenda sem restrição do número de recursos permite obter uma solução de configuração para o SDP. Como foi referido, o número de recursos utilizados nesta configuração é definido como o número máximo de recursos da experimentação, na medida em que o aumento do número de recursos não permite diminuir o instante de conclusão. No caso de teste, este número é igual a 18 recursos. Como o caso de teste é composto por 3 transformações, no mínimo são necessários 3 recursos para executar a encomenda. Assim sendo, são realizados 16 testes, variando o número de recursos entre 3 e 18.

A variação do número de recursos, e a conseqüente variação do Grau de Simultaneidade, permite analisar a forma de variação do instante mínimo de conclusão da encomenda relativamente ao Grau de Simultaneidade. A variação do número de recursos permite ainda, analisar a variação dos custos de

preparação e de posse, calculados da forma descrita na secção V - 1.4, em função do Grau de Simultaneidade. Executa-se o processo de experimentação variando o número de recursos entre 3 e 18 recursos para o sistema total, o que corresponde a uma variação do Grau de Simultaneidade entre 0.14 e 0.86. Estes valores estão representados na coluna de parâmetros da Tabela 20. Em cada teste fixa-se o número de recursos máximo para o conjunto das três transformações e é efectuada a minimização do instante de conclusão da encomenda, isto é, de  $t_3$ .

Tabela 20: Resultados - Selecção de Recursos para todas as Transformações.

Teste	Parâmetros		Resultados		
	GS	res	Variáveis Medidas		Critério
			Cs	Ch	Tn+
1	0,14	3	3	600	20
2	0,19	4	4	400	15
3	0,24	5	5	364	14
4	0,29	6	6	300	10
5	0,33	7	7	260	9
6	0,38	8	8	220	8
7	0,43	9	9	180	7
8	0,48	10	10	210	7
9	0,52	11	11	164	6
10	0,57	12	12	174	6
11	0,62	13	13	144	5
12	0,67	14	14	144	5
13	0,71	15	15	144	5
14	0,76	16	16	144	5
15	0,81	17	17	144	5
16	0,86	18	18	144	5

Na Tabela 20 apresentam-se os resultados de experimentação para a afectação global de recursos a todas transformações. A coluna “res” apresenta o número de recursos máximo imposto para cada teste, a coluna “GS” o respectivo Grau de Simultaneidade e a coluna “Tn+” o instante de conclusão de  $t_3$ , em unidades de tempo. Os custos de preparação e posse, de acordo com a secção V - 1.4, são medidos em cada solução de configuração, sendo indicados em unidades monetárias, respectivamente por “Cs” e “Cp”.

Para efeitos de apresentação das curvas efectuou-se uma normalização dos dados que permite visualizar os custos de posse e de preparação no mesmo eixo. Em ambos os casos utilizou-se um valor majorante que permite obter resultados abaixo de uma unidade: para Cs utiliza-se o seu valor máximo possível, isto é o número máximo de recursos 21; para Ch um majorante  $K$  superior ao valor máximo medido. Neste caso, para normalização dos valores, utiliza-se um valor  $K$  tal que, a soma dos custos, quando Ch é máximo, seja igual a uma unidade de custo. Os resultados normalizados desta experimentação estão representados na Tabela 21. A soma dos custos de preparação e de posse está representada na coluna *Custos*.

Na Figura 51 representam-se os instantes de conclusão da encomenda em função do Grau de Simultaneidade e a curva de aproximação a esses instantes. É possível verificar que o instante de conclusão do processamento da encomenda  $Tn+$ , tem uma variação similar ao da função  $1/x$ . A variação do Grau de Simultaneidade em valores próximos de zero tem um maior impacto na diminuição do instante de conclusão do que a mesma variação para valores próximos da unidade.

Tabela 21: Resultados normalizados - Seleção de Recursos para todas as Transformações.

Teste	Parâmetro GS	Resultados			Critério Tn+
		Variáveis Medidas			
		Cs (/21)	Ch (/K)	Custos	
1	0,14	0,14	0,86	1,00	20
2	0,19	0,19	0,57	0,76	15
3	0,24	0,24	0,52	0,76	14
4	0,29	0,29	0,43	0,72	10
5	0,33	0,33	0,37	0,71	9
6	0,38	0,38	0,32	0,70	8
7	0,43	0,43	0,26	0,69	7
8	0,48	0,48	0,30	0,78	7
9	0,52	0,52	0,24	0,76	6
10	0,57	0,57	0,25	0,82	6
11	0,62	0,62	0,21	0,83	5
12	0,67	0,67	0,21	0,87	5
13	0,71	0,71	0,21	0,92	5
14	0,76	0,76	0,21	0,97	5
15	0,81	0,81	0,21	1,02	5
16	0,86	0,86	0,21	1,06	5

Para a experiência realizada, um valor de GS de 0,43 (Tabela 21) permite atingir valores do instante de conclusão da encomenda próximo do mínimo. A variação do Grau de Simultaneidade de aproximadamente 3 décimos, entre 0,14 e 0,43, permite reduzir o instante de conclusão da encomenda em 13 UT, de 20 para 7 UT. A mesma variação do GS de aproximadamente 3 décimos, entre 0,43 e 0,76, permite reduzir o instante de conclusão da encomenda apenas 2 UT, de 7 para 5 UT. Com o mesmo valor de GS de 0,43 obtém-se um custo operacional, i.e. custo de posse mais preparação, mínimo para esta experimentação. Valores superiores de GS permitem reduzir marginalmente o instante de conclusão, por se encontrar na zona de ganhos fortemente decrescentes, mas penalizam o custo operacional, i.e. aquele que resulta da preparação da produção e posse de materiais.

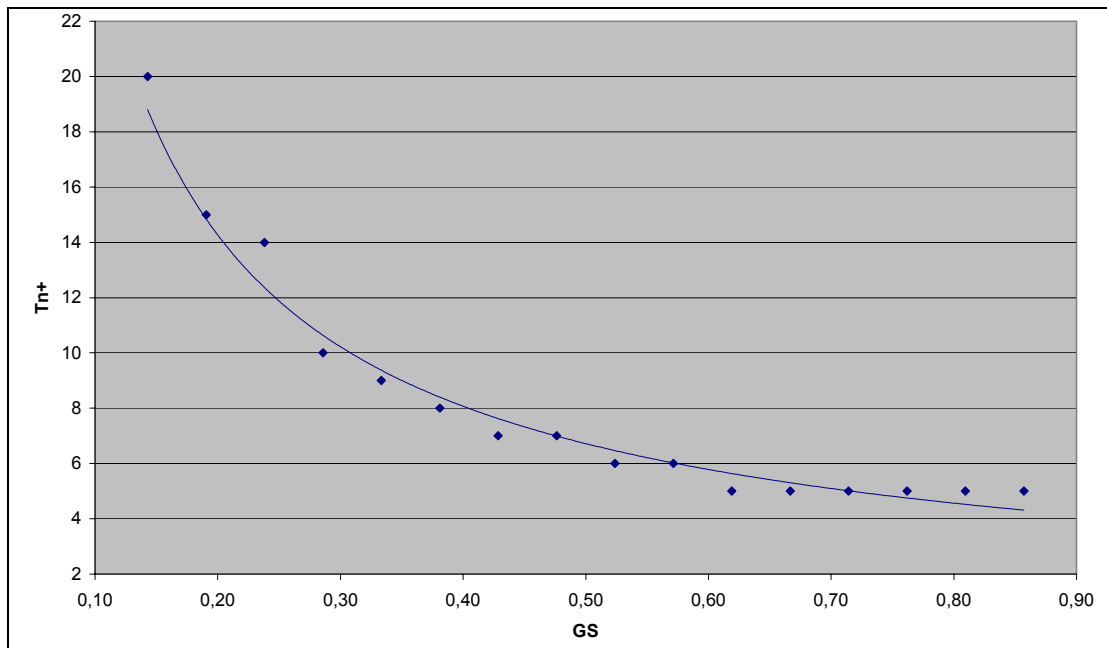


Figura 51: Curva de Variação do Instante de Conclusão

Na Figura 52 representam-se os custos operacionais de execução de cada encomenda associados a cada configuração do sistema resultante da aplicação do modelo global de afectação de recursos. Nesta figura também se apresentam as curvas de aproximação desses custos. Com a forma de cálculo de custos estabelecida no capítulo anterior, é possível verificar que, relativamente ao Grau de Simultaneidade, a variação do custo de preparação é linear e a variação do custo de posse é similar ao da função  $1/x$ . Uma pequena variação do GS para valores próximos de zero tem um maior impacto na diminuição do custo de posse do que a mesma variação para valores próximos da unidade. O custo operacional mínimo, conforme referido atrás obtém-se para um valor GS na ordem de grandeza de 0,40, em particular de 0,43.

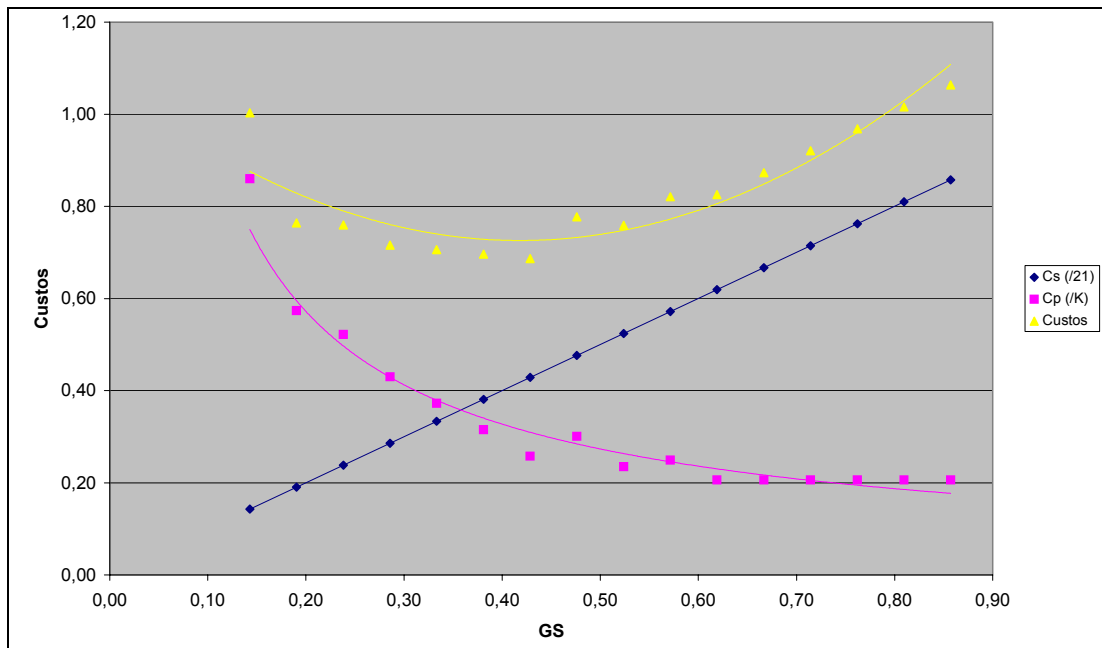


Figura 52: Curvas de Variação dos Custos e do Instante de Conclusão

### 1.5 Considerações Finais sobre Possibilidades de Configuração do Sistema

A aplicação da Produção Simultânea para projecto de um Sistema Distribuído de Produção, permite efectuar a afectação de recursos para satisfação de uma encomenda. A configuração do sistema depende da aplicação de modelos de afectação locais ou globais. Nas possibilidades de configuração definidas não se pretende analisar a influência da utilização de diferentes tipos de recurso e com diferentes agendas. Por esta razão utilizam-se recursos idênticos para a execução de todas as transformações, o que permite analisar as configurações do sistema sem influência do tipo de recurso ou da sua agenda. A existência de um reduzido número de recursos face à quantidade de trabalho a executar em cada transformação, faz com que na maioria dos testes, a minimização do instante de conclusão, resulte na afectação de recursos a mais do que uma transformação. Considerando todas as quantidades de utilização de objectos iguais a uma unidade, a quantidade de trabalho a executar, nos testes realizados, é igual à quantidade encomendada do produto, pelo que se define como limite para o número de recursos um número idêntico ao de objectos encomendados.

A utilização de recursos do mesmo tipo e com a mesma agenda associada à definição de quantidades iguais de utilização dos objectos pelas transformações, faz com que todas as transformações tenham o mesmo peso sobre o sistema. Isto significa que existe um fluxo equilibrado, isto é, todas as transformações

têm que executar a mesma quantidade de trabalho e os recursos demoram o mesmo tempo a executar essa quantidade de trabalho. Assim sendo, não existe, à partida uma associação recurso / transformação que restrinja o fluxo de todo o sistema, isto é, que seja um estrangulamento do sistema. O estrangulamento do sistema SDP, baseado no conceito de estrangulamento referido por Goldratt and Fox (1986), é a transformação que demora mais tempo a ser executada pelos recursos que lhe são afectados, em relação a outras que processam a mesma encomenda ou o mesmo conjunto de encomendas. Ainda segundo Goldratt and Fox (1986) não se tira vantagem da execução mais rápida de outras transformações que não sejam estrangulamentos, apenas se aumentam os custos operacionais, por se aumentar as despesas com matérias primas e aumentar os trabalhos em curso. De acordo com esta noção de estrangulamento, a aplicação da simultaneidade a algumas transformações pode ter mais influência sobre o instante de conclusão da encomenda do que a aplicação a outras transformações.

A minimização do instante de conclusão de cada transformação, variando o número de recursos utilizados, permite obter uma curva de variação daquele instante em função do Grau de Simultaneidade. Sendo o GS directamente proporcional ao número de recursos utilizados. A minimização do instante de conclusão de cada transformação pressupõe a utilização do modelo de afectação de recursos a uma transformação seguido da actualização da agenda dos recursos, e assim sucessivamente até terminar a afectação a todas as transformações. Para a configuração do sistema, obtida pela afectação de recursos para uma transformação de cada vez conclui-se que, no caso em análise, o instante de conclusão da encomenda tem uma variação, relativamente ao Grau de Simultaneidade, similar ao da função  $1/x$ . Nesta curva, valores de GS à volta de 0.40 permitem obter instantes de conclusão próximos do mínimo.

A minimização do instante de conclusão da encomenda, variando o número de recursos utilizados, permite obter uma curva de variação daquele instante em função do Grau de Simultaneidade. O estudo das configurações do sistema obtidas pela selecção de recursos para todas as transformações simultaneamente permite concluir que, no caso em análise, o instante de conclusão da encomenda e o custo de posse têm uma variação, relativamente ao Grau de Simultaneidade, similar ao da função  $1/x$ . Os resultados obtidos mostram que, para o caso em análise, o instante de conclusão é próximo do mínimo, para valores de GS da mesma ordem de grandeza dos que permitem obter um mínimo custo operacional. Nesta análise, a um valor de GS de 0.43, corresponde um mínimo custo operacional, e um instante de conclusão próximo do mínimo.

## **2. APLICAÇÃO DO MODELO MSDP EM AMBIENTE DE PS A VÁRIAS ENCOMENDAS**

A representação formal do Sistema Distribuído de Produção apresentada na secção III - 2, permite utilizar modelos de afectação de recursos a todas transformações de um produto, criando assim, uma instância de um nível do sistema de produção para uma encomenda desse produto. Pretende-se verificar a possibilidade de aplicação do modelo MSDP em ambiente de PS, para determinar soluções de programação da produção de uma sequência de processamento de várias encomendas.

## **2.1 Plano de Experimentação**

Os modelos de optimização baseados em diferentes critérios e restrições, que permitem criar uma instância do sistema de produção para uma encomenda de um produto, foram formalizados, de acordo com a Produção Simultânea, na secção IV - 3.2. A determinação de instâncias para outras encomendas requer repetidamente, a execução da actualização das agendas dos recursos seguida da aplicação do modelo de optimização com base nas novas agendas, numa sequência predeterminada. Este procedimento conjunto de escalonamento, utilizando o modelo referido e a actualização sistemática e sucessiva das agendas dos processadores após o escalonamento de uma encomenda, designa-se neste trabalho por método DISMAN, acrónimo derivado de “DIStributed MANufacturing”. O método DISMAN utiliza o modelo de optimização com o critério de minimização do instante de conclusão da encomenda e com as restrições genéricas e de agenda definidas nas secções IV - 3.2.1 e IV - 3.2.2, respectivamente. Neste plano de experimentação, um processo de *experimentação* consiste na aplicação do método DISMAN a um caso de teste composto por um conjunto de encomendas. Em cada processo de experimentação, podem realizar-se vários *testes*, correspondentes à alteração da forma de processamento das encomendas. Isto significa que se aplica o método DISMAN para uma encomenda composta por um lote de todas as unidades encomendadas, ou por vários lotes correspondentes à partição do lote original.

Consideramos neste processo de experimentação que a sequência pela qual as encomendas ou ordens de fabrico são escalonadas foi previamente estabelecida. Tal sequência pode interpretar-se como aquela em que os trabalhos são lançados para produção. Fica, assim, por resolver o problema da selecção de recursos para cada encomenda, que resolvido define uma configuração específica do sistema para a encomenda, e ainda o da calendarização dos trabalhos, i.e. definição dos instantes de execução de cada transformação dos produtos encomendados.

Cada solução obtida pelo método DISMAN para cada encomenda, é óptima sob o ponto de vista do critério de optimização seleccionado e das restrições impostas à afectação de recursos. Há por isso necessidade de avaliar a solução total que integre o escalonamento de todas as encomendas. Tal *avaliação* pode ser equacionada segundo o critério de optimização seleccionado e outras *medidas de desempenho*. O desempenho é avaliado por medidas consideradas relevantes para a análise deste método, como sejam medidas relacionados com a rapidez de conclusão das encomendas, que é um dos objectivos da filosofia de Produção Simultânea. Assim sendo, consideram-se as medidas de instante de conclusão de todas as encomendas, somatório dos instantes de conclusão e uma medida da Qualidade de Serviço de Produção (QMS) ao cliente, introduzida por Silva, S. C. (2003). Além destas medidas também se considera a medida de Utilização dos recursos, que tende a aumentar com a diminuição do instante de conclusão das encomendas. Estas medidas são definidas formalmente numa secção posterior.

A análise baseia-se em *casos de testes* utilizados no trabalho de Almeida (2002). O interesse desta opção para estudo comparativo de soluções reside no facto do trabalho referido utilizar a abordagem ao escalonamento que se equaciona nesta análise, isto é, a orientação ao produto e à Produção Simultânea. Não se encontraram referências a esta dupla orientação noutros trabalhos. Almeida (2002) utiliza um método, designado MEOP, baseado na criação de um padrão de escalonamento para um produto, onde se considera a

utilização de lotes de transferência variáveis e limitados ao tamanho do lote de produção. A sequência de lançamento das encomendas é predeterminada pelo utilizador do método. O escalonamento sucessivo de encomendas de produtos é resolvido pelo encaixe dos respectivos padrões, nas agendas dos processadores, até se obter um escalonamento global para todos os produtos considerados. Deve realçar-se que, no trabalho de Almeida (2002), se utiliza o princípio de processamento não interrompido de cada encomenda, explorando-se o processo de sobreposição de lotes (“batch overlapping”). Desta forma, consegue-se, normalmente, reduzir o número de vezes que se executam trabalhos de preparação dos meios de produção em comparação com o que se teria de executar no caso de se considerar a partição dos lotes (“batch splitting”). Tal abordagem simplifica, relativamente à partição de lotes, o processo de controlo dos trabalhos em curso porque, em cada recurso, o lote é processado sem interrupção, não existindo necessidade de criação de mecanismos de sincronização entre diferentes partes do mesmo lote.

Dos problemas apresentados por Almeida (2002), seleccionaram-se três, de aparente maior complexidade, por envolverem mais tarefas de transformação, recursos e/ou produtos. Os problemas seleccionados são do tipo “Flexible Job Shop” (FJS), definido por Pinedo and Chao (1999) como uma “Job Shop” que utiliza centros de trabalho com máquinas alternativas em paralelo. Numa “Job Shop” os vários trabalhos podem ter sequências de fabrico variadas. Em todos estes problemas utilizam-se recursos alternativos para pelo menos uma operação de cada encomenda. A existência de recursos alternativos permite explorar a capacidade do método DISMAN atribuir simultaneamente a recursos alternativos a produção do mesmo lote de uma encomenda, usando esquemas de partição.

Os *resultados* relativos ao método DISMAN foram obtidos através da experimentação baseada na aplicação do modelo MSDP em ambiente de PS, desenvolvido neste trabalho. Os resultados do método MEOP são retirados do trabalho de experimentação resultante da aplicação do modelo desenvolvido e apresentado por Almeida (2002). Todos os resultados apresentados, relativos a outros métodos de escalonamento conhecidos, resultam da utilização do programa LEKIN (Pinedo and Chao (1999)), desenvolvido na Stern School of Business, New York University sob a supervisão dos Professores Michael L. Pinedo, Xiuli Chao e Joseph Leung. As heurísticas seleccionadas por Almeida (2002) e neste trabalho de experimentação para comparação com os métodos MEOP e DISMAN são descritas detalhadamente por Pinedo and Chao (1999) e implementados no sistema LEKIN. As referidas heurísticas são as seguintes: SPT - “Shortest Processing Time”; EDD - “Earliest Due Date”; FCFS - “First Come First Served”; SB/ $C_{max}$  - “Shifting Bottleneck” com o critério de minimização do tempo máximo de conclusão de todos os trabalhos; SB/ $\sum(WT)$  - “Shifting Bottleneck” com o critério de minimização do “atraso positivo”<sup>42</sup> pesado total de todos os trabalhos.

## **2.2 Medidas de Desempenho**

Para efeito de comparação do método DISMAN com outros métodos de programação da produção seleccionaram-se medidas de desempenho relacionadas com a conclusão do processamento das encomendas / ordens de produção e com a utilização dos recursos. O método DISMAN não tem como objectivo o aumento da utilização de recursos, mas a partição dos lotes de produção deverá permitir obter bons resultados.

---

<sup>42</sup> “Tardiness”.



As duas medidas directamente relacionadas com o instante de conclusão de processamento das encomendas referidas por Pinedo and Chao (1999) são o instante de conclusão  $C_{\max}$  de processamento de todas as encomendas e o somatório de todos os instantes de conclusão  $\sum C$  de processamento de todas as encomendas. Adicionalmente avaliam-se os resultados utilizando uma medida directamente relacionada com o instante de conclusão de processamento das encomendas referida por Silva, S. C. (2003): medida de Qualidade de Serviço de Produção QMS<sup>43</sup>. Em seguida apresenta-se a definição formal de cada uma destas medidas.

### **2.2.1. Instante de conclusão do Conjunto das Encomendas**

Considera-se, de acordo com o modelo apresentado nesta dissertação que cada encomenda resulta do processamento das transformações  $T_j$  de entrega de cada produto / componente que compõe o produto final, cujo instante de conclusão é representado por  $T_j^+$ . O instante de conclusão  $C_{enc}$  do processamento de uma encomenda  $enc$  de um produto é dado por:

$$\forall j \in \{1, \dots, m_{enc}\} : C_{enc} = \max(T_j^+) \quad \text{Equação 91}$$

Nesta experimentação considera-se que o sistema satisfaz um número máximo  $N\_ENC$  de encomendas  $enc$  sucessivas, sendo cada uma destas de várias unidades de um produto  $P_{enc}$ :

$$\forall enc \in \{1, \dots, N\_ENC\} : P_1 \leq P_{enc} \leq P_{N\_ENC} \quad \text{Equação 92}$$

Sendo assim, define-se o instante de conclusão de cada encomenda  $C_{enc}$  como o instante de conclusão da última transformação do produto encomendado  $P_{enc}$ , e dessa forma o instante de conclusão de todas as encomendas resulta de:

$$\forall enc \in \{1, \dots, N\_ENC\} : C_{\max} = \max(C_{enc}) \quad \text{Equação 93}$$

Esta medida permite verificar, em termos absolutos, qual o método de programação de produção que fornece uma solução com menor instante de conclusão do conjunto das encomendas.

### **2.2.2. Somatório dos Instantes de Conclusão das Encomendas**

Esta medida permite agregar os instantes de conclusão de todas as encomendas permitindo verificar qual dos métodos se comporta melhor perante o conjunto das encomendas, isto é, aquele que permite minimizar a média dos instantes de conclusão.

Considerando a Equação 91 e a Equação 92 o Somatório dos Instantes de Conclusão das Encomendas resulta de:

$$\sum C = \sum_{enc=1}^{N\_ENC} C_{enc} \quad \text{Equação 94}$$

<sup>43</sup> "Quality of Manufacturing Service".

### 2.2.3. Qualidade de Serviço de Produção (QMS)

O desenvolvimento desta medida por Silva, S. C. (2003) tem como objectivo avaliar a qualidade de serviço prestado aos clientes de várias encomendas do ponto de vista do sistema de produção. Pressupõe-se que a diminuição do tempo entre a colocação da encomenda de um produto e a sua entrega aumenta a satisfação do cliente. O sistema de produção pode contribuir para esta diminuição diminuindo o intervalo de tempo para processar a encomenda. O processamento de várias encomendas por um método terá uma maior qualidade de serviço relativamente a outro método se num determinado horizonte de tempo  $H$ , o somatório dos períodos de tempo que medeiam entre a conclusão do processamento das encomendas e o horizonte for maior. Neste caso, haverá no sistema mais tempo livre de processamento e em geral as encomendas concluem-se mais cedo. Considerando o número máximo de  $N\_ENC$  encomendas e a definição do instante de tempo de conclusão  $C_{enc}$  de uma encomenda  $enc$  representada na Equação 91, a medida de Qualidade de Serviço de Produção de Silva, S. C. (2003) é formulada pela Equação 95.

$$QMS = \frac{\sum_{enc=1}^{N\_ENC} (H - C_{enc})}{H} \quad \text{Equação 95}$$

A medida de Qualidade de Serviço de Produção, normalizada de forma a apresentar valores entre zero e um para instantes de conclusão inferiores a  $H$ , resulta de:

$$QMS = \frac{\sum_{enc=1}^{N\_ENC} (H - C_{enc})}{n \cdot H} \quad \text{Equação 96}$$

Em todos os exemplos de experimentação apresentados o valor estabelecido para horizonte temporal foi o máximo tempo de conclusão de todas as encomendas por aplicação de todos os métodos utilizados. Considerando o conjunto de todos os métodos definido por *Metodos*, então o horizonte temporal  $H$  resulta de:

$$\forall met \in Metodos : H = \max(C_{\max,met}) \quad \text{Equação 97}$$

### 2.2.4. Utilização de Recursos

Esta medida permite medir o tempo de utilização dos recursos durante o tempo de satisfação de todas as encomendas. Um determinado recurso  $k$  é utilizado durante um período para a execução de uma transformação de processamento de um produto numa encomenda, sendo esse tempo de utilização determinado pela subtracção entre o tempo de conclusão desse processamento  $T_{jk}^+$  e o tempo de início  $T_{jk}^-$ :

$$\forall j \in \{1, \dots, m\}, \forall k \in \{1, \dots, r\} : T_{jk}^+ - T_{jk}^- \quad \text{Equação 98}$$

Considerando o processamento ininterrupto de um lote num recurso, a medida de utilização associada a um determinado método corresponde ao quociente do somatório de todos os períodos de processamento de todos os  $k$  recursos pelo tempo total disponível no intervalo em que se está a medir a sua utilização, isto é, pelo produto entre o número de recursos e o instante máximo de conclusão de todas as encomendas.

Depreende-se que a minimização do instante de conclusão de todas as encomendas conduz à minimização da medida de utilização.

$$Utilização = \frac{\sum_{j=1}^n \sum_{k=1}^r (T_{jk}^+ - T_{jk}^-)}{r \cdot C_{\max}} \quad \text{Equação 99}$$

No método DISMAN aplica-se partição com separação de lotes que leva à existência de períodos não contíguos de processamento de uma transformação num recurso, obrigando ao cálculo da utilização do recurso pelo somatório das durações de todos esses períodos. Considerando para a execução de cada transformação  $j$  a utilização do recurso  $k$  num número  $per_{jk}$  de períodos de processamento:

$$Utilização = \frac{\sum_{j=1}^n \sum_{k=1}^r \sum_{z=1}^{per_{jk}} (T_{jkz}^+ - T_{jkz}^-)}{r \cdot C_{\max}} \quad \text{Equação 100}$$

### 2.3 Descrição da Experimentação

De acordo com a descrição efectuada anteriormente, coloca-se uma encomenda para execução pelo sistema de produção, isto é, para todas as transformações. A satisfação desta encomenda requer que se efectue a afectação das quantidades de trabalho necessárias de cada transformação aos recursos disponíveis. Para esta afectação utiliza-se o modelo de optimização global com minimização do instante de conclusão da encomenda para afectação global de recursos às transformações, com as restrições genéricas, de agenda apresentadas, respectivamente, nas secções IV - 3.2.1 e IV - 3.2.2. Depois da programação desta encomenda efectua-se a programação das outras encomendas numa sequência predefinida e que nesta experimentação coincide sempre com a sequência utilizada por Almeida (2002).

A experimentação realiza-se com base em três casos de teste. Teremos assim, três processos de experimentação, cada um relacionado com um destes casos, em que estão definidas as estruturas do produto e respectivas estruturas do sistema, as encomendas realizadas e a respectiva sequência de lançamento e os recursos disponíveis. Para cada caso, determina-se uma solução de programação da produção para todas as encomendas, consideradas uma de cada vez na sequência predefinida.

Os dados de cada experimentação são introduzidos em folhas de cálculo Excel. A execução da experimentação é realizada em Excel com recurso a alguns procedimentos desenvolvidos em VBA e a processos externos de optimização. Entre cada processo de optimização realiza-se uma actualização das agendas dos recursos. Para execução dos processos de optimização foi utilizado o programa Xpress-MP 2003 com o código Mosel indicado em anexo. Os processos de optimização executados no programa Xpress-MP recebem, a partir de um ficheiro de texto exemplificado em anexo, os seguintes dados de entrada:

- Especificação dos produtos, definindo nomeadamente, o número de transformações, recursos utilizados, tempos de processamento unitário em cada recurso por cada transformação, quantidade de trabalho requerida em cada transformação, agenda livre de cada recurso, plano de processamento do produto, i.e. transformações consecutivas e paralelas, e limites superiores de algumas variáveis de decisão do modelo.

- Ordem de processamento dos produtos, respectiva quantidade requerida que permite calcular a carga necessária de cada recurso, e alguns dados auxiliares.
- Especificação do produto em processamento.
- Dados resultantes do último processo de optimização e Gráfico de Gantt com a respectiva ocupação dos recursos.
- Agenda de ocupação dos recursos dinamicamente actualizada, com dados acumulados de ocupação dos recursos, resultantes de todos os processos de optimização e respectivo gráfico de Gantt.
- O tipo de optimização que se pretende, nestes casos, de minimização do instante de conclusão da encomenda.

A experimentação é composta fundamentalmente pela aplicação sucessiva de processos de optimização. A especificação do processo de experimentação representa-se no diagrama de fluxo da Figura 53 e em anexo (Experimentação - CButton\_Go\_Click). É composta por inicialização dos dados e em seguida, para todas as optimizações requeridas, pela actualização do produto a processar, definição da agenda livre de cada recurso, gravação da especificação do produto, execução da optimização na linguagem Mosel utilizando o Xpress-MP 2003, leitura do resultado da optimização e actualização desses dados na folha de acumulação dos dados de ocupação dos recursos.

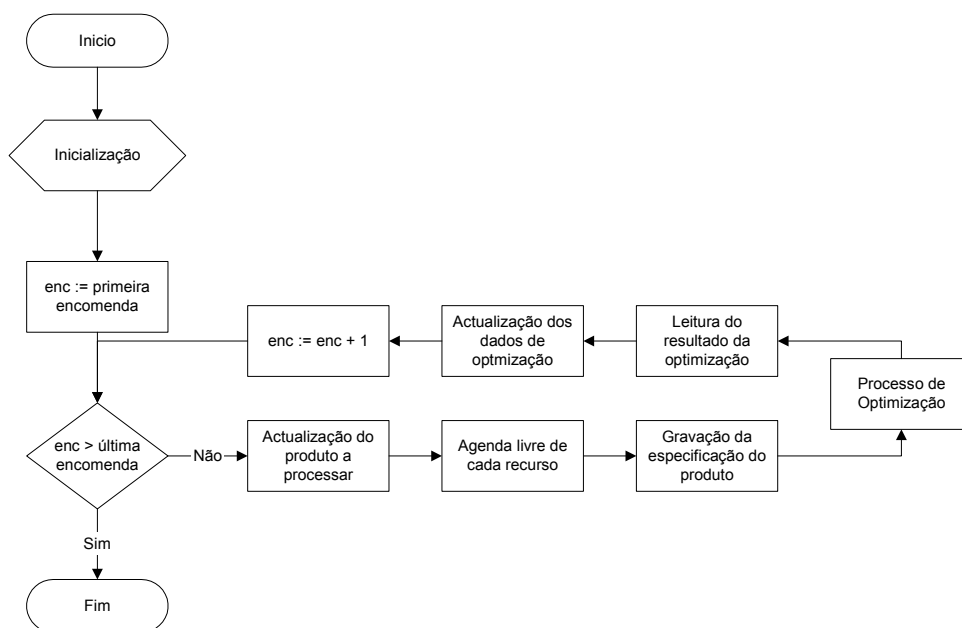


Figura 53: Diagrama de Fluxo do processo de experimentação

No processo de gravação da especificação do produto cria-se um ficheiro de texto com dados de inicialização do problema de optimização, lidos pelo Xpress-MP que por sua vez cria um outro ficheiro de texto com o resultado da optimização. Cada experimentação é executada sobre um conjunto de encomendas de um caso de teste. A experimentação mais longa executada nesta secção demorou, no total, aproximadamente 11 minutos. A maior parte do tempo de execução é utilizado pelos procedimentos realizados em Excel – VBA. Por não ser um objectivo do trabalho, o tempo de execução destes procedimentos não foi alvo de optimização, nem ao nível da selecção da ferramenta, nem da linguagem de programação, nem da codificação dos procedimentos. O processo de optimização é iterativamente executado em Xpress-MP

para afectação de recursos às transformações executadas para satisfação de uma encomenda de um produto. Este processo é o processo crítico relativamente ao aumento de dimensão, ou complexidade, do problema e utiliza, no pior dos casos analisados nesta secção, em média, 0,1 segundos. Para estes mesmos casos o processo total relativo a cada encomenda de um produto, de inicialização dos dados, execução do processo de optimização, actualização das agendas dos recursos, gravação e leitura de ficheiros auxiliares, demora em média 10 segundos. Sendo o tempo de execução da optimização igual a 1% do tempo total de execução, existe uma margem elevada de possível redução do tempo total de execução sem alterar a formulação do problema de optimização.

## 2.4 Caso AA\_FJS3x4

Este caso, apresentado por Almeida (2002), é baseado num sistema produtivo do tipo FJS, em que existe pelo menos um centro de trabalho da “Job Shop” provido com 2 ou mais processadores paralelos. Este sistema produtivo permite processar 4 produtos (P1, P2, P3 e P4) e é composto por 3 transformações ( $1 \leq j \leq n=3$ ) referentes a estágios de transformação sucessivos, que recorrem a 4 recursos ( $1 \leq k \leq r=4$ ) disponíveis. Todos os produtos têm uma estrutura similar, de tal forma que se pode representar a configuração genérica dos sistemas que os produzem da mesma forma, conforme representado na Figura 54. Uma vez instanciados os recursos nas transformações do sistema fica definida a configuração específica do sistema produtivo para a encomenda do produto. Os produtos são obtidos como resultado da terceira transformação  $t_3$ , isto é, correspondem ao objecto  $o_4$ . Uma unidade de quantidade de trabalho consome uma unidade de objecto e produz uma unidade de outro objecto, sendo as quantidades  $q_l$ ,  $1 \leq l \leq 6$  unitárias.

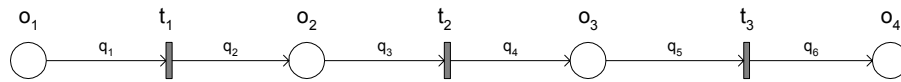


Figura 54: Representação idêntica para todos os produtos do caso AA\_FJS3x4

O início da produção dos produtos P1 e P4 só se pode dar no instante 5, altura em que as suas encomendas são lançadas:

$$P_1, P_4 : T_1^- \geq 5 ; P_2, P_3 : T_1^- \geq 0$$

O tempo de processamento unitário  $c_{jk}$  de uma transformação por um recurso, isto é, o tempo que o recurso  $k$  necessita para executar a transformação  $j$  uma vez, é apresentado na Tabela 22. Nesta tabela, os espaços vazios significam a não possibilidade de utilização desse recurso para processamento dessa transformação. Apenas os recursos  $r_2$  e  $r_3$  são paralelos e funcionalmente idênticos, podendo ser utilizados de uma forma indiferenciada.

Tabela 22: Tempo  $c_{jk}$  de processamento unitário de cada transformação por cada recurso

$c_{jk}$	$r_k$				$c_{jk}$	$r_k$				$c_{jk}$	$r_k$				$c_{jk}$	$r_k$				
	P1	1	2	3	4	P2	1	2	3	4	P3	1	2	3	4	P4	1	2	3	4
$T_j$	1	5				1				4	1				5	1	5	5		
	2		10	10		2	5				2		3	3		2	7			
	3				4	3		6	6		3	7				3				3

A experimentação realizada por Almeida (2002) com base neste caso de teste, é dividida em várias experimentações relacionadas com diferentes encomendas. A primeira é relacionada com encomendas unitárias de todos os produtos (caso AA\_FJS3x4\_01). As outras são relacionadas com encomendas de 10 unidades para todos os produtos. Estas são analisadas nas três seguintes perspectivas: produção em lotes de 10 unidades (AA\_FJS3x4\_10), produção em 10 lotes de 1 unidade (AA\_FJS3x4\_10\_1) e produção em 2 lotes de 5 unidades (AA\_FJS3x4\_10\_5). Nos testes de Almeida (2002) não se aplica a partição de lotes mas explora-se a sobreposição com lotes de transferência de uma e de 5 unidades.

O escalonamento sucessivo do fabrico de vários produtos implica o escalonamento de encomendas sucessivas implica, no método DISMAN, a definição da sequência pela qual se trata o problema do escalonamento de cada encomenda. Como se referiu atrás segue-se a ordem de lançamento das encomendas utilizada no trabalho de Almeida (2002). Nas encomendas unitárias aplica-se a sequência de processamento das encomendas de P2, P4, P3 e P1, representada por P2-4-3-1, e nas encomendas de 10 unidades a sequência P3-2-1-4.

### **2.4.1. Caso AA\_FJS3x4\_01**

Definiu-se a sequência de produção P2-4-3-1, e correram-se optimizações sucessivas da atribuição de recursos, actualizando a agenda dos recursos entre cada optimização. Os resultados da aplicação deste método podem ser visualizados no gráfico de Gantt da Figura 55 onde não existe processamento paralelo de nenhum produto por se considerarem lotes de uma unidade.

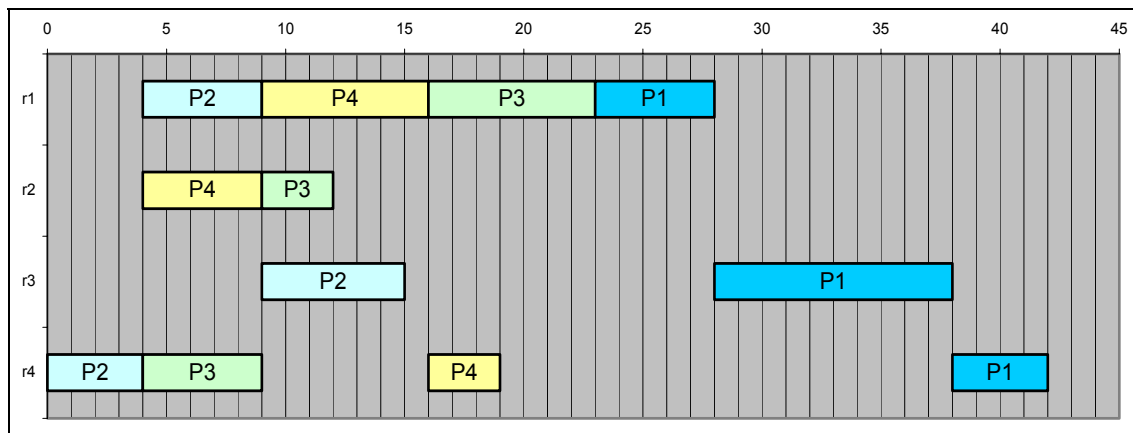


Figura 55: AA\_FJS3x4\_01 – Gráfico de Gantt

Os resultados deste método, DISMAN, são similares (Tabela 23) aos do método MEOP (Almeida (2002)) para todos os produtos, diferindo apenas o início da produção para o produto P3. Como foi referido atrás o método MEOP explora a agregação da produção de cada produto, com base nos padrões de escalonamento, não permitindo a separação dos períodos de produção. Assim sendo, as operações de cada lote nos diferentes recursos são encadeadas umas nas outras não permitindo a separação temporal do processamento do mesmo lote entre recursos. Tal não é imposto pelo método DISMAN, que assim alarga as opções de solução, explorando situações não consideradas pelo método MEOP e assim podendo, eventualmente, encontrar melhores soluções.

Tabela 23: AA\_FJS3x4\_01 - Intervalos de tempo de selecção dos recursos.

Produto	Método	r1		r2		r3		r4		R K	Total	
		Start	Finish	Start	Finish	Start	Finish	Start	Finish		Start	Finish
P1	EDD	27	32	-	-	32	42	42	46	7	27	46
	FCFS	9	14	14	24	-	-	24	28	1	9	28
	SB/Cmax	9	14	14	24	-	-	24	28	1	9	28
	SB/Σ(WT)	9	14	14	24	-	-	24	28	1	9	28
	SPT	24	29	-	-	29	39	39	43	6	24	43
	MEOP	23	28	28	38	-	-	38	42	4	23	42
	Disman	23	28	-	-	28	38	38	42	4	23	42
P2	EDD	15	20	20	26	-	-	5	9	7	5	26
	FCFS	4	9	-	-	9	15	0	4	1	0	15
	SB/Cmax	4	9	-	-	12	18	0	4	6	0	18
	SB/Σ(WT)	4	9	-	-	9	15	0	4	1	0	15
	SPT	4	9	9	15	-	-	0	4	1	0	15
	MEOP	4	9	9	15	-	-	0	4	1	0	15
	Disman	4	9	-	-	9	15	0	4	1	0	15
P3	EDD	8	15	5	8	-	-	0	5	1	0	15
	FCFS	21	28	10	13	-	-	4	9	6	4	28
	SB/Cmax	21	28	-	-	9	12	4	9	6	4	28
	SB/Σ(WT)	14	21	9	12	-	-	4	9	2	4	21
	SPT	17	24	-	-	10	13	4	9	5	4	24
	MEOP	16	23	-	-	13	16	8	13	3	8	23
	Disman	16	23	9	12	-	-	4	9	3	4	23
P4	EDD	20	27	-	-	5	10	27	30	5	5	30
	FCFS	14	21	5	10	-	-	28	31	6	5	31
	SB/Cmax	14	21	5	10	-	-	21	24	4	5	24
	SB/Σ(WT)	21	28	-	-	15	20	28	31	6	15	31
	SPT	10	17	-	-	5	10	17	20	3	5	20
	MEOP	9	16	4	9	-	-	16	19	1	4	19
	Disman	9	16	4	9	-	-	16	19	1	4	19

Na coluna RK (“rank”) da Tabela 23 ordenam-se os métodos pelo instante de conclusão do processamento de cada produto. O melhor resultado corresponde ao 1 e o pior ao 7. É assim, possível verificar que, neste exemplo, o método DISMAN em conjunto com o método MEOP apresentam de uma forma geral bons resultados. Se determinarmos a média dos RK podemos verificar que estes dois métodos obtêm o valor mais baixo de todos os métodos, igual a 2,3. Esta média dá uma indicação da forma como o método se comporta em relação aos outros, do ponto de vista dos resultados dos instantes de conclusão de cada encomenda.

Aplicando as medidas de desempenho referidas atrás, é possível comparar os resultados de todos os métodos. Como se pode ver na Tabela 24, o método SB/Cmax apresenta um instante inferior de conclusão de processamento de todos os produtos e uma maior utilização de recursos, e SB/Σ(WT) apresenta melhor resultado relativamente ao somatório de tempos de conclusão e QMS. Deve realçar-se que os resultados obtidos pelo método DISMAN são muito dependentes da sequência de produção seleccionada e ainda que, neste teste não foi explorado o paralelismo de processamento dos lotes nas tarefas de transformação, por as encomendas serem unitárias. Tal exploração pode melhorar substancialmente o desempenho produtivo.

Tabela 24: AA\_FJS3x4\_01 - Resultados agregados

Método	EDD	FCFS	SB/Cmax	SB/Σ(WT)	SPT	MEOP	Disman
Cmax	46	31	<b>28</b>	31	43	42	42
ΣC	117	102	98	<b>95</b>	102	99	99
Utilização	46,4%	68,8%	<b>76,2%</b>	68,8%	49,6%	50,8%	50,8%
QMS	0,36	0,45	0,47	<b>0,48</b>	0,45	0,46	0,46

### 2.4.2. Caso AA\_FJS3x4\_10

Neste caso consideram-se quatro encomendas de 10 unidades de cada um dos produtos, executando-se sucessivas optimizações para afectação de recursos para cada encomenda, com actualização da agenda dos recursos entre cada uma. O método DISMAN é aplicado, considerando lotes de 10 unidades (DISMAN 10\_0), lotes de 1 unidade (DISMAN 10\_1) e lotes de 5 unidades (DISMAN 10\_5). Em todos estes casos aplica-se a

sequência P3-2-1-4. Os resultados obtidos pelo métodos DISMAN, pelo método MEOP com lotes de transferência de 1 unidade (MEOP 10\_1) e sequência P3-2-1-4, e por todos os outros métodos referidos são apresentados na Tabela 25. Em geral, com a aplicação do método DISMAN a este caso de teste, obtêm-se bons resultados dos instantes de conclusão de cada encomenda. Este facto pode verificar-se pela obtenção, na maioria das encomendas, de boas classificações RK para o método DISMAN. Na realidade, a média destes índices no método DISMAN 10\_1 é igual a 2,8 e corresponde ao menor valor médio dos RK, indicando boas soluções, do ponto de vista do instante de conclusão, para a maioria das encomendas.

Pela aplicação do método DISMAN 10\_1, o de máxima partição do lote, é possível verificar que se obtém a solução de menor tempo de conclusão de todas encomendas  $C_{max}$ , i.e. do grupo de encomendas. O  $C_{max}$  de um método corresponde ao máximo dos tempos de conclusão de cada encomenda, indicados na coluna “finish”, resultantes da aplicação desse método. Neste caso, o tempo de conclusão de todas encomendas  $C_{max}$  resultante da aplicação do método DISMAN 10\_1 corresponde a 254 UT. Na Tabela 25 de apresentação dos resultados da experimentação aqui descrita é possível confirmar o instante de conclusão na linha relativa ao produto P4, e ao método DISMAN 10\_1.

Tabela 25: AA\_FJS3x4\_10\_1/5 – Intervalos de tempo de selecção dos recursos.

Produto	Método	r1		r2		r3		r4		R	Total		
		Start	Finish	Start	Finish	Start	Finish	Start	Finish		K	Start	Finish
P1	EDD	5	55			55	155	155	195	3	5	195	
	FCFS	5	55			55	155	155	195	3	5	195	
	SB/Cmax	5	55	55	155			205	245	6	5	245	
	SB/Σ(WT)	5	55	55	155			205	245	6	5	245	
	SPT	230	280	280	380			380	420	10	230	420	
	P3-2-1-4 MEOP 10_1	146	196			151	251	165	205	8	146	251	
	P2-4-3-1 MEOP 10_5	210	260	215	315			290	330	9	210	330	
P3-2-1-4 DISMAN 10_0	5	55	65	115	65	115	115	155	1	5	155		
P3-2-1-4 DISMAN 10_1	128	178	138	188	133	183	143	192	2	128	192		
P3-2-1-4 DISMAN 10_5	9	179	60	199	70	209	90	229	5	9	229		
P2	EDD	150	200	200	260			50	90	10	50	260	
	FCFS	55	105	120	180			0	40	6	0	180	
	SB/Cmax	55	105			120	180	0	40	6	0	180	
	SB/Σ(WT)	55	105			120	180	0	40	6	0	180	
	SPT	40	90			90	150	0	40	3	0	150	
	P3-2-1-4 MEOP 10_1	96	146	101	161			56	96	4	56	161	
	P2-4-3-1 MEOP 10_5	20	70	45	105			0	40	1	0	105	
P3-2-1-4 DISMAN 10_0	135	185	185	215	185	215	50	90	9	50	215		
P3-2-1-4 DISMAN 10_1	78	128	88	134	83	129	50	90	2	50	134		
P3-2-1-4 DISMAN 10_5	104	154	129	166	129	172	50	90	5	50	172		
P3	EDD	80	150	50	80			0	50	5	0	150	
	FCFS	175	245	90	120			40	90	8	40	245	
	SB/Cmax	175	245			90	120	40	90	8	40	245	
	SB/Σ(WT)	175	245			90	120	40	90	8	40	245	
	SPT	160	230	90	120			40	90	7	40	230	
	P3-2-1-4 MEOP 10_1	26	96	23	53			0	50	2	0	96	
	P2-4-3-1 MEOP 10_5	140	210	105	135			70	120	6	70	210	
P3-2-1-4 DISMAN 10_0	65	135	50	65	50	65	0	50	4	0	135		
P3-2-1-4 DISMAN 10_1	8	78	40	43	5	53	0	50	1	0	78		
P3-2-1-4 DISMAN 10_5	34	104	25	56	28	59	0	50	3	0	104		
P4	EDD	200	270			5	55	270	300	9	5	300	
	FCFS	105	175	5	55			195	225	5	5	225	
	SB/Cmax	105	175	5	55			175	205	3	5	205	
	SB/Σ(WT)	105	175	5	55			175	205	3	5	205	
	SPT	90	160	5	55			160	190	2	5	190	
	P3-2-1-4 MEOP 10_1	196	266			251	301	274	304	10	196	304	
	P2-4-3-1 MEOP 10_5	70	140			95	145	130	160	1	70	160	
P3-2-1-4 DISMAN 10_0	185	255	5	50	45	50	255	285	8	5	285		
P3-2-1-4 DISMAN 10_1	178	251	5	237	68	188	192	254	6	5	254		
P3-2-1-4 DISMAN 10_5	179	249	5	115	23	28	229	264	7	5	264		

Em seguida apresentam-se os gráficos de Gantt resultantes da aplicação das sucessivas optimizações. Nos gráficos de Gantt apresentados nestas secções utiliza-se a seguinte nomenclatura para indicação dos lotes: Pij, em que “i” designa o produto e “j” a ordem de escalonamento do lote. No gráfico da Figura 56 é possível verificar a exploração do processamento paralelo ou simultâneo em recursos funcionalmente equivalentes, de forma a minimizar o tempo de entrega de cada produto. Tal simultaneidade de processamento verifica-se nesta instância de escalonamento para as ordens de fabrico de P1, P2 e P3 nos



recursos  $r_2$  e  $r_3$ . A comparação do gráfico da Figura 56 com o da Figura 55 realça este aspecto, na medida em que neste não é possível explorar esse paralelismo por estar a ser processada uma encomenda de apenas uma unidade.

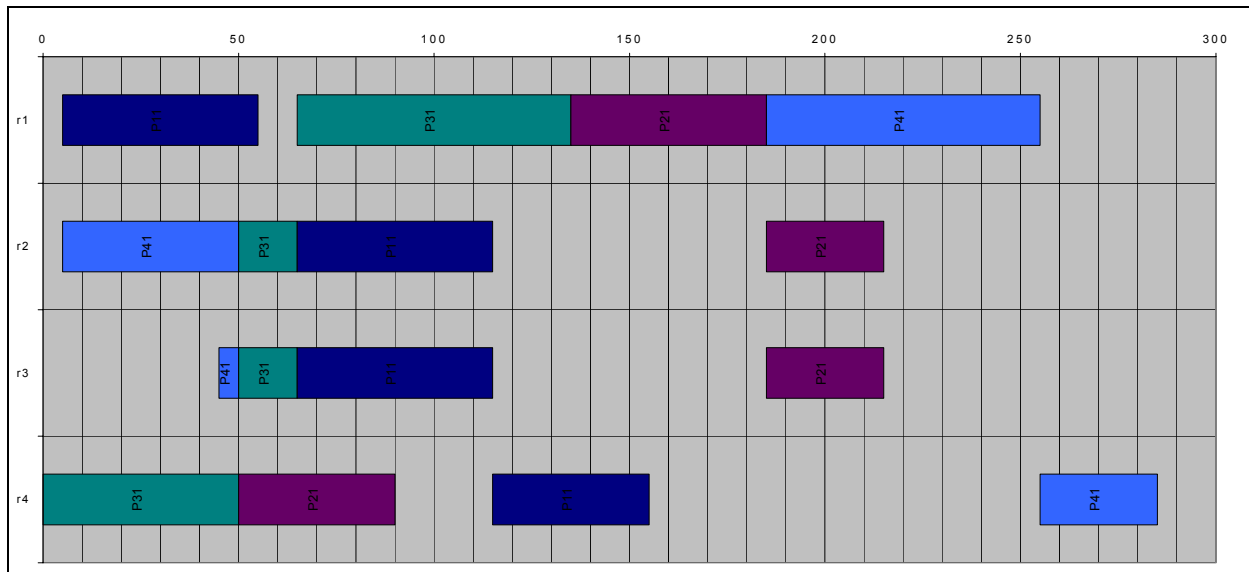


Figura 56: AA\_FJS3x4\_10 - Gráfico de Gantt - P3-2-1-4

No gráfico da Figura 56 verifica-se uma situação especial, em que o processo de escalonamento de um produto (P1) é posterior ao de outro (P2) e, no entanto, é escalonado com instantes de início e conclusão anteriores. Este facto deve-se, neste caso, à sequência de processamento seleccionada, à exploração de processamento paralelo e ao facto de se admitir que o processamento da encomenda de um produto não tem que resultar do encadeamento contínuo do processamento dos diversos lotes, em que é dividido por partição da encomenda. Note-se que no caso do método MEOP, Almeida (2002), tal encadeamento é imposto pelo padrão de escalonamento.

No gráfico de Gantt da Figura 57 visualiza-se o resultado da programação da produção considerando 2 lotes de 5 unidades por produto, com exploração do processamento paralelo em 2 recursos,  $r_2$  e  $r_3$ .

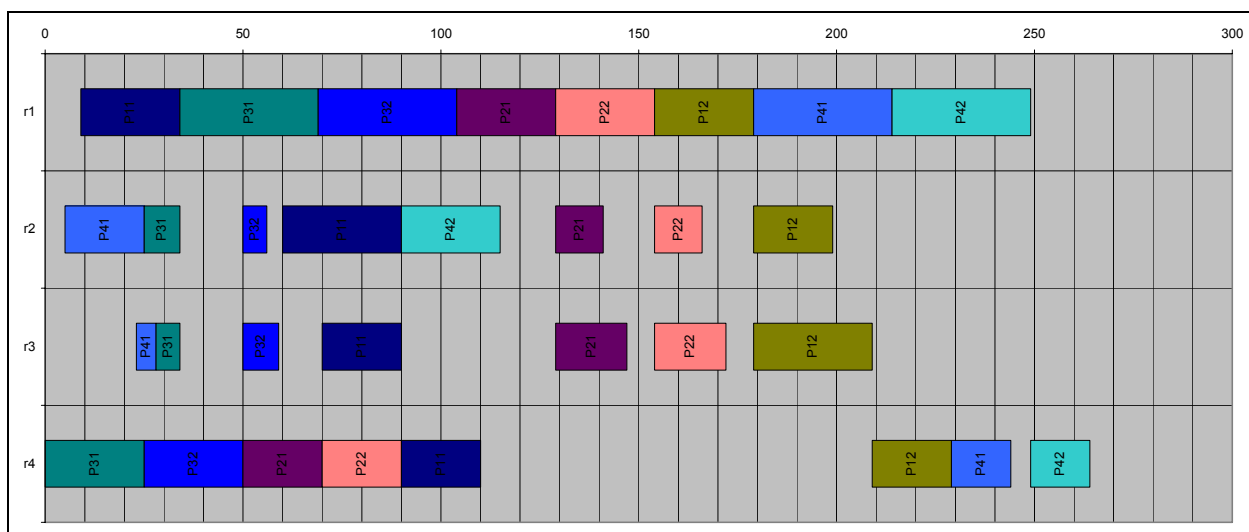


Figura 57: AA\_FJS3x4\_10\_5 - Gráfico de Gantt - P3-2-1-4

O gráfico de Gantt da Figura 58 permite visualizar o resultado da aplicação de sucessivas otimizações com partição das encomendas em lotes de uma unidade. O processamento paralelo advém do facto de se processar cada lote de uma unidade como uma encomenda independente, logo, diferentes encomendas do mesmo produto poderão ser processados simultaneamente em diferentes recursos diferentes, paralelos ou não. Este é o caso em que se verifica a maior fragmentação da encomenda de um produto e dos períodos de utilização dos recursos.

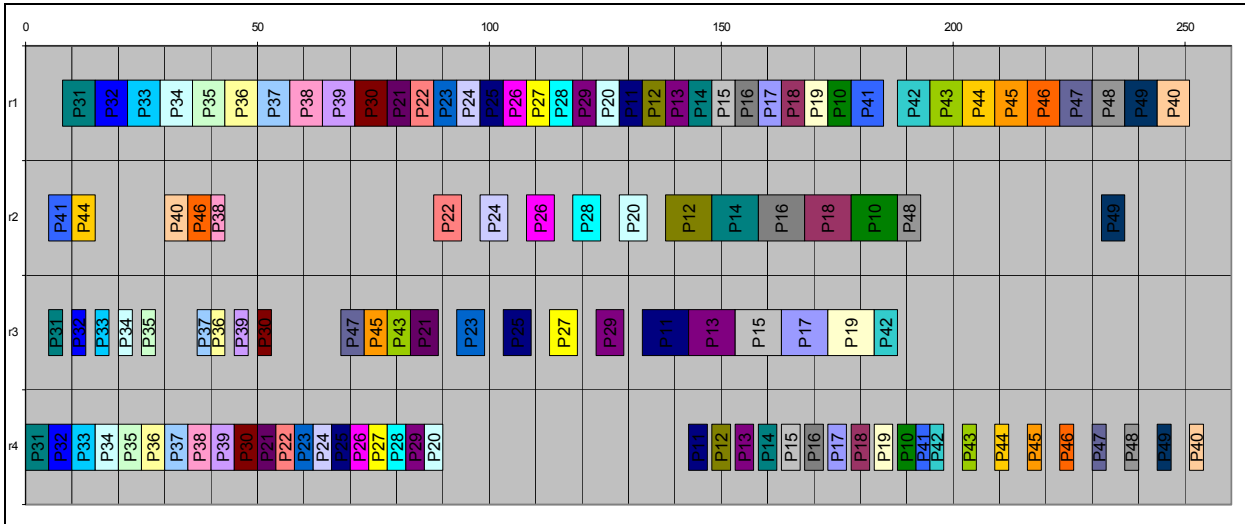


Figura 58: AA\_FJS3x4\_10\_1 - Gráfico de Gantt - P3-2-1-4

Na Tabela 26 apresentam-se resultados agregados para todos os métodos. O instante de conclusão de todos os trabalhos  $C_{max}$ , isto é, de todas as encomendas, no método DISMAN 10\_1 é igual a 254 e superior relativamente a três dos outros métodos. No entanto tem um resultado muito próximo do mínimo obtido de 245. Quanto ao critério de somatório dos instantes de conclusão  $\sum C$ , o método DISMAN 10\_1 obtém o melhor resultado. Mas o mesmo método, com qualquer partição obtém melhores resultados de  $\sum C$  do que todos os outros métodos. O facto do somatório ser inferior indica que, em média, os trabalhos permanecem menos em curso de fabrico e que no conjunto a conclusão dos trabalhos é mais rápida, libertando mais cedo os recursos e possibilitando a entrega mais rápida das encomendas. Na utilização dos recursos o método DISMAN 10\_1 é o quarto melhor com um valor muito próximo dos melhores. Finalmente, relativamente ao critério de Qualidade de Serviço de Produção, o método DISMAN 10\_1 é substancialmente melhor do que todos os outros, sendo mesmo melhor nos casos sem partição e com partição de 5. O método DISMAN permite obter resultados em geral bons comparativamente com os outros métodos testados, favorecendo a rápida execução das encomendas e por conseguinte, a entrega mais rápida ao cliente.

Tabela 26: AA\_FJS3x4 – Resultados agregados

Método	EDD	FCFS	SB/Cmax	SB/ $\sum(WT)$	SPT	MEOP 10_1	MEOP 10_5	DISMAN 10_0	DISMAN 10_1	DISMAN 10_5
$C_{max}$	300	<b>245</b>	<b>245</b>	<b>245</b>	420	304	330	285	254	264
$\sum C$	905	845	875	875	990	812	805	<b>790</b>	<b>658</b>	<b>769</b>
Utilização	53,3%	<b>65,3%</b>	<b>65,3%</b>	<b>65,3%</b>	38,1%	52,6%	48,5%	56,1%	63,0%	60,6%
QMS	0,46	0,50	0,48	0,48	0,41	0,52	0,52	<b>0,53</b>	<b>0,61</b>	<b>0,54</b>

A análise dos resultados obtidos com o método DISMAN e várias partições de lote, permite verificar uma melhoria das medidas de desempenho com o aumento da partição. Como foi visto na secção V - 1, estes ganhos tendem a ser decrescentes com o aumento da simultaneidade de processamento. Com o objectivo de analisar essa evolução, realizou-se um processo de experimentação adicional de aplicação do método DISMAN com partição da encomenda em 5 lotes de duas unidades. Os resultados agregados, resultantes da aplicação deste método DISMAN 10\_2, são apresentados na Tabela 27. Estes resultados confirmam, em geral, um decréscimo de ganhos com o aumento da partição, i.e. com o aumento do processamento simultâneo. Este facto sugere que a utilização de partições de valor relativamente moderado permite obter resultados próximos dos resultados de máxima partição.

Tabela 27: AA\_FJS3x4 – Resultados agregados do método DISMAN.

Método	DISMAN 10_0	DISMAN 10_5	DISMAN 10_2	DISMAN 10_1
Cmax	285	264	259	254
$\Sigma C$	790	769	682	658
Utilização	56,1%	60,6%	61,8%	63,0%
QMS	0,53	0,54	0,59	0,61

## 2.5 Caso AA\_FJS4x6

Este caso de estudo, tal como o anterior, é baseado num sistema produtivo do tipo “Flexible Job Shop” (FJS) referido por Pinedo and Chao (1999), em que existe pelo menos um centro de trabalho da “Job Shop” provido com 2 ou mais processadores paralelos. Este sistema de produção permite processar 6 produtos (P1 ... P6) e é composto por 4 transformações sucessivas ( $1 \leq j \leq n=4$ ), que recorrem a 5 ( $1 \leq k \leq r=5$ ) recursos disponíveis. Todos os produtos têm uma estrutura similar do ponto de vista do processamento que está representada na Figura 59. Nesta estrutura os objectos (componentes) são sucessivamente processados nas 4 transformações até se obter o produto final. Em todas as transformações uma unidade de quantidade de trabalho consome uma unidade de um objecto e produz uma unidade de outro objecto ( $q_l, 1 \leq l \leq 8$ ).

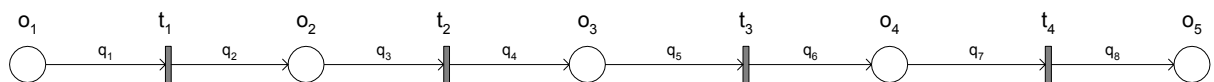


Figura 59: Representação idêntica para todos os produtos do caso AA\_FJS4x6

O início da produção de todos os produtos está restringido pelos seguintes instantes de lançamento:

$$P_1, P_3, P_5 : T_1^- \geq 0 ; \quad P_2 : T_1^- \geq 5 ; \quad P_4 : T_1^- \geq 2 ; \quad P_6 : T_1^- \geq 6 .$$

O tempo de processamento unitário de uma transformação por um recurso é apresentado na Tabela 28, em que os espaços vazios significam a não possibilidade de utilização desse recurso para processamento dessa transformação. Neste caso apenas os processadores 1 e 2 são funcionalmente equivalentes.

Tabela 28: Tempo  $c_{jk}$  de processamento unitário de cada transformação por cada recurso  $r_k$ .

$C_{jk}$	P1	1	2	3	4	5	$C_{jk}$	P2	1	2	3	4	5	$C_{jk}$	P3	1	2	3	4	5
$T_j$	1	4	4				1	1				2		1	1					3
	2			1			2	2	3	3				2	2	4	4			
	3				3		3	3			1			3	3				1	
	4					2	4	4				2		4	4			2		

$C_{jk}$	P4	1	2	3	4	5	$C_{jk}$	P5	1	2	3	4	5	$C_{jk}$	P6	1	2	3	4	5
$T_j$	1	3	3				1	1			3			1	2	2				
	2					2	2	2	2			2		2	2					
	3				2		3	3	4	4				3	3			1		
	4			3			4	4				2		4	4				4	

### 2.5.1. Caso AA\_FJS4x6\_1

Nesta secção analisam-se os resultados da programação da produção obtidos com o método DISMAN para lotes de uma unidade, com a sequência de lançamento de encomendas P3-6-2-4-5-1, podendo visualizar-se o resultado no gráfico de Gantt da Figura 60. Na Tabela 29 apresentam-se os resultados obtidos pela aplicação do método DISMAN a este caso, e por diversos outros métodos conhecidos. Os resultados do método MEOP foram retirados de Almeida (2002).

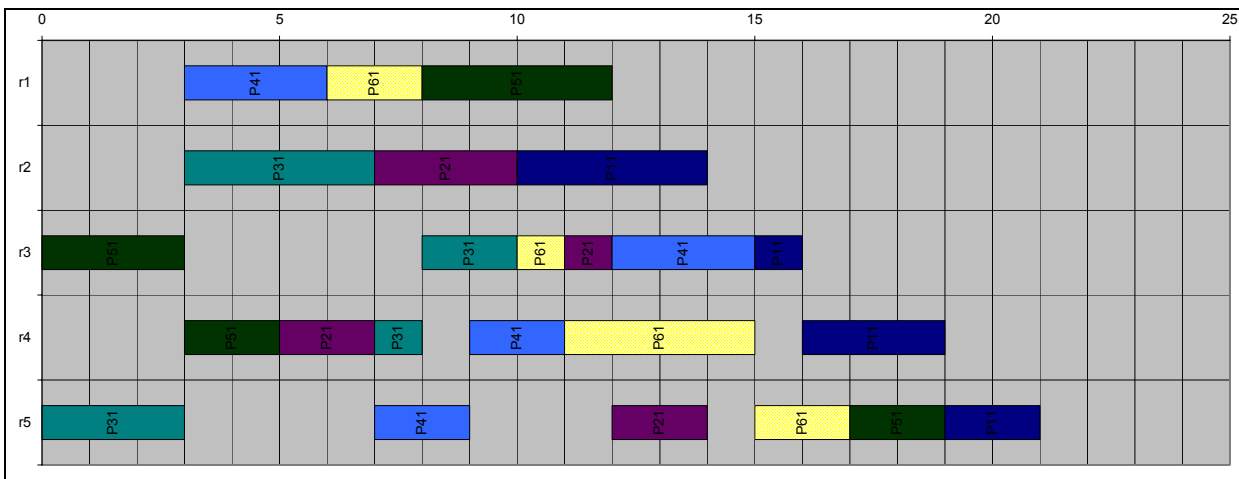


Figura 60: AA\_FJS4x6\_1 - Gráfico de Gantt - P3-6-2-4-5-1

Os instantes de conclusão do processamento de cada encomenda obtidos com o método DISMAN são, relativamente ao método MEOP, na maioria dos casos melhores ou iguais. Estes resultados podem ser confirmados pelo índice RK (“rank”) da Tabela 29, correspondente à ordenação dos métodos pelo instante de conclusão do processamento de cada produto. O melhor resultado corresponde ao 1 e o pior ao 7. É assim, possível verificar que o método DISMAN é o método que apresenta de uma forma consistente melhores resultados, obtendo a menor média dos RK, igual a 2.5.

Tabela 29: AA\_FJS4x6 - Intervalos de tempo de selecção dos recursos.

Produto	Método	r1		r2		r3		r4		r5		R K	Total	
		Start	Finish	Start	Finish	Start	Finish	Start	Finish	Start	Finish		Start	Finish
P1	EDD	0	4			4	5	5	8	8	10	1	0	10
	FCFS	0	4			4	5	7	10	11	13	5	0	13
	SB/Cmax			0	4	4	5	5	8	8	10	1	0	10
	SB/Σ(WT)	0	4			4	5	5	8	8	10	1	0	10
	SPT	0	4			4	5	5	8	8	10	1	0	10
	MEOP			16	20	20	21	21	24	24	26	7	16	26
	Disman			10	14	15	16	16	19	19	21	6	10	21
P2	EDD	10	13			13	14	8	10	14	16	3	8	16
	FCFS			9	12	12	13	5	7	13	15	2	5	15
	SB/Cmax			10	13	13	14	8	10	14	16	3	8	16
	SB/Σ(WT)	10	13			20	21	8	10	21	23	7	8	23
	SPT			11	14	14	15	9	11	15	17	5	9	17
	MEOP	11	14			14	15	9	11	15	17	5	9	17
	Disman			7	10	11	12	5	7	12	14	1	5	14
P3	EDD			3	7	11	13	10	11	0	3	4	0	13
	FCFS	4	8			16	18	12	13	0	3	5	0	18
	SB/Cmax	13	17			18	20	17	18	10	13	6	10	20
	SB/Σ(WT)			13	17	18	20	17	18	10	13	6	10	20
	SPT			3	7	9	11	8	9	0	3	3	0	11
	MEOP	3	7			8	10	7	8	0	3	1	0	10
	Disman			3	7	8	10	7	8	0	3	1	0	10
P4	EDD	4	7			14	17	12	14	10	12	3	4	17
	FCFS			2	5	13	16	10	12	5	7	2	2	16
	SB/Cmax			4	7	20	23	18	20	16	18	5	4	23
	SB/Σ(WT)	13	16			21	24	18	20	16	18	6	13	24
	SPT	9	12			18	21	16	18	12	14	4	9	21
	MEOP	14	17			21	24	19	21	17	19	6	14	24
	Disman	3	6			12	15	9	11	7	9	1	3	15
P5	EDD			7	11	0	3	3	5	12	14	4	0	14
	FCFS			5	9	0	3	3	5	9	11	1	0	11
	SB/Cmax	17	21			0	3	3	5	21	23	6	0	23
	SB/Σ(WT)			31	35	25	28	29	31	35	37	7	25	37
	SPT	5	9			0	3	3	5	10	12	3	0	12
	MEOP			5	9	0	3	3	5	9	11	1	0	11
	Disman	8	12			0	3	3	5	17	19	5	0	19
P6	EDD	7	9			9	10	14	18	18	20	5	7	20
	FCFS	8	10			10	11	13	17	17	19	3	8	19
	SB/Cmax			7	9	9	10	20	24	24	26	6	7	26
	SB/Σ(WT)	6	8			24	25	25	29	29	31	7	6	31
	SPT			7	9	11	12	12	16	17	19	3	7	19
	MEOP	8	10			10	11	15	17	11	15	1	8	17
	Disman	6	8			10	11	11	15	15	17	1	6	17

Na Tabela 30 apresentam-se os resultados de alguns parâmetros globais de análise para todos os métodos. O processamento de uma única unidade encomendada não permite explorar a possibilidade de processamento simultâneo, por parte do método DISMAN. Dessa forma, este método ainda fica mais dependente da sequência de lançamento de encomendas, face a outros métodos. Nesta sequência de lançamento, o método DISMAN, para uma encomenda de uma unidade obtém o terceiro melhor resultado em todas as medidas. O instante de conclusão do processamento de todos os produtos Cmax e a utilização são melhores pela aplicação da regra FCFS e o somatório dos instantes de conclusão é melhor pelas regras EDD e SPT. A qualidade de serviço é melhor como resultado da aplicação das regras FCFS, EDD e SPT. Deve notar-se, no entanto, que em todas as medidas, o método DISMAN obtém resultados próximos dos melhores.

Tabela 30: AA\_FJS4x6 - Resultados agregados

Método	EDD	FCFS	SB/Cmax	SB/Σ(WT)	SPT	MEOP	DISMAN
Cmax	20	19	26	37	21	26	21
ΣC	90	92	118	145	90	105	96
Utilização	58,0%	61,1%	44,6%	31,4%	55,2%	44,6%	55,2%
QMS	0,59	0,59	0,47	0,35	0,59	0,53	0,57

### 2.5.2. Caso AA\_FJS4x6\_10

Neste caso consideram-se 6 encomendas de 10 unidades de cada um dos 6 produtos, executando-se sucessivas optimizações para afectação de recursos para cada encomenda, com actualização da agenda dos recursos entre cada uma. O método DISMAN 10\_0 considera 1 lote de 10 unidades para cada produto, o DISMAN 10\_1 considera 10 lotes de 1 unidade e o DISMAN 10\_5 considera 2 lotes de 5 unidades. Em

todos estes casos aplica-se a sequência de lançamento de encomendas P3-6-2-4-5-1, por ser a sequência utilizada por Almeida (2002). Todos os resultados obtidos pela aplicação do método DISMAN e dos outros métodos conhecidos são apresentados na Tabela 31. Nesta tabela também são apresentados os resultados do método MEOP com lotes de transferência de 1 unidade, MEOP 10\_1, e de 5 unidades, MEOP 10\_5, retirados de Almeida (2002).

A aplicação do método DISMAN a este caso de teste, permite obter bons resultados dos instantes de conclusão de cada encomenda. Na realidade, a média do índice de ordenação de cada encomenda RK, dos métodos em função do instante de conclusão, no método DISMAN 10\_1 é igual a 2,0 e corresponde ao menor valor médio dos RK. Esta média para o método DISMAN 10\_5 é igual a 4,2 e corresponde ao segundo menor valor médio dos RK. Estes valores médios permitem verificar que o método DISMAN se concentra na redução do instante de conclusão de cada encomenda.

Tabela 31: AA\_FJS4x6\_10\_1/\_5 – Intervalos de tempo de selecção dos recursos.

Produto	Método	r1		r2		r3		r4		r5		R	Total	
		Start	Finish	Start	Finish	Start	Finish	Start	Finish	Start	Finish		Start	Finish
P1	EDD	0	40			40	50	50	80	80	100	1	0	100
	FCFS	0	40			40	50	50	80	80	100	1	0	100
	SB/Cmax	0	40			40	50	50	80	80	100	1	0	100
	SB/Σ(WT)	0	40			40	50	72	102	102	122	4	0	122
	SPT			0	40	40	50	90	120	120	140	5	0	140
	MEOP 10_1			178	218	219	229	220	250	232	252	9	178	252
	MEOP 10_5			200	240	219	229	220	250	232	252	9	200	252
	DISMAN 10_0	88	100	2	30	100	110	140	170	180	200	8	2	200
	DISMAN 10_1	61	143	72	147	65	148	126	157	132	159	6	61	159
	DISMAN 10_5	86	122	5	9	112	127	127	157	142	167	7	5	167
P2	EDD			32	62	62	72	5	25	100	120	5	5	120
	FCFS	40	70			70	80	5	25	100	120	5	5	120
	SB/Cmax	40	70			70	80	5	25	100	120	5	5	120
	SB/Σ(WT)			32	62	62	72	5	25	72	92	3	5	92
	SPT	26	56			56	66	5	25	66	86	2	5	86
	MEOP 10_1	100	130			121	131	98	118	122	142	9	98	142
	MEOP 10_5	15	45			121	131	98	118	122	142	9	15	142
	DISMAN 10_0	-	-	80	110	110	120	30	50	120	140	8	30	140
	DISMAN 10_1	34	55	46	55	40	77	5	34	41	79	1	5	79
	DISMAN 10_5	27	30	45	72	67	77	5	27	77	107	4	5	107
P3	EDD	40	80			90	110	80	90	0	30	6	0	110
	FCFS			52	92	150	170	140	150	0	30	9	0	170
	SB/Cmax			56	96	150	170	140	150	0	30	9	0	170
	SB/Σ(WT)	40	80			112	132	102	112	0	30	8	0	132
	SPT			40	80	96	116	80	90	0	30	7	0	116
	MEOP 10_1	3	43			35	55	34	44	0	30	2	0	55
	MEOP 10_5	3	43			35	55	34	44	0	30	2	0	55
	DISMAN 10_0	30	50	30	50	60	80	50	60	0	30	5	0	80
	DISMAN 10_1	6	34	3	31	8	37	7	35	0	30	1	0	37
	DISMAN 10_5	15	42	19	42	32	57	27	47	0	30	4	0	57
P4	EDD			2	32	110	140	90	110	32	52	4	2	140
	FCFS			2	32	100	130	80	100	32	52	3	2	130
	SB/Cmax			26	56	100	130	80	100	56	76	3	26	130
	SB/Σ(WT)			2	32	72	102	52	72	32	52	1	2	102
	SPT	56	86			180	210	160	180	140	160	10	56	210
	MEOP 10_1	139	169			146	176	144	164	142	162	8	139	176
	MEOP 10_5	50	80			146	176	144	164	142	162	8	50	176
	DISMAN 10_0	-	-	50	80	120	150	100	120	80	100	7	50	150
	DISMAN 10_1	3	103	58	82	77	110	75	107	30	105	2	3	110
	DISMAN 10_5	4	57	95	107	97	142	87	127	57	117	6	4	142
P5	EDD	80	120			0	30	30	50	120	140	3	0	140
	FCFS	70	110			0	30	30	50	120	140	3	0	140
	SB/Cmax	70	110			0	30	30	50	120	140	3	0	140
	SB/Σ(WT)			62	102	0	30	30	50	122	142	6	0	142
	SPT	140	180			66	96	120	140	180	200	8	66	200
	MEOP 10_1	190	230			176	206	188	208	212	232	9	176	232
	MEOP 10_5	145	185			176	206	188	208	212	232	9	145	232
	DISMAN 10_0	140	160	140	160	0	30	120	140	160	180	7	0	180
	DISMAN 10_1	111	130	87	127	0	124	77	126	95	132	1	0	132
	DISMAN 10_5	-	-	72	127	12	97	32	107	117	137	2	12	137
P6	EDD			62	82	140	150	150	190	190	210	10	62	210
	FCFS			32	52	52	62	100	140	140	160	6	32	160
	SB/Cmax			6	26	30	40	100	140	140	160	6	6	160
	SB/Σ(WT)	80	100			102	112	112	152	152	172	8	80	172
	SPT	6	26			26	36	36	76	86	106	4	6	106
	MEOP 10_1	44	64			55	65	78	98	56	96	3	44	98
	MEOP 10_5			175	195	55	65	78	98	56	96	9	55	195
	DISMAN 10_0	10	30	-	-	30	40	60	100	100	120	5	10	120
	DISMAN 10_1	10	30	7	66	10	71	35	75	39	77	1	7	77
	DISMAN 10_5	7	15	9	34	27	47	47	87	67	97	2	7	97

Pela aplicação do método DISMAN 10\_1, o de máxima partição do lote, é possível verificar que se obtém a solução de menor tempo de conclusão de todas encomendas  $C_{max}$ , i.e. do grupo de encomendas. O  $C_{max}$  de um método corresponde ao máximo dos tempos de conclusão de cada encomenda, indicados na coluna “finish”, resultantes da aplicação desse método. Neste caso, o tempo de conclusão de todas encomendas  $C_{max}$  resultante da aplicação do método DISMAN 10\_1 corresponde a 159 UT. Na Tabela 31 de apresentação dos resultados da experimentação aqui descrita é possível confirmar o instante de conclusão na linha relativa ao produto P1, e ao método DISMAN 10\_1.

No gráfico de Gantt da Figura 61 é possível visualizar o resultado da programação da produção para o método DISMAN 10\_0, seguindo a sequência de lançamento de encomendas P3-6-2-4-5-1, em que se obtém o instante de conclusão de todas as encomendas igual a 200 UT.

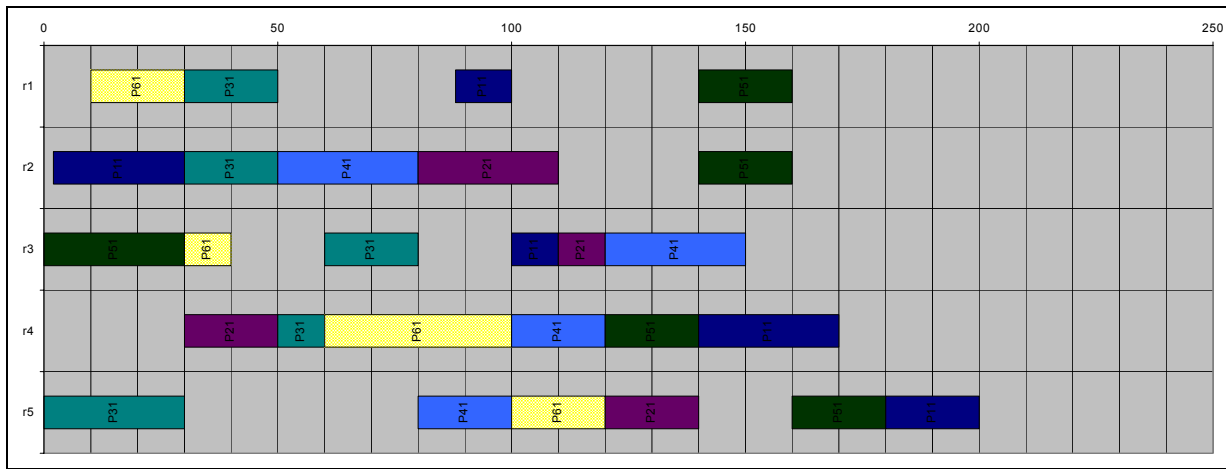


Figura 61: AA\_FJS4x6\_10 - Gráfico de Gantt - P3-6-2-4-5-1

No gráfico de Gantt da Figura 62 apresenta-se o resultado da programação da produção para lotes de 1 unidade, com extrema fragmentação das encomendas e do tempo de ocupação dos recursos. O instante de conclusão da produção de todos os produtos é de 159 UT, obtendo-se uma redução em relação ao caso anterior de 20.5%.

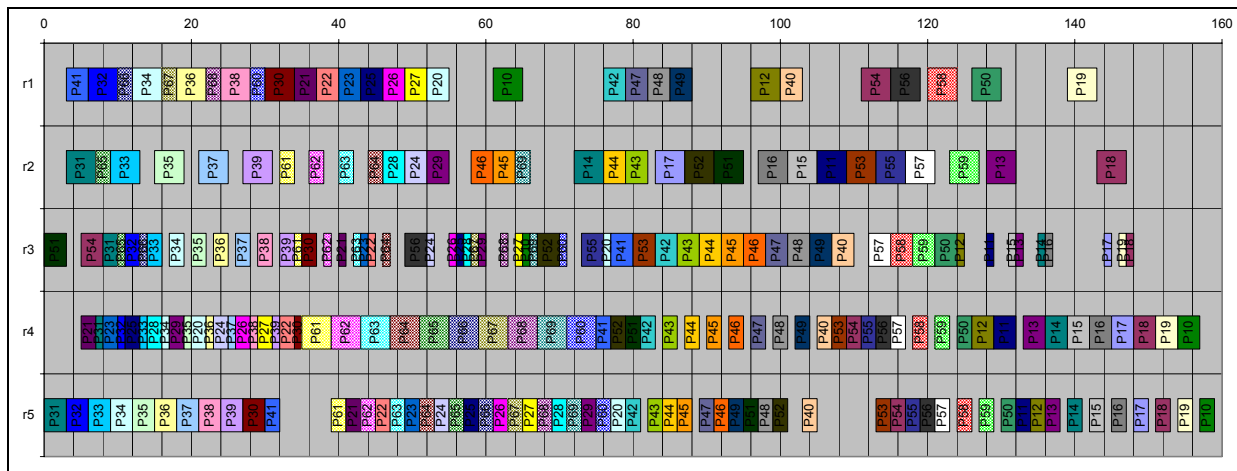


Figura 62: AA\_FJS4x6\_10\_1 - Gráfico de Gantt - P3-6-2-4-5-1

No gráfico de Gantt da Figura 63 apresenta-se o resultado da programação da produção para lotes de 5 unidades. O instante de conclusão da produção de todos os produtos é de 167 UT, obtendo-se uma redução em relação ao caso inicial de 16.5%.

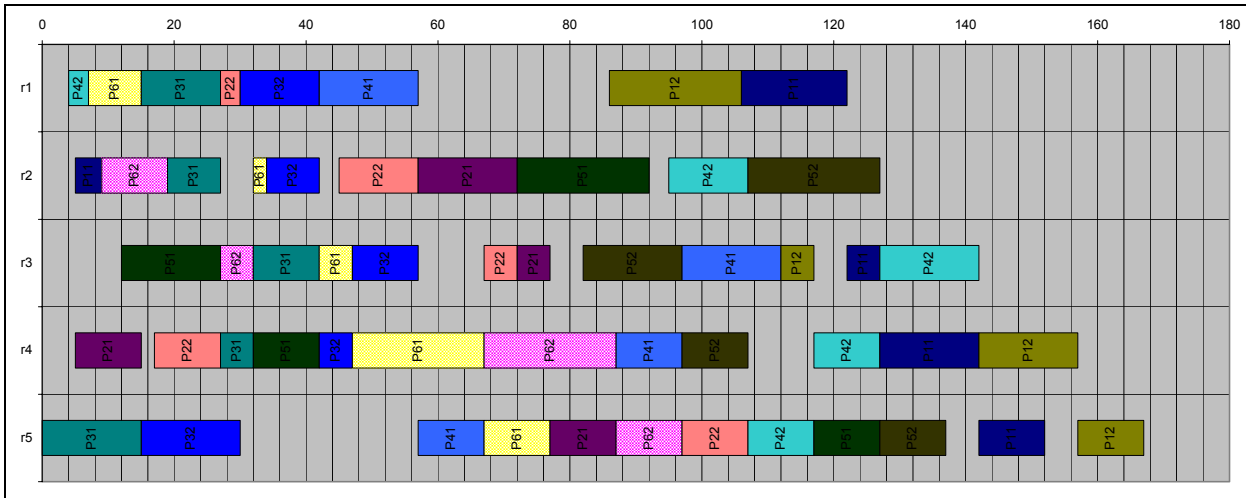


Figura 63: AA\_FJS4x6\_10\_5 - Gráfico de Gantt - P3-6-2-4-5-1

O método DISMAN 10\_1 e DISMAN 10\_5 permitem, como se pode verificar na Tabela 32, obter os melhores resultados relativamente a todos os critérios de comparação. Isto mostra que o método DISMAN aplicado a este caso de teste, com partição e separação de lotes, permite obter bons resultados relativamente ao instante de conclusão das encomendas, absoluto e em média, e bons resultados de utilização dos recursos e de serviço ao cliente.

Tabela 32: AA\_FJS4x6 - Resultados agregados

Método	EDD	FCFS	SB/Cmax	SB/ $\sum(WT)$	SPT	MEOP 10_1	MEOP 10_5	DISMAN 10_0	DISMAN 10_1	DISMAN 10_5
Cmax	210	170	170	172	210	252	252	200	<b>159</b>	<b>167</b>
$\sum C$	820	820	820	762	858	955	1052	870	<b>594</b>	<b>707</b>
Utilização	55,2%	68,2%	68,2%	67,4%	55,2%	46,0%	46,0%	58,0%	<b>73,0%</b>	<b>69,5%</b>
QMS	0,46	0,46	0,46	0,50	0,43	0,37	0,30	0,42	<b>0,61</b>	<b>0,53</b>

A análise dos resultados obtidos pelo método DISMAN neste caso, aplicado com várias dimensões de partição de lote, parece confirmar uma tendência verificada na secção V - 1 para uma única encomenda: com o aumento da simultaneidade de processamento os ganhos de redução dos instantes de conclusão vão diminuindo. Esta análise levou à execução de um processo adicional de experimentação com a aplicação do método DISMAN 10\_2, considerando uma partição de 5 lotes de 2 unidades. Com este método obtiveram-se os resultados agregados apresentados na Tabela 33. A análise dos resultados deste método vêm confirmar a tendência referida. Existe um ganho elevado nas medidas efectuadas, na primeira partição, que é menor para a segunda e quase inexistente ou mesmo não existente para a última partição.



Tabela 33: AA\_FJS4x6 - Resultados agregados apenas para o método DISMAN.

Método	DISMAN 10_0	DISMAN 10_5	DISMAN 10_2	DISMAN 10_1
Cmax	200	167	158	159
$\sum C$	870	707	608	594
Utilização	58,0%	69,5%	73,4%	73,0%
QMS	0,42	0,53	0,60	0,61

## 2.6 Caso AA\_STRUCT

Neste caso, denominado AA\_STRUCT, pretende-se escalonar a produção de 5 encomendas de 4 produtos (P1 ... P4) com estruturas completamente diferentes. O sistema produtivo tem um total de 6 transformações de produção ( $1 \leq j \leq n=6$ ), que recorrem a 9 recursos disponíveis ( $1 \leq k \leq r=9$ ). Os produtos correspondem a alguns equipamentos desportivos, sendo de P1 a P4 respectivamente: Trave Olímpica, Plinto, Espalдар e Bok<sup>44</sup>. A programação da produção é baseada, de acordo com Almeida (2002), na sequência de produção pré-seleccionada P4-3-2-1, imediatamente seguida do escalonamento de uma nova encomenda do produto do tipo P1 (a partir daqui denominada P5), tendo todos os recursos a agenda inicial completamente livre. Isto significa que se considera a sequência P4-3-2-1-5 de lançamento de encomendas. As quantidades encomendadas de cada produto estão representados na Tabela 34. Almeida (2002) analisou o escalonamento em vários cenários de utilização, ou não, dos recursos alternativos, com diferentes lotes de transferência. A comparação baseia-se no cenário que considera a possibilidade de utilização de todos os recursos alternativos, e nos resultados mais favoráveis obtidos por Almeida (2002), relativos à utilização de lotes de transferência de uma unidade ( $l=1$ ).

Tabela 34: AA\_STRUCT - Quantidades Encomendadas

Encomenda	P1	P2	P3	P4	P5
Quantidade	8	5	3	3	12

As sequências de transformações que permitem produzir os 4 produtos estão representada nas redes de Petri da Figura 64. Para obtenção de uma unidade de produto é necessário executar uma unidade de trabalho em qualquer uma das transformações de todos os produtos. Em todos os produtos uma unidade de quantidade de trabalho consome uma unidade de objecto e produz uma unidade de um ou mais objectos, com excepção das transformações  $t_1$  e  $t_3$  do produto P4. Assim:

- Para  $P_1$ :  $q_l = 1, 1 \leq l \leq 9$ ;
- Para  $P_2$ :  $q_l = 1, 1 \leq l \leq 14$ ;
- Para  $P_3$ :  $q_l = 1, 1 \leq l \leq 11$ ;
- Para  $P_4$ :  $\begin{cases} q_l = 1, & l = \{3, 4, 7, \dots, 14\} \\ q_l = 2, & l = \{1, 2, 5, 6\} \end{cases}$ .

<sup>44</sup> "Vault Buck".

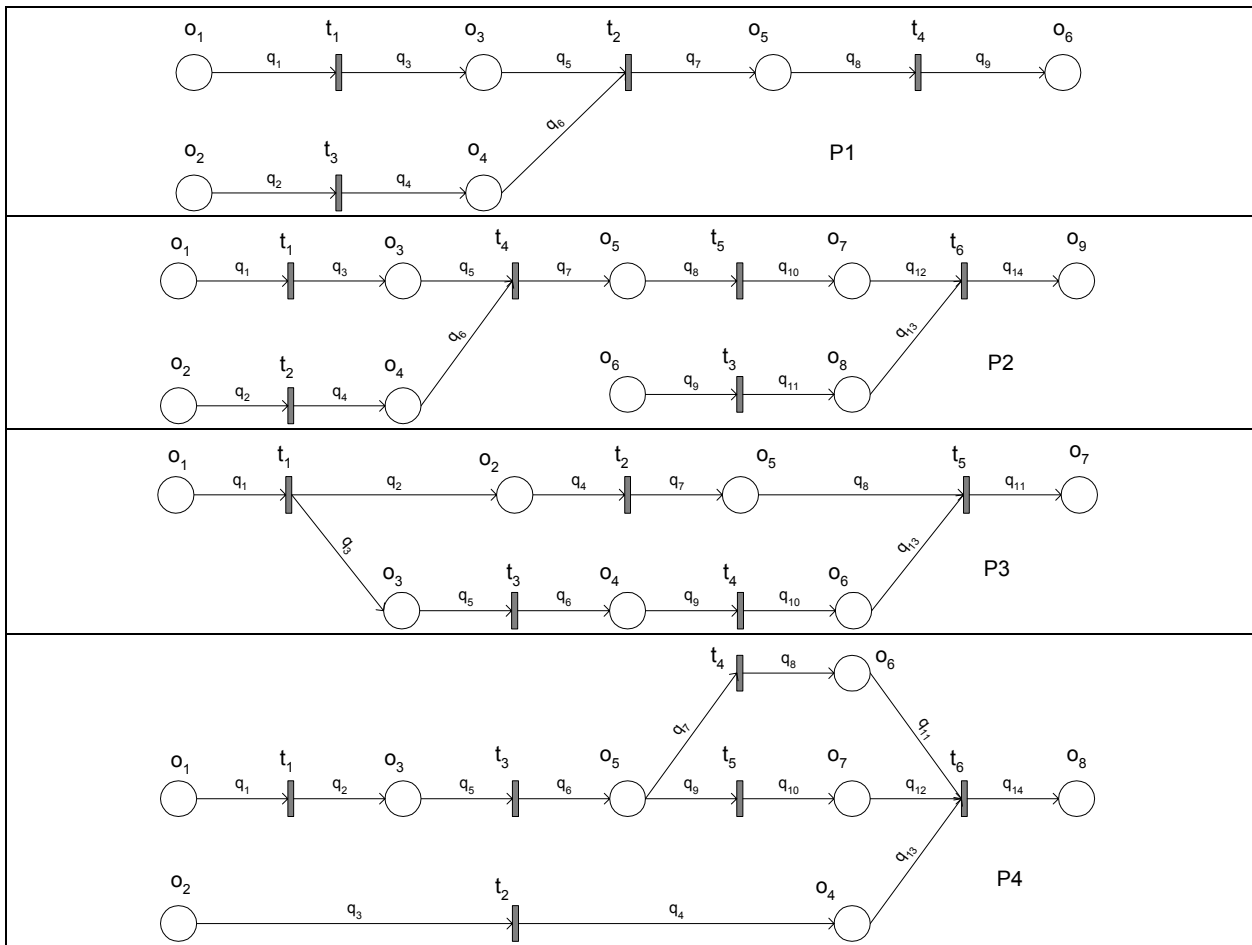


Figura 64: AA\_STRUCT – Redes de Petri dos produtos P1 a P4

Os tempos de execução de uma unidade de quantidade de trabalho em cada transformação por cada recurso estão representados na Tabela 35. Os espaços vazios representam a não possibilidade de utilização desse recurso. Os recursos 1, 2 3 são funcionalmente equivalentes aos recursos 7, 8, e 9.

Tabela 35: Tempo  $c_{jk}$  de processamento unitário de cada transformação por cada recurso

$c_{jk}$	P1	1	2	3	4	5	6	7	8	9
$T_j$	1	1						1		
	2		3						3	
	3			1						1
	4				2					
	5									
	6									

$c_{jk}$	P2	1	2	3	4	5	6	7	8	9
$T_j$	1			3						3
	2	2							2	
	3		2							2
	4						4			
	5				2					
	6								3	

$c_{jk}$	P3	1	2	3	4	5	6	7	8	9
$T_j$	1				2					
	2					3				
	3		1						1	
	4			3						3
	5	2						2		
	6									

$c_{jk}$	P4	1	2	3	4	5	6	7	8	9
$T_j$	1		2							2
	2	3							3	
	3			1						1
	4					3				
	5						2			
	6				2					

O gráfico de Gantt da Figura 65 permite visualizar a ocupação dos recursos como resultado da aplicação sucessiva do método DISMAN na sequência P4-3-2-1-5, em que o 5º produto (aqui designado P5) corresponde ao produto P1.

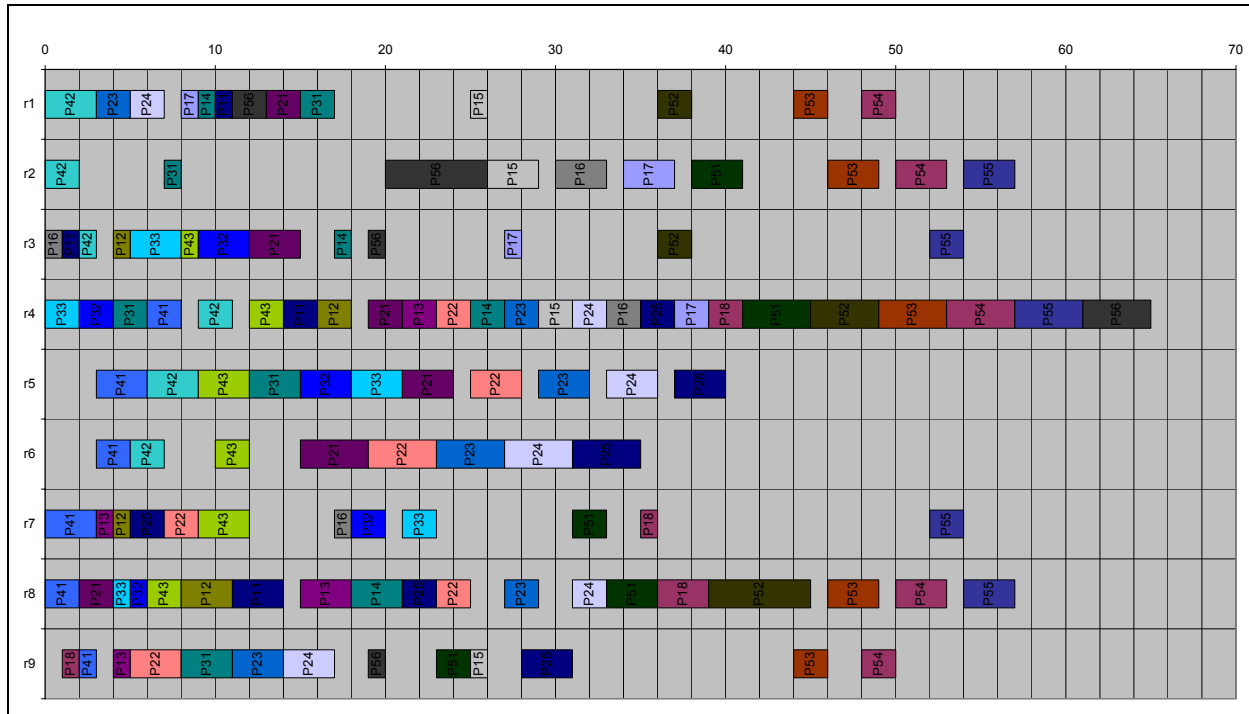


Figura 65: AA\_STRUCT - Gráfico de Gantt – P4-3-2-1-5

Os resultados da Tabela 36 obtêm-se por aplicação do método DISMAN com partição das encomendas em lotes de uma unidade e são comparados com os resultados do método MEOP, com lotes de transferência também de uma unidade. Em todos os casos o processamento dos produtos termina em instantes de tempo inferiores no método DISMAN com ganhos percentuais de 26.8%, 13%, 28.1%, 12.5% e 30.9% para os produtos P1 a P5 respectivamente.

Se medirmos a diferença entre o início e a conclusão de cada encomenda de cada produto, o método MEOP obtém menores diferenças para todos os produtos excepto para P4. Isto resulta do facto do método DISMAN adoptar a partição com separação do processamento das encomendas, explorando períodos de tempo livres mas, previsivelmente alargando o período de tempo entre o início e a conclusão da encomendas. O método DISMAN adopta esta partição por se concentrar no objectivo de conclusão de cada encomenda o mais cedo possível, enquanto o método MEOP adopta, em conjunto com este objectivo, o de não separação do processamento das encomendas.

Tabela 36: AA\_STRUCT - Resultados - P4-3-2-1-5

Prod.	Mét.	r1		r2		r3		r4		r5		r6		r7		r8		r9		Total	
		St	Fin	St	Fin	St	Fin	St	Fin	St	Fin	St	Fin	St	Fin	St	Fin	St	Fin	St	Fin
P1	Meop	29	37	-	-	-	-	30	54	29	37	-	-	40	56	-	-	-	-	29	56
	Dism	8	26	26	37	0	28	14	41	-	-	-	-	3	36	8	39	1	26	0	41
P2	Meop	14	24	-	-	29	39	-	-	14	29	-	-	29	39	31	46	17	37	14	46
	Dism	3	15	-	-	12	15	19	37	21	40	15	35	5	9	2	33	5	31	2	40
P3	Meop	-	-	26	32	20	23	-	-	-	-	21	30	16	22	21	30	-	-	16	32
	Dism	15	17	7	8	5	12	0	6	12	21	-	-	18	23	4	6	8	11	0	23
P4	Meop	5	14	-	-	0	6	-	-	4	7	-	-	10	16	5	14	8	14	0	16
	Dism	0	3	0	2	2	9	6	14	3	12	3	12	0	12	0	8	2	3	0	14
P5	Meop	53	65	-	-	-	-	54	90	53	65	-	-	68	94	-	-	-	-	53	94
	Dism	11	50	20	57	19	54	41	65	-	-	-	-	31	54	33	57	19	50	11	65
Total	Meop	5	65	26	32	0	39	30	90	4	65	21	30	10	94	5	46	8	37	0	94
	Dism	0	50	0	57	0	54	0	65	3	40	3	35	0	54	0	57	1	50	0	65

Relativamente aos critérios de comparação dos resultados agregados do conjunto das encomendas, apresentados na Tabela 37, o método DISMAN obtém melhores resultados do que o método MEOP, com diferenças substanciais. O instante de conclusão de todas as encomendas  $C_{max}$  é inferior aproximadamente 30% no método DISMAN. O somatório dos instantes de conclusão para o método DISMAN é de 183 UT e para o método MEOP de 242 UT. Isto significa que termina mais cedo, tanto o conjunto das encomendas como, em média, todas as encomendas. Além destas medidas, pode verificar-se na Tabela 37, que o método DISMAN obtém uma utilização de recursos superior e que presta um melhor serviço ao cliente, medido por QMS. Pode então concluir-se que o método DISMAN permite a libertação de recursos mais cedo, aumentando a sua utilização pelo processamento simultâneo. Daqui também resulta uma mais rápida entrega das encomendas ao cliente, aumentando a sua possível satisfação e possibilitando um retorno do investimento mais rápido.

Tabela 37: AA\_STRUCT - Resultados agregados

Método	MEOP	DISMAN
$C_{max}$	92	<b>65</b>
$\sum C$	242	<b>183</b>
Utilização	35,3%	<b>49,9%</b>
QMS	0,47	<b>0,60</b>

## 2.7 Considerações Finais sobre o Processamento de Várias Encomendas

Na análise de aplicação do modelo de optimização são assumidos custos de preparação desprezáveis, pelo que a aplicação destes modelos só é válida em sistemas de grande flexibilidade, por exemplo Sistemas de Fabrico Flexível - SFF. Nestes sistemas os equipamentos têm grande versatilidade e o tempo de preparação por mudança de artigo é geralmente pequeno podendo ser desprezável face ao tempo de processamento.

O método desenvolvido neste trabalho permite obter resultados bons na programação da produção para todos os casos aqui analisados. Em todos os casos conseguiram obter-se resultados melhores do que os do método MEOP desenvolvido por Almeida (2002), relativamente ao instante de conclusão do processamento de todos os produtos e ao somatório dos instantes de conclusão. O método permite também obter bons resultados, quanto ao instante de conclusão da produção de cada produto, quando comparado com outros métodos tradicionais.

O instante de conclusão de cada encomenda de um produto é, no método DISMAN, dependente da sequência de lançamento das encomendas. Na realidade, para a primeira encomenda, este método obtém a solução óptima do ponto de vista da formulação do problema. A programação assim obtida para esta encomenda, resulta na ocupação parcial dos recursos, que vai afectar a solução obtida para a encomenda seguinte e assim sucessivamente. Há outros métodos que programam a produção directamente para o conjunto das encomendas, podendo obter uma solução em função de medidas de desempenho para o conjunto das encomendas. No entanto, verificou-se nestes casos de teste a obtenção de soluções pelo método DISMAN, na maioria das encomendas, melhor que outros métodos comparativos ou numa posição próxima do melhor método. Este facto reflecte-se na obtenção sistemática da melhor média do índice de ordenação dos métodos, segundo o instante de conclusão de cada encomenda.

A utilização do conjunto dos recursos aumenta com a diminuição do instante de conclusão, pelo que, como seria de esperar pelos resultados obtidos do instante de conclusão do conjunto de encomendas, este método também permitiu obter bons resultados de utilização dos recursos.

No método DISMAN obtiveram-se soluções sistematicamente melhores do que as de todos os outros métodos relativamente ao somatório dos instantes de conclusão de todas as encomendas e à medida de Qualidade de Serviço de Produção ao cliente QMS. Estas duas medidas estão relacionadas com a redução dos instantes de conclusão médios de todas as encomendas, isto é, melhoram, quando em média, cada trabalho termina mais cedo.

O aumento da partição das encomendas permitiu obter, pela aplicação do método DISMAN aos casos analisados, sempre melhores resultados das medidas utilizadas. No entanto, verificou-se que a grande parte dos ganhos se verifica para valores moderados de partição. Valores baixos de partição permitem também, obter um menor número de operações de preparação e de transporte, pelo que haverá uma tendência de obtenção de maiores vantagens pela utilização de pequenos níveis de partição, que também resultam, como se verificou, em bons resultados relativamente às medidas de desempenho utilizadas.



## **VI - MODELO MSDP BASEADO EM AGENTES DE SOFTWARE**

O modelo de Sistemas Distribuídos de Produção pode ser implementado com diferentes tecnologias alternativas. Neste capítulo considera-se a tecnologia de Agentes de Software para projecto do sistema. Consideram-se ferramentas específicas para projecto e implementação do modelo de Sistemas Distribuídos de Produção, apresentando-se a sua especificação parcial, nomeadamente a especificação do modelo lógico estático de informação e das actividades principais de cada agente.





A definição de Sistema Distribuído de Produção seguida neste trabalho baseia-se em características de Distribuição, Autonomia e Reconfiguração do sistema. O projecto de um sistema deste tipo requer a utilização de tecnologias adequadas. A pesquisa bibliográfica efectuada permitiu constatar que existe um elevado número de trabalhos de investigação baseados nos paradigmas de Agentes. A análise de trabalhos baseados nestes paradigmas, levou à sua consideração para aplicação no projecto de um modelo computacional do modelo de Sistemas Distribuídos de Produção (MSDP).

Este capítulo divide-se em duas secções principais: a primeira de fundamentação teórica sobre Agentes e a segunda de apresentação do projecto de um modelo de Agentes que satisfaz os requisitos do modelo de Sistemas Distribuídos de Produção apresentado em capítulos anteriores.

## **1. AGENTES**

Agente, segundo o dicionário de Português da Porto Editora, é um substantivo que corresponde, entre outros, à *pessoa encarregada de praticar certas operações materiais ou representar os interesses de outrem*. Ampliando este conceito, pode-se generalizar dizendo que um agente é uma entidade *encarregada de praticar certas operações ou representar os interesses de outra entidade*. O conceito de Agentes, em pleno desenvolvimento na comunidade de computação, baseia-se em parte nesta definição, e tem sido utilizado para a modelação de sistemas distribuídos de produção.

Existem trabalhos de estudo explícito sobre Agentes, relatados por Huhns and Singh (1998) e Bradshaw (1997), desde o início da segunda metade do século XX. Estes trabalhos introduziram conceitos que foram, de certa forma, negligenciados durante várias décadas, mas que voltaram a germinar de uma forma intensa na comunidade de Inteligência Artificial desde o início dos anos 80 do século XX, conforme se pode constatar pelos relatos de vários autores como Bradshaw (1997), Huhns and Singh (1998) e Nwana and Ndumu (1999).

A comunidade de Engenharia de Software, nomeadamente das Bases de Dados, Sistemas Operativos e Redes, também utilizam agentes desde a década de 80 do século XX (Huhns and Singh (1998)). Apresentam-se alguns exemplos desta utilização em seguida.

O protocolo de gestão de redes locais SNMP (“Simple Network Management Protocol”) utiliza agentes de monitorização (“High-Tech Dictionary” - <http://www.computeruser.com/resources/dictionary/>), que recolhem informação dos dispositivos da rede. Estes agentes armazenam a informação recolhida numa base de informação de gestão, que é utilizada para verificação do correcto funcionamento desses dispositivos.

O Agente de Utilizador (“User Agent”) é uma aplicação OSI (“Open Systems Interconnection”), que cria, envia e recebe mensagens, representando um utilizador ou uma organização no sistema de gestão de mensagens X.400. O modelo OSI, desenvolvido pela instituição ISO (“International Organization for Standardization”) para permitir a comunicação entre diferentes dispositivos, é a base da maioria dos protocolos de rede existentes. Este é um exemplo de agentes de utilizador para correio electrónico. São agentes que permitem ao utilizador a escrita e leitura de mensagens de correio electrónico. Estas mensagens vão do agente de utilizador para o agente de transferência de correio electrónico e daí são enviados para o destino.

Os agentes Web são componentes de software que podem representar o utilizador em diversas acções avisando-o sempre que ocorre um evento. Alguns dos agentes WEB (<http://botspot.com/>) mais comuns são: agentes de compras com procura de produtos e comparação de preços; agentes de procura de informação; gestores de carregamento<sup>45</sup> de informação pela rede; assistentes pessoais; agentes de jogos; agentes que simulam comportamento humano. Alguns destes agentes WEB são desenvolvidos com características de inteligência artificial, que lhes fornecem capacidades adicionais, nomeadamente: interpretação de linguagem natural; expressão de emoções; execução de raciocínios lógicos. Outros agentes não têm estas características, sendo necessário exprimir explicitamente todas as acções que o utilizador quer executar.

## **1.1 Enquadramento da Utilização de Agentes**

Existe uma relação próxima entre os sistemas baseados em agentes e os sistemas distribuídos. Com a utilização de agentes existe a intenção de explorar a distribuição de recursos, separar e distribuir a resolução de problemas complexos, e aumentar a autonomia e cooperação dos subsistemas. Existem várias razões para o recurso a Sistemas Multiagentes, havendo um paralelismo evidente entre algumas dessas razões e as que são subjacentes ao modelo de Sistemas Distribuídos de Produção apresentado nesta dissertação.

A aplicação de sistemas multiagentes tem vindo a ser fortemente explorada na área industrial e, em geral, na implementação de sistemas complexos, modulares, distribuídos e alteráveis. Pode verificar-se este facto pela existência de inúmeras publicações da comunidade dos Sistemas de Produção e da Inteligência Artificial abordando a aplicação de sistemas multiagente na resolução de inúmeros problemas de afectação distribuída de recursos. Uma revisão exhaustiva de sistemas multiagente aplicados à programação e controlo de produção é efectuada por Baker (1998). Além deste, pode citar-se o seguinte conjunto adicional de exemplos de aplicação aos sistemas de produção, nomeadamente à Engenharia Concorrente, projecto colaborativo, integração da produção, controlo da produção, programação da produção, manuseamento de materiais e sistemas de produção holónicos que se podem encontrar em: Wiendahl and Ahrens (1997); Gunasekaran (1998); Jennings and Wooldridge (1998); Kádár, Monostori and Szelke (1998); Ming *et al.* (1998); Miyashita (1998); Sluga, Butala and Bervar (1998); Sousa and Ramos (1998); Trentesaux, Damien, Dindeleux and Tahon (1998); Parunak *et al.* (1999); Sousa and Ramos (1999); Shen, Maturana and Norrie (2000a); Shen, Maturana and Norrie (2000b); Shen, Norrie and Barthès (2001); Shen (2001); Mondal and Tiwari (2003).

Algumas das razões justificativas da utilização do paradigma de agentes para a modelação de sistemas são:

- Jennings and Wooldridge (1998) – modelação de sistemas abertos cuja estrutura se pode alterar dinamicamente; modelação de sistemas de elevada complexidade através da modularidade e abstracção do paradigma de agentes; modelação de sistemas cooperantes com o utilizador, com maior autonomia, pró-actividade, reactividade e adaptabilidade.
- Nwana and Ndumu (1999) – modelação de sistemas que requerem a interligação e operação de recursos distribuídos.

---

<sup>45</sup> "Download".

A modelação de sistemas distribuídos de produção apresenta alguns requisitos, que estão associados, de uma forma mais estreita, aos requisitos de duas das razões referidas: Sistema Aberto e Distribuição de Recursos.

### **1.1.1. Sistema Aberto**

Um sistema aberto, segundo Huhns and Singh (1998) e Jennings and Wooldridge (1998), é um sistema composto por componentes heterogéneos que comunicam entre si e que se podem alterar ao longo do tempo. A heterogeneidade do sistema coloca problemas de integração que podem ser resolvidos pelo recurso a normas de integração e a agentes de representação de cada componente.

De acordo com a definição de sistema distribuído de produção (capítulo II), a heterogeneidade e a capacidade de reconfiguração do sistema, são características que devem ser consideradas. O sistema distribuído de produção, em função de falhas de subsistemas, alterações do ambiente e/ou reavaliação de estratégias, pode mudar a sua estrutura. Numa situação ideal, essa mudança seria instantânea. Além da característica de reconfigurabilidade, o sistema distribuído de produção é composto por recursos heterogéneos autónomos que cooperam para atingir os objectivos do sistema. Estes recursos podem ser representados por agentes cooperantes, com definição dos conceitos do sistema e da utilização de plataformas de comunicação integradas.

### **1.1.2. Distribuição**

O Sistema Distribuído de Produção envolve um conjunto de entidades de resolução de problemas que estão física ou logicamente distribuídas, e que precisam de interagir entre elas. Nwana and Ndumu (1999) afirmam que a concepção de um sistema como sendo do tipo multiagentes se deve aplicar quando existem problemas de interligação de entidades distribuídas, ou que, no mínimo, tenha subproblemas de afectação de recursos distribuídos. Os problemas de distribuição da produção podem dividir-se em subproblemas de distribuição de controlo, de informação, de materiais, de pessoas e de meios de produção.

#### **i. Distribuição do Controlo**

Ao considerar células autónomas de produção e recursos distribuídos de produção, consideram-se diferentes unidades distribuídas lógica e espacialmente, com capacidade de executarem as operações de transformação sem intervenção directa de outras unidades do sistema. Estas unidades respondem a acontecimentos do seu meio ambiente para satisfazerem os seus próprios objectivos e os da comunidade ou sistema a que pertencem. A selecção de recursos para uma determinada transformação pode resultar da decisão tomada localmente por uma célula de produção.

#### **ii. Distribuição da Informação**

A informação sobre produtos e sobre processos de produção pode estar em locais distintos daqueles onde se situam os meios mais adequados para execução das operações de produção dos produtos. Esta informação pode estar, por exemplo, em diferentes computadores, oficinas, fábricas, empresas ou países, assim como, os recursos de produção seleccionados, seja por razões económicas, técnicas ou de estratégia de produção, podem estar noutros locais completamente diferentes.

### **iii. Distribuição de Materiais, Pessoas e Meios de Produção**

Os materiais, pessoas e meios de produção mais adequados são seleccionados por razões de ordem económica, técnica ou de estratégia empresarial. Tal deve ter em conta que os materiais, pessoas e meios de produção podem não estar todos situados ou disponíveis no mesmo local, podendo, por essa razão, existir a necessidade de cooperação distribuída. A um determinado nível, como foi referido na apresentação da definição de sistemas distribuídos de produção, os meios de produção estão sempre distribuídos e poderão ser modelados dessa forma.

## **1.2 Definições de Agentes**

Os agentes têm servido de tema de investigação em diferentes áreas científicas e com diferentes abordagens. Discutem-se dois tipos fundamentais de agentes relacionados com a Inteligência Artificial e a Engenharia de Software, aqui referidos por agentes inteligentes e agentes de software, respectivamente.

### **1.2.1. Agentes de Software**

Agentes de software são agentes que não incluem mecanismos explícitos de representação de conhecimento, de inferência ou de aprendizagem. São componentes de software que representam utilizadores com o intuito de automatizar as suas acções. Na Tabela 38 apresentam-se definições de três autores diferentes.

Tabela 38: Definições de Agentes de Software

<b>Autor</b>	<b>Definição de Agente de Software</b>
Nwana (1996)	<i>Um agente é um componente de software capaz de actuar de forma a cumprir com tarefas em nome do seu utilizador.</i>
Jennings and Wooldridge (1998)	<i>Sistema computacional com a capacidade de execução de acções autónomas num determinado ambiente de forma a cumprir os seus objectivos de projecto.</i>
Silva, Alberto (1999)	<i>Agentes de software são unidades de código que executam num determinado processo computacional de controlo independente dos programas que com ele interactuam.</i>
Parunak et al. (1999)	<i>Agentes adicionam duas características a objectos passivos: um processo computacional de controlo independente e iniciativa local, usualmente expressa por objectivos locais. Em conjunto, isto permite que o agente monitorize e responda a alterações do ambiente de uma forma autónoma, isto é, sem ser invocado externamente.</i>

Estas definições têm algumas características idênticas, nomeadamente, todas elas referem de alguma forma, a autonomia e os objectivos introduzidos no projecto do agente. A Figura 66 pode servir de ilustração do conceito de agente, com as seguintes capacidades: percepção do ambiente; actuação; representação do estado interno; decisão; comunicação.

Neste contexto, autonomia é a capacidade do componente de software se manter a executar os seus processos computacionais, independentemente dos programas ou utilizadores que com ele interactuam. Isto implica que o agente executa o seu código num processo computacional independente (“thread”) e sem interferência directa do meio ambiente na forma de cumprir determinadas acções. Se o agente não está sujeito a interferências directas, então pode recusar alguns pedidos de execução de tarefas. Ao contrário de um objecto de software, o agente não responde a invocações directas de métodos por mensagens, mas sim a requisições, que pode recusar, colocadas por algum mecanismo de comunicação. Não existindo interferência

directa do meio ambiente, tem que existir algum tipo de capacidade de monitorização que permita ao agente reagir em conformidade com os seus objectivos.

A acção ou reacção executada pelos agentes depende dos objectivos introduzidos no seu projecto. Em função das requisições colocadas ou das alterações do ambiente, o agente terá que avaliar o seu estado interno e os seus objectivos para fornecer uma resposta. Embora não esteja explícito em todas as definições apresentadas, os objectivos dos agentes estão directamente ligados aos utilizadores que representam.

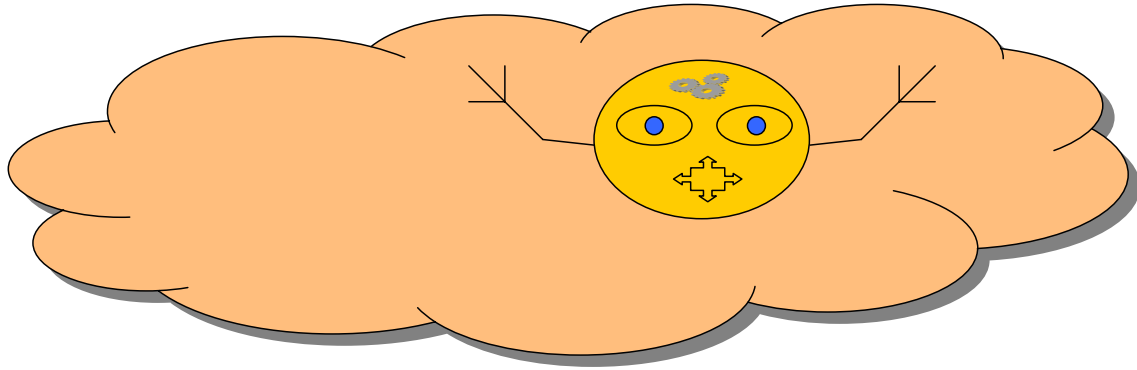


Figura 66: Ilustração de Agente num determinado ambiente.

A abordagem aos agentes segundo diferentes perspectivas ajuda a clarificar o conceito em causa. Silva, Alberto (1999) apresenta os agentes segundo três perspectivas diferentes, descritas a seguir.

### ***i. Perspectiva Genérica***

Um agente é uma entidade de software com uma identidade, estado e comportamento bem definidos, e representa, de alguma forma, o seu utilizador nas tarefas que realiza. Um agente deverá apresentar, pelo menos, características de autonomia, sociabilidade, persistência, pró-actividade e/ou reactividade. A autonomia foi discutida atrás. A sociabilidade é uma característica necessária para implementar a comunicação com o utilizador do agente e para construir sistemas multiagente com interacção entre diferentes agentes. A persistência é uma característica técnica necessária para uma das vertentes da autonomia: a de permitir manter o agente activo ao longo do tempo. Um agente, em função dos seus objectivos de projecto, pode ser pró-activo e/ou reactivo em relação ao seu ambiente, ao utilizador e a outros agentes. Se o agente for pró-activo, pode desencadear acções em função dos seus objectivos sem reacção a nenhum evento. Se o agente for reactivo desencadeia acções em função dos seus objectivos e de eventos.

### ***ii. Perspectiva Técnica***

Um agente é um objecto activo de média granularidade. Isto significa que é uma instância de uma determinada classe representada por um identificador único e global, com o seu próprio conjunto de processos, código e dados. Visto como um novo paradigma, o agente providencia o conceito de objecto activo e autónomo, podendo por isso, ser adequado à concepção de aplicações dinâmicas, distribuídas e/ou complexas.

### ***iii. Perspectiva de Utilização***

Para o utilizador, o agente pode ser visto como um novo paradigma de interacção homem máquina, baseado no modelo de delegação, ou de gestão indirecta, por oposição ao modelo tradicional de interacção

directa. O agente não corresponde a um objecto que responde apenas a invocações directas, mas tem capacidade para monitorizar o ambiente, para actuar em nome do seu utilizador, para emitir avisos, ou para colocar questões sobre acções a executar.

### **1.2.2. Agentes Inteligentes**

A definição de Inteligência Artificial varia com os autores, havendo correntes que apresentam definições baseadas no comportamento humano e outras baseadas num conceito ideal de Racionalidade ou capacidade de raciocínio. Numa definição baseada no comportamento humano, apresentada por Rich and Knight (1991), segundo os quais a Inteligência Artificial corresponde ao estudo da forma de fazer com que os computadores executem acções que, de momento, são melhor executadas por pessoas. Russell and Norvig (1995), dentro de uma corrente relacionada com a Racionalidade, apresentam uma outra definição de Inteligência Artificial, como sendo o estudo e a construção de Agentes Racionais, que são “coisas” que apreendem o ambiente que os rodeia e, baseando-se nas suas convicções, agem de forma a atingirem os seus objectivos. Estes autores realçam a importância de efectuar inferências correctas a partir de formas de representação do conhecimento. No entanto, reconhecem o facto de nem sempre ser possível provar que existe uma decisão correcta, mesmo que seja necessário tomar uma decisão. Por outro lado, pode ser mais rápido tomar uma decisão por reflexo do que uma decisão por dedução.

Turing (1950) apresenta um dos mais conhecidos testes de Inteligência Artificial, o teste de Turing (<http://www.turing.org.uk/>), que pode ser ilustrado da forma apresentada na Figura 67: Uma máquina inteligente deve ser capaz de desempenhar, como uma pessoa, tarefas cognitivas, de forma que um interrogador humano não consiga distinguir uma máquina de uma pessoa. Com base nesta definição uma máquina inteligente deve ter as seguintes capacidades:

- Processamento de linguagem natural, para permitir a comunicação com o interlocutor.
- Representação de conhecimento, para armazenar informação durante o projecto e durante o interrogatório.
- Raciocínio automático, para responder às questões colocadas com base no conhecimento armazenado e em conhecimento deduzido.
- Aprendizagem, para se adaptar a novas circunstâncias e para detectar e extrapolar padrões.



Figura 67: Ilustração do teste de inteligência de Alan Turing.

Na comunidade de Inteligência Artificial existe muito trabalho desenvolvido ao longo dos últimos anos na área dos agentes Inteligentes. Na Tabela 39 apresenta-se um conjunto de definições de Agentes Inteligentes. Nestas definições referem-se duas características fundamentais que são autonomia e objectivos próprios. Estas características também foram referidas nas definições de agentes de software. A autonomia,

relativamente à capacidade do componente actuar sem controlo directo de outros elementos do sistema, está associada à capacidade de monitorização do ambiente e à persistência do componente de software. Os objectivos próprios do agente estão associados à delegação de poderes, sendo definidos a partir dos objectivos dos utilizadores do agente. No entanto, existem diferenças na abordagem efectuada, pela Inteligência Artificial, a estas características, que se descrevem mais à frente.

Tabela 39: Definições de Agentes Inteligentes

Autor	Definição de Agente de Software
Russell and Norvig (1995)	<i>Um Agente é qualquer coisa que pode ser vista como apreendendo o seu ambiente através de sensores e actuando nesse ambiente através de accionadores. Um Agente Inteligente ideal é aquele que, em função do conhecimento projectado e adquirido, escolhe a acção que se espera que maximize a sua medida de desempenho.</i>
Franklin and Graesser (1997)	<i>Um Agente (autónomo) é um sistema situado dentro e participante de um ambiente, que apreende esse ambiente e actua nele, de forma a concretizar a sua própria agenda e como consequência, de forma a afectar a sua visão sobre o futuro.</i>
Bradshaw (1997)	<i>Espera-se que um agente de software aja em nome de alguém para executar uma tarefa que lhe foi delegada.</i>
Shoham (1997)	<i>Um agente é uma entidade contendo componentes mentais como crenças, aptidões, alternativas e compromissos.</i>
Huhns and Singh (1998)	<i>São componentes de software activos e persistentes que apreendem, raciocinam, agem e comunicam.</i>
Jennings and Wooldridge (1998)	<i>Um Agente é um sistema de computador com a capacidade de acção autónoma flexível de forma a atingir os seus objectivos de projecto.</i>
Luck, Mcburney and Preist (2003)	<i>Agentes podem ser definidos como entidades computacionais autónomas de resolução de problemas, capazes de operar em ambientes abertos e dinâmicos</i>

O agente não monitoriza apenas o ambiente, mas tenta compreender esse ambiente, isto é, apreende. Tem capacidade para interpretar novas situações que detectou e adicionar esse conhecimento à sua base de conhecimento. A partir da sua base de conhecimento e da informação apreendida, tem capacidade para inferir novo conhecimento, ou seja, tem capacidade de raciocínio. Essa capacidade de raciocínio facilita o processo de delegação de poderes do utilizador, que não tem que ser totalmente explícito, e permite a dedução de desejos, i.e. objectivos, do seu utilizador. Assim sendo, os objectivos do agente não são apenas de projecto, mas podem evoluir ao longo do tempo de vida do agente. A comunicação com outros agentes também não tem que ser totalmente explícita, e o agente pode aprender novos conceitos ao longo do tempo.

Na Figura 68 apresenta-se a ilustração de um tipo de agente inteligente definido por Russell and Norvig (1995) que, baseado numa função utilidade, mede o grau de satisfação de objectivos relativamente a possibilidades de acção e ao estado do sistema. Um agente tem a capacidade de monitorização do ambiente e de actuação sobre o mesmo. Por outro lado, a partir do estado actual do sistema, do conhecimento das suas acções e da implicação das mesmas sobre o estado do sistema, é capaz de seleccionar acções que maximizem a sua função utilidade.

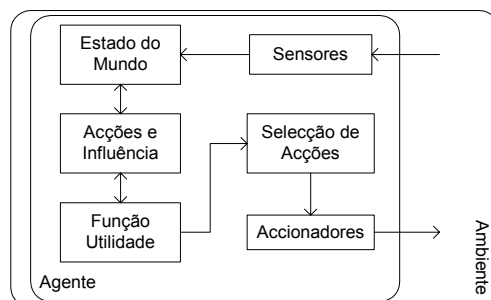


Figura 68: Agente Inteligente - Russell and Norvig (1995).

O conceito de autonomia flexível colocado por Jennings and Wooldridge (1998), aplica-se a sistemas que incluam as seguintes características:

1. **Autonomia:** a capacidade de agir sem intervenção directa de humanos ou outros agentes, tendo controlo sobre as suas acções e estado interno.
2. **Reactividade:** os agentes devem detectar alterações no seu ambiente (pode ser um ambiente físico, um utilizador, uma colecção de agentes, a Internet, etc.) e responder em tempo útil. Baseia-se em três componentes principais: percepção, acção e comunicação.
3. **Pró-actividade:** os agentes devem tomar a iniciativa quando apropriado, não actuando apenas em resposta a alterações no seu ambiente, mas também apresentando comportamentos conduzidos por objectivos.
4. **Sociabilidade:** os agentes devem interagir, quando considerarem apropriado, com outros agentes ou humanos para completarem as suas acções e/ou ajudar outros com as suas actividades.

A coexistência destas quatro características define um agente Inteligente, numa noção *fraca* de agente. O agente pró-activo tem a capacidade de exhibir um comportamento oportunista, conduzido pelos seus objectivos, e tomar a iniciativa quando apropriado. O facto de um agente ter autonomia sobre as suas acções e estado interno, implica que pode negar uma determinada acção sobre a requisição de outros intervenientes do sistema. Esta possibilidade de controlo sobre as suas acções afasta os agentes inteligentes claramente da modelação Orientada aos Objectos. Nesta teoria, a reacção dos objectos à invocação de métodos que alteram o seu estado interno, não pode ser negada. Existe mesmo uma distinção na forma de solicitar as acções destas duas entidades: uma requisição nos agentes e uma invocação nos objectos. Neste contexto, uma requisição corresponde a um pedido e uma invocação corresponde a um chamamento.

Numa noção *forte* de agente, o agente inteligente pode incluir algumas das seguintes características adicionais:

- **Mobilidade:** a capacidade de um agente se mover numa rede de modo a realizar as suas tarefas e cumprir os seus objectivos.
- **Intencionalidade:** capacidade de representação explícita dos objectivos de um agente. Estes agentes, ditos intencionais ou cognitivos, apresentam quatro componentes. Para além da percepção, acção e comunicação, apresentam ainda a capacidade de raciocínio sobre uma base de conhecimento.
- **Aprendizagem:** capacidade de manipulação e geração de conhecimento, actualizando a sua base de conhecimento através do reconhecimento de padrões de comportamentos, de preferências, etc.
- **Veracidade:** não comunicar (deliberadamente) informação falsa.
- **Racionalidade:** não aceitar objectivos impossíveis de concretizar, contraditórios, ou não compensadores em termos de risco / custo / esforço envolvido.
- **Benevolência:** adopção dos objectivos dos outros desde que estes não sejam contraditórios com os seus.
- **Características mentais:** definição de um modelo de agente baseado em noções mentais com conhecimento, crenças, intenções ou desejos.



### 1.3 Sistemas Multiagente

Os sistemas multiagente são caracterizados por comunidades de agentes que interactivam de forma a atingir os objectivos do sistema. Segundo Nwana and Ndumu (1999), pretende-se criar um sistema para interligar agentes desenvolvidos separadamente, permitindo que o conjunto funcione para além das capacidades de qualquer um dos agentes individuais da composição.

Nos sistemas multiagente tem que existir uma noção comum dos conceitos do sistema, necessários para atingir os seus objectivos. Estes conceitos podem ser explicitamente definidos numa ontologia do sistema, ou podem estar implicitamente definidos na base de dados ou base de conhecimento de cada agente. Além dos agentes terem um conhecimento comum de conceitos do sistema, podem também ter, conhecimento sobre o comportamento e reacção do sistema a acções dos agentes e a alterações do ambiente.

A forma como as pessoas compreendem e prevêm o comportamento de outras pessoas pode servir de modelo à forma como os agentes o fazem. As diferentes formas das pessoas preverem o comportamento dos sistemas podem ser, segundo Bradshaw (1997), classificadas pelas Posições de Predição de *Dagget*, representadas na Tabela 40. Se as pessoas compreenderem bem o comportamento físico de um sistema vão tentar prever o comportamento do sistema em função das leis e características físicas. Este é o caso de um jogo de bilhar, em que as pessoas tentam prever o comportamento do sistema em função do conhecimento do comportamento físico que têm desse sistema. Se as pessoas conhecerem um sistema pelas funções para as quais ele foi projectado, como o caso dos automóveis, então podem prever o comportamento do sistema em função desse conhecimento. Há outros sistemas, como pessoas, animais, robots e agentes, perante os quais as pessoas tomam uma posição baseada num determinado modelo de agência racional. As pessoas criam um modelo de uma entidade que se comporta de uma forma racional para poderem prever o seu comportamento. Pode dizer-se que os agentes de software adoptam posições de predição do tipo física e de projecto. Estes agentes podem ter modelos de outros agentes e dos sistemas em que se integram, baseados nas suas funcionalidades projectadas e no seu comportamento físico. Os agentes inteligentes, além destas posições de predição, também podem incluir posições de predição de intenção. Nesta forma de predição, o agente espera que outros agentes se comportem de acordo com um modelo de comportamento racional.

Tabela 40: Posições de Predição de *Dagget*.

Posição de Predição	Descrição
Posição de Predição Física	A predição baseia-se nas leis e características físicas
Posição de Predição de Projecto	A predição baseia-se no projecto do sistema
Posição de Predição de Intenção	A predição baseia-se na assunção de agência racional

Para que os agentes interactuem entre si e com o ambiente, tem que haver uma forma de comunicação estabelecida. A forma de comunicação utilizada depende da arquitectura do sistema, podendo efectuar-se por utilização de linguagens específicas de comunicação entre agentes ou por envio de mensagens ou objectos entre agentes. A monitorização do sistema depende dos seus objectivos, da sua arquitectura e dos seus agentes, podendo ser efectuada por interceptação de requisições que circulem nos sistema, por leitura de sensores, ou por inquéritos directos ao utilizador ou a outros agentes.

### **1.3.1. Arquitectura da Estrutura de Ligação**

A estrutura de ligação entre agentes, segundo Shen, Norrie and Barthès (2001) e Shen (2001), pode basear-se numa das seguintes arquitecturas:

- Hierarquia.
- Federação.
- Agentes Autónomos.

A hierarquia caracteriza-se por relações de domínio de alguns agentes sobre outros. Neste caso, a autonomia dos agentes é restringida por algum grau de controlo imposto pelos agentes dominantes. Este tipo de arquitectura distingue-se de sistemas centralizados porque existe autonomia de alguns agentes.

Numa federação existe a formação de grupos de agentes que comunicam entre si através de agentes que fornecem um serviço de ligação. Estes agentes de ligação podem, fundamentalmente, ser de três tipos: organizador<sup>46</sup>; mediador<sup>47</sup>; intermediário de união<sup>48</sup>. O organizador está ligado a um conjunto de agentes e comunica com outros organizadores para fornecer serviços de ligação que permitam colocar o sistema a funcionar (Shen, Maturana and Norrie (2000b); Sun, Zhang and Nee (2001)). O mediador é um agente que actua num determinado mercado, entre agentes vendedores e agentes compradores, fornecendo serviços de ligação que tornam transparente a ligação entre eles (Parunak *et al.* (1999); Baker, Parunak and Erol (1999); Kim, Choi and Yoo (2001); Carvalho, Putnik and Cunha (2003)). O intermediário de união fornece serviços similares aos do organizador, podendo em seguida abandonar o sistema, porque a partir daí os agentes comunicam directamente entre si (Parunak *et al.* (1999); Shen, Maturana and Norrie (2000b)).

A arquitectura baseada em agentes autónomos é uma arquitectura em que os agentes podem comunicar entre si sem recorrerem a agentes de ligação. Este tipo de arquitectura obriga a que os agentes se conheçam, ou que se utilize uma plataforma comum para transmitir informação a que todos acedam (Wiendahl and Ahrens (1997); Parunak, Baker and Clark (1997); Shen, Norrie and Barthès (2001); Leitão, Restivo and Putnik (2001)).

### **1.3.2. Comunicação / Coordenação**

Um dos principais problemas a resolver na concepção de sistemas distribuídos é a forma de coordenação das acções dos agentes individuais, de modo a fazer com que eles efectivamente trabalhem em conjunto. Algumas das abordagens possíveis a este problema são:

- Um agente está no comando. Este agente constrói um plano e distribuí partes do plano a agentes "escravos". Estes fazem o que lhes mandam fazer e depois relatam o resultado. Se necessário, estes agentes podem comunicar com outros agentes "escravos", para atingir os seus objectivos.
- Um agente está no comando e decompõe o problema em subproblemas, mas nesse momento, através de negociação, os agentes decidem qual deles toma a responsabilidade de cada subtarefa.
- Nenhum agente está no comando, embora exista um único objectivo comum. Os agentes têm que cooperar tanto na formação de um plano como na sua execução.

---

<sup>46</sup> "Facilitator".

<sup>47</sup> "Broker".

<sup>48</sup> "Matchmaker".

- Nenhum agente está no comando, e não há garantia de partilha de um único objectivo. Os agentes podem mesmo competir entre si.

A comunicação entre agentes depende da arquitectura do sistema e da linguagem utilizada. Pode-se utilizar uma linguagem normalizada de comunicação entre agentes, ou uma linguagem desenvolvida não normalizada embebida nos próprios processos dos agentes. Qualquer um destes dois tipos de linguagem pode ser utilizado com qualquer arquitectura. A utilização de uma linguagem normalizada pode, em teoria, simplificar o desenvolvimento de agentes para integração num sistema multiagente existente.

A comunicação entre agentes de um sistema com *arquitectura hierárquica* limita a comunicação às ligações hierarquicamente estabelecidas. Nestes casos, a coordenação do trabalho dos agentes é controlada por agentes em posição hierárquica superior.

Nas *federações*, a comunicação é efectuada através de agentes de ligação e a coordenação pode depender desses agentes. Em termos gerais, o agente de ligação do tipo organizador, fornece serviços de ligação que possibilitam a comunicação entre os agentes; os agente do tipo mediador, além destes serviços de ligação, podem também fornecer mecanismos de negociação relacionados com as regras do mercado em que se inserem; os agentes do tipo intermediário de ligação, fornecem os serviços de ligação e em seguida deixam que os agentes comuniquem directamente entre eles.

Nos sistemas baseados em *agentes autónomos*, a coordenação é desejável desde que permita aumentar a eficiência do sistema ou garantir o cumprimento de objectivos globais, que podem resultar da partilha de informação entre agentes ou da conjugação de recursos e competências de diversos agentes. Neste sistemas multiagente teremos três abordagens principais à comunicação:

- sem comunicação;
- com transmissão de mensagens;
- com um sistema partilhado de armazenamento de informação.

Na primeira destas abordagens, mais limitada do ponto de vista da cooperação, o agente tem um modelo dos outros agentes e da forma como ele se comporta para prever os seus estados e pode ter também alguma capacidade de monitorização do ambiente.

Na segunda abordagem, existe um protocolo estabelecido entre os agentes que conhecem alguns dos agentes existentes no sistema, havendo troca de mensagens directamente entre os agentes ligados.

Na terceira abordagem, os agentes não precisam de saber que agentes é que existem no sistema porque colocam mensagens numa área partilhada, visível a todos os agentes do sistema. Neste caso a comunicação pode ser directa ou indirecta, dependendo do facto de se identificar o receptor ou não. Este sistema de comunicação é similar a um sistema de quadro negro<sup>49</sup>, que conforme descrito por Rich and Knight (1991), corresponde a uma estrutura de memória partilhada, através da qual as fontes de conhecimento comunicam umas com as outras.

Nas duas últimas abordagens, tanto se pode ter comunicação para um único agente como para vários agentes simultaneamente, havendo a necessidade de utilizar um protocolo de coordenação que permita que os agentes cheguem a um entendimento.

---

<sup>49</sup> "Blackboard System".

Além da utilização de meios de comunicação, e de formas de coordenação é necessário que os agentes sejam capazes de compreenderem a informação recebida. Para este efeito é necessário que os agentes conheçam os conceitos comunicados e tenham a mesma interpretação sobre esses conceitos.

A definição de *ontologia* apresentada pelo dicionário de português da Porto Editora é a seguinte: *parte da metafísica que estuda o ser em si, as suas propriedades e os modos por que se manifesta; ciência que estuda os seres considerados em geral*. Esta é uma definição que evoluiu a partir de um ramo da filosofia fundado pelo filósofo grego Parménides de Élea<sup>50</sup>. Este estudo das propriedades do ser inspirou a aplicação deste termo pela comunidade da Inteligência Artificial, que o usa para denominar o conjunto de conceitos que têm que ser conhecidos pelos intervenientes de um sistema, para poderem comunicar.

## **1.4 Ferramentas de Modelação e Desenvolvimento**

Variadas ferramentas utilizadas nos sistemas de informação e na inteligência artificial têm vindo a ser aplicadas nos sistemas baseados em agentes. Algumas destas ferramentas fornecem mecanismos explícitos de inteligência artificial, como linguagens baseadas em lógica de primeira ordem, mecanismos de inferência, demonstradores de teoremas. As ferramentas dos sistemas de informação mais utilizadas são linguagens de representação da informação, como aquelas que são fornecidas pela norma UML, linguagens de programação como C++ e Java, e normas de implementação de objectos distribuídos, como DCOM e CORBA que fornecem mecanismos de ligação e mobilidade de objectos através da rede.

Existem vários organismos e instituições com trabalho desenvolvido em arquitecturas, linguagens e plataformas de suporte a sistemas multiagente, nomeadamente: AgentLink, FIPA; UMBC AgentWeb.

Uma das instituições mais citadas é a Fundação para Agentes Físicos Inteligentes (FIPA – “Foundation for Intelligent Physical Agents”), cujo objectivo fundamental é o de produzir normas de software para sistemas baseados em agentes e para agentes heterogéneos que interactuam entre si nestes sistemas. O grupo Agentcities, de individualidades e organizações, forma uma rede aberta que apresenta ferramentas para sistemas baseados em agentes que respeitam as normas FIPA. Alguns exemplos de sistemas que se podem aqui encontrar são: April Agent Platform; Comtec Agent Platform; FIPA-OS; JADE; LEAP; ZEUS.

A AgentLink, Rede Europeia de Excelência para a Computação Baseada em Agentes, é uma organização fundada pela Comissão Europeia, composta por um conjunto de investigadores e produtores com interesses comuns na tecnologia de agentes.

O UMBC AgentWeb, é um repositório de recursos na Internet com interesse para os sistemas baseados em agentes, mantido por Tim Finin e Yannis Labrou do Laboratório para Tecnologia de Informação Avançada, de Engenharia Electrotécnica e Ciências da Computação da Universidade de Maryland, Baltimore (UMBC – “University of Maryland Baltimore County”).

Nesta secção pretende-se apresentar um pequeno conjunto de ferramentas que podem ser utilizadas para projecto e desenvolvimento de sistemas multiagente.

---

<sup>50</sup> «O que se pode dizer e pensar é forçoso que seja» e «não é para ser dito nem pensado o que não é». Parménides de Élea. In *Infopédia*. Porto: Porto Editora, 2003. [Consulta: 2003-06-25]. Disponível na WWW: <URL: <http://www.infopedia.pt/E1.jsp?id=93312>>.

### **1.4.1. Linguagens**

As linguagens mais referidas no desenvolvimento de sistemas baseados em agentes inteligentes são as linguagens KIF e KQML. KIF (“Knowledge Interchange Format”) é uma linguagem orientada ao intercâmbio de conhecimento entre diversos programas de software, cujas principais características são:

- Semântica declarativa – o significado de expressões nas representações pode ser entendido sem recurso a um intérprete para manipular essas expressões;
- Compreensão lógica – permite a expressão de frases em lógica de primeira ordem;
- Permite a representação de conhecimento sobre a representação de conhecimento;
- Permite a definição de objectos, funções e relações.

KQML (“Knowledge Query and Manipulation Language”) é uma linguagem e um protocolo para trocar informação e conhecimento. KQML é, em simultâneo, um protocolo de formato e de manuseamento de mensagens, que permite partilhar conhecimento entre agentes de um sistema.

As ontologias, referidas atrás, são utilizadas para definição de conceitos que todos os intervenientes devem conhecer, podendo ser construídas com linguagens de representação de informação e conhecimento como KIF, KQML, XML e UML (Cranefield, Hausteiny and Purvis (2001)).

### **1.4.2. Ambientes**

Arquitectura Aberta de Agentes - OAA é um ambiente de integração de agentes heterogéneos de software em ambientes distribuídos. Este ambiente é de acesso livre através do site da empresa criadora SRI (<http://www.ai.sri.com/~oaa>). As bibliotecas de agentes OAA existem para diferentes sistemas operativos e diferentes linguagens de programação, incluindo Java, Delphi e Visual Basic. OAA tem as seguintes características:

- Aberta: os agentes podem ser criados em diferentes linguagens de programação.
- Extensível: possibilidade de adição ou substituição de agentes durante o funcionamento.
- Distribuída: os agentes podem ser espalhados através de quaisquer computadores ligados em rede.
- Paralelismo: os agentes podem cooperar ou competir por tarefas em paralelo.
- Mobilidade: a utilização de interfaces leves permite o funcionamento em dispositivos móveis.

O modelo de objecto componente - COM é uma arquitectura de software, desenvolvida pela Microsoft, que permite construir aplicações a partir de componentes binários de software, i.e. prontos a serem executados. COM é a arquitectura de alicerce que constitui a fundação para níveis mais elevados de serviço de software em sistemas Windows. O DCOM é um protocolo que permite aos componentes de software comunicarem directamente através da rede. Foi concebido para funcionar através de diferentes protocolos de rede, incluindo protocolos de Internet.

CORBA (“Common Object Request Broker Architecture”) é a solução da OMG (“Object Management Group”) para a necessidade de interoperabilidade entre diferentes produtos de hardware e software (Nelson (1999)). Permite às aplicações comunicarem umas com as outras, independentemente da sua localização ou de quem as projectou. Os ORB simplificam este processo de comunicação, pela utilização de interfaces de

aplicação. No caso do CORBA utiliza-se a linguagem de definição de interfaces de objectos IDL, que permite a integração de componentes já existentes.

JATLite (“Java Agent Template Lite”) é um pacote de programas escrito em Java, desenvolvido na Universidade de Stanford (Shen, Norrie and Barthès (2001)). Permite aos utilizadores a criação rápida de agentes de software que comunicam robustamente através da Internet. JATLite proporciona uma infraestrutura básica, na qual os agentes se registam com um agente de ligação para as mensagens dos agentes, ligam-se e desligam-se da Internet, enviam e recebem mensagens, transferem ficheiros e invocam outros programas ou acções nos vários computadores em que estão a funcionar. JATLite facilita, em especial, a construção de agentes que enviam e recebem mensagens em KQML.

Voyager é um sistema proprietário, da empresa *ObjectSpace* (<http://www.objectspace.com>), de desenvolvimento de agentes, em linguagem Java, baseados num sistema ORB<sup>51</sup>. Um ORB providencia a capacidade de criar objectos num sistema remoto e de invocar métodos desses objectos. O sistema Voyager permite criar agentes segundo a norma CORBA, invocar métodos desses agentes remotamente e adicionar mobilidade aos agentes (Nelson (1999)). Os agentes Voyager têm mobilidade e autonomia, ambas fornecidas pela classe base de agente. Um agente pode mover-se de um local para outro e deixar para trás um endereço de envio para que mensagens, que lhe são destinadas, possam chegar ao seu destino. As mensagens podem ser sincronizadas, só de ida, ou futuras. As mensagens futuras são não sincronizadas, mas estão associadas a um local de armazenamento que pode ser usado para mais tarde recuperar um valor devolvido. O Voyager utiliza serialização para enviar o estado do agente enquanto ele se desloca de uma localização para outra. O Voyager também inclui um gestor de segurança que pode ser utilizado para restringir as operações que um agente pode desempenhar. Os agentes Voyager não têm inteligência inerente mas podem ser aumentados com técnicas de inteligência artificial.

DESIRE (“DEsign and Specification of Interacting REasoning components”) é um ambiente de modelação para desenvolver sistemas baseados em agentes, desenvolvido na Universidade Vrije de Amsterdão, que inclui:

- Um método de desenho para sistemas multiagente compostos.
- Uma linguagem formal de especificação para projecto de sistemas – apoiando o projecto conceptual até ao projecto detalhado.
- Ferramentas de software para apoiar o projecto de sistemas.
- Um gerador de implementação para traduzir automaticamente as especificações para código, num ambiente específico de implementação.
- Ferramentas de verificação de propriedades estáticas dos componentes, como consistência e correcção.

A estrutura das especificações DESIRE é baseada na noção da arquitectura composta: uma arquitectura composta por componentes com relações hierárquicas entre si. Cada componente tem a sua própria interface especificando a dinâmica do sistema, incluindo saídas, entradas e conhecimento de controlo de tarefas e tornando o sistema estruturado e descentralizado. Os componentes (compostos ou primitivos) podem ser reutilizados como blocos de construção; as suas interfaces de entrada e saída são definidas, mas a sua

---

<sup>51</sup> “Object Request Broker”.

estrutura interna está escondida do resto do sistema. Os agentes, que desempenham parte de uma tarefa complexa, são modelados como componentes compostos. A funcionalidade de um componente primitivo é, em geral, especificado de uma forma declarativa através de uma base de conhecimento.

## **2. MODELO MULTIAGENTE DO MSDP**

A utilização de um sistema multiagente permite explorar algumas das características destes sistemas para representação do Sistema Distribuído de Produção, de acordo com o que ficou definido na secção II - 3.4 do modelo MSDP. As razões para se utilizarem sistemas baseados em agentes e em particular para se justificar a utilização destes sistemas para modelação do MSDP, foram apresentadas na secção VI - 1.1. Estas razões podem resumir-se da seguinte forma:

- Distribuição - Os sistemas multiagente são sistemas adequados à implementação de sistemas distribuídos e complexos com problemas de atribuição de recursos.
- Autonomia - A autonomia é uma das características fundamentais da definição de agente, sendo, portanto, um paradigma adequado de representação de elementos autónomos de processamento.
- Reconfiguração – A capacidade de reconfiguração do sistema SDP pode ser implementada por mecanismos de coordenação do comportamento dos agentes.

O projecto de um modelo multiagente do MSDP permite criar uma representação de um modelo do SDP que pode ser implementado com agentes. Este modelo baseado em agentes é complementar do modelo MSDP e da respectiva representação formal, desenvolvidos nos capítulos III e IV. O projecto deste modelo baseado em agentes, fornece assim, uma representação adicional do MSDP. Esta representação permite implementar um SDP computacional, que por sua vez permite que se realizem estudos do sistema em ambiente dinâmico, isto é, com atrasos e falhas de comunicação e/ou de recursos. Com este sistema SDP computacional é possível verificar, por exemplo: o impacto do aumento de recursos e da distância no processo de comunicação e coordenação; formas de modelação de recursos com diferentes graus de autonomia; aplicabilidade da utilização do SDP em ambiente industrial.

O projecto e desenvolvimento de um sistema baseado em agentes coloca, à partida, duas possibilidades principais: sistema baseado em agentes de software ou em agentes inteligentes. As características dos agentes de software requerem a utilização de linguagens de modelação de sistemas de informação e ambientes de desenvolvimento e implementação que forneçam serviços adequados ao ciclo de vida dos agentes e à comunicação em rede entre esses agentes. Os agentes inteligentes, além destas características, também requerem a utilização de linguagens de formalização e manipulação de conhecimento, mecanismos de inferência que permitam deduzir conclusões correctas, capacidade de aprendizagem e capacidade de dedução de novos objectivos. Este conjunto adicional de características torna mais complexo o projecto e o desenvolvimento de sistemas baseados em agentes inteligentes.

O modelo MSDP desenvolvido neste trabalho não apresenta características que obriguem a seleccionar um destes tipos de agente. Por esta razão, pode optar-se pela modelação do SDP baseado em agentes de software. Esta opção permite modelar o sistema sem a complexidade adicional dos sistemas baseados em agentes inteligentes.

As metodologias clássicas de Engenharia de Software, segundo Pressman (1987), seguem, em geral, um modelo de ciclo de vida em cascata<sup>52</sup>, composto por um conjunto de etapas fundamentais representadas na Figura 69. Outras metodologias, como a de utilização de protótipos ou orientadas aos objectos mantêm estas fases com alterações que permitam abreviar e/ou flexibilizar o seu fluxo de controlo.

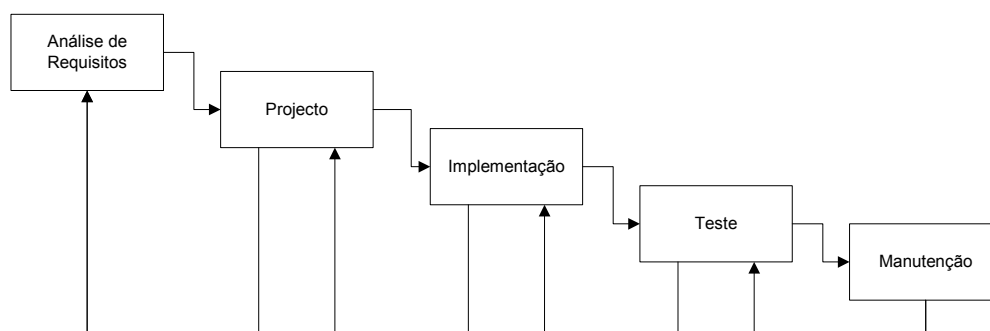


Figura 69: Engenharia de Software - Ciclo de Vida de Clássico.

Seguindo as duas primeiras fases do ciclo de vida clássico, efectua-se uma análise de requisitos e projecta-se um modelo parcial do sistema. O modelo parcial do SDP baseado em agentes é composto por: um modelo dos agentes, descrevendo o tipo de agentes e a sua forma de coordenação; um modelo estático de informação do sistema, que todos os agentes utilizam; modelos das principais actividades dos agentes. Foi desenvolvido trabalho de implementação que permitiu verificar a possibilidade de utilização de algumas ferramentas, mas não foi efectuado um teste de implementação completa do projecto.

## 2.1 Requisitos do Sistema

O projecto do sistema multiagentes baseia-se num conjunto de requisitos fundamentais do modelo MSDP:

1. Introdução de produtos no sistema.
  - i. Definição e especificação do produto.
  - ii. Construção da estrutura do produto.
  - iii. Especificação de processos.
2. Introdução de clientes no sistema.
  - i. Identificação.
  - ii. Capacidade de gerar encomendas.
3. Introdução de encomendas no sistema.
  - i. Definição do produto.
  - ii. Introdução de quantidades.
  - iii. Selecção de recursos.
  - iv. Registo de abandono de um recurso do sistema e correcção do problema.
  - v. Controlo do andamento da encomenda
4. Os recursos deverão ter capacidade de resposta a diferentes situações, nomeadamente:
  - i. Registo dos Tipos de Processo Produtivo que pode executar.
  - ii. Resposta a pedido de encomendas: viabilidade, tempo de entrega e custo.
  - iii. Resposta a pedido de agenda.
  - iv. Introdução de encomendas contratadas pelos clientes.

---

<sup>52</sup> "Waterfall".



- v. Programação dos seus trabalhos.
  - vi. Controlo dos trabalhos programados.
  - vii. Processamento de encomendas efectuadas.
  - viii. Representação de um recurso físico.
  - ix. Objectivo de utilização eficiente do seu recurso.
5. Critérios de Avaliação
- i. Tempo de resposta aos pedidos de encomenda.
  - ii. Tempo de entrega.
  - iii. Cumprimento de prazos.
  - iv. Custo.
6. Algumas destas funções exigem capacidade de:
- i. Comunicação com outros recursos.
  - ii. Actuar sobre o meio ambiente.
  - iii. Reagir a alterações do meio ambiente.

## **2.2 Ferramentas Seleccionadas**

Com base no modelo MSDP e na análise de requisitos pode utilizar-se a tecnologia de agentes de software para criar um modelo conceptual do sistema. Para representação e modelação dos agentes do sistema utiliza-se a norma UML de especificação de sistemas de informação. Fundamentalmente, serão utilizados os mecanismos de representação estática da informação e diagramas de actividade de objectos desta norma. Para desenvolvimento preliminar utiliza-se uma ferramenta de suporte a agentes de software de utilização livre: JavaSpaces. Esta ferramenta fornece mecanismos de registo, procura, comunicação e armazenamento de objectos num local da rede. Este local denomina-se JavaSpace.

### **2.2.1. Especificação de Sistemas de Informação**

Um dos requisitos do projecto de sistemas é o de utilização de linguagens de modelação do sistema, nas vertentes estrutural e funcional. O Grupo de Gestão de Objectos – OMG<sup>53</sup> – é uma instituição que reúne inúmeros elementos da industria de software com objectivo de criar normas de suporte à industria. Uma das normas da OMG para modelação de sistemas é o UML, que permite especificar, visualizar e documentar modelos de sistemas de software. Esta linguagem, embora tenha sido desenvolvida para sistemas de software, pode ser utilizada para a modelação de outros sistemas.

O UML é composto por três categorias de diagramas (Omg (2003)) representando diferentes aspectos do modelo:

- Diagramas de Estrutura: Diagramas de Classes, Objectos, Componentes e Implementação<sup>54</sup>.
- Diagramas de Comportamento: Diagramas de Casos de Uso, Diagramas de Sequência, Diagramas de Colaboração, Diagramas “Gráfico de Estado”<sup>55</sup> e Diagramas de Actividade.
- Diagramas de Gestão de Modelos: Diagramas de Pacotes, Subsistemas e Modelos.

O primeiro grupo destes diagramas permite modelar aspectos estruturais do modelo, nomeadamente: quais as classes que o compõem e o respectivo inter-relacionamento existente; estados do sistema

---

<sup>53</sup> “Object Management Group”.

<sup>54</sup> “Deployment”.

<sup>55</sup> “Statechart”.

representados por instâncias de classes, i.e. objectos, em determinado instante; componentes de software; diagramas utilizados para entrega do sistema.

Os diagramas de comportamento permitem modelar o comportamento do sistema com diversas ferramentas diferentes. Jackson, Liddle and Woodfield (1999) classificam os Diagramas de Casos de Uso, de Colaboração e de Sequência como diagramas de interação, utilizados para especificar a interação entre objectos externos e internos, e entre objectos do sistema. Os Diagramas de Casos de Uso descrevem a interação dos utilizadores externos com o sistema. Diagramas de Colaboração e de Sequência descrevem a interação entre objectos do sistema, os primeiros realçando o relacionamento entre eles e os segundos realçando aspectos temporais entre mensagens. Jackson, Liddle and Woodfield (1999) classificam os Diagramas “Gráfico de Estado” e de Actividade como diagramas de descrição de fluxo de processos baseando-se em máquinas de estado. O primeiro destes diagramas é mesmo uma variante da máquina de estados utilizado para representação do comportamento de um objecto. O segundo é um caso especial do primeiro, onde se representam acções imediatamente seguidas de estados para descrição de casos de uso ou operações.

Na última categoria incluem-se diagramas que apoiam o gestor do sistema de software a agrupar classes e funcionalidades do sistema, permitindo uma visão mais clara do sistema e facilitando o planeamento e controlo dos projectos de software.

No sistema especificado neste capítulo utilizam-se apenas diagramas de modelação estrutural e de comportamento, nomeadamente, Diagramas de Classes e Diagramas de Actividade. Com estes diagramas é possível representar o modelo lógico estrutural da informação utilizada pelos agentes e as principais actividades dos agentes do sistema.

### **2.2.2. Ambiente de Suporte a Agentes de Software**

Os agentes de software são componentes de software autónomos, capazes de monitorizar o seu ambiente e executar acções de acordo com os objectivos definidos. Os sistemas multiagente terão que se basear em componentes com estas características e com a capacidade de comunicação entre os agentes. Os agentes de software são especificados como objectos de software a correr num processo de controlo próprio do sistema operativo, capazes de comunicar através dum espaço partilhado de armazenamento de informação, com base num modelo comum de informação sobre o SDP. Para linguagem de desenvolvimento dos objectos optou-se pela linguagem Java, pela sua característica de independência da plataforma e para exploração do ambiente de partilha de objectos JavaSpace. Este ambiente é fornecido livremente em ambiente de código aberto e permite desenvolver sistemas distribuídos de software. Embora o sistema aqui descrito não esteja implementado, considera-se importante a selecção e descrição de mecanismos de implementação que permitam visualizar a viabilidade do modelo. Foram consideradas outras possibilidades como por exemplo, o sistema Voyager, similar a este. O facto de ser um sistema proprietário com um custo associado levou ao seu abandono após alguma experimentação baseada em Nelson (1999) e em Silva, Alberto (1999). O sistema Desire também foi sumariamente utilizado, mas não foi seleccionado por ser de desenvolvimento de agentes inteligentes. Desire, JATLite e OAA são sistemas a analisar em trabalho posterior.

O ambiente JavaSpaces é descrito por Halter (2002) e por Bishop and Warren (2003) como um sistema que fornece um conjunto de serviços para gerir objectos de software distribuídos. Os objectos podem ser colocados num “espaço”, denominado JavaSpace, onde se mantêm de uma forma persistente. O JavaSpace funciona como um serviço Jini, que possibilita, aos clientes do espaço, o armazenamento e a partilha de informação sobre a forma de objectos. A Figura 70 ilustra a utilização de um JavaSpace para comunicação entre agentes, onde estes podem colocar objectos de comunicação que são lidos por outros agentes.

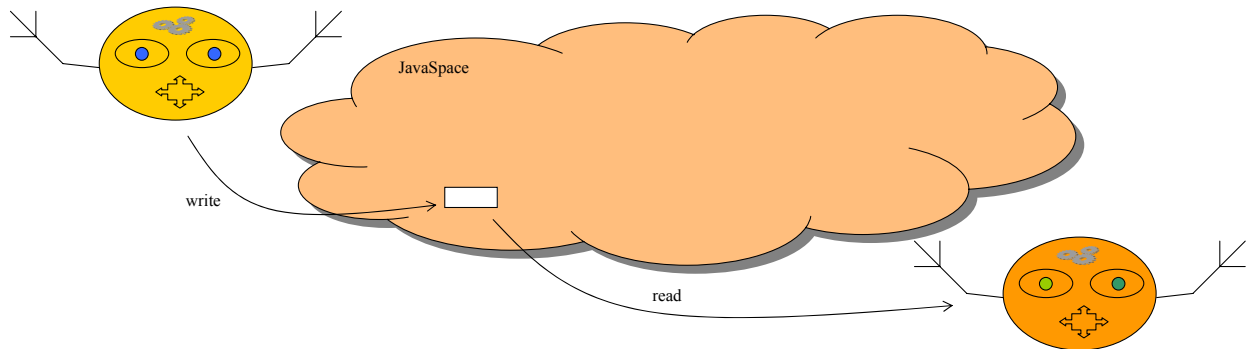


Figura 70: JavaSpaces como espaço de partilha de objectos.

Jini é um sistema construído com base na linguagem Java. O sistema JavaSpaces é um sistema construído com base na linguagem Java e nas características adicionadas pelo Jini. Esta relação é ilustrada pela Figura 71. A principal característica do Jini é possibilitar que as aplicações possam encontrar e utilizar serviços através da rede. Jini fornece mais um conjunto de características de suporte ao funcionamento em rede, como sejam: Empréstimo<sup>56</sup>, que assegura a libertação de recursos ao fim de algum tempo; Eventos remotos, que permitem que um objecto seja notificado por um serviço Jini do acontecimento de algum evento; Transações, que asseguram a execução completa de um processo ou o retorno ao estado anterior ao início do processamento.

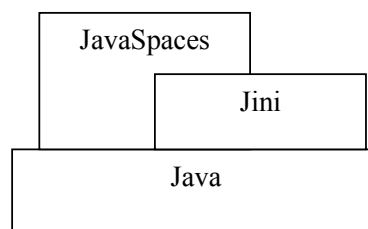


Figura 71: JavaSpaces e Jini - retirado de Halter (2002).

### **2.2.3. Exemplo de Aplicação das Ferramentas Seleccionadas**

Para teste da aplicabilidade destas ferramentas desenvolveu-se um exemplo simples de utilização dos JavaSpaces, para um sistema de encomenda e oferta. Neste exemplo utiliza-se um objecto de informação, representando uma encomenda, para colocar uma requisição de encomenda no espaço partilhado. Esta requisição é lida por objectos recurso que efectuem uma oferta aleatória para a satisfação daquela encomenda. Neste exemplo não se utiliza nenhum método de selecção de ofertas, verificando-se apenas as possibilidades de comunicação e identificação de objectos através do espaço partilhado.

<sup>56</sup> “Leasing”.

A Figura 72 representa uma parte do modelo lógico estático do exemplo desenvolvido. O objecto representativo do cliente, apresentado na Figura 72 como “Customer”, gere a colocação das suas encomendas através de um objecto “OrderMngmt” de gestão de encomendas no espaço utilizado. O objecto “CustomerEntry” é o objecto que o cliente pode colocar no espaço partilhado para a sua identificação. Todos os objectos que podem ser colocados no JavaSpace têm que ser descendentes da classe “Entry” definida pelo ambiente Jini e, neste exemplo, no nome atribuído à classe têm o subconjunto de caracteres “Entry”.

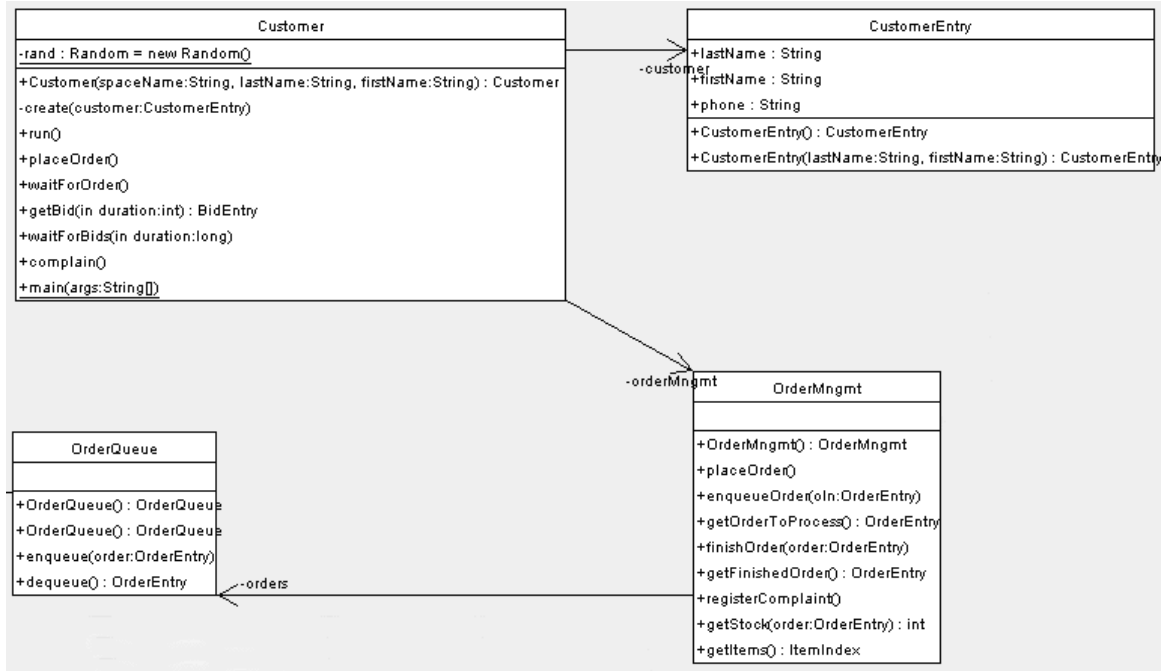


Figura 72: Exemplo de Utilização de JavaSpaces – Relações do Objecto “Customer”.

A Figura 73 representa os objectos que são colocados no JavaSpace para comunicação entre diferentes elementos do sistema. Além do objecto “CustomerEntry” referido atrás, são utilizados objectos “OrderEntry” para publicação das requisições de encomenda, “ProcessorEntry” para possível identificação do recurso processador das encomendas e “BidEntry” para publicação das ofertas do recursos. O objecto “Customer” cria uma encomenda e coloca a requisição “OrderEntry” no JavaSpace, com um tempo de vida, “lease”, predefinido. Em seguida fica à espera de ofertas, tentando ler do espaço objectos “BidEntry” referentes à encomenda, e colocados por objectos dos recursos.

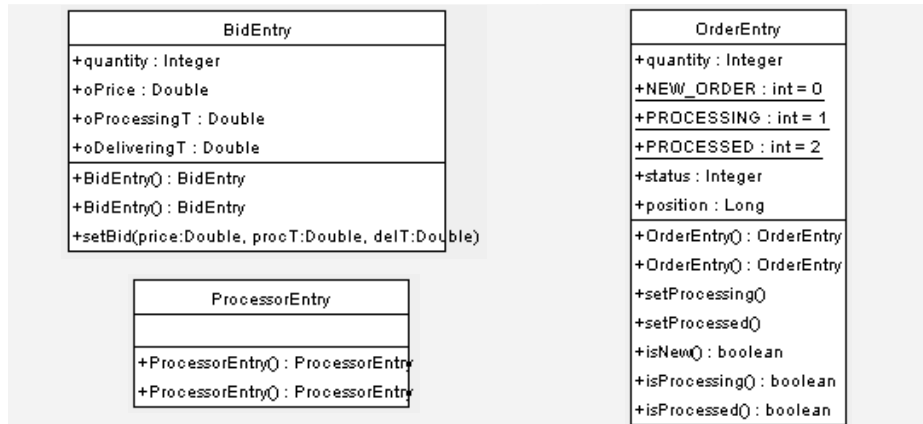


Figura 73: Exemplo de Utilização de JavaSpaces – Objectos do tipo “Entry”.

Para correr qualquer sistema JavaSpaces é necessário correr processos desse ambiente para fornecer serviços de acesso ao espaço partilhado. Em anexo encontra-se uma imagem de écran com este exemplo a correr, sendo visíveis cinco janelas de linha de comando. As janelas superiores correspondem a processos do ambiente JavaSpaces que fornecem serviços necessários. A janela inferior esquerda apresenta um processo de um cliente, objecto “Customer”, a criar encomendas, a colocá-la no espaço partilhado e a esperar por ofertas. A janela inferior direita apresenta um processo de um recurso, objecto “Processor”, a ler encomendas, a criar ofertas e a colocá-las no espaço partilhado.

## **2.3 Descrição Genérica do Modelo**

Nesta secção apresenta-se a descrição genérica do sistema multiagente distribuído de produção baseado no modelo MSDP apresentado nos capítulos III - Modelo do Sistema Distribuído de Produção e IV - Produção Simultânea. Todos os elementos de um Sistema Distribuído de Produção são representados por agentes, que comunicam com, e em nome de, cada elemento de uma forma automática ou interactiva. Os objectos de produção (produtos ou componentes) serão representados por objectos de informação utilizados por diversos componentes do sistema. Os recursos de produção, isto é, processadores do sistema de produção, delegam a sua representação em Agentes. Os clientes, utilizadores directos ou indirectos dos processadores, também serão representados por agentes. Utilizam-se Agentes Encomenda para gestão do sistema, que assume a responsabilidade de coordenação do sistema, através da utilização de estratégias de gestão definidas pelos clientes.

Neste modelo considera-se que os Agentes Recurso conhecem as aptidões dos recursos e a sua capacidade de processamento de execução de transformações de entrega de produtos conhecidos. Este conhecimento pode aumentar por acção dos utilizadores dos agentes. Conhecendo as transformações que os recursos podem executar, têm a capacidade de tentar obter encomendas para esse recurso.

O modelo estrutural do sistema multiagente distribuído de produção está ilustrado na Figura 74. Os agentes podem estar distribuídos por vários locais e comunicar através da rede pelo envio de mensagens para um repositório partilhado. Esta forma de comunicação é idêntica a um quadro negro, que é um sistema utilizado na inteligência artificial distribuída e que foi referido na secção VI - 1.3.2. Todos os agentes do sistema têm acesso a esse repositório e podem assim monitorizar a actividade do sistema. Embora não seja, nesta fase, explorada a mobilidade dos agentes, esta é uma possibilidade que pode trazer vantagens de utilização de recursos de processamento de informação mais libertos, ou de utilização de outros recursos de processamento de informação com outras aptidões noutros locais. Por exemplo, poderá ser importante a deslocação de um agente para uma máquina com um determinado software que lhe permita obter nova informação ou executar algoritmos complexos.

### **2.3.1. Agente Cliente - Descrição Genérica**

Cada cliente tem um Agente Cliente que o representa no sistema e que lhe permite introduzir encomendas no sistema. A introdução de uma encomenda, através da criação de um Agente Encomenda, será o evento iniciador da formação de cada sistema de produção. Assim sendo, o Agente Cliente cria um Agente Encomenda para cada encomenda de um único produto com quantidades diversas.

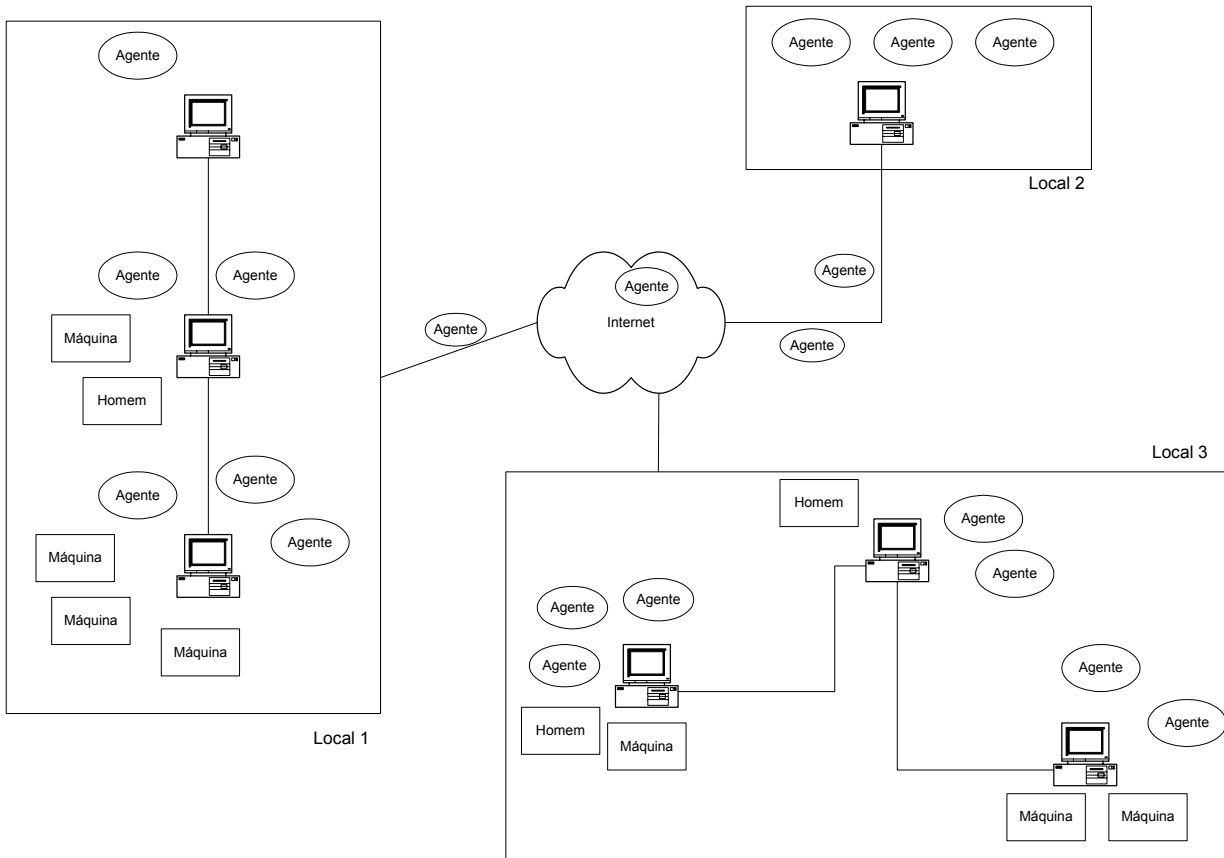


Figura 74: Modelo Estrutural do Sistema na Rede.

### **2.3.2. Agente Encomenda - Descrição Genérica**

O Agente Encomenda irá negociar / requisitar o produto (componente) pretendido pelo cliente. O Agente Encomenda irá pesquisar a rede à procura de recursos que possam executar os processos de produção necessários. Esta pesquisa poderia ser realizada por Agentes de Pesquisa de Informação, que percorrem a rede à procura de informação publicada por diferentes recursos, tentando encontrar informação sobre operações e/ou processos relevantes. Numa arquitectura federada podem utilizar-se agentes de ligação que teriam a responsabilidade de encontrar os recursos apropriados. Neste modelo utiliza-se um repositório de informação partilhado, onde se publica o interesse na obtenção de um determinado produto, e no qual se vai recolher informação sobre os meios que podem executar esse pedido.

### **2.3.3. Agente Recurso - Descrição Genérica**

Os Agentes Recurso representam os recursos produtivos disponíveis, com o objectivo de conquistarem quantidades de trabalho, para utilização desses recursos. Quando algum Agente Encomenda coloca informação sobre uma encomenda no espaço partilhado, o Agente Recurso pode responder colocando uma oferta para essa encomenda no espaço. Esta oferta depende da análise ao pedido e de uma tomada de decisão, baseada na agenda, capacidade e aptidões do recurso que representa. Esta oferta deverá incluir toda a informação do recurso necessária para avaliação da oferta. A informação necessária depende da estratégia utilizada no sistema para selecção de recursos. Uma das funções do agente, é a de manter a sua agenda actualizada em função do recurso real e das encomendas seleccionadas.

### **2.3.4. Coordenação dos Agentes**

Não foi até este momento indicada a forma de decisão da selecção de recursos para cada encomenda. Há, no entanto, um conjunto de decisões relacionadas com a definição de SDP, com o modelo MSDP, com a representação formal do MSDP e com as estratégias de gestão da produção equacionadas pelo modelo, que colocam diversas balizas orientadoras da coordenação dos agentes. Assim sendo, podem estabelecer-se duas formas fundamentais de coordenação e controlo, directamente relacionados com os modelos, local e global, de afectação de recursos, introduzidos na secção III - 1.2.3 e formalizados nas secções IV - 3.1 e IV - 3.2:

- Controlo Global - o Agente Encomenda conhece as aptidões e a agenda de cada recurso, e selecciona recursos para todas as transformações associadas à satisfação da encomenda *completa*, podendo entregar a execução dessas transformações em diferentes quantidades, a diferentes recursos.
- Controlo Local - o sistema de produção resulta da criação de sucessivas encomendas colocadas na rede, relativas às várias transformações associadas ao sistema de produção que satisfaz a encomenda global. Nesta forma de controlo efectua-se a selecção de recursos para cada uma dessas encomendas *parciais*, satisfazendo-se a encomenda completa por agrupamento destas soluções.

Em qualquer dos casos, a encomenda pode ser executada toda pelo mesmo Agente Recurso ou por diversos agentes, dependendo das vantagens do paralelismo de processamento resultante da Produção Simultânea, considerando restrições de número, capacidade e agenda dos recursos.

Um modelo de informação do processamento de encomendas pode ser extremamente detalhado conforme se pode verificar pela especificação de Scheer (1999), com apresentação detalhada de funções, fluxo de objectos, fluxo de controlo, fluxo de materiais, utilização e responsabilidade de funções, objectivos, e elementos organizacionais. Neste caso apenas se especifica a criação da encomenda, com publicação e selecção de recursos, deixando totalmente encapsulada, pelo Agente Recurso, a forma de processamento da encomenda. Interessa, posteriormente, o controlo do estado desse processamento. Há um conjunto de acções que não são tratadas neste ponto por não serem relevantes para o modelo em causa, nomeadamente acções de controlo de qualidade, gestão administrativa, planeamento interno do recurso.

## **2.4 Especificação do Modelo**

Um dos objectivos da apresentação deste modelo é o de criar um projecto de um sistema multiagente que corresponda ao modelo do Sistema Distribuído de Produção apresentado. Este projecto terá que estar de acordo com os requisitos e com a descrição genérica apresentados, e mostrar o relacionamento efectivo entre o modelo do Sistema Distribuído de Produção e o modelo do sistema baseado em agentes.

Apresenta-se a seguir um modelo conceptual da informação utilizada pelos diferentes agentes distribuídos. A infra-estrutura de armazenamento e comunicação dos agentes é fornecida por espaços do sistema JavaSpaces, através do qual se enviam objectos de informação.

### **2.4.1. Modelo Lógico Estático**

Os agentes têm um modelo lógico estático idêntico do seu ambiente, representado na Figura 75. Este modelo corresponde à sua própria informação sobre os agentes que conhece e sobre as suas relações. Os

agentes que conhece serão representado pelos objectos Cliente, Encomenda e Recurso. Existem quatro relações directas entre classes de informação que representam agentes do sistema: Cliente – Encomenda; Oferta; Selecção.

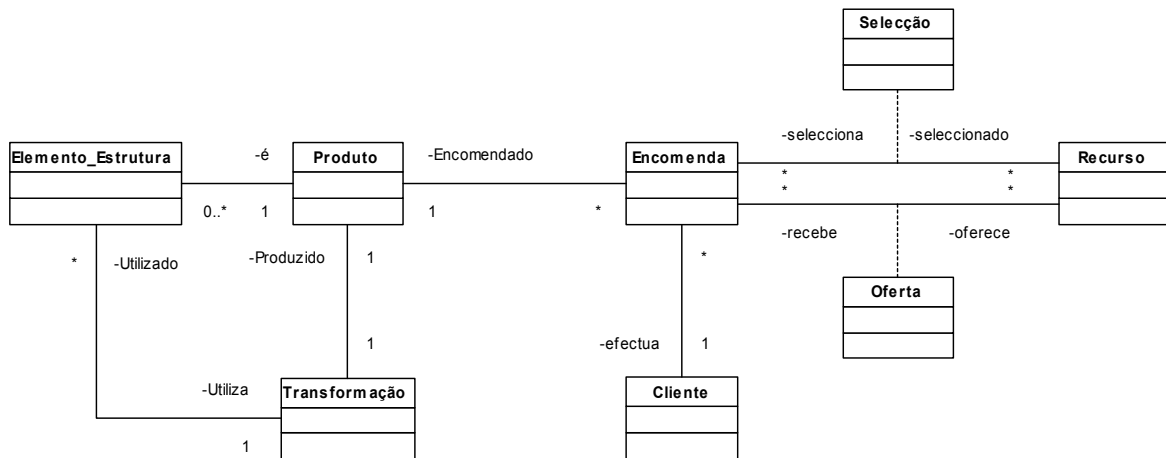


Figura 75: Modelo Lógico Estático da Informação de Agentes Recurso.

O cliente é caracterizado por um conjunto de propriedades, entre as quais está a sua identificação real e a sua identificação no sistema de agentes. Na relação entre clientes e encomendas, um cliente pode efectuar várias encomendas, sendo cada uma gerida por um Agente Encomenda até à sua conclusão.

A classe Produto representa a informação relativa ao produto. Um produto resulta da execução de uma transformação. Esta transformação pode utilizar vários elementos de estrutura para produzir aquele produto. Um produto pode pertencer a muitas estruturas do produto, isto é, pode ser elemento de muitas estruturas. Esta forma de representação da estrutura do produto está de acordo com o conceito de produto do modelo MSDP. Numa determinada Rede de Petri, associada a um sistema criado para a satisfação de uma encomenda, cada objecto  $o_i$  da rede corresponde a um Produto. De acordo com o modelo MSDP, a própria estrutura do produto representa o processo de produção, isto é, as transformações para mudar objectos de um estado para outro.

As encomendas são representadas pela classe Encomenda, que é o elemento estruturante do Sistema Distribuído de Produção. Um Sistema Distribuído de Produção, de acordo com o MSDP, é criado para cada encomenda de um único produto final. Esta encomenda está associada a uma transformação de produção do produto pretendido. Esta transformação corresponde a uma das transformações  $t_j$  da Rede de Petri do Sistema Distribuído de Produção. O Agente Encomenda é responsável pela atribuição de índices aos objectos e às transformações, cuja informação utiliza na selecção de recursos.

A classe Recurso é utilizada para representar os Agentes Recurso, e a informação necessária para aplicação do modelo do Sistema Distribuído de Produção. Este agente tem informação sobre a sua capacidade unitária  $c_{jk}$  do recurso  $k$  para executar a transformação  $j$ , conforme descrito na secção VI - 2.3. Dependendo da estratégia de selecção de recursos utilizada, pode ser necessário conhecer as aptidões, os tempos unitários de execução de cada transformação e a agenda dos recursos.



Os recursos podem efectuar várias ofertas para várias encomendas e estas podem receber ofertas de diversos recursos. Estas ofertas contêm informação sobre a quantidade de trabalho  $w_{jk}$ , e sobre os instantes de início,  $T_{jk}^-$ , e conclusão,  $T_{jk}^+$ , de transformações executadas pelos recursos.

A encomenda aplica estratégias de selecção de recursos, requisitando em seguida a execução de uma transformação  $j$  numa determinada quantidade de trabalho  $w_{jk}$  a alguns dos  $r$  recursos. Um recurso pode ser seleccionado por diversas encomendas.

### **2.4.2. Comunicação / Coordenação**

A comunicação entre agentes é efectuada através da inserção de objectos de comunicação no espaço partilhado fornecido pelos mecanismos dos JavaSpaces. Esses objectos permitem publicar informação sobre encomendas, ofertas de satisfação de encomendas, pedidos de agenda para encomendas, agendas de recursos, requisição de trabalho, aceitação de trabalho e estado dos trabalhos.

Aos objectos das classes Oferta e Selecção, representados na Figura 75, correspondem objectos que são colocados no espaço partilhado, JavaSpace, para comunicação entre agentes. Para poderem ser colocados no espaço partilhado, estes objectos têm que ser instâncias de classes herdeiras da classe "Entry" do ambiente JavaSpaces. Ao serem colocados no espaço JavaSpace podem ser lidos por qualquer objecto.

O processo de procura de ofertas, e selecção de recursos descrito no modelo local de afectação de recursos do modelo MSDP, é um processo de coordenação entre agentes similar ao mecanismo de coordenação de Redes de Contrato<sup>57</sup>. Segundo Rich and Knight (1991), nesta forma de coordenação um agente decompõe o problema e depois negocia com os outros agentes a atribuição das subtarefas. Nesta rede de contrato os agentes podem desempenhar duas funções: Gestor, que decompõe o problema, procura contratantes para resolverem partes do seu problema, e supervisiona a execução; Contratante, que executa a subtarefa, realizando o trabalho, ou que se torna gestor subcontratando partes do seu trabalho a outros contratantes. Existem vários trabalhos de investigação que abordam o problema da negociação entre agentes, para resolver problemas de programação de produção, através da utilização deste protocolo, nomeadamente: Monostori, Kádár and Hornyák (1998); Rabelo, Camarinha-Matos and Afsarmanesh (1998); Sousa and Ramos (1999).

No modelo baseado em agentes do SDP, o agente representante da encomenda funciona como gestor que anuncia uma transformação e os agentes representantes dos recursos avaliam a transformação e efectuem ofertas que o gestor vai seleccionar.

### **2.4.3. Agente Cliente**

Cada Agente Cliente deve ser capaz de identificar o cliente, armazenar informação sobre produtos, sobre encomendas, sobre recursos candidatos e sobre recursos seleccionados. Esta informação está representada no modelo lógico da Figura 75.

O Agente Cliente pode efectuar várias encomendas, e cada uma, pode apenas incluir um dos objectos  $o_i$  de produção, correspondente a um produto que é pretendido em quantidades diversas. O produto

---

<sup>57</sup> "Contract Net".

encomendado resulta da execução de uma das transformações  $t_j$  do modelo de Sistemas Distribuídos de Produção que podem ser executados por vários recursos, se estes tiverem essas aptidões.

Esta informação não é definitiva e vai sendo actualizada no funcionamento do próprio sistema multiagente. Quando um componente é fabricado por um recurso de produção que ainda não estava na base de dados, então actualiza-se a mesma para conter mais um registo.

Cada encomenda efectuada vai ser realizada por um agente criado para esse efeito. O Agente Encomenda criado com este objectivo vai ter o tempo de vida coincidente com o tempo de vida da encomenda, porque vai ser responsável pela selecção de ofertas, isto é, de recursos, pela atribuição de uma quantidade de trabalho a esses recursos, e pela monitorização da conclusão dessa encomenda.

Um Agente Cliente cria uma encomenda para um produto pretendido e lança um Agente Encomenda para a gestão dessa encomenda, conforme está representado pela modelo lógico da actividade de lançamento de encomenda representada na Figura 76.

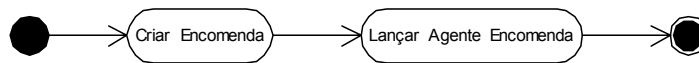


Figura 76: Modelo Lógico da Actividade de Lançamento de Encomenda do Agente Cliente.

O sucesso de uma encomenda, isto é, da sua entrega com qualidade, tem que se reflectir na gestão da informação realizada pelo cliente, que deve actualizar a sua base de dados. O insucesso de uma encomenda gera um conflito interno que o Agente tem que resolver, abdicando da encomenda ou gerando nova encomenda. A primeira destas soluções tem implicações sobre o utilizador deste agente, que deve ser o responsável da decisão, por inquérito ou por delegação. A decisão de gerar nova encomenda leva novamente à execução da actividade de lançamento de encomenda.

As outras actividades associadas a clientes não são aqui especificadas por não serem alvo de análise neste estudo. Directamente relacionado com a gestão dos seus fornecedores, o cliente tem que ter capacidade de processar o pagamento de encomendas, gerir conflitos, analisar fornecedores.

#### **2.4.4. Agente Encomenda**

Cada Agente Encomenda tem uma única encomenda atribuída que deve satisfazer, composta pelo produto pretendido, em quantidades especificadas pelo cliente. Para gerir a entrega da encomenda ao seu cliente, o agente tem que procurar fornecedores e seleccionar, pelo menos um deles, para a execução da transformação cujo resultado é o produto pretendido. O cliente, ao criar o Agente Encomenda, vai preencher alguma informação representada no modelo lógico da Figura 75, nomeadamente sobre a sua identificação, os dados da encomenda, o produto e a respectiva transformação. O cliente deve definir a estratégia para selecção de recursos e a estratégia a seguir no caso do insucesso de algumas das actividades do Agente Encomenda. Os dados relativos aos processos de produção e recursos candidatos são geridos por este agente.

De acordo com o modelo MSDP, a afectação de recursos pode ser local ou global. A selecção de recursos global foi especificada formalmente na secção IV - 3.2 e depende do conhecimento das agendas dos recursos candidatos à satisfação de qualquer uma das transformações de uma encomenda. A selecção de recursos local, especificada formalmente na secção IV - 3.1, depende das ofertas efectuadas por todos os recursos a

encomendas de um único produto. As diferenças fundamentais entre as duas abordagens estão relacionadas com a informação necessária. No modelo global, os recursos interessados têm que publicar a sua agenda, enquanto que, no modelo local têm que publicar uma oferta de execução do trabalho. No primeiro caso, os tempos de execução serão determinados pelo Agente Encomenda e no segundo caso, serão fornecidos pelos Agentes Recurso.

Uma actividade de selecção de recursos é composta por diversas acções, sendo as principais acções: selecção de recursos, monitorização da encomenda e comunicação com o cliente.

No modelo local, a selecção de recursos de produção é efectuada pelo Agente Encomenda (Figura 77), que começa por publicar pedidos de satisfação dos requisitos da encomenda, seguido da recolha de ofertas. Se houver ofertas, terá que passar o controlo para a acção de selecção de recursos local, senão terá que comunicar o facto ao cliente.

Na selecção dos recursos de produção pelo modelo global, este agente tem que começar por recolher informação sobre as agendas dos recursos interessados na encomenda. Se houver respostas, passa o controlo para a acção de selecção de recursos pelo modelo global.

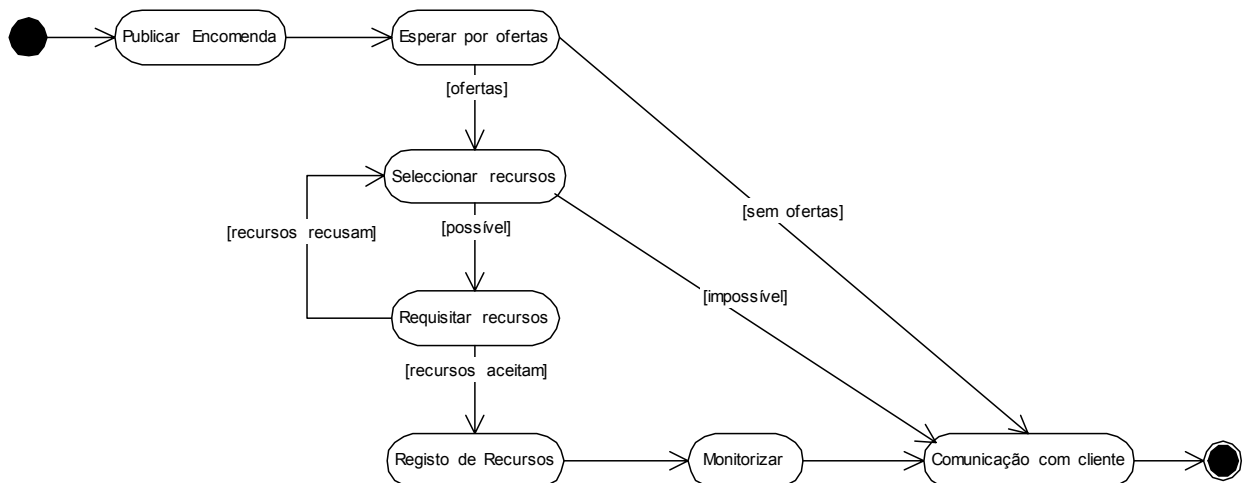


Figura 77: Modelo Lógico de Actividade de Gestão Local de Encomenda de Agentes Encomenda.

O Agente Encomenda, depois de recolher ofertas, utiliza estratégias predefinidas para selecção dos recursos. Pode mesmo comparar estratégias para obtenção de melhores resultados. Se a afectação de recursos for executada com sucesso, executa a actividade de requisição efectiva dos recursos. Se não for possível seleccionar recursos, por incapacidade dos recursos ou pela aplicação da estratégia de selecção, então comunica a falha da encomenda ao cliente e este tem que agir em conformidade com os seus objectivos.

O sucesso da requisição dos recursos depende da aceitação por parte destes da execução da quantidade de trabalho pretendida de determinadas transformações. Se aceitarem, então regista-se este facto e executa-se a monitorização da encomenda. Se os agentes representantes dos recursos recusarem a execução do trabalho, então é necessário seleccionar novos recursos.

Com o intuito de clarificação da relação entre a especificação do modelo multiagentes e o modelo formal de representação e afectação de recursos do SDP, representa-se, na Tabela 41, a relação entre as variáveis

do modelo de optimização, secção IV - 3.2.1.i, e os objectos do modelo lógico estático da Figura 75. Entre as variáveis indicadas, estão os períodos de tempo de trabalho de execução de cada transformação  $j$ , atribuídos a cada recurso  $k$ . Estes períodos são definidos pelos instantes de início,  $T_{jk}^-$ , e conclusão,  $T_{jk}^+$ , desse trabalho. O período de trabalho definido por estes instantes de tempo depende da quantidade de trabalho  $w_{jk}$  atribuída a cada recurso. Um recurso é considerado seleccionado se tiver alguma quantidade de trabalho atribuída, sendo representado por  $wz_{jk} > 0$ . Na Tabela 41 representam-se os objectos que contêm informação sobre as variáveis indicadas. Estes objectos representam, no modelo lógico estático, os objectos de comunicação entre agentes. É preciso notar, no entanto, que aquelas variáveis não são definidas por estes objectos, sendo sempre da responsabilidade dos Agentes Encomenda, ou dos agentes representantes dos recursos.

Na aplicação do modelo global, os agentes representantes dos recursos transmitem informação sobre a agenda de ocupação do recurso utilizador, que é utilizado pelo Agente Encomenda para determinar períodos de trabalho a atribuir a cada recurso.

Na aplicação do modelo local os agentes representantes dos recursos transmitem ofertas de trabalho para períodos definidos por eles com quantidades de trabalho que satisfaçam a encomenda. Esta informação é utilizada pelo Agente Encomenda para seleccionar as ofertas mais vantajosas.

Tabela 41: Agente Encomenda - Relação entre variáveis de optimização e o modelo lógico.

Variáveis dos modelos de selecção de recursos		Objectos do Modelo Lógico	
Descrição	Representação	Seleccção Global	Seleccção Local
V1 – Quantidade de Trabalho	$w_{jk}$	Seleccção	Oferta / Seleccção
V5 – Instante de início de uma transformação executada por um recurso	$T_{jk}^-$	Seleccção	Oferta / Seleccção
V6 – Instante de conclusão de uma transformação executada por um recurso	$T_{jk}^+$	Seleccção	Oferta / Seleccção
V7 – Tempo de execução de uma transformação executada por um recurso	$T_{jk}$	Seleccção	Oferta / Seleccção
V8 – Recurso seleccionado	$wz_{jk}$	Seleccção	Seleccção

### **2.4.5. Agente Recurso**

O Agente Recurso armazena informação sobre o recursos que representa, os processos que pode executar, as encomendas para as quais faz ofertas e sobre aquelas em que foi seleccionado, de acordo com o modelo lógico estático da informação representado na Figura 75.

Na Figura 78 representa-se a actividade de criação de oferta para uma encomenda por parte do Agente Recurso. Esta actividade inicia-se com a recepção de um aviso de encomenda seguida da acção de actualização de agenda, de acordo com a sua informação interna e a informação do recurso real. Em paralelo com esta acção decorre a acção de procura de encomenda no espaço partilhado. Se qualquer uma destas acções falhar, a actividade termina sem criação nem publicação de oferta.

O sucesso de ambas as acções permite analisar a possibilidade de satisfação da encomenda, isto é, analisar a sua aptidão, a sua disponibilidade de capacidade material e temporal e a satisfação dos objectivos definidos. Se a análise for negativa, então termina a actividade. Um Agente Recurso pode ter capacidade livre e interesse na satisfação de uma encomenda, mas pode não ter materiais necessários para a execução da

mesma. Nesse caso terá que criar um Agente Cliente que o represente com o objectivo de criar a encomenda dos materiais necessários. Este mecanismo permite construir o sistema de produção baseado na selecção individual de recursos. Só depois de obter respostas para essa encomenda é que poderá efectuar a aceitação interna desta encomenda.

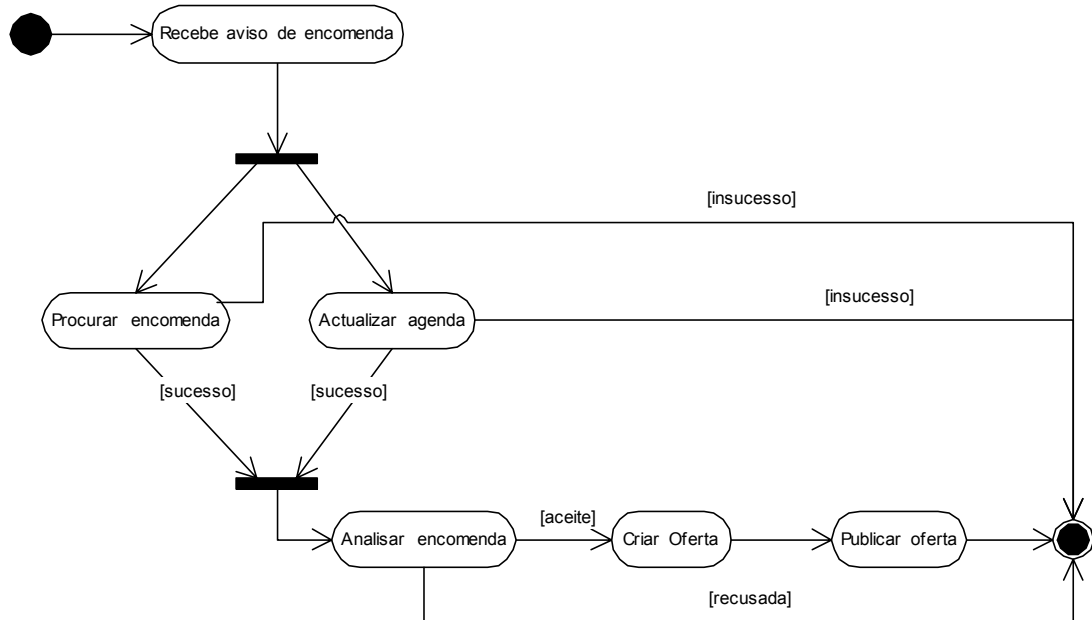


Figura 78: Modelo Lógico de Actividade de “Criação de Oferta” de Agentes Recurso.

A aceitação interna da encomenda conduz à criação de uma oferta e posterior publicação. A oferta deve incluir o instante de entrega e o custo associado.

A aplicação de determinadas estratégias de selecção de recursos implica o conhecimento da agenda dos recursos disponíveis para a satisfação de uma encomenda. Nesse caso, o Agente Recurso processa o pedido de agenda de uma forma similar ao processamento do pedido de encomenda. O diagrama de actividade é estruturalmente idêntico ao da Figura 78, mas as acções executadas são diferentes. Em vez de pedidos de encomendas, existem pedidos de agenda para a realização de encomendas. O agente procura pedidos, actualiza a agenda, analisa esses pedidos e, eventualmente, cria e publica uma oferta.

Se uma oferta publicada pelo Agente Recurso for seleccionada por um Agente Encomenda, então existe uma requisição de processamento de uma determinada quantidade de trabalho pelo recurso. O Agente Recurso executa a actividade representada na Figura 79 para análise do pedido e possível aceitação do mesmo.

Esta actividade inicia-se com a recepção de um aviso de pedido de trabalho, seguindo-se a execução, em paralelo, de acções de procura do pedido e actualização da agenda do recurso. A existência de um pedido baseado numa oferta anterior é analisado em paralelo com a verificação da agenda. O sucesso destas duas actividades permite inquirir o utilizador sobre a aceitação do pedido. Esta consulta ao utilizador pode ser negligenciada se o utilizador assim o quiser. Finalmente, efectua-se o registo e a publicação da aceitação do

trabalho. O registo desta aceitação implica o comprometimento por parte do recurso com a execução do trabalho.

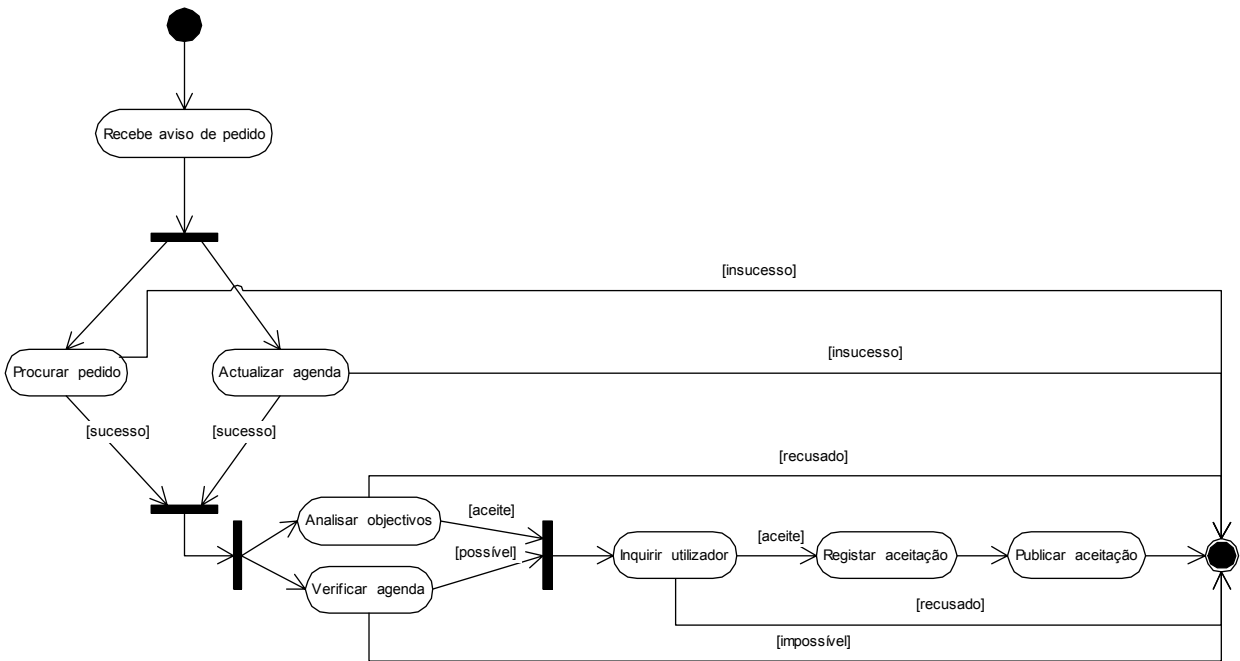


Figura 79: Modelo Lógico de Actividade de “Aceitação de Pedido” de Trabalho.

### 3. CONSIDERAÇÕES FINAIS

A utilização de tecnologias adequadas ao desenvolvimento de um sistema pode ditar o seu sucesso. Os Sistemas Distribuídos de Produção, baseados em elementos de processamento distribuídos, autónomos e com a possibilidade de alterarem a ligação entre eles, são sistemas abertos, com propriedades de distribuição. Por estas razões, e de acordo com um conjunto de autores da área dos agentes, defende-se a utilização da tecnologia de agentes de software para representação do modelo MSDP. Esta representação pode ser utilizado para implementar um SDP computacional, que por sua vez pode ser utilizado para a realização de estudos do sistema em ambiente dinâmico e em ambiente industrial.

Neste capítulo, apresenta-se um projecto parcial de um modelo MSDP baseado em Agentes. A análise de algumas ferramentas levou à consideração do UML, como linguagem de especificação do sistema, e o ambiente JavaSpaces, para desenvolvimento de agentes e do mecanismo de comunicação. Algum trabalho de desenvolvimento permitiu construir um pequeno exemplo demonstrativo da possibilidade de aplicação destas ferramentas.

Uma parte do projecto é representado por um modelo lógico estático da informação utilizada por todos os agentes, isto é, um modelo da informação em que todos os agentes se vão basear. Essa informação resulta do projecto do sistema e da informação adquirida com o funcionamento do mesmo. O mapeamento entre parte da informação utilizada nos modelos formais do Sistema Distribuído de Produção e o modelo de informação deste sistema permite mostrar e clarificar a ligação entre os dois sistemas. Descreve-se ainda a forma de comunicação por envio de objectos para um espaço partilhado e a forma de coordenação do trabalho por

aplicação do mecanismo de “Rede de Contrato”. Os três agentes fundamentais do sistema, Agente Cliente, Agente Encomenda e Agente Recurso utilizam o mesmo modelo de informação do ambiente que os rodeia, que lhes permite conhecer os conceitos em causa e dessa forma estabelecer comunicação. A especificação parcial apresentada destes agentes é baseada na descrição das principais actividades executadas por cada um deles e que resultam da sua interacção.





## **VII - CONCLUSÃO**

Na conclusão apresentada analisa-se o trabalho desenvolvido e os resultados obtidos à luz dos objectivos definidos e apresentam-se sugestões para trabalho futuro.



O trabalho desenvolvido no âmbito desta dissertação, intitulada Sistemas Distribuídos de Produção em Ambiente de Produção Simultânea, teve como motivação principal a abordagem a conceitos de sistemas de produção que permitissem responder a condições de mercado progressivamente mais exigentes. Estas condições estão relacionadas com o aumento da personalização de produtos com ciclos de vida cada vez mais reduzidos e com a necessidade dos sistemas de produção se adaptarem rapidamente a alterações do ambiente em que se inserem, i.e. da procura e do mercado de recursos de produção.

## **1. CONCLUSÕES E CONTRIBUIÇÕES DO TRABALHO DESENVOLVIDO**

No âmbito desta dissertação, desenvolveu-se trabalho de investigação, com vista ao cumprimento de um conjunto de objectivos parciais definidos com base num objectivo genérico. Foi desenvolvido um modelo de Sistemas Distribuídos de Produção que permite projectar o sistema, efectuar análises sobre a coordenação dos elementos do sistema e avaliar o seu desempenho em ambiente de Produção Simultânea.

### **1.1 Definição de Sistema Distribuído de Produção**

Defende-se a ideia de que a definição de Sistema Distribuído de Produção não se deve apenas basear na extensão de conceitos tradicionais sobre sistemas de produção, com a noção de distribuição dos elementos constituintes do sistema. As transformações de mercado, conduzem à exploração de sistemas de produção com características adicionais à distribuição dos elementos constituintes do sistema, nomeadamente, uma crescente autonomia, capacidade de auto-organização e modelação dos sistemas por replicação de estruturas similares. Esta noção sobre sistemas distribuídos de produção é introduzida por trabalhos anteriores de estudo de novos paradigmas de organização dos sistemas de produção, como os Sistemas de Produção Biónicos, Holónicos e Fractais. Estes paradigmas servem de motivação ao desenvolvimento de grande parte deste trabalho.

Na secção II - 3.4 apresenta-se uma definição de Sistema Distribuído de Produção que serve de base ao desenvolvimento de um modelo de Sistemas Distribuídos de Produção. Esta definição baseia-se em conceitos de sistemas, sistemas distribuídos de informação heterogéneos e sistemas de produção, no contexto dos novos paradigmas de organização da produção referidos. A definição apresentada caracteriza-se por três noções fundamentais, a de autonomia dos elementos do sistema, distribuição desses mesmos elementos e capacidade do sistema se configurar e reconfigurar para dar resposta a alterações do ambiente.

A distinção entre Sistemas Distribuídos de Produção e não distribuídos faz-se na abordagem explícita aos conceitos de autonomia, distribuição e reconfiguração, no projecto e gestão do sistema. A *autonomia* dos elementos do Sistema Distribuído de Produção baseia-se na modelação dos elementos, considerando a capacidade de *satisfação total* dos requisitos de um trabalho que o elemento assumiu, com *encapsulamento da forma de execução*. A *distribuição* dos elementos do sistema, por natureza distribuído, deve ser considerada no modelo do sistema. Um modelo MSDP deve permitir *ligar os elementos*, fornecendo meios de comunicação, coordenação e integração. Devem existir mecanismos que permitam interligar recursos de produção distribuídos que devem comunicar de uma forma integrada, com base em conhecimento uniforme sobre o ambiente em que se inserem. A *reconfiguração* de sistemas distribuídos de produção pode ser

repetidamente executada em horizontes temporais curtos, até ao *limite de reconfiguração instantânea* do sistema.

## **1.2 Modelo do Sistema Distribuído de Produção**

Neste trabalho propõe-se um modelo de Sistemas Distribuídos de Produção que satisfaz a definição apresentada. Este modelo deve permitir a aplicação da filosofia de Produção Simultânea, que tem como premissa a satisfação de uma encomenda de cada vez. Assim sendo, o modelo é baseado no projecto de Sistemas Distribuídos de Produção, que são criados tantas vezes quantas as encomendas. Assume-se ainda, que *cada encomenda se refere a apenas um dado produto*. Por criação ou projecto do sistema entende-se a geração de uma configuração genérica e posterior configuração específica. A configuração genérica associa, através da estrutura do produto, os produtos com as transformações do sistema, i.e. com os processos de produção representados por essas transformações. A configuração específica corresponde à configuração genérica do sistema associada à representação dos recursos de produção que executam cada transformação.

O produto encomendado executa-se através de um conjunto de processos produtivos que transformam ou convertem matérias primas no produto final. Cada *passo de transformação do produto* pode ser executado por recursos de produção diferentes, podendo ser seleccionados entre recursos alternativos. Cada um destes *passos* pode ser visto como o processo de satisfação de uma encomenda. De facto, é possível modelar o Sistema Distribuído de Produção como um conjunto de sistemas dentro de sistemas, isto é, sistemas que agregam outros sistemas, cada um deles satisfazendo uma encomenda que pode ser componente de outra encomenda, estando cada uma associada a uma *parte* do produto final.

Cada transformação é da responsabilidade de uma célula autónoma de produção. Esta célula é composta por um ou mais recursos autónomos com a responsabilidade de execução daquela transformação, isto é, com a responsabilidade de entrega do produto que resulta da transformação.

Cada *passo de alteração do produto* resulta de uma *transformação de estado do produto*, baseada numa estrutura do produto final. Neste modelo todos os elementos da estrutura são produtos que resultam da execução de uma transformação que altera o produto. Esta alteração pode ser resultado de um serviço, quer seja auxiliar, como um transporte, quer seja principal, como uma transformação física. Considera-se que os serviços prestados resultam num novo estado do produto, por exemplo, produto movimentado, testado ou transformado. Esta forma de modelação do sistema depende da modelação da estrutura do produto e das fronteiras que se pretendam impor.

O Sistema Distribuído de Produção baseia-se em células virtuais autónomas que existem no âmbito de cada sistema e que têm a responsabilidade de satisfação da encomenda de um único produto. É esta modelação de produtos compostos por outros produtos e da correspondência com células compostas por outras células, baseada na estrutura do produto, que permite criar um mecanismo de reconfiguração do sistema. Se existir alguma alteração do ambiente que impeça a satisfação de uma das encomendas associadas a uma das transformações do produto, então pela repetição da encomenda é possível seleccionar novos recursos e reconfigurar o sistema. Além disso, os mesmos recursos físicos podem estar associados a

diferentes células de produção, fazendo assim parte de diversas configurações de sistemas de produção, que se estão continuamente a reconfigurar para satisfazer novas encomendas.

O desenvolvimento de modelos matemáticos para geração de configurações e para o estudo do desempenho do sistema, a funcionar de acordo com princípios de Produção Simultânea, requer a representação formal do referido modelo. Esta representação formal permite, ainda, a especificação do sistema de uma forma precisa e não ambígua. O modelo formal MSDP, isto é, a formalização dos conceitos do modelo MSDP referidos atrás, é baseado na representação de sistemas de produção através de Redes de Petri e no relacionamento entre sistemas agregados. A agregação pode simplificar a gestão do sistema por divisão do problema e conseqüente redução da complexidade. Esta agregação baseia-se em produtos dentro de produtos e células dentro de células, reflectindo um carácter fractal do sistema, possibilitando a agregação de informação e dos processos de gestão do sistema. A determinação de necessidades de materiais e de quantidade de trabalho pode ser executada em qualquer nível de agregação, com menor quantidade de dados e menor complexidade de cálculo.

Apresentaram-se diversos exemplos de aplicação do modelo MSDP desenvolvido, e da respectiva representação formal baseada em Redes de Petri. Pode-se concluir que é possível utilizar o modelo MSDP, baseado na estrutura do produto, para representar e projectar um conjunto alargado de casos de sistemas de produção. O modelo também se mostrou útil na execução de funções de gestão de produção, nomeadamente para determinação de quantidades de produtos e de trabalho requeridos para a execução de encomendas. A utilidade do modelo é ilustrada pelos inúmeros exemplos, com configurações simples e mais complexas, que foram apresentados em diversos capítulos, e em particular na secção 3 do capítulo III - Modelo do Sistema Distribuído de Produção e em todo o capítulo V - Experimentação do Modelo MSDP. Explorou-se ainda, em alguns exemplos, a representação do sistema por agregação de sistemas, que mostra a possibilidade de representar o sistema desta forma. Esta possibilidade reduz a complexidade de cálculo de necessidade de materiais e de quantidade de trabalho em níveis de agregação superior.

### **1.3 Produção Simultânea**

A formalização do modelo de Sistemas Distribuídos de Produção permite especificar formas de selecção de recursos para cada transformação, isto é, de geração de configurações de sistemas, que recorram a princípios de Produção Simultânea. A Produção Simultânea é um conceito introduzido por Silva, S. C. and Putnik (1995) com o objectivo de aumento da satisfação dos clientes através da redução do prazo de entrega de encomendas, resultante da redução do prazo de fabrico. Neste conceito preconiza-se a construção de um escalonamento dos recursos para a satisfação de uma encomenda de cada vez.

Desenvolveu-se um trabalho de especificação de princípios de utilização simultânea de recursos em função da abordagem adoptada perante as quantidades de cada ordem de produção. Alguns destes princípios foram explorados noutros trabalhos, nomeadamente o princípio de sobreposição dos lotes a produzir por Almeida (2002).

Desenvolveu-se uma medida de simultaneidade de utilização de recursos em cada transformação que permite aferir a simultaneidade do sistema. Esta medida, o Grau de Simultaneidade *GS*, pode ser utilizada para analisar o comportamento dos sistemas de produção em ambiente de Produção Simultânea, e em particular, analisar a variação do prazo de entrega de uma encomenda relativamente ao grau de simultaneidade.

No modelo desenvolvido neste trabalho de investigação, construiu-se a especificação formal do problema de optimização que permite afectar recursos a cada transformação, programando a produção. Nesta especificação explora-se a possibilidade de execução simultânea da transformação por diversos recursos. Considera-se que cada encomenda é constituída por uma determinada quantidade de um produto e que existe a possibilidade de partição da encomenda para exploração do processamento simultâneo. Esta especificação leva em consideração a possibilidade de atribuição da quantidade de trabalho a executar numa transformação por diversos recursos que a executam simultaneamente, considerando restrições de procura, oferta, disponibilidade, agenda dos recursos e execução de transformações em paralelo. A especificação deste problema de programação da produção, permite estender o modelo formal MSDP, para determinação de uma configuração específica do sistema em ambiente de Produção Simultânea.

O modelo formal MSDP em ambiente de Produção Simultânea, foi aplicado para determinação de configurações específicas do sistema em diversos casos de estudo. Esta aplicação permite concluir que é possível aplicar este modelo para projectar SDPs aplicando a filosofia de PS. Esta aplicabilidade permite ainda analisar o comportamento dos sistemas de produção em ambiente de Produção Simultânea.

Apresentou-se um projecto do Sistema Distribuído de Produção baseado em Agentes de Software, com descrição da ligação com o modelo formal do sistema. A especificação deste modelo multiagente é efectuada na linguagem de representação UML, que pode ser desenvolvido no ambiente JavaSpaces. Este ambiente permite implementar um espaço de comunicação partilhado por agentes de software distribuídos na rede e desenvolvidos na linguagem Java. Apresenta-se a especificação do modelo lógico estático de informação utilizado por todos os agentes para representação do ambiente em que se inserem e modelos de actividade dos agentes de acordo com o modelo MSDP.

A coordenação da actividade de sistemas de produção requer formas de monitorização de eventos que exijam coordenação e controlo, nomeadamente avarias de recursos ou incumprimento de datas de entrega estabelecidas. Neste trabalho não se trata directamente este problema, mas fornece-se um modelo MSDP que permite gerar novas configurações de sistema em função dos eventos referidos. Isto não inibe a necessidade de se fazer um estudo próprio da coordenação da actividade produtiva no ambiente distribuído equacionado, naturalmente em trabalho futuro.

#### **1.4 Estudo do Modelo MSDP em ambiente de PS**

O modelo de Sistemas Distribuídos de Produção em ambiente de Produção Simultânea é utilizado para simular e verificar o comportamento do sistema. Este modelo tem duas variantes, o modelo de afectação local e o modelo de afectação global de recursos às transformações do sistema. Foi possível verificar, para ambas

as variantes do modelo que, numa estrutura genérica de um sistema SDP, o instante de conclusão de uma encomenda, denominado neste trabalho por prazo de entrega, diminui com o aumento do grau de simultaneidade, isto é, com o aumento do número de recursos utilizados simultaneamente em cada transformação. A curva de relacionamento entre o instante de conclusão e o grau de simultaneidade tem uma variação do tipo da função  $1/x$ , em que os ganhos na redução do instante de conclusão de cada encomenda são decrescentes com o aumento do grau de simultaneidade. Em particular, na estrutura genérica analisada, verificou-se que o aumento do grau de simultaneidade acima de determinado valor, no caso à volta de 0,43, traz ganhos muito pequenos e decrescentes de uma forma acentuada.

Assumindo modelos de custos de posse e de preparação simplificados, no modelo global de afectação de recursos, foi possível verificar o comportamento da variação destes custos com a variação do grau de simultaneidade. De acordo com estes modelos de custos, o custo de posse tende a diminuir com o aumento do grau de simultaneidade, por acção da diminuição do instante de conclusão da encomenda e o custo de preparação tende a aumentar por acção do maior número de recursos envolvidos, logo maior número de actividades de preparação da produção. Verificou-se, então, que no mesmo sistema genérico em análise, a soma dos custos de preparação e posse toma valores mínimos para um determinado valor de Grau de Simultaneidade. Este valor é da mesma ordem de grandeza do valor acima referido, em que se verifica uma redução de ganhos acentuada.

Se atendermos a que o grau de simultaneidade definido é directamente proporcional ao número de recursos usados pelo sistema de produção em cada transformação, podemos concluir que existe uma tendência de obtenção de um elevado desempenho para um valor baixo do número de recursos alternativos. Este facto diminui, ainda, relativamente à utilização de todos os recursos disponíveis, os custos de preparação e eventualmente de transporte entre recursos necessários.

O modelo de Sistemas Distribuídos de Produção em ambiente de Produção Simultânea é direccionado à satisfação dos requisitos de produção de uma encomenda de um produto de cada vez. É possível aplicar este modelo para satisfação de várias encomendas, desde que se considere a aplicação sucessiva do modelo a cada uma das encomendas. Neste caso, criam-se modelos do sistema para cada encomenda que permitem determinar uma configuração específica para cada uma delas, isto é, permitem resolver o problema de programação da produção para cada uma delas. Para aplicar este modelo assume-se o funcionamento em ambientes flexíveis com custos de preparação da produção desprezáveis face aos custos de processamento. Esta abordagem permite analisar o comportamento do modelo em vários casos e comparar com outros modelos de afectação de recursos e de programação de produção. Em particular, aplicou-se sucessivamente o modelo MSDP em ambiente de PS a casos de estudo apresentados por Almeida (2002). Na comparação com vários métodos de programação da produção verificou-se a obtenção, de uma forma consistente, de melhores resultados quanto ao instante de conclusão do processamento de todas as encomendas, quanto ao somatório dos instantes de conclusão, quanto à utilização dos recursos e quanto à medida de Qualidade de Serviço na Produção. Pela análise destes resultados conclui-se que o modelo MSDP em ambiente de PS, aplicado em vários exemplos, se comporta de forma favorável quanto à redução de instantes de conclusão de várias encomendas. Esta redução permite entregar encomendas mais cedo aos clientes e libertar os recursos para processamento de outras encomendas.

## **2. LIMITAÇÕES DO TRABALHO DESENVOLVIDO E PERSPECTIVAS DE TRABALHO FUTURO**

Analisando o trabalho desenvolvido é possível vislumbrar um conjunto de questões merecedoras de aprofundamento em trabalho futuro. Podem atribuir-se essas questões a várias partes desta dissertação.

### **2.1 Definição de Sistema Distribuído de Produção**

O sistema distribuído de produção é um tema em profundo desenvolvimento internacional, e o seu estudo e análise deve ser aprofundado e continuamente actualizado. Terá interesse verificar a relação existente ou que poderá vir a existir com outros modelos de Sistemas de Produção Biónicos, Holónicos e Fractais, e com modelos de Empresas Virtuais, Redes de Empresas, Sistemas Inteligentes de Produção, Produção Ágil, Produção do tipo “Lean”.

A comparação da definição apresentada com outras definições, poderá abrir novos horizontes de investigação e projectar a evolução do tema. É do maior interesse verificar em que sentido evoluem os sistemas produtivos industriais relativamente à autonomia, distribuição e capacidade de reconfiguração e simultaneidade. Neste trabalho, considera-se que estas características têm um valor relativo, que se poderia / deveria medir. Métricas de cada uma destas características, independentes e/ou interrelacionadas, podem servir para avaliar ou comparar sistemas distribuídos de produção. O Grau de Simultaneidade, adiantado nesta dissertação, pode mesmo servir para medir a distribuição de um sistema, do ponto de vista do número de elementos do sistema que é necessário interligar. A análise de sistemas produtivos em função destas medidas, poderá levar à classificação de tipos de sistemas e ambientes de mercado mais adequados à implementação de Sistemas Distribuídos de Produção.

### **2.2 Modelo do Sistema Distribuído de Produção**

O modelo do Sistema Distribuído de Produção deve ser aplicado a mais casos, académicos ou industriais, de forma a confirmar a sua validade. O modelo aparenta ser suficientemente genérico para se poder aplicar em diversos casos e de diferentes formas. A aplicação do modelo, aparentemente, fornece uma robusta ferramenta de modelação de sistemas de planeamento e controlo de produção, com uma relação estreita entre a configuração do sistema e a estrutura do produto, e com a especificação de níveis de agregação do sistema. Esta modelação por níveis de agregação permite reduzir a quantidade de informação e dessa forma a complexidade de cálculo, e permite criar sistemas de planeamento e controlo de produção com criação de programas de produção, de uma forma dinâmica para cada encomenda. Neste trabalho desenvolveram-se modelos de aplicação da Produção Simultânea que permitem obter programas da produção de uma forma dinâmica para cada encomenda.

Nos sistemas do tipo MRP os prazos de entrega das encomendas são definidos em função dos prazos de entrega (LT) dos materiais. Estes prazos (LT) são definidos pela experiência e pelo conhecimento da carga média agregada do sistema e do seu relacionamento com a capacidade de produção. Estes prazos LT não são definidos em função da carga real do sistema, distribuída no tempo, levando a prazos de entrega que não são precisos. Esta abordagem é normalmente requerida para um planeamento de médio ou longo prazo, devido à



volatilidade ou imprecisão dos dados, permitindo atrasar a programação detalhada da produção para períodos mais próximos do início de produção e portanto, com dados mais precisos.

No modelo desenvolvido não se considera a possibilidade de modelação baseada em produtos com processos de fabrico alternativos ou processos de fabrico divergentes, i.e. com transformações que possam entregar vários objectos diferentes.

A possibilidade de modelar produtos com processos de fabrico alternativos poderia ser resolvida pela replicação de estruturas de produto com diferentes transformações ou redes de transformação. Uma possibilidade de desenvolvimento é a de modelar a alternativa de processos de fabrico numa única estrutura e desenvolver um modelo para o sistema que considere esta possibilidade. Deve notar-se que esta possibilidade é facilmente modelada por uma Rede de Petri, com caminhos alternativos para entregar um mesmo objecto. A existência de caminhos alternativos aumenta o número de transformações e assim a possível complexidade de cálculo. Na configuração genérica do sistema, algumas das transformações não seriam utilizadas.

A possibilidade de modelar produtos com processos de fabrico divergentes pode ser, aparentemente, facilmente considerada no modelo desenvolvido. Pode considerar-se que a transformação associada a esse processo de fabrico entrega vários produtos que podem ser utilizados em outros produtos. A determinação de quantidades de materiais e quantidades de trabalho, e a afectação de recursos, seriam realizados considerando essa única transformação de produção de vários objectos.

Estas duas possibilidades de modelação de produtos, consideradas para desenvolvimento de trabalho futuro, trazem implicações sobre o modelo MSDP e de geração de configurações genéricas. Esta geração de configurações baseia-se na existência de uma única célula, directamente associada a uma transformação de entrega de um produto encomendado. Seria portanto necessário, realizar um estudo adicional de alteração do modelo para considerar estas possibilidades de modelação.

### **2.3 Produção Simultânea**

Desenvolveram-se modelos locais e globais de afectação de recursos a transformações, baseados no princípio de simultaneidade de partição com separação da encomenda, i.e. de um lote de uma encomenda em lotes independentes (IV - 1.4), existindo a necessidade de desenvolver modelos para outros princípios. A utilização híbrida de mais do que um princípio de simultaneidade poderá mesmo fornecer soluções de afectação de recursos com uma superior utilização dos recursos e com menores instantes de conclusão. Uma das vias possíveis de trabalho futuro decorre da possibilidade de exploração destas opções de investigação.

A utilização deste modelo, conforme se verificou nos testes realizados, provoca um aumento do número de lotes de produção associados a uma mesma encomenda. Estes lotes são processados em períodos de tempo e recursos totalmente diferentes, o que aumenta a necessidade de coordenação dos trabalhos executados por cada recurso sobre cada lote. É necessário desenvolver mecanismos adequados para manter a unicidade de cada encomenda, nomeadamente, de transporte entre recursos, identificação dos lotes, monitorização e controlo dos trabalhos.

## **2.4 Sistema Multiagentes**

Uma outra via de desenvolvimento de trabalho, decorre da possibilidade de implementação do projecto do Sistema Distribuído de Produção baseado em agentes de software, apresentado nesta dissertação. Esta possibilidade permitiria efectuar uma análise ao comportamento dinâmico do sistema de produção, avaliando por exemplo, a capacidade de configuração de sistemas, o período de tempo entre instantes de conclusão das encomendas, a reconfigurabilidade do sistema, face a variáveis como o número de Agentes Recurso alternativos, autonomia dos agentes, carga dos recursos e tempos de preparação. O projecto e desenvolvimento de um sistema baseado em Agentes Inteligentes permitiria comparar as duas alternativas, por exemplo relativamente a tempos de resposta, prazos de entrega, custos, cooperação entre agentes, autonomia e reconfigurabilidade.

## **REFERÊNCIAS BIBLIOGRÁFICAS**

Apresentação da listagem das referências bibliográficas consultadas para a realização deste trabalho. Os inúmeros autores aqui referidos desenvolveram trabalho que se considerou relevante no âmbito dos Sistemas Distribuídos de Produção em Ambiente de Produção Simultânea.



- Adiga, S., Ed. (1997). *Object-oriented Software for Manufacturing System*, Chapman & Hall.
- Agrawal, A. and Graves, R. J. (1999). "A distributed systems model for estimation of printed circuit board fabrication costs." *Production Planning & Control* **10**(7): 650-658.
- Aguiar, M. W., Murgatroyd, I. S. and Edwards, J. M. (1996). "Object-Oriented resource Models: their Role in Specifying Components of Integrated Manufacturing Systems." *Computer Integrated Manufacturing Systems* **9**(1): 33-48.
- Almeida, A. (2002). Análise e Desenvolvimento de Mecanismos e Algoritmos de Apoio ao Escalonamento Orientado ao Produto. Dissertação para obtenção do Grau de Doutor em Engenharia de Produção (in portuguese). Escola de Engenharia, Universidade do Minho.
- Al-Mubarak, F., Khumawala, B. M. and Canel, C. (2003). "Focused cellular manufacturing: an alternative to cellular manufacturing." *International Journal of Operations & Production Management* **23**(3): 277-299.
- Alves, A., Lima, R. M. and Silva, S. C. (2003). "Sistemas de Produção Orientados ao Produto: Integrando Células e Pessoas." *Revista de Inovação Organizacional* **1**(1).
- Archimède, B., Charbonnaud, P. and Mercier, N. (2003). "Robustness evaluation of multisite distributed schedule with perturbed virtual jobshops." *Production Planning & Control* **14**(1): 55-67.
- Athey, T. H. (1982). *Systematic System Approach: An Integrated Method for Solving Systems Problems*, Prentice Hall.
- Azevedo, A. L. (1999). Apoio à Decisão na Negociação de Encomendas em Redes de Empresas. Dissertação para obtenção do Grau de Doutor em Engenharia Electrotécnica e de Computadores (in portuguese). Faculdade de Engenharia, Universidade do Porto.
- Baker, A. D. (1998). "A Survey of Factory Control Algorithms That Can Be Implemented in a Multi-Agent Hierarchy: Dispatching, Scheduling, and Pull." *Journal of Manufacturing Systems* **17**(4): 297-320.
- Baker, A. D., Parunak, H. V. D. and Erol, K. (1999). "Agents and the Internet: Infrastructure for Mass Customization." *Ieee Internet Computing*: 62-69.
- Barata, J., Camarinha-Matos, L. M., Boissier, R., Leitão, P., Restivo, F. and Raddadi, M. (2001). Integrated and Distributed Manufacturing, a Multi-agent Perspective, in *3rd Workshop on European Scientific and Industrial Collaboration*, Netherlands.
- Bishop, P. and Warren, N. (2003). *JavaSpaces in Practice*, Addison-Wesley.
- Bourrières, J. P. (1998). Multilevel algebraic modelling of complex manufacturing tasks, in *IFAC IMS'98*, Porto Alegre, Brasil, 261-265.
- Bradshaw, J. M. (1997). An Introduction to Software Agents, in *Software Agents (J. M. Bradshaw)*, The MIT Press: 3-46.
- Brun, A. and Portioli, A. (1999). "Agent-based shop-floor scheduling of multistage systems." *Computers & Industrial Engineering* **37**: 457-460.
- Butala, P. and Sluga, A. (2002). "Dynamic structuring of distributed manufacturing systems." *Advanced Engineering Informatics* **16**(2): 127-133.
- Camarinha-Matos, L. M., Afsarmanesh, H., Garita, C. and Lima, C. (1998). "Towards an Architecture for Virtual Enterprise." *Journal of Intelligent Manufacturing* **9**(2): 189-199.
- Camarinha-Matos, L. M., Afsarmanesh, H. and Osório, A. L. (2001). "Flexibility and safety in a web-based infrastructure for virtual enterprises." *International Journal of Computer Integrated Manufacturing* **14**(1): 66-82.
- Carmichael, A. (1994). Objects: Changing Software Production and Procurement, in *Object Development Methods (A. Carmichael)*, SIGS Books: 1-12.
- Carvalho, J. D. A., Putnik, G. D. and Cunha, M. (2003). "Infrastructures for Virtual Enterprises." *Cadernos do Departamento de Produção e Sistemas, Escola de Engenharia, Universidade do Minho (disponível em <http://www.dps.uminho.pt/cad-dps>)* **DPS-03**(4).
- Coulouris, G., Dollimore, J. and Kindberg, T. (2001). *Distributed Systems: Concepts and Design*, Third Edition. Addison-Wesley.
- Cranefield, S., Hausteiny, S. and Purvis, M. (2001). UML-Based Ontology Modelling for Software Agents, in *Proc. of Ontologies in Agent Systems Workshop, Agents 2001*, Montreal, 21-28.
- Crowe, T. J. and Stahlman, E. J. (1995). "A Proposed Structure for Distributed Shopfloor Control." *Integrated Manufacturing Systems* **6**(6): 31-36.
- Davidrajuh, R. and Deng, Z. (2000). "An autonomous data collection system for virtual manufacturing systems." *International Journal of Agile Management Systems* **2**(1): 7-15.
- Franklin, S. and Graesser, A. (1997). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, in *Intelligent agents III : agent theories, architectures, and languages (J. P. Müller, M. Wooldridge and N. R. Jennings)*, Springer-Verlag: 21-35.
- Fujii, S., Kaihara, T. and Morita, H. (2000). "A distributed virtual factory in agile manufacturing environment." *International Journal of Production Research* **38**(17): 4113-4128.

- Gale, T. and Eldred, J. (1996). *Getting Results with the Object-Oriented Enterprise Model*, Sigs Books.
- Gertsen, F. (1991). Product Development for Production, in *Advances in Production Management Systems (E. Eloranta)*, Elsevier Science: 87-95.
- Gjerdrum, J., Shah, N. and Papageorgiou, L. G. (2001). "A combined optimization and agent-based approach to supply chain modelling and performance assessment." *Production Planning & Control* **12**(1): 81-88.
- Goldratt, E. M. (1990). *What is this thing called theory of constraints and how should it be implemented?*, The North River Press.
- Goldratt, E. M. and Fox, R. E. (1986). *The Race*, The North River Press.
- Gunasekaran, A. (1998). "Agile manufacturing: enablers and an implementation framework." *International Journal of Production Research* **36**(5): 1223-1247.
- Gunasekaran, A., Mcgaughey, R. and Wolstencroft, V. (2001). Agile Manufacturing: Concepts and Framework, in *Agile Manufacturing: The 21st Century Competitive Strategy (A. Gunasekaran)*, Elsevier Science: 25-51.
- Gupta, J. N. D., Bourne, D. H., Kim, K. H. and Krishnan, S. S. (1998). "Automated process planning for sheet metal bending operations." *Journal of Manufacturing Systems* **17**(5): 338-360.
- Halter, S. L. (2002). *JavaSpaces: Example by Example*, Prentice Hall.
- Herrmann, J. W. and Chincholkar, M. M. (2001). "Reducing Throughput Time during Product Design." *Journal of Manufacturing Systems* **20**(6): 416-428.
- Hillier, F. S. and Lieberman, G. J. (1989). *Introduction to Operations Research*, 4th. McGraw-Hill.
- Hitomi, K. (1996). *Manufacturing Systems Engineering*, 2nd. Taylor & Francis.
- Hoche, O., Broomhead, P. and Grieve, R. J. (1996). "Developing an Object Oriented Framework for the Control of Small Manufacturing Cells." *Computer Integrated Manufacturing Systems* **9**(4): 217-229.
- Hoche, O., Broomhead, P., Grieve, R. J. and Chestney, J. A. (1996). "Application of an Object-Oriented Framework Methodology to the Control of Manufacturing Cells." *Computer Integrated Manufacturing Systems* **9**(4): 231-243.
- Huhns, M. N. and Singh, M. P. (1998). Agents and Multiagent Systems: Themes, Approaches, and Challenges, in *Readings in Agents (M. N. Huhns and M. P. Singh)*, Morgan Kaufmann: 1-23.
- Jackson, R. B., Liddle, S. W. and Woodfield, S. N. (1999). An Analysis of the Unified Modeling Language: UML Compared with OSM, in *Handbook of Object Technology (S. Zamir)*, CRC Press.
- Jennings, N. R. and Wooldridge, M. (1998). Applications of Intelligent Agents, in *Agent Technology Foundations, Applications and Markets (N. R. Jennings and M. Wooldridge)*, Springer Verlag.
- Kádar, B., Monostori, L. and Szelke, E. (1998). "An Object Oriented Framework for Developing Distributed Manufacturing Architectures." *Journal of Intelligent Manufacturing* **9**(2): 173-179.
- Kim, Y., Choi, Y. and Yoo, S. B. (2001). "Brokering and 3D collaborative viewing of mechanical part models on the Web." *International Journal of Computer Integrated Manufacturing* **14**(1): 28-40.
- Kusiak, A. and Wang, J. (1993). Decomposition in Concurrent Design, in *Concurrent Engineering: Automation, Tools, and Techniques (A. Kusiak)*, John Wiley & Sons.
- Lecompte, T. and Bourrières, J. P. (1999). Calcul de besoins et calcul de charge en production discrète par une technique de pseudo-inversion, in *JDA'99 CNRS / GDR Automatique*, Nancy, France, 325-328.
- Lecompte, T., Deschamps, J. C. and Bourrières, J. P. (2000). "A data model for generalized scheduling for virtual enterprise." *Production Planning & Control* **11**(4): 343-348.
- Leitão, P. and Restivo, F. (2000). A framework for Distributed Manufacturing Applications, in *ASI'2000 International Conference*, Bordeaux, France.
- Leitão, P., Restivo, F. and Putnik, G. D. (2001). A Multi-Agent Based Cell Controller, in *Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation - ETFA 2001*, Antibes - Juan les Pins, France, 463-470.
- Lima, R. M. and Silva, S. C. (1998). Object Oriented Modelling of Product Oriented Manufacturing Systems, in *Intelligent Systems for Manufacturing: Multi-Agent Systems and Virtual Organization (L. M. Camarinha-Matos, H. Afsarmanesh and V. Marik)*. Prague, Czech Republic, Kluwer Academic Publishers: 325-334.
- Lima, R. M., Silva, S. C. and Martins, P. M. (1999a). Sistemas Distribuídos de Produção, in *1º Congresso Luso-Moçambicano de Engenharia (J. F. S. Gomes, A. Matos and C. Afonso)*, Maputo, Moçambique, B13-B20 (in portuguese).
- Lima, R. M., Silva, S. C. and Martins, P. M. (1999b). Sistemas Distribuídos de Produção, in *4ª Jornadas de Engenharia de Produção: A Gestão Industrial na Viragem do Milénio*, Guimarães, Portugal, Departamento de Produção e Sistemas, 20-21.
- Luck, M., Mcburney, P. and Preist, C. (2003). *Agent Technology: Enabling Next Generation Computing - A Roadmap for Agent-Based Computing*, AgentLink.
- Maturana, F., Shen, W. and Norrie, D. H. (1999). "MetaMorph: an adaptive agent-based architecture for intelligent manufacturing." *International Journal of Production Research* **37**(10): 2159-2173.
- Mclean, C. R. and Brown, P. F. (1987). The Automated Manufacturing Research Facility at the National Bureau of

- Standards, in *New Technologies for Production Management systems* (H. Yoshikawa and J. L. Burbidge), Elsevier Science.
- Mertins, K., Sussenguth, W. and Jochem, R. (1992). An Object Oriented Method for Enterprise Modelling as a Basis for Enterprise Coordination, in *Enterprise Modelling: The First International Conference*, The MIT Press, 249-258.
- Meyer, B. (1988). *Object Oriented Software Construction*, Prentice Hall.
- Ming, L., Xiaohong, Y., Tseng, M. M. and Shuzi, Y. (1998). "A CORBA-based agent-driven design for distributed intelligent manufacturing systems." *Journal of Intelligent Manufacturing* **9**(5): 457-465.
- Miyashita, K. (1998). "CAMPS: a constraint-based architecture for multiagent planning and scheduling." *Journal of Intelligent Manufacturing* **9**(2): 147-154.
- Mondal, S. and Tiwari, M. K. (2003). "Formulation of mobile agents for integration of supply chain using the KLAIM concept." *International Journal of Production Research* **41**(1): 97-119.
- Monostori, L., Kádár, B. and Hornyák, J. (1998). "Approaches to managing changes and uncertainties in manufacturing." *CIRP Annals* **47**(1): 365-368.
- Nelson, J. (1999). *Programming Mobile Objects with Java*, John Wiley & Sons.
- Nwana, H. S. (1996). "Software Agents: A Overview." *The Knowledge Engineering Review* **11**(3): 205-244.
- Nwana, H. S. and Ndumu, D. T. (1999). "A Perspective on Software Agents Research." *The Knowledge Engineering Review* **14**(2): 1-18.
- Omg (2003). Introduction to OMG's Unified Modeling Language™ (UML™) ([http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm)), OMG.
- Parunak, H. V. D., Baker, A. D. and Clark, S. J. (1997). The AARIA Agent Architecture: An example of requirements-driven agent-based system design, in *Proceedings of the First International Conference on Autonomous Agents*, Maryna del Rey, CA, ACM Press, 482-483.
- Parunak, H. V. D., Sauter, J., Fleischer, M. and Ward, A. (1999). "The RAPPID Project: Symbiosis between Industrial Requirements and MAS Research." *Autonomous Agents and Multi-Agent Systems* **2**: 111-140.
- Pinedo, M. and Chao, X. (1999). *Operations Scheduling with Applications in Manufacturing and Services*, McGraw-Hill.
- Pressman, R. S. (1987). *Software Engineering: A Practitioner's Approach*, 2nd. McGraw-Hill.
- Product Data Technology Advisory, G. (1997). Standardisation Product Data Model and Associated Economic Benefits: Tutorial.
- Proth, J.-M. and Xie, X. (1996). *Petri Nets: A Tool for Design and Management of Manufacturing Systems*, John Wiley and Sons.
- Putnik, G. D. and Silva, S. C. (1995). One-Product-Integrated-Manufacturing, in *Balanced Automation Systems: Architectures and Design Methods* (L. M. Camarinha-Matos and H. Afsarmanesh).
- Putnik, G. D., Sousa, R. M., Moreira, F. M., Carvalho, J. D., Spasic, Z. and Babic, B. (1998). Distributed/Virtual Manufacturing System Cell: An Experimental Installation, in *The 4th International Intelligent Manufacturing Systems: Theory and Practice*, 59-68.
- Rabelo, R. J. and Camarinha-Matos, L. M. (1996). Towards Agile Scheduling in Extended Enterprise, in *Balanced Automation Systems II: Implementation Challenges for Anthropocentric Manufacturing* (L. M. Camarinha-Matos and H. Afsarmanesh), Chapman & Hall, 413-422.
- Rabelo, R. J., Camarinha-Matos, L. M. and Afsarmanesh, H. (1998). Multiagent Perspectives to Agile Scheduling, in *Intelligent Systems for Manufacturing: Multi-Agent Systems and Virtual Organization* (L. M. Camarinha-Matos, H. Afsarmanesh and V. Marik), Kluwer Academic Publishers: 51-66.
- Rich, E. and Knight, K. (1991). *Artificial Intelligence*, 2nd. McGraw-Hill.
- Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*, Prentice Hall.
- Sanchez, L. M. and Nagi, R. (2001). "A review of agile manufacturing systems." *International Journal of Production Research* **39**(16): 3561-3600.
- Scheer, A.-W. (1999). *ARIS - Business Process Frameworks*, 3rd. Springer-Verlag.
- Shen, W. (2001). "Agent-Based Cooperative Manufacturing Scheduling: an Overview." *COVE News*(2).
- Shen, W., Maturana, F. and Norrie, D. H. (2000a). "Enhancing the performance of an agent-based manufacturing system through learning and forecasting." *Journal of Intelligent Manufacturing* **11**(4): 365-380.
- Shen, W., Maturana, F. and Norrie, D. H. (2000b). "MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing." *Journal of Intelligent Manufacturing* **11**(3): 237-251.
- Shen, W., Norrie, D. H. and Barthès, J. P. (2001). *Multi-agent Systems for Concurrent Intelligent Design and Manufacturing*, Taylor & Francis.
- Shoham, Y. (1997). An Overview of Agent-Oriented Programming, in *Software Agents* (J. M. Bradshaw), The MIT Press: 271-290.
- Shuey, R., Spooner, D. and Frieder, O. (1997). *The Architecture of Distributed Computer Systems*, Addison-Wesley.
- Silva, A. (1999). *Agentes de Software na Internet*, Centro Atlântico.

- Silva, S. C. (2003). "Measuring the quality of service in manufacturing industries (draft paper)." *Cadernos do Departamento de Produção e Sistemas, Escola de Engenharia, Universidade do Minho*.
- Silva, S. C. and Alves, A. (1997). Diferentes Perspectivas de Sistemas de Produção Orientados ao Produto, in *1º Encontro Nacional de Engenharia e Gestão Industrial (in portuguese)*.
- Silva, S. C. and Alves, A. (2002). Design of Product Oriented Manufacturing Systems, in *5th IEEE / IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Systems - BASYS'2002*, Cancun, Mexico, Kluwer.
- Silva, S. C. and Putnik, G. D. (1995). Project on Intelligent Manufacturing Cell at University of Minho, in *The First World Congress on Intelligent Manufacturing Processes and Systems*.
- Skinner, W. (1974). "The Focused Factory." *Harvard Business Review*: 113-121.
- Sluga, A. and Butala, P. (2001). "Self-organization in a distributed manufacturing system based on constraint logic programming." *CIRP Annals* **50**(1): 323-326.
- Sluga, A., Butala, P. and Bervar, G. (1998). "A multi-agent approach to process planning and fabrication in distributed manufacturing." *Computers & Industrial Engineering* **35**(3-4): 455-458.
- Sousa, P. and Ramos, C. (1998). "A dynamic scheduling holon for manufacturing orders." *Journal of Intelligent Manufacturing* **9**(2): 107-112.
- Sousa, P. and Ramos, C. (1999). "A distributed architecture and negotiation protocol for scheduling in manufacturing systems." *Computers in Industry* **38**(2): 103-113.
- Sun, J., Zhang, Y. F. and Nee, A. Y. C. (2001). "A distributed multi-agent environment for product design and manufacturing planning." *International Journal of Production Research* **39**(4): 625-645.
- Tharumarajah, A. (2001). "Survey of resource allocation methods for distributed manufacturing systems." *Production Planning & Control* **12**(1): 58-68.
- Tharumarajah, A., Wells, A. J. and Nemes, L. (1996). "Comparison of the Bionic, Fractal and Holonic Manufacturing System Concepts." *International Journal of Computer Integrated Manufacturing* **9**(3): 217-226.
- Trentesaux, D., Dindeleux, R. and Tahon, C. (1998). "A multicriteria decision support system for dynamic task allocation in a distributed production activity control structure." *International Journal of Computer Integrated Manufacturing* **11**(1): 3-17.
- Trentesaux, D., Tahon, C. and Ladet, P. (1998). "Hybrid production control approach for JIT scheduling." *Artificial Intelligence in Engineering* **12**(1-2): 49-67.
- Tsukada, T. K. and Shin, K. G. (1998). "Distributed tool sharing in flexible manufacturing systems." *Robotics and Automation, IEEE Transactions on* **14**(3): 379-389.
- Tu, Y. (1997). "Production planning and control in a virtual One-of-a-Kind Production company." *Computers in Industry* **34**(3): 271-283.
- Turing, A. (1950). "Computing Machinery and Intelligence." *MIND* **59**(236): 433-460.
- Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, P. (1998). "Reference architecture for holonic manufacturing systems: PROSA." *Computers in Industry* **37**(3): 255-274.
- Vernadat, F. (1999). "Research agenda for agile manufacturing." *International Journal of Agile Management Systems* **1**(1): 37-40.
- Warnecke, H. J. (1993). *The Fractal Company*, Springer-Verlag.
- Wentorf, R. and Shephard, M. S. (1993). Automated Analysis Idealization Control, in *Concurrent Engineering: Automation, Tools, and Techniques (A. Kusiak)*, John Wiley & Sons.
- Wiendahl, H. P. and Ahrens, V. (1997). "Agent-based control of self-organised production systems." *Annals CIRP* **46**(1): 365-368.
- Wu, B. (1994). *Manufacturing System Design and Analysis: Context and Techniques*, 2nd. Chapman & Hall.
- Wu, J. (1999). *Distributed System Design*, CRC Press.
- Wu, N., Mao, N. and Qian, Y. (1999). "An approach to partner selection in agile manufacturing." *Journal of Intelligent Manufacturing* **10**(6): 519-529.
- Yoo, C.-H. (1997). An Object Oriented Database for Information Management in Computer Integrated Manufacturing, in *Advances in Industrial Engineering Applications and Practice II (J. Chen and A. Mital)*, 207-213.
- Yourdon, E. (1994). *Object-Oriented Systems Design, An Integrated Approach*, Prentice Hall.
- Zaremba, M. B., Obuchowicz, A., Banaszak, Z. A. and Jędrzejek, K. J. (1997). "A max-algebra approach to the robust distributed control of repetitive AGV systems." *International Journal of Production Research* **35**(10): 2667-2688.
- Zhao, J., Cheung, W. M. and Young, R. I. M. (1999). "A consistent manufacturing data model to support virtual enterprises." *International Journal of Agile Management Systems* **1**(3): 150-158.
- Zhao, X., Wang, J. and Luo, Z. (2000a). "A stochastic model of a reconfigurable manufacturing system. Part 1: A framework." *International Journal of Production Research* **38**(10): 2273-2285.



- Zhao, X., Wang, J. and Luo, Z. (2000b). "A stochastic model of a reconfigurable manufacturing system. Part 2: Optimal configurations." *International Journal of Production Research* **38**(12): 2829-2842.
- Zhao, X., Wang, J. and Luo, Z. (2001a). "A stochastic model of a reconfigurable manufacturing system. Part 3: Optimal selection policy." *International Journal of Production Research* **39**(4): 747-758.
- Zhao, X., Wang, J. and Luo, Z. (2001b). "A stochastic model of a reconfigurable manufacturing system. Part 4: Performance measure." *International Journal of Production Research* **39**(6): 1113-1126.
- Zolghadri, M., Lecompte, T. and Bourrières, J. P. (1998). A Generalised Scheduling Concept for Multi-level Control of Complex Manufacturing Systems, in *Life Cycle Approaches to Production Systems: Management, Control, Supervision, ASI'98*, Bremen, Germany.



## **ANEXOS**



## 1. EXPERIMENTAÇÃO

Apresentação anexos relativos ao capítulo de experimentação, contendo várias listagens de programas e imagens dos sistemas utilizados.

### 1.1 Xpress-MP 2003 - Código Mosel

```
(!*****
DisMan Example Problems
=====

file Ex3_demX_X_X.mos
*****

EEng - UMinho - DPS
author: R.M. Lima, 2003.06.02
*****!)

model "Ex3_demX_X_X"
uses "mxxprs"           ! Use Xpress-Optimizer

! *****
! Index and dynamic variables declarations
! *****
declarations
  n: integer                ! Number of transformations
  r: integer                ! Number of resources
  fr_k: integer            ! Max. number of free slots per resource
  ntpre: integer          ! Number of precedence transformation tuples
  ntpar: integer          ! Number of parallel transformation tuples

  M: integer                ! BigM

  PType: string

  jj: set of integer       ! Transformation index set
  kk: set of integer       ! Resource index set
  ffk: set of integer      ! Number of free slots index set
  stpar: set of integer    ! parallel transformation tuples index set

  ! Decision variables
  w: array(jj, kk) of mpvar ! Quantity of work wjk
  wz: array(jj, kk) of mpvar ! number of cells
  TjP: array(jj) of mpvar   ! End time for each transformation Tj+
  TjKM: array(jj, kk) of mpvar ! Start time for each transformation by each resource Tjk-
  TjKP: array(jj, kk) of mpvar ! Finish time for each transformation by each resource Tjk+
  bjkfk: array(jj, kk, ffk) of mpvar ! Aux. binary disjoint var. for free periods match
  b_par: array(stpar, kk) of mpvar ! Aux. binary disjoint var. for parallel transformations
end-declarations

! *****
! Index Setup
! *****
initializations from 'Ex3_demX_X_X_3.dat'
  n r fr_k ntpre ntpar
  M PType
end-initializations

! *****
!Data and Variable Declarations
! *****
declarations
  ! Data
  t_last: integer
  cells: integer
  MaxT: array(1..n) of integer
  Maxw: array(1..n, 1..r) of integer
  MaxTjk: array(1..n, 1..r) of integer
  c: array(1..n, 1..r) of integer ! Unit processing times
  W: array(1..n) of integer      ! Required quantity of work by transformation
  Tf: array(1..r) of integer     ! Available free time by resource
  FrM: array(1..r, 1..fr_k) of integer ! Start time for each free slot Fr-
  FrP: array(1..r, 1..fr_k) of integer ! End time for each free slot Fr+
  Tpre: array(1..ntpre, 1..2) of integer ! Transformation precedence tuples
  Tpar: array(1..ntpar, 1..2) of integer ! Transformation parallel tuples

  ! Auxiliary variables
  Tjk: array(1..n, 1..r) of integer ! Processing time for each transformation by each resource Tjk
end-declarations

! *****
! Data setup
! *****
initializations from 'Ex3_demX_X_X_3.dat'
  W Tf c
  FrM FrP
  MaxT
  Tpre Tpar
end-initializations

forall(j in 1..n) do
```

*Sistemas Distribuídos de Produção em Ambiente de Produção Simultânea*

```

forall(k in 1..r) do
    if c(j, k)=0
        then Maxw(j, k) := 0
        else Maxw(j, k) := floor(Tf(k)/c(j, k))
        end-if
        MaxTjk(j, k) := MaxT(j)*Maxw(j, k)
end-do
end-do

! *****
! Decision Variables Creation and Type Definition
! *****
t_last := 0
forall(j in 1..n) do
    create(TjP(j))          !TjP(j) is_integer
    forall(k in 1..r | c(j,k)>0) do
        t_last := j          !Last transformation index (usually equal to n)
        ! Quantity of work variables creation and type definition
        create(w(j,k))
        w(j,k) is_integer
        ! Cell utilization variables creation and type definition
        create(wz(j,k))
        wz(j,k) is_binary
        ! Tjk- and Tjk+ variables creation; doesn't need to define type (it will be integer)
        create(TjkM(j,k))    !TjkM(j,k) is_integer
        create(TjkP(j,k))    !TjkP(j,k) is_integer
        forall(fk in 1..fr_k | FrM(k,fk)>0 or FrP(k,fk)>0) do
            !Period match binary disjoint variables
            create(bjkfk(j, k, fk))
            bjkfk(j, k, fk) is_binary
        end-do
    end-do
end-do

end-do
if( not(ntpar=1 and Tpar(1,1)=Tpar(1,2)) )
then
    forall(g in 1..ntpar) do
        forall(k in 1..r | c(Tpar(g,1),k)>0 and c(Tpar(g,2),k)>0) do
            create(b_par(g, k))
            b_par(g, k) is_binary
        end-do
    end-do
end-if

! *****
! Restrictions
! *****
forall(j in 1..n) do
    UB2(j) := TjP(j) <= MaxT(j)          ! UB2
    R1(j) := sum(k1 in 1..r) w(j, k1) = W(j)          ! R1 - Quantity of work required

    forall(k in 1..r) do
        UB1(j, k) := w(j, k) <= Maxw(j, k)          ! UB1
        UB3(j, k) := TjkM(j, k) <= MaxTjk(j, k)      ! UB3

        R2(k) := sum(j1 in 1..n) c(j1, k)*w(j1, k) <= Tf(k)          ! R2 - Resource free time
        R3(j, k) := TjkP(j, k) = TjkM(j, k)+c(j, k)*w(j, k)          ! R3 - Tjk+ definition
        R5(j, k) := TjP(j) >= TjkP(j, k)              ! R5 - Tj+ = Max(Tjk+)

        UB4(j, k) := sum(fk in 1..fr_k) bjkfk(j, k, fk) <= 1          ! UB4 -
        R4M(j, k) := TjkM(j, k) >= sum(fk in 1..fr_k) FrM(k, fk) * bjkfk(j, k, fk) ! R4 - Free periods
        R4P(j, k) := TjkP(j, k) <= sum(fk in 1..fr_k) FrP(k, fk) * bjkfk(j, k, fk) ! R4 - Free slots

        if c(j, k)=0
            then R8(j, k) := wz(j, k) = 0          ! Cell counter
            else R8(j, k) := wz(j, k) >= w(j, k) / Maxw(j, k)          ! Cell counter
            end-if
    end-do
end-do

!Precedence restrictions
if( not(ntpre=1 and Tpre(1,1)=Tpre(1,2)) )
then
    forall(f in 1..ntpre) do
        forall(k in 1..r | c(Tpre(f,2),k)>0 ) do !and c(Tpre(f,1),k)>0
            R6(f,k) := TjkM(Tpre(f,2), k) >= TjP(Tpre(f,1))          ! R6 - Precedence
        end-do
    end-do
end-if

!Start times for all products
forall(k in 1..r | c(1, k) > 0) do
    R10(k) := TjkM(1, k) >= 0
end-do

!Parallel restrictions
if( not(ntpar=1 and Tpar(1,1)=Tpar(1,2)) )
then
    forall(g in 1..ntpar) do
        forall(k in 1..r | c(Tpar(g,1),k)>0 and c(Tpar(g,2),k)>0) do
            R71P(g,k) := TjkP(Tpar(g,2), k) <= TjkM(Tpar(g,1), k)+b_par(g,k)*M          ! R7 - Paralell
            R71M(g,k) := TjkM(Tpar(g,2), k) >= TjkP(Tpar(g,1), k)-(1-b_par(g,k))*M ! R7 - Paralell
        end-do
    end-do
end-if

! *****
! Solve the problem
! *****
case PType of
    "finish":          do
                        minimize( TjP(t_last) )
                        !exportprob( EP_MPS, "Ex3_demX", TjP(3) )
                    end-do

```

## Sistemas Distribuídos de Produção em Ambiente de Produção Simultânea

```

"work": minimize( sum(j in 1..n, k in 1..r) c(j, k)*w(j, k) )
"cells": minimize( sum(j in 1..n, k in 1..r) wz(j,k) )
"finish_cells": do
    initializations from 'Ex3_demX_X_X_3.dat'
    cells
    end-initializations
    sum(j in 1..n, k in 1..r) wz(j,k) <= cells
    minimize( TjP(t_last) )
end-do
"work_finish": do
    TjP(t_last) <= 6
    minimize( sum(j in 1..n, k in 1..r) c(j, k)*w(j,k) )
end-do
"cells_finish": do
    TjP(t_last) <= 6
    minimize( sum(j in 1..n, k in 1..r) wz(j,k) )
end-do
end-case

! *****
! Solution output
! *****
def_out:=getfid(F_OUTPUT) ! save file ID of default output
fopen("mylog.txt",F_OUTPUT)! switch output to 'mylog.txt'
my_out:=getfid(F_OUTPUT) ! save ID of current output stream

fselect(def_out) ! select default ouput...
fselect(my_out) ! select the file 'mylog.txt'

writeln("MIP_Solution")
writeln(PType)
writeln("Objective: ", getobjval)

writeln("w=[")
forall(j in 1..n) do
    write(strfmt(" ",5))
    forall(k in 1..r) write(strfmt(getsol(w(j,k)),3),", " )
    writeln("")
end-do
writeln(" ]")
writeln("")

writeln("Tj+=[")
!forall(j in 1..n) write(strfmt(getsol(TjP(j)),3),", " )
forall(j in 1..n) do
    write(strfmt(" ",7))
    write(strfmt(getsol(TjP(j)),3),", " )
    writeln("")
end-do
writeln(" ]")
writeln("")

writeln("Tjk-=[")
forall(j in 1..n) do
    write(strfmt(" ",7))
    forall(k in 1..r) write(strfmt(getsol(TjkM(j, k)),3),", " )
    writeln("")
end-do
writeln(" ]")
writeln("")

writeln("Tjk+=[")
forall(j in 1..n) do
    write(strfmt(" ",7))
    forall(k in 1..r) write(strfmt(getsol(TjkP(j, k)),3),", " )
    writeln("")
end-do
writeln(" ]")
writeln("")

writeln("Tjk=[")
forall(j in 1..n) do
    write(strfmt(" ",7))
    forall(k in 1..r) write(strfmt(getsol(TjkP(j, k)-TjkM(j, k)),3),", " )
    writeln("")
end-do
writeln(" ]")
writeln("")

writeln("")

writeln("wz=[")
forall(j in 1..n) do
    write(strfmt(" ",6))
    forall(k in 1..r) write(strfmt(getsol(wz(j,k)),3),", " )
    writeln("")
end-do
writeln(" ]")

fclose(F_OUTPUT) ! close current output (= 'mylog.txt')
end-model

```

## 1.2 Xpress-MP 2003 – Dados de Entrada

```
!Datafile_for_'Ex3_demX_X_X.mos'-P11          1          P          1          1
!Prob3_transf3_res5_dem10

n:      1
r:      5
fr_k:   1

M:      100000

PType:  finish
!PType: work
!PType: cells
!PType: finish_cells
!PType: work_finish
!PType: cells_finish

!Data_setup
W: [
    10
  ]
c: [
    1          1          1          1
  ] !Unit_Processing_time

!Janelas_temporais_disponiveis_para_utilização
!
FrM: [
    0
    0
    0
    0
    0
  ]
!
FrP: [
    100
    100
    100
    100
    100
  ]

MaxT: [
    100
  ]

Tf: [
    100
    100
    100
    100
    100
  ] !Free_time_by_resource

!Precedence_and_parallel_data
ntpre: 1
ntpar: 1
Tpre: [
    1          1
  ] !Transformation_precedence
Tpar: [
    1          1
  ] !Transformation_parallel
```



## 1.3 Experimentação AA

### 1.3.1. Folha de Cálculo

Ordem	Produto	Ordem_Prd	Qtyd	Mosel_file
1		4	1	1 cap7AA_6.mos
2		4	2	1 cap7AA_6.mos
3		4	3	1 cap7AA_6.mos
4		3	1	1 cap7AA_6.mos
5		3	2	1 cap7AA_6.mos
6		3	3	1 cap7AA_6.mos
7		2	1	1 cap7AA_6.mos
8		2	2	1 cap7AA_6.mos
9		2	3	1 cap7AA_6.mos
10		2	4	1 cap7AA_6.mos
11		2	5	1 cap7AA_6.mos
12		1	1	1 cap7AA_6.mos
13		1	2	1 cap7AA_6.mos
14		1	3	1 cap7AA_6.mos
15		1	4	1 cap7AA_6.mos
16		1	5	1 cap7AA_6.mos
17		1	6	1 cap7AA_6.mos
18		1	7	1 cap7AA_6.mos
19		1	8	1 cap7AA_6.mos
20		5	1	2 cap7AA_6.mos
21		5	2	2 cap7AA_6.mos
22		5	3	2 cap7AA_6.mos
23		5	4	2 cap7AA_6.mos
24		5	5	2 cap7AA_6.mos
25		5	6	2 cap7AA_6.mos

### 1.3.2. Programação em VBA

```

VERSION 1.0 CLASS
BEGIN
  MultiUse = -1 'True
END
Attribute VB_Name = "Sheet7"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = True

Private Sub CButton_Go_Click()
  Dim i As Integer
  Dim order As Integer

  order = get_order

  read_prod_data
  Calculate
  Application.Wait (Now + TimeValue("0:00:03"))
  save_text
  solve_prob
  Calculate
  Application.Wait (Now + TimeValue("0:00:02"))

  For i = 2 To order
    read_text
    Calculate
    Application.Wait (Now + TimeValue("0:00:03"))
    copy_parcial_sol
    read_prod_data
    Calculate
    Application.Wait (Now + TimeValue("0:00:03"))
    free_periods
    save_text
    solve_prob
  
```

## *Sistemas Distribuídos de Produção em Ambiente de Produção Simultânea*

```
        Calculate
        Application.Wait (Now + TimeValue("0:00:02"))
    Next i

    read_text
    Calculate
    Application.Wait (Now + TimeValue("0:00:02"))
    copy_parcial_sol
    save_xls
End Sub

Private Sub CommandButton1_Click()
    save_text
End Sub
```

---

## 2. MODELO DE AGENTES DE SOFTWARE

Apresentam-se alguns anexos relativos ao modelo multiagente do Sistema Distribuído de Produção.

### 2.1 Exemplo de Aplicação das Ferramentas Seleccionadas

A imagem seguinte corresponde a uma imagem do écran representando processos relacionados com o exemplo de implementação, que permitiu verificar a aplicabilidade dos JavaSpaces como ferramenta de implementação de um espaço de comunicação partilhado.

The screenshot displays a desktop environment with several windows illustrating the system's operation:

- Top Left Window (C:\j2sdk1.4.1\_01\bin\java.exe):** Shows a list of requests from various BATMAN agents (e.g., 1038, 1046, 1054, 1056, 1061, 1064, 1069, 1075, 1082, 1086, 1090, 1092, 1106, 1111, 1115, 1126, 1130, 1135) for files like reggie-dl.jar, mahalo-dl.jar, and outrigger-dl.jar.
- Top Right Window (C:\WINDOWS\System32\cmd.exe):** Shows the execution of a Java command: `C:\java\JavaSpacesBasics\classes\jsbe\ch3-d1>java -jar \java\jini1_2_1\lib\tools.jar -port 8082 -dir \java\JavaSpacesBasics\classes\jsbe\ch3-d1 -verbose net/jini1/lookup/ServiceDiscoveryManager$LookupCacheImpl$LookupListener_Stub.class requested from BATMAN:1077`
- Middle Left Window:** A desktop icon for 'Sun ONE Studio 4 CE'.
- Middle Right Window (C:\WINDOWS\System32\cmd.exe):** Shows the command to start the 'OrderSpace' Java Space: `C:\java\JavaSpacesBasics\classes\jsbe\disman>java -jar -Djava.security.policy=\java\jini1_2_1\policy\policy.all -Djava.rmi.server.codebase=http://batman:8081/outrigger-dl.jar -Dcom.sun.jini.outrigger.spaceName=OrderSpace \java\jini1_2_1\lib\transient-outrigger.jar public`
- Bottom Left Window (Command Prompt - Customer OrderSpace zz z):** Displays bid and processor information:
 

```
Delivering Time:36.757713317871094
This bid doesn't exist, so add it.
Bid
Bid ID jsbe.ch6.Stamp@992bae
Bid ID - counter: 10
Bid ID - High: 1057424231926
Bid ID - Low: 2985402354070591873
Customer ID counter: 4
Customer ID high: 1057424231926
Customer ID low: 2985402354070591873
Order ID counter: 5
currentOrder ID counter: 5
Item ID counter: 0
Quantity: 38
Processor
Processor ID counter: 3
Processing Price:34.602420806884766
Processing Time:18.914133071899414
Delivering Time:13.859687805175781
```
- Bottom Right Window (Command Prompt - Processor OrderSpace):** Displays bid and processing information:
 

```
Bid
Bid ID jsbe.ch6.Stamp@11e67ac
Bid ID - counter: 9
Bid ID - High: 1057424231926
Bid ID - Low: 2985402354070591873
Processing Price:28.048791885375977
Processing Time:4.713225841522217
Delivering Time:36.757713317871094
Start processing order
Allocated 38 units of LoquatMaceraterCuberJuicerMincerCookeryMasherShredderPeelerDicerSlicerShredder
Bid
Bid ID jsbe.ch6.Stamp@f864fe
Bid ID - counter: 10
Bid ID - High: 1057424231926
Bid ID - Low: 2985402354070591873
Processing Price:34.602420806884766
Processing Time:18.914133071899414
Delivering Time:13.859687805175781
```